Exercice 1. Le monoïde syntaxique $M(L)$ comprend les 3 éléments $L_k = \{a \in \Sigma^* \mid \sum_{i=1}^{|a|} a_i \equiv k \pmod 3\}$ pour $k \in \{0,1,2\}$. Leur produit est $L_{k_1} \cdot L_{k_2} = L_{(k_1+k_2) \bmod 3}$. Le monoide syntaxique $M(L)$ est donc le groupe d'ordre 3.

Tout groupe fini d'ordre supérieur à 2 n'est pas un monoïde apériodique. Donc $M(L)$ n'est pas apériodique qui est équivalent à L n'étant pas sans étoile.

Pour tout mot $w \in \Sigma^*$, on définit trois ensembles de positions $P_k = \{x | \sum_{i=0}^{x-1} \equiv k \pmod 3\}$ pour $k \in \{0,1,2\}$. Ces ensembles peuvent être décrits en MSO par conjonction de trois formules (partout k et j sont dans $\{0,1,2\}$) :

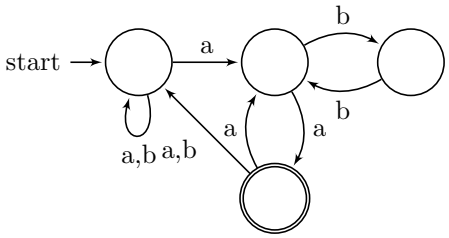
$$\begin{aligned}\text{mutex} =& \forall x \bigwedge_{k \neq j} \neg(x \in P_k \wedge x \in P_j)\\ \text{init} =& P_0(0)\\ \text{succ} =& \forall x \bigwedge_k \bigwedge_j (x \in P_k \wedge w_x = j \implies x+1 \in P_{k+j \mod 3})\end{aligned}$$

Pour que w soit dans L il faut et il suffit que la fin du mot soit dans P_2, ce qui donne la formule recherchée :

$$\text{dansL} = \exists P_1, P_2, P_3 \colon (\text{mutex} \wedge \text{init} \wedge \text{succ} \wedge \text{Fin} \in P_2)$$

Exercice 2. Une expression ω-régulière est $\big(\{a,b\}^* a (bb)^* a\big)^\omega$.

Un automate de Büchi (non déterministe) est le suivant :



Pour construire une formule MSO, on définit les sous-formules

$$\begin{aligned}\text{b2n}(i,j) \quad = \quad & i \leq j \wedge \forall k \in [i,j] : w_k = b \wedge \exists S \colon i \in S \wedge j-1 \in S \wedge \\ & \forall k \in S \colon \big(k+1 \notin S \wedge (k+2 \leq j \implies k+2 \in S)\big)\end{aligned}$$

qui est vraie si et seulement si le sous-mot $w_i \cdots w_j$ est de la forme b^{2n} et

$$\text{ab2na}(i,j) \quad = \quad i \leq j \wedge w_i = a \wedge \text{b2n}(i+1, j-1) \wedge w_j = a$$

qui est vraie si et seulement si le sous-mot $w_i \cdots w_j$ est de la forme $ab^{2n}a$. Il reste de vérifier si cette dernière formule est vraie pour un nombre infini d'intervalles $[i,j]$:

$$\text{dans\_L} \quad = \quad \forall k \exists i,j \colon \big(k < i < j \wedge \text{ab2na}(i,j)\big)$$

Exercice 3. On considère d'abord le cas où A **est fini.** Dans ce cas il est bien sûr décidable. Soit $a_0, a_1, \ldots a_n$ ses éléments dans l'ordre croissant. Alors on définit la fonction g croissante avec $Im(g) = A$ comme suit :

$$g(i) = \begin{cases} a_i, & \text{si} i < n \\ a_n, & \text{sinon.} \end{cases}$$

Cette définition par cas montre que g est récursive primitive ce qui termine la preuve pour ce cas.

Soit maintenant A **infini.** S'il est décidable, sa fonction caractéristique χ_A est récursive totale. On définira la fonction g comme suit :

$$g(i) = \mu x.(x \geq i \wedge \chi_A(x) = 1).$$

On observe que

– par construction elle est récursive partielle ;
– puisque A n'est pas bornée elle est totale ;
– par construction elle est croissante ;
– pour chaque $i \in A$ on a que $g(i) = i$, donc $Im(g) \supset A$;
– toujours $\chi_A(g(i)) = 1$, donc $Im(g) \subset A$.

donc on a toutes les propriétés requises.

