

Partie I - Logique et calcul des propositions

La notation $\overline{p}$ signifie $\neg p$.

Question I.1 $E = (I_1 \wedge I_2 \wedge I_3) \vee (\overline{I_1} \wedge \overline{I_2} \wedge \overline{I_3})$.

Question I.2 $I_1 = (C_1 \wedge C_3)$, $I_2 = (C_2 \implies \overline{C_3}) \equiv (\overline{C_2} \vee \overline{C_3})$, $I_3 = (C_1 \wedge \overline{C_2})$.

Question I.3

C_1	C_2	C_3	I_1	I_2	I_3	$I_1 \wedge I_2 \wedge I_3$	$\overline{I_1} \wedge \overline{I_2} \wedge \overline{I_3}$	E
0	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0
0	1	0	0	1	0	0	0	0
0	1	1	0	0	0	0	1	1
1	0	0	0	1	1	0	0	0
1	0	1	1	1	1	1	0	1
1	1	0	0	1	0	0	0	0
1	1	1	1	0	0	0	0	0

$$E \equiv (\overline{C_1} \wedge C_2 \wedge C_3) \vee (C_1 \wedge \overline{C_2} \wedge C_3) \equiv (C_1 \oplus C_2) \wedge C_3.$$

Ainsi $(E \implies C_3)$: Socrate doit donc prendre le troisième couloir.

Question I.4 Si Cerbère a menti, c'est que $\overline{I_1} \wedge \overline{I_2} \wedge \overline{I_3}$ est vrai, ce qui a lieu si $\overline{C_1} \wedge C_2 \wedge C_3 = 1$, donc $C_2 = 1$ et $C_3 = 1$.
Socrate a donc le choix entre le second et le troisième couloir.

Partie II - Algorithmique et programmation en CaML

L'énoncé définit l'enveloppe convexe $\mathcal{E}_C(\mathcal{P})$ de $\mathcal{P}$ comme étant un polygone "plein" alors que dans la suite du problème, $\mathcal{E}_C(\mathcal{P})$ désigne plutôt son contour.

Question II.1

Posons $a = \|\overrightarrow{qp}\|$ et $b = \|\overrightarrow{qr}\|$. a et b sont non nuls car p, q, r sont distincts.

Posons $\overrightarrow{u} = \frac{1}{a} \cdot \overrightarrow{qp}$. Ainsi $\overrightarrow{u}$ est normé et $\overrightarrow{qp} = a \cdot \overrightarrow{u}$. Notons $\overrightarrow{v}$ le vecteur normé directement orthogonal à $\overrightarrow{u}$.

Alors $\overrightarrow{qr}$ se décompose en : $\overrightarrow{qr} = b \cdot (\cos \theta \cdot \overrightarrow{u} + \sin \theta \cdot \overrightarrow{v})$ avec $\theta \in [0, 2\pi[$.

Le calcul du produit mixte $[p, q, r]$ ne dépend pas de la base orthonormée directe dans laquelle on se place,

$$\text{donc } [p, q, r] = \begin{vmatrix} a & b \cos \theta \\ 0 & b \sin \theta \end{vmatrix} = ab \sin \theta, \text{ d'où la conclusion :}$$

- . $[p, q, r] > 0 \iff \sin \theta > 0 \iff \theta \in]0, \pi[$.
- . $[p, q, r] < 0 \iff \sin \theta < 0 \iff \theta \in]\pi, 2\pi[$.
- . $[p, q, r] = 0 \iff \sin \theta = 0 \iff \theta = 0 \text{ ou } \pi \iff p, q, r \text{ alignés.}$

Question II.2

```
let croise (p:point) (q:point) (r:point) =
  let (p1,p2) = p and (q1,q2) = q and (r1,r2) = r
  in (p1 - q1) * (r2 - q2) - (p2 - q2) * (r1 - q1) ;;
```

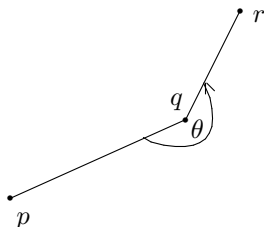
Question II.3

```

let a_gauche (p:point) (q:point) (e:suite) =
  let rec aux liste (accu:suite) =
    match liste with
    | [] -> accu
    | r::queue -> if (croise p q r) < 0 then aux queue (r::accu)
                  else aux queue accu
  in aux e [] ;;

```

Question II.4



Les points situés à gauche de la droite (p, q) orientée de p vers q sont tels que l'angle $\theta = \widehat{pqr} \in]\pi, 2\pi[$.

