DS n° 04 — corrigé

I Questions de cours

Q 1) Soit $L \subseteq \Sigma^*$ un langage régulier. Alors il existe $N \in \mathbb{N}^*$ tel que pour tout mot $u \in L$ avec $|u| \geq N$, il existe $x, y, z \in \Sigma^*$ tel que $xyz = u$ avec $|xy| \leq N$ et $y \neq \epsilon$ et tel que $xy^*z \subseteq L$.

Q 2)
- L_1 est dénoté par l'expression régulière aa^*b^2 et est donc régulier.
- Supposons que L_2 soit régulier, notons $N > 0$ la constante du lemme de l'étoile et considérons le mot $u = a^N b^{2N}$. Il existe une décomposition $u = xyz$ avec $|xy| \leq N$ et $y \neq \epsilon$ et donc $i \in \mathbb{N}$ et $j \in \mathbb{N}^*$ tels que $x = a^i$, $y = b^j$ et $z = a^{N-i-j}b^{2N}$. La conclusion du lemme de l'étoile indique que $xz = a^{N-j}b^{2N} \in L_2$ ce qui est absurde puisque $2|u|_a = 2(N-j) \neq 2N = |u|_b$ puisque $j \neq 0$. Le langage L_2 n'est donc pas régulier.
- $L_3 = \Sigma^*\Sigma^* = \Sigma^*$ qui est bien sûr régulier.
- On applique le lemme de l'étoile avec le mot $u = a^N b a^N b$ où N est la constante donnée par le lemme. On en déduit que $a^{N-j}ba^Nb$ est un carré, avec $j \neq 0$, ce qui est absurde. Le langage n'est donc pas régulier.

Q 3) La commande `ls` permet de lister le contenu du répertoire courant.

Q 4) Il est tout à fait possible d'écrire un programme qui vérifie qu'un texte est un programme OCaml valide. C'est d'ailleurs une des premières étapes effectuées par un compilateur.

Q 5) On utilise la fonction `termine` et une fonction qui ne termine pas.

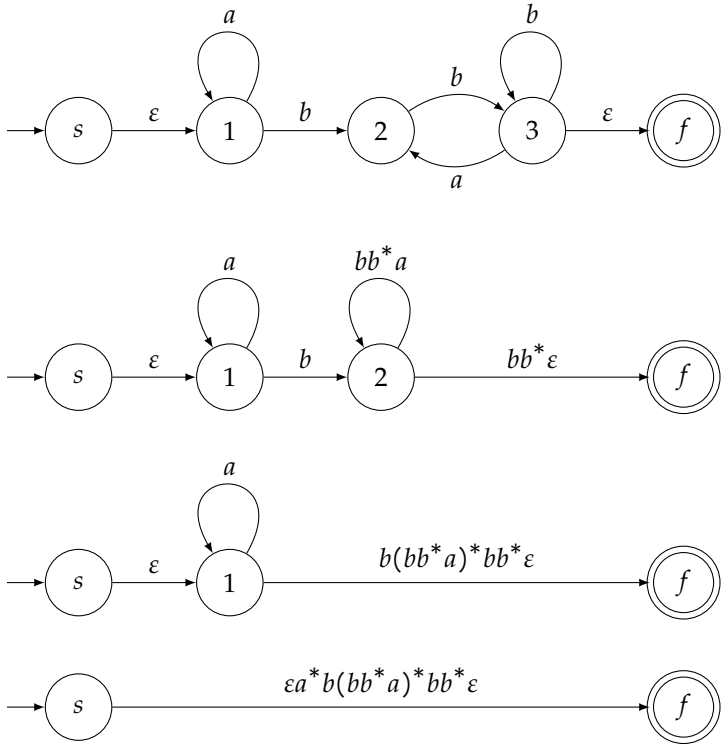
OCaml
```
let terminaison_inverse nom_fichier =
  let rec loop () = loop () in
  if termine nom_fichier then loop ()
```

Q 6) La compilation de ce programme OCaml valide ne pose aucun problème et on obtient un exécutable. Si cet exécutable termine, c'est que le programme de nom `"paradoxe.ml"` termine. Or ce programme se termine par l'exécution de `terminaison_inverse "paradoxe.ml"` qui ne termine pas. Ceci est absurde. Inversement, si ce programme ne termine pas, comme il comporte deux définitions de fonction, ce qui termine, c'est que l'appel `terminaison_inverse "paradoxe.ml"` ne termine pas et donc que ce programme termine. C'est également absurde.

Q 7) Un tel programme ne peut donc pas exister, nous avons pourtant réussi à l'écrire en supposant l'existence de la fonction `termine`, cette fonction ne peut donc pas exister.

Q 8) On obtient l'arbre de la figure 1.