Réciproquement, si un ensemble infini $A = Im(g)$ avec g totale et croissante, alors

$$\chi_A(x) = \begin{cases} 1, & \text{si } g(\mu i.g(i) \geq x) = x \\ 0, & \text{sinon.} \end{cases}$$

et le μ dans cette formule est toujours défini. Donc χ_A est récursive totale, donc A décidable.

(Informellement, pour décider si $x \in A$ on énumère les éléments de A jusqu'à ce qu'on tombe sur x - dans ce cas on dit OUI, ou on dépasse x et la réponse est NON).

Exercice 4. On montre que M n'est pas hors contexte en utilisant le lemme de pompage. Cela implique a fortiori que M n'est pas régulier.

On suppose donc que M est hors contexte. D'après le lemme de pompage, il existe un entier positif N tel que tout mot $w \in M$ de longueur au moins N peut s'écrire comme $w = xuyvz$ avec $|uyv| \leq N$ et $uv \neq \epsilon$ tel que $xu^n yv^n z \in M$ pour tout $n \geq 0$. On utilise le mot $w = a^N b^N a^N b^N \in M$. Soit $w = xuyvz$ la décomposition selon le lemme de pompage. Il est évident que $u, v \in a^* \cup b^*$ car $xu^2yv^2z \notin M$ sinon. On distingue les cas suivants :

1. $u, v \in a^*$: On a soit $xu^2yv^2z = a^{N+|u|+|v|}b^N a^N b^N$, soit $xu^2yv^2z = a^N b^N a^{N+|u|+|v|}b^N$ et donc $xu^2yv^2z \notin M$. Contradiction.
2. $u, v \in b^*$: On arrive à une contradiction dans la même manière que dans le cas 1.
3. $u \in b^*$ et $v \in a^*$: Nécessairement, $xu^2yv^2z = a^N b^{N+|u|}a^{N+|v|}b^N$ qui n'est pas dans M. Contradiction.
4. $u \in a^*$ et $v \in b^*$: Dans ce cas, soit $xu^2yv^2z = a^{N+|u|}b^{N+|v|}a^N b^N$, soit $xu^2yv^2z = a^N b^N a^{N+|u|}b^{N+|v|}$, dont aucun est dans M. Contradiction.

On a donc montré que M n'est pas hors contexte.

La grammaire suivante décrit $\bar{M}$:

$$\begin{aligned} \text{Start} &\to \text{Impair} \mid \text{Pair} \\ \text{Impair} &\to \text{Centre}_a \mid \text{Centre}_b \\ \text{Pair} &\to \text{Centre}_a\, \text{Centre}_b \mid \text{Centre}_b\, \text{Centre}_a \\ \text{Centre}_a &\to a \mid a\, \text{Centre}_a\, a \mid a\, \text{Centre}_a\, b \mid b\, \text{Centre}_a\, a \mid b\, \text{Centre}_a\, b \\ \text{Centre}_b &\to b \mid a\, \text{Centre}_b\, a \mid a\, \text{Centre}_b\, b \mid b\, \text{Centre}_b\, a \mid b\, \text{Centre}_b\, b \end{aligned}$$

Il est évident que $w \in \bar{M}$ si $|w|$ est impair. Pour tout mot z de longueur impair, soit centre(z) la lettre au centre de z, c'est-à-dire la lettre $z_{(|z|+1)/2}$.

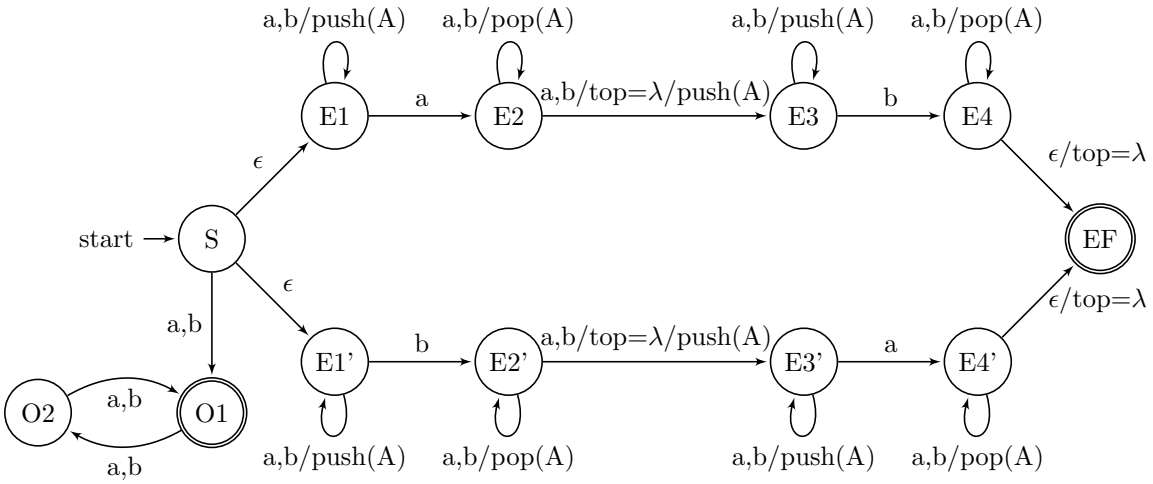
Pour un mot w de longueur paire appelons un couple (x, y) une décomposition impaire de si $w = xy$ et les deux longueurs $|x|$ et $|y|$ sont impaires.