On parcourt la liste e (grâce à l'instruction `match` et à la récursivité) en ajoutant dans la variable `accu` (initialisée avec la liste vide) tous les points r rencontrés qui sont situés à gauche de la droite (p, q) . Les appels récursifs s'arrêtent lorsqu'on arrive à une liste vide.

La fonction auxiliaire `aux` est à récursivité terminale.

Question II.5 La complexité est évidemment proportionnelle à la taille de la liste e .

Question II.6

```

let separation (p:point) (q:point) (e:suite) =
  let rec aux liste (g:suite) (d:suite) (a:suite) =
    match liste with
    | [] -> (g,d,a)
    | r::queue -> let s = croise p q r in
                  if s < 0
                  then aux queue (r::g) d a
                  else if s > 0
                  then aux queue g (r::d) a
                  else if (r=p) || (r=q)
                  then aux queue g d a
                  else aux queue g d (r::a)
  in aux e [] [] [] ;;

```

Question II.7

```

let ordre_polaire (o:point) (p:point) (q:point) =
  let s = croise p o q in
  if s = 0
  then let (o1,o2) = o and (p1,p2) = p and (q1,q2) = q in
        let a1 = p1-o1 and a2 = p2-o2 and b1 = q1-o1 and b2 = q2-o2
        in b1 * b1 + b2 * b2 <= a1 * a1 + a2 * a2
  else (s > 0) ;;

```

Question II.8

```

let extreme (p1:point) (p2:point) =
  let (x1,y1) = p1 and (x2,y2) = p2 in
    if y1 < y2 then (p1,p2)
      else if y2 < y1 then (p2,p1)
        else if x1 < x2 then (p1,p2)
          else (p2,p1) ;;

```

Question II.9

```

let extremes (e:suite) =
  let rec aux liste b h =
    match liste with
    | [] -> (b,h)
    | p::queue -> aux queue (fst (extreme b p)) (snd (extreme h p))
  in aux (tl e) (hd e) (hd e) ;;

```

Question II.10 On suppose la liste e non vide et on choisit pour initialiser les variables b et h de la fonction auxiliaire aux le couple (t, t) où t désigne l'élément de tête de la liste e .

Après chaque appel récursif, les variables b et h contiennent les deux points extrêmes de la sous-liste déjà traitée.

Chaque fois qu'on examine un point p de la liste e , on met à jour b et h en utilisant la formule d'associativité :

$$extremes(\{p\} \cup L) = extreme(p, extremes(L)).$$

Question II.11 Raisonnons par l'absurde. Supposons que B par exemple ne soit pas sur $\mathcal{E}_C(\mathcal{P})$ (considéré comme étant le polygone frontière). B se trouve donc soit à "l'intérieur" du polygone, soit éventuellement sur un des cotés $[p, q]$ (comme le point $(5, 3)$ de l'exemple II.1).

. S'il est à l'intérieur, c'est qu'il existe un point de $\mathcal{P}$ plus bas que B , ce qui contredit que $y_B = \min\{y_M, M \in \mathcal{P}\}$.

. S'il est sur un des cotés $[p, q]$, alors une des extrémités p ou q est située plus bas à gauche que B , ce qui contredit que $y_B = \min\{y_M, M \in \mathcal{P}\}$.

Ainsi B et H sont des sommets de $\mathcal{E}_C(\mathcal{P})$.

Question II.12 On suppose bien sûr que $\text{Card}(\mathcal{P}) \geq 2$.

