Mémoire

Auteur : Testa, Coralie
Promoteur(s) : Massuir, Adeline
Faculté : Faculté des Sciences
Diplôme : Master en sciences mathématiques, à finalité approfondie
Année académique : 2021-2022
URI/URL : http://hdl.handle.net/2268.2/14656

Université de Liège

Faculté des Sciences

Département de Mathématique

La complexité en états d'opérations régulières

Auteur :
Coralie Testa

Promotrice :
Adeline Massuir

Mémoire de fin d'études en vue de l'obtention du diplôme de Master en Sciences Mathématiques, à finalité approfondie

Année académique 2021 – 2022

Remerciements

Je tiens avant tout à remercier ma promotrice madame Adeline Massuir. Merci pour son aide précieuse, son encadrement, ses relectures et tous ses conseils avisés.

Un grand merci à Alan Heye pour son soutien et ses encouragements qui m'ont permis de ne jamais abandonner.

Merci à ma famille qui a toujours été là pour moi et qui a cru en mes capacités.

Enfin, je tiens à remercier mesdames E. Charlier, M. Stipulanti, C. ESSER d'avoir accepté de faire partie de mon jury.

Table des matières

Introduction

L'étude de la complexité en états a débuté en 1970 avec l'article [12] dans lequel Maslov a donné la valeur de la complexité en états de certaines opérations (la racine carrée, le déplacement cyclique et l'élimination proportionnelle) mais n'a pas fourni de preuves. Suite à un renouvellement de l'intérêt pour les langages formels au début des années 1990, Yu, Zhuang et Salomaa [15] ont poursuivi l'étude. Après ça, de nombreux autres documents ont été rédigés à propos de la complexité en états. Plusieurs sous-domaines ont été créés en fonction que les automates utilisés soient déterministes ou non, que les langages soient finis ou infinis, etc. Nous allons nous intéresser au cas des automates déterministes pour n'importe quel langage.

La complexité en états d'un langage régulier est la taille de son automate minimal et la complexité en états d'une opération régulière est la plus grande complexité en états de langages obtenus en appliquant cette opération sur des langages de complexité en états fixée. Ainsi, pour calculer la complexité en états, l'approche générale est de calculer une borne supérieure à partir des caractéristiques de l'opération considérée et de fournir un témoin, c'est-à-dire un exemple spécifique atteignant la borne qui devient alors la complexité en états recherchée.

Ainsi, la complexité en états de beaucoup d'opérations unaires et binaires a été déterminée de cette façon, comme par exemple dans [10], [14] et [6]. Dernièrement, la complexité en états de combinaisons d'opérations a été étudiée (et dans la plupart des cas ce n'est pas simplement une composition des complexités individuelles) comme par exemple dans [11] et [7].

Nous allons mettre en lumière les idées de deux articles récents [2] et [3] ; tous deux écrits par Pascal Caron, Edwin Hamel-De le Court, Jean-Gabriel Luque et Bruno Patrou (ce dernier n'a contribué qu'au premier article).

Ce document est organisé de la façon suivante. Le premier chapitre rappelle les notions importantes de la théorie des langages formels et fournit les définitions de base ainsi que les notations utilisées par la suite.

Dans le chapitre 2, les opérations 1-uniformes sont définies. Nous allons voir qu'on peut

calculer la complexité en états d'opérations régulières en faisant des calculs directement sur des AFDs. Ainsi, il est pratique de lier ces opérations sur des langages avec des opérations agissant directement sur des AFDs qui sont appelées modificateurs. Les opérations qui peuvent être décrites par un modificateur sont dites descriptibles. Divers modificateurs d'opérations classiques sont présentés et des exemples d'applications sont donnés. Nous présenterons les modificateurs 1-uniformes qui sont des modificateurs qui peuvent être associés à une opération régulière. Ces modificateurs se comportent bien par rapport à la composition.

Dans les deux chapitres suivants, nous allons définir des AFDs particuliers, les monstres. On les appelle "les monstres" car ce sont des AFDs avec un très grand alphabet. Leur alphabet est composé de toutes les fonctions de transitions possibles. Nous allons utiliser les monstres pour montrer la correspondance entre les opérations et les modificateurs 1-uniformes. Il se trouve qu'un modificateur 1-uniforme décrit toujours une opération 1-uniforme, et chaque opération 1-uniforme est décrite par un modificateur 1-uniforme. Ainsi, chaque opération 1-uniforme correspond à une construction sur des AFDs. Finalement, un résultat majeur sera démontré qui permettra de concevoir une méthode pour calculer la complexité en états des opérations descriptibles en utilisant les monstres. Pour calculer la complexité en états d'une opération régulière, les monstres sont de bons candidats pour être des témoins. Après ça, nous appliquerons cette méthode pour calculer la complexité en états de l'étoile, la concaténation, l'étoile de l'intersection et la racine carrée.