On montre qu'un mot w de longueur paire $2n$ est dans $\bar{M}$ si et seulement s'il existe une décomposition impaire (x, y) de w telle que centre(x) $\neq$ centre(y) (voir le dessin de l'énoncé).

Le point crucial est de noter que pour toute décomposition impaire (x, y) de w avec $|x| = 2r - 1$ on a centre(x) $= w_r$ et centre(y) $= w_{r+n}$ (donc les centres de x et de y ont la même position dans les deux moitiés de w) Effectivement $x = w_1 \dots w_{2r-1}$ et $y = w_{2r} \dots w_{2n}$ et les indices des centres de x et de y sont respectivement

$$\frac{1 + (2r - 1)}{2} = r \text{ et } \frac{2r + 2n}{2} = r + n.$$

Un automate à pile pour $\bar{M}$ (avec λ symbole du bas de pile) est :



Exercice 5, question 1.

Solution 1. On construit d'abord un automate non-déterministe reconnaissant le langage défini par e (comme dans la preuve du théorème de Kleene). Puis on utilise le parcours de graphe pour s'assurer qu'il n'y a pas de chemin d'un état initial vers un état final – dans ce cas le langage est vide. La taille de l'automate, la complexité de sa construction, et la complexité de la recherche de chemin sont polynomiales, donc EMPTY $\in$ P.

Solution 2. L'algorithme récursif suivant résoud le problème :

$$\begin{aligned} Vide(a) &= faux \\ Vide(\varepsilon) &= faux \\ Vide(\emptyset) &= vrai \\ Vide(f+g) &= Vide(f) \wedge Vide(g) \\ Vide(f \cdot g) &= Vide(f) \vee Vide(g) \\ Vide(f^*) &= faux \end{aligned}$$

(le dernier cas s'explique par $\varepsilon \in f^*$). Pour implémenter cet algorithme on construira l'arbre syntaxique de e et remontera des feuilles vers la racine en temps polynomial.

Exercice 5, question 2. On remarque d'abord que pour une expression e de taille n, la taille de l'automate équivalent non-déterministe est polynomiale en n, tandis que celle de l'automate déterministe est exponentielle. Le mot le plus court qui n'est pas dans le langage ne dépasse pas le diamètre de l'automate déterministe mais peut aussi avoir la taille exponentielle. On conclut donc, que l'algorithme de classe NSPACE

- devrait utiliser l'automate non-déterministe ;
- ne devrait pas écrire un mot w n'appartenant pas au langage.

Malheureusement dans des nombreuses copies il y avait des erreurs sur ces deux points.

On décrit maintenant un algorithme NSPACE pour le problème ¬UNIVERSAL. Soit e une expression. On construit d'abord un automate non-déterministe A (de taille polynomiale) qui reconnaît son langage, soit I son ensemble d'états initiaux et F états finals. On essaye de deviner un mot non reconnu par l'automate (on ne l'écrit pas, sa taille ne pose pas de problème).

```
E=I
while (E intersects F)
    a= guess any letter
    E=Succ(E,a)
// we exit from the loop whenever E is disjoint from F
return true
```

Ce programme nondéterministe a besoin d'espace polynomial (pour stocker l'automate et l'ensemble M des états atteints). Montrons qu'il est correct.