- Montrons que $\leq_B$ est une relation d'ordre sur $E = \mathcal{P}_D \cup \mathcal{P}_{BH} \cup \{H\}$.
 - ★ Elle est réflexive car $\forall M \in E$, $\widehat{MBM} = 0$ et $\|\overrightarrow{BM}\| \geq \|\overrightarrow{BM}\|$, donc $M \leq_B M$.
 - ★ Si $M \leq_B N$ et $N \leq_B M$, alors B, M, N sont alignés car on ne peut pas avoir à la fois $\widehat{MBN} \in]0, \pi[$ et $\widehat{NBM} \in]0, \pi[$. Mais alors $\|\overrightarrow{BM}\| \geq \|\overrightarrow{BN}\|$ et $\|\overrightarrow{BN}\| \geq \|\overrightarrow{BM}\|$, donc $\|\overrightarrow{BM}\| = \|\overrightarrow{BN}\|$.
 M, B, N étant alignés, il reste deux possibilités. Mais on ne peut pas avoir M et N symétriques par rapport à B car B est le point le plus bas à gauche de $\mathcal{P}$. Donc nécessairement $M = N$, ce qui prouve que la relation $\leq_B$ est antisymétrique.
 - ★ Si $M \leq_B N$ et $N \leq_B P$, alors $\widehat{MBP} = \widehat{MBN} + \widehat{NBP}$ est ≥ 0 (comme somme de deux mesures d'angles ≥ 0) et est $\leq \pi$ du fait que les points M et P sont dans E .
 Le cas où $\widehat{MBP} = \pi$ est impossible car B serait entre M et P , donc B ne serait pas le point le plus bas de $\mathcal{P}$.
 Le cas particulier où $\widehat{MBP} = 0$ ne pose pas de problème car alors les points sont alignés dans l'ordre B, P, N, M .
 On a donc $M \leq_B P$ dans tous les cas de figures, ce qui prouve que la relation $\leq_B$ est transitive.
- Montrons maintenant que $\leq_B$ est une relation d'ordre total sur E .
 Considérons deux points distincts M et N de E .
 . S'ils sont alignés avec B , alors ils sont du même coté de B et alors $M \leq_B N$ ou $N \leq_B M$ selon que $\|\overrightarrow{BM}\| > \|\overrightarrow{BN}\|$ ou $\|\overrightarrow{BN}\| > \|\overrightarrow{BM}\|$.
 . S'ils ne sont pas alignés, alors $M \leq_B N$ si $\widehat{MBN} \in]0, \pi[$, ou $N \leq_B M$ si $\widehat{MBN} \in]\pi, 2\pi[$ car alors $\widehat{NBM} \in]0, \pi[$.
 En résumé M et N sont comparables pour la relation d'ordre $\leq_B$.
Conséquence : toute partie de E admet un minimum (car E est fini).

Question II.13

Notons $(s_1, \dots, s_n)$ les sommets consécutifs du contour positif de $\mathcal{E}_C(\{B, H\} \cup \mathcal{P}_D)$ avec $s_1 = B$ et $s_n = H$.

Soit p le minimum de $\mathcal{P}_D \cup \{H\}$ selon $\leq_B$.

Montrons par l'absurde que $p = s_2$.

Sinon, d'après le théorème II.2, p serait situé à gauche de l'arête $[s_1, s_2]$: on aurait donc $\widehat{s_1 s_2 p} \in]\pi, 2\pi[$, d'où $\det(\overrightarrow{s_2 s_1}, \overrightarrow{s_2 p}) < 0$.

Alors $\det(\overrightarrow{s_1 s_2}, \overrightarrow{s_1 p}) = -\det(\overrightarrow{s_2 s_1}, \overrightarrow{s_1 s_2} + \overrightarrow{s_2 p}) = -\det(\overrightarrow{s_2 s_1}, \overrightarrow{s_2 p}) > 0$, donc $\widehat{s_2 s_1 p} \in]0, \pi[$, d'où $s_2 \leq_B p$ et $s_2 \neq p$, ce qui contredit que p est le minimum de $\mathcal{P}_D \cup \{H\}$ selon $\leq_B$.

Question II.14

En raisonnant de même, on montrerait que s_{i+1} est égal au minimum de $\mathcal{P}_D \cup \{H\} \setminus \{s_1, \dots, s_i\}$ selon $\leq_{s_i}$.