Q 9) On obtient successivement :



Le langage reconnu par l'automate est donc dénoté par l'expression régulière $\epsilon a^*b(bb^*a)^*bb^*\epsilon$.

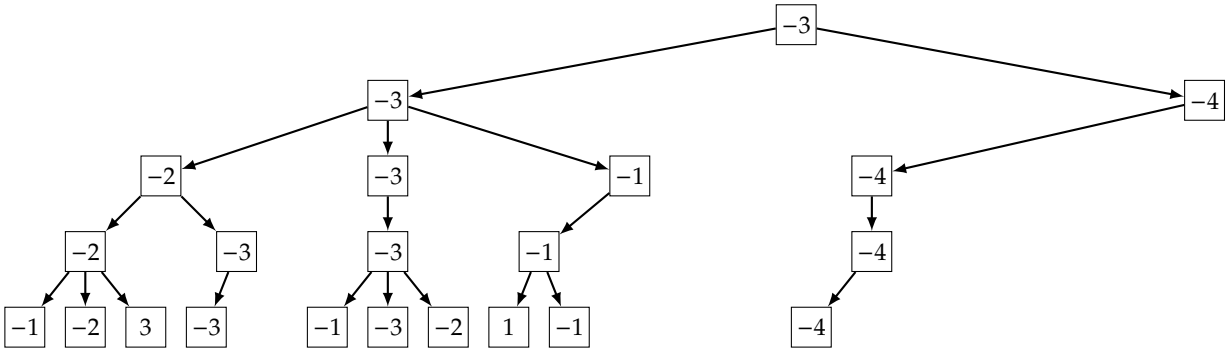
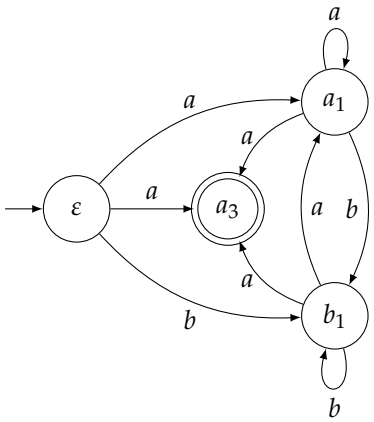


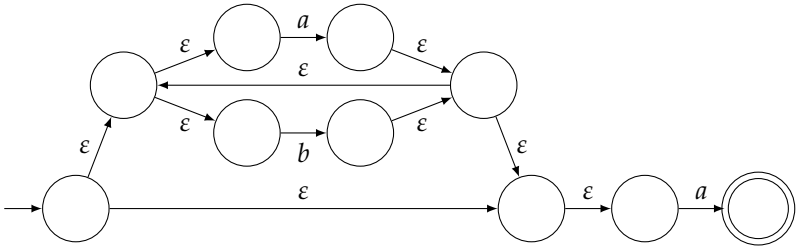
FIGURE 1 – Arbre des configurations jusqu'à une profondeur 4 avec valeur de l'heuristique aux feuilles élagué avec élagage $\alpha\beta$. La racine appartient au joueur 1 qui cherche à maximiser le score.

Q 10) On commence par linéariser l'expression en $(a_1|b_2)^\star a_3$. On calcule $P = \{a_1, b_2, a_3\}$, $S = \{a_3\}$ et $F = \{a_1a_1, a_1b_2, b_2b_2, b_2a_1, a_1a_3, b_2a_3\}$. On construit ensuite l'automate locale suivant où l'on a directement enlevé les marques sur les transitions.

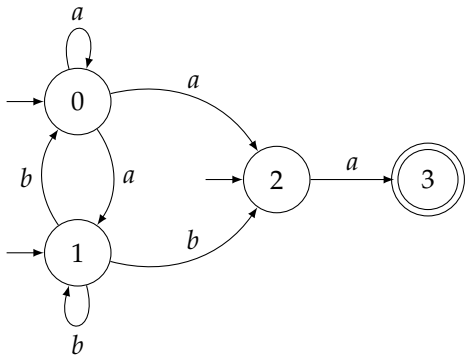


Q 11) On obtient l'automate suivant :

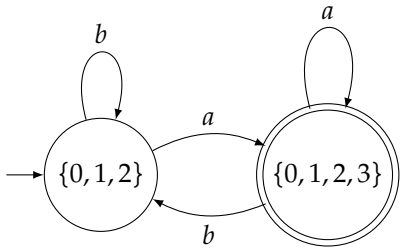
Q 12) On obtient :



Q 13) On obtient :



Q 14) On obtient :



II Le jeu de Nim

Q 15) Le nombre total d'allumettes restantes est un variant pour ce jeux. À chaque coup le nombre d'allumettes décroît strictement (on enlève au moins une allumette dans un tas). Toute partie est donc finie. Toute configuration finale est gagnante pour l'un ou pour l'autre des deux joueurs, il n'y a pas de match nul. Tout état est donc dans l'attracteur d'un des deux joueurs et il en va de même de la configuration initiale qui est donc gagnante pour l'un des deux joueurs. Il existe donc une stratégie gagnante pour l'un ou pour l'autre des deux joueurs.