- Si $e \in \neg$UNIVERSAL, alors il existe un mot w qui n'est pas dans le langage. En devinant successivement les lettres de ce mot, on arrivera à un ensemble E qui ne contient aucun état final. Un tel calcul de notre programme renverra true.
- Sinon (si $e \in$ UNIVERSAL) tous les mots sont acceptés par A et pour toute séquence de lettres devinées M contiendra un état final, le programme ne s'arrêtera jamais.

(petit exercice : comment modifier ce programme non-déterministe pour qu'il s'arrête toujours ?)

On a montré que $\neg$UNIVERSAL est dans NSPACE, ce dernier coïncide avec PSPACE et est clos par complément. Donc UNIVERSAL $\in$ PSPACE.

Exercice 5, question 3.

Soit $L \subset \Sigma^*$ un langage de la classe PSPACE, reconnu en espace $p(n)$ (où p est un polynôme) par une machine de Turing M avec états Q et un seul ruban. Soit w un mot de n lettres. Pour réduire L à $\neg$UNIVERSAL, on écrira une expression régulière e de taille polynomiale en n qui est non-universelle ssi $w \in L$. Informellement, e décrira tous les mots qui ne correspondent pas au calculs accepteurs de M sur w.

On prend l'alphabet $\Gamma = \Sigma \cup Q \cup \{\#, \$, \llcorner\}$.

Soit $s = p(n)$ l'espace suffisant pour le calcul sur w. On peut donc faire le calcul sur le ruban fini de $2s$ cases en plaçant initialement la tête sur la case numéro $s+1$. On décrit le calcul de longueur t par une séquence

$$\sigma = \$\alpha_1 q_1 \beta_1 \# \alpha_2 q_1 \beta_2 \# \dots \# \alpha_t q_1 \beta_t \# \$$$

(t peut être très grand et cette séquence très longue). Ici les q_i sont des états successifs de la machine, α_i, β_i le contenu du ruban à gauche et à droite de la tête.

On rappelle que w et S sont fixés. Soit l'expression $e =$ `struct` + `init` + `accept` + `instr_tête` + `copie_ruban` avec les sous-expressions :

- `struct` les mots qui n'ont pas la bonne structure : $\$\alpha_1 q_1 \beta_1 \# \alpha_2 q_1 \beta_2 \# \dots \# \alpha_t q_1 \beta_t \# \$$ avec $\alpha, \beta \in (\Sigma \cup \{\llcorner\})^*$ avec $|\alpha_i| + |\beta_i| = 2s$ et $q_i \in Q$;
- `init` les mots qui ne commencent pas par la configuration initiale $\$\llcorner^s i w \llcorner^{s-n} \#$ avec i l'état initial de la machine ;
- `accept` les mots où dans la dernière configuration l'état accepteur de M n'apparaît pas ;
- `instr_tête` les mots où sur deux configuration consécutives l'état q ou le symbole observé ne changent pas comme prévu par le programme de M.
- `copie_ruban` les mots où sur deux configuration consécutives le ruban n'est pas recopié correctement.

Pour chacun de ces cas il s'agit de longues expressions (de taille $O(s)$) qui traitent chaque position du ruban de $-s$ à s séparément.

On a donc que $w \in L$ ssi il existe un calcul accepteur de M sur w en espace s ssi il existe un mot σ qui décrit un tel calcul ssi l'expression e n'est pas universelle. Donc $L \prec \neg$UNIVERSAL.

Pour passer à UNIVERSAL on peut raisonner comme ceci : soit $L \in$ PSPACE, alors $\neg L$ aussi, et donc (comme on vient de prouver) $\neg L \prec \neg$UNIVERSAL. Par la même réduction $L \prec$ UNIVERSAL cqfd.

Exercice 5, question 4.

- $f \cap e = \emptyset$. On construit un automate (non-déterministe) pour chaque expression, puis l'automate produit pour $f \cap e$, et on teste si son langage est vide comme dans 5.1. Cela donne un algorithme polynomial, le problème est donc dans P.
- $f = e$. Un algorithme NSPACEpour le problème $f \neq e$ devine lettre à lettre un mot qui est dans f mais pas dans e et vérifie que c'est vraiment le cas comme dans 5.2. Puisque NSPACEcoïncide avec PSPACEet est clos par complément, on peut conclure que $(f = e) \in$PSPACE.
  Pour prouver que ce problème est PSPACE-complet, on réduira UNIVERSALà ce problème : e est universel ssi $\Sigma^* = e$.
- $f \subset e$. Le même raisonnement que dans le point précédent montre que ce problème est PSPACE-complet