Question II.15 Évident. On s'arrête lorsque $s_i = H$.**Question II.16**

```
let min_polaire (o:point) (e:suite) =
  let rec aux liste m (accu:suite) =
    match liste with
    | [] -> (m, accu)
    | p::queue -> if ordre_polaire o p m
                  then aux queue p (m::accu)
                  else aux queue m (p::accu)
  in aux (tl e) (hd e) [] ;;
```

Question II.17 Les variables `m`, `liste`, `accu` sont initialisées respectivement avec le premier élément de la liste `e`, la liste `e` amputée de son premier élément et la liste vide.

Après chaque appel récursif, la variable `m` contient le minimum selon $\leq_o$ de la sous-liste déjà traitée et la variable `accu` tous les éléments de cette sous-liste sauf `m`.

Lors de chaque appel récursif, on compare p avec m :

- si $p \leq_o m$, alors la valeur de m est rajoutée dans `accu` et p remplace m .
- sinon on conserve m et on rajoute p à `accu`.

Question II.18 Le nombre d'appels récursifs est égal à $|e| - 1$.**Question II.19**

```
let demi_jarvis b h (e:suite) =
  let rec aux liste (accu:suite) si =
    if si = h then accu
    else let (suivant, sousliste) = min_polaire si liste
        in aux sousliste (suivant::accu) suivant
  in let (s:suite) = b::(rev (aux (h::e) [] b)) in s ;;
```

Question II.20 On applique l'algorithme mis en évidence à la question II.14 selon lequel s_{i+1} est égal au minimum de $\mathcal{P}_D \cup \{H\} \setminus \{s_1, \dots, s_i\}$ selon $\leq_{s_i}$.

On s'arrête lorsque $s_i = h$. La variable `accu` contient alors la suite inversée $(s_n, \dots, s_2)$. On la retourne et on ajoute b en tête pour obtenir la liste voulue $(s_1, s_2, \dots, s_n)$.

Remarquons que cette fonction est encore correcte dans le cas où e est vide car elle renvoie alors (b, h) .

Question II.21 Notons $N = |e|$.

Il y a $n - 1$ appels de la fonction récursive `aux`, chacun appelant la fonction `min_polaire`.

. le premier appel de `min_polaire` porte sur une liste de $N + 1$ éléments a pour complexité N .

. le deuxième appel de `min_polaire` porte sur une liste de N éléments a pour complexité $N - 1$.

. le dernier appel de `min_polaire` porte sur une liste de $N - n + 3$ éléments a pour complexité $N - n + 2$.

La complexité totale est donc $\sum_{k=1}^{n-1} (N + 1 - k) = (n - 1)(N + 1) - \frac{n(n - 1)}{2} \leq nN$.

Question II.22

```

let jarvis (e:suite) =
  let (b,h) = extremes e
  in let (g,d,a) = separation b h e
  in let s1 = demi_jarvis b h d and s2 = demi_jarvis h b g
  in let (s:suite) = (tl s1)@(tl s2) in s ;;

```

Question II.23 On commence par déterminer les deux points extrêmes b et h de e , puis on sépare e avec la droite (b, h) pour obtenir la partition (g, d, a) . Les points de a autres que b et h n'interviennent pas dans la détermination de l'enveloppe convexe de e .

On calcule la suite S_1 donnant l'enveloppe convexe de $g \cup \{b, h\}$ en partant de b et la suite S_2 donnant l'enveloppe convexe de $d \cup \{h, b\}$ en partant de h conformément à une remarque de l'énoncé.

Enfin, on réunit ces deux listes en tenant compte du fait que S_1 se termine par h et que S_2 commence aussi avec h qu'on supprime donc de S_2 (et S_2 se termine par b et que S_1 commence aussi avec b qu'on supprime donc de S_1).

La liste finale donne le contour positif de $\mathcal{E}_C(\mathcal{P})$ se terminant par b .

Question II.24 Notons $N = |\mathcal{P}|$ et $n = |\mathcal{E}_C(\mathcal{P})|$.