Q 16) Le premier élève à me signaler qu'il manque le corrigé pour cette question obtient un point de plus au prochain devoir.

Q 17) Dans la variante classique du jeux $\star n$, toute configuration $\star p$ avec $1 \leq p \leq n$ est gagnante pour le joueur qui la contrôle, puisqu'il suffit de prendre toutes les allumettes. Les attracteurs d'un joueurs sont donc exactement les configurations non terminales qu'il contrôle. Dans la variante misère la configuration $\star p$ est perdante pour le joueur qui la contrôle si $p = 1$ et gagnante pour le joueur qui la contrôle si $2 \leq p \leq n$, puisque l'on peut prendre toutes les allumettes sauf une. Les attracteurs d'un joueur sont donc exactement les configurations qu'il contrôle ne comportant pas exactement une allumette.

Q 18) Pour la variante classique le premier joueur possède donc toujours une stratégie gagnante au jeu $\star n$ pour $n \geq 1$. Pour la variante misère le premier joueur possède une stratégie gagnante au jeu $\star n$ si $n \geq 2$ et si $n = 1$ c'est le deuxième joueur qui possède une stratégie gagnante.

Q 19) On propose la stratégie suivante pour le joueur 2 : pour une configuration $\star k + \star p$ avec $1 \leq k < p$, enlever $p - k$ allumettes du deuxième tas ; pour une configuration à un seul tas, prendre toutes les allumettes, pour une autre configuration jouer un coup arbitraire. Montrons par récurrence sur $n \in \mathbb{N}^*$ qu'une configuration $\star n + \star n$ est gagnante pour le joueur 2. Pour $n = 1$, il y a un seul coup possible pour le joueur 1 qui amène dans la configuration $\star 1$ qui est gagnante pour le joueur 2. La configuration $\star 1 + \star 1$ est donc aussi gagnante pour le joueur 2. Supposons le résultat vrai pour $n \in \mathbb{N}^*$ et considérons le jeu $\star(n+1) + \star(n+1)$. Le premièr joueur doit enlever un certain nombre d'allumettes x d'un tas. Si $x = n + 1$, on tombe sur une configuration à un seul tas qui est gagnante pour le joueur qui la contrôle d'après la question précédente. Sinon, on arrive à une la configuration $\star(n+1-x) + \star(n+1)$ avec $n + 1 - x \geq 1$. La stratégie du joueur 2 est d'enlever x allumettes du deuxième tas ce qui amène à la configuration $\star(n+1-x) + \star(n+1-x)$ qui par hypothèse de récurrence est gagnante pour le joueur 2. La configuration $\star(n+1-x) + \star(n+1)$ est donc gagnante pour le joueur 2, et ceci pour tout coup $x \in [\![1, n+1]\!]$ du joueur 1. La configuration $\star(n+1) + \star(n+1)$ est donc gagnante pour le joueur 2. La stratégie proposée est donc gagnante pour le joueur 2.

Q 20) Le première joueur peut enlever $n - m$ allumettes du deuxième tas, ce qui ramène le jeu à une configuration $\star m + \star m$ avec $m \geq 1$ contrôlée par le joueur 2. En suivant la stratégie ci-dessus et en inversant joueur 1 et joueur 2, le premier joueur dispose d'une stratégie gangante à partir de cette position.

Q 21) Déterminer, en justifiant, quel joueur possède une stratégie gagnante dans la variante misère pour les configurations initiales :

- Pour $\star 1 + \star n$ pour $n \geq 1$, le premier joueur peut prendre les n allumettes du deuxième tas et dispose d'une stratégie gagnante ;
- Pour $\star 2 + \star 2$, quel que soit le coup du premier joueur, le deuxième joueur peut ramener le jeu dans la configuration $\star 1$ qui est gagnante pour le deuxième joueur, qui dispose donc d'une stratégie gagnante ;
- Pour la configuration $\star 2 + \star n$ pour $n \geq 3$, le premier joeur peut enlever $n - 2$ allumettes du deuxième tas pour se ramener au cas précédent en échangeant les joueurs et le premier joueur dispose donc d'une stragégie gagnante.

Q 22) Le deuxième joueur possède une stratégie gagante dans la variante misère du jeu $\star m + \star n$ pour $n, m \in \mathbb{N}^*$ si et seulement si $n = m$ et $n \geq 2$. Dans les autres cas c'est le premier joueur qui possède une stratégie gagnante.