Le calcul des points extrêmes et la séparation ont une complexité en $O(N)$.

Dans le pire des cas où tous les points de $\mathcal{P}$ sont sur $\mathcal{E}_C(\mathcal{P})$, alors $N = n$.

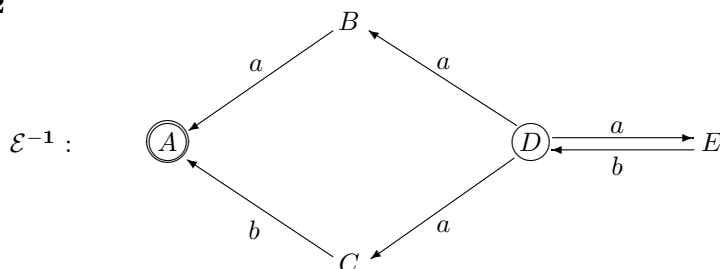
La complexité de la fonction `jarvis` est donc en $O(N^2)$.

Partie III - Automates et langages

Question III.1

$$E_1 = (a + b)a(ba)^* \equiv (a + b)(ab)^*a \equiv a(ab)^*a + b(ab)^*a.$$

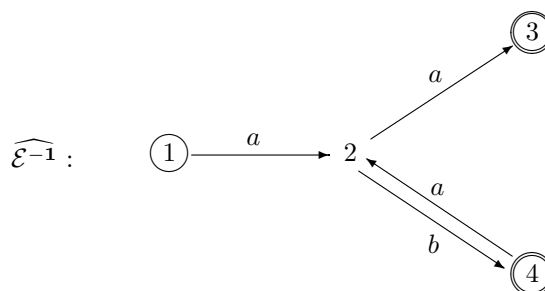
Question III.2



Question III.3 L'automate $\mathcal{E}^{-1}$ est non déterministe car de D partent plusieurs transitions étiquetées par a .

Construisons la table de transition de l'automate déterminisé et émondé $\widehat{\mathcal{E}^{-1}}$ équivalent à $\mathcal{E}^{-1}$.

	a	b
$1 = \{D\}$	$\{B, C, E\}$	
$2 = \{B, C, E\}$	$\{A\}$	$\{A, D\}$
$3 = \{A\}$		
$4 = \{A, D\}$	$\{B, C, E\}$	



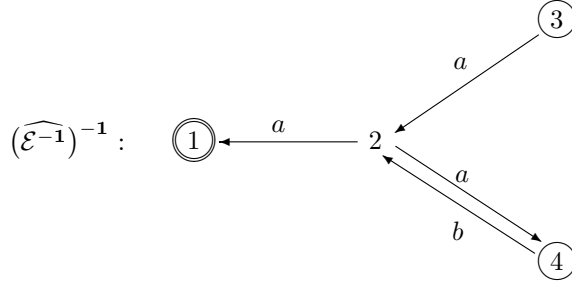
Question III.4 Le langage reconnu est la réunion de deux langages :

- celui formé des mots qui conduisent à l'état final 3 décrit par l'expression rationnelle $a(ba)^*a$.
- celui formé des mots qui conduisent à l'état final 4 décrit par l'expression rationnelle $a(ba)^*b$.

Donc $E_2 = a(ba)^*a + a(ba)^*b \equiv a(ba)^*(a+b).$

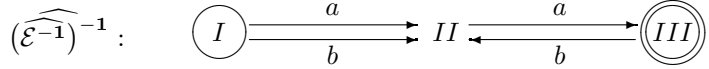
On constate qu'il s'agit du langage miroir : $a_1 \dots a_n \in L(\mathcal{E}) \iff a_n \dots a_1 \in L(\widehat{\mathcal{E}^{-1}}).$

Question III.5



Question III.6 L'automate $(\widehat{\mathcal{E}^{-1}})^{-1}$ n'est pas déterministe car il a deux états initiaux 3 et 4. De plus, de l'état 2 partent deux transitions étiquetées par a .

	a	b
$I = \{3, 4\}$	$\{2\}$	$\{2\}$
$II = \{2\}$	$\{1, 4\}$	
$III = \{1, 4\}$		$\{2\}$



Question III.7

Une expression rationnelle du langage reconnu par ce dernier automate est : $E_3 = (a+b)a(ba)^* = E_1$.
C'est normal, car le langage miroir de $L(E_2)$ est E_1 .

On a donc obtenu ainsi un automate "minimal" équivalent à $\mathcal{E}$.

Question III.8 Montrons la propriété par récurrence sur $k = |m|$.

. Si $m = 0$, alors $m = \Lambda$ et l'équivalence est vérifiée pour $q'' = q$.

. Supposons la propriété vraie pour tout mot u de longueur k .

Soit m un mot de longueur $k+1$: il s'écrit $m = y.u$ avec $|u| = k$ et $y \in X$.

C.N. : supposons que $(m.x, q, q') \in \gamma^*$, c'est à dire $(y.u.x, q, q') \in \gamma^*$.

Par définition, $\exists p \in Q / (y, q, p) \in \gamma \wedge (u.x, p, q') \in \gamma^*$.

D'après l'hypothèse de récurrence, $\exists q'' \in Q / (u, p, q'') \in \gamma^* \wedge (x, q'', q') \in \gamma$.

Or par définition, comme $(y, q, p) \in \gamma \wedge (u, p, q'') \in \gamma^*$, on a $(y.u, q, q'') \in \gamma^*$. Ainsi $(m, q, q'') \in \gamma^* \wedge (x, q'', q') \in \gamma$.

C.S. : supposons que $\exists q'' \in Q / (m, q, q'') \in \gamma^* \wedge (x, q'', q') \in \gamma$. Alors, par définition de γ^* appliquée à m , on a : $\exists p \in Q / (y, q, p) \in \gamma \wedge (u, p, q'') \in \gamma^*$, donc, d'après l'hypothèse de récurrence, $(u.x, p, q') \in \gamma^*$ et enfin, par définition, $(y.u.x, q, q') \in \gamma^*$, c'est à dire $(m.x, q, q') \in \gamma^*$.

Question III.9 Montrons de même la propriété par récurrence sur $k = |m|$.

. Si $m = 0$, alors $m = \Lambda$ et on a : $\Lambda^{-1} = \Lambda$, $(\Lambda, q, q) \in \gamma^*$ et $(\Lambda^{-1}, q, q) \in (\gamma^{-1})^*$.

. Supposons la propriété vraie pour tout mot u de longueur k .

Soit m un mot de longueur $k+1$: il s'écrit $m = y.u$ avec $|u| = k$ et $y \in X$.

C.N. : supposons que $(m, q, q') \in \gamma^*$. Par définition, $\exists q'' \in Q / (y, q, q'') \in \gamma \wedge (u, q'', q') \in \gamma^*$.

D'après l'hypothèse de récurrence, $(u^{-1}, q', q'') \in (\gamma^{-1})^*$. De plus, par définition, $(y, q'', q) \in \gamma^{-1}$.

Or $m^{-1} = u^{-1}.y$: il résulte de **III.8** que $(m^{-1}, q', q) \in (\gamma^{-1})^*$.

C.S. : supposons que $(m^{-1}, q', q) \in (\gamma^{-1})^*$ avec $m^{-1} = u^{-1}.y$. D'après **III.8**, $\exists q'' \in Q / (u^{-1}, q', q'') \in (\gamma^{-1})^*$ et $(y, q'', q) \in \gamma^{-1}$, donc, par définition, $(u, q'', q') \in \gamma^*$ et $(y, q, q'') \in \gamma$.

Finalement, $(y.u, q, q') \in \gamma^*$, c'est à dire $(m, q, q') \in \gamma^*$.

Question III.10 Montrons que $L(\mathcal{A}^{-1})$ est le langage "miroir" de $L(\mathcal{A})$.

$m \in L(\mathcal{A})$ $\iff \exists q_I \in I, \exists q_T \in T / (m, q_I, q_T) \in \gamma^* \iff (m^{-1}, q_T, q_I) \in (\gamma^{-1})^* \iff$ $m^{-1} \in L(\mathcal{A}^{-1})$.

Fin du corrigé