La suite du document se penchera particulièrement sur l'article [3]. Dans le chapitre 5, les modificateurs amicaux vont être présentés. A tout modificateur 1-uniforme amical est associé un modificateur standard qui est un autre modificateur décrivant la même opération. Afin de montrer une propriété de régularité sur les états finaux d'un modificateur amical standard 1-uniforme, la suite caractéristique sera définie. En fait, une fonction caractéristique est associée à chaque état de l'automate de sortie. Un résultat important de ce document est celui de la correspondance entre les modificateurs amicaux standards et les opérations amicales obtenues en combinant des racines et des opérations booléennes. Pour finir, nous calculerons la complexité en états maximale d'opérations amicales, en fonction de leur arité dans le Chapitre 6.

Chapitre 1

Préliminaires

Dans ce chapitre, nous allons rappeler les notions importantes de la théorie des langages formels, définir la complexité en états et fournir les notations utilisées dans la suite du document. Des explications plus complètes à propos des langages formels et des automates se trouvent dans [13].

1.1 Mots et langages

Définition 1.1.1 (alphabet)**.** Un alphabet est un ensemble fini.

Définition 1.1.2 (mot)**.** Soit Σ un alphabet. Un mot sur Σ est une suite finie et ordonnée de symboles. La longueur d'un mot w est le nombre de symboles constituant ce mot ; on la note $|w|$. L'unique mot de longueur 0 est le mot correspondant à la suite vide. Ce mot s'appelle le mot vide et on le note ε. L'ensemble des mots sur Σ est noté Σ^*.

Définition 1.1.3. Si σ est une lettre de l'alphabet Σ, pour tout mot $w = w_1 \cdots w_l \in \Sigma^*$, on dénote par

$$|w|_\sigma = \#\{i \in \{1, \ldots, l\} | w_i = \sigma\}$$

le nombre de lettre σ apparaissant dans le mot w.

Définition 1.1.4 (préfixe)**.** Soit $w = w_1 \cdots w_l$ un mot sur Σ. Les mots

$$\varepsilon, w_1, w_1 w_2, \ldots, w_1 \cdots w_{l-1}, w_1 \cdots w_l = w$$

sont les préfixes de w. L'ensemble des préfixes de w est noté $\mathrm{Pref}(w)$.

Définition 1.1.5 (monoïde)**.** Soient A un ensemble et $\circ : A \times A \to A$ une opération binaire interne et partout définie. L'ensemble A muni de l'opération $\circ$ possède une structure de monoïde si les propriétés suivantes sont satisfaites :

— L'opération $\circ$ est associative :

$$\forall x, y, z \in A : (x \circ y) \circ z = x \circ (y \circ z).$$

— Il existe un neutre unique $e \in A$ tel que

$$\forall x \in A : x \circ e = e \circ x = x.$$

Définition 1.1.6 (concaténation)**.** Soit Σ un alphabet. On définit l'opération de concaténation sur Σ^* de la façon suivante. Pour tous mots $u = u_1 \cdots u_k$ et $v = v_1 \cdots v_l, u_i, v_i \in \Sigma$, la concaténation de u et v, notée $u \cdot v$ ou simplement uv, est le mot

$$w = w_1 \cdots w_{k+l} \text{ où } \begin{cases} w_i = u_i & , 1 \leq i \leq k \\ w_{k+i} = v_i & , 1 \leq i \leq l \end{cases}$$

Ainsi, Σ^* muni de l'opération de concaténation est un monoïde de neutre ε. En particulier, on définit la puissance n-ième d'un mot w comme la concaténation de n copies de w,

$$w^n = \underbrace{w \cdots w}_{n \text{ fois}}.$$

On pose $w^0 = \varepsilon$.

Définition 1.1.7 (langage)**.** Un langage sur Σ est simplement un ensemble (fini ou infini) de mots sur Σ. En d'autres termes, un langage est une partie de Σ^*. On distingue en particulier le langage vide $\emptyset$.

Passons à présent en revue quelques opérations sur les langages. Tout d'abord, puisqu'un langage est un ensemble, on dispose des opérations ensemblistes usuelles comme le complémentaire, l'union, l'intersection ou encore l'union disjointe.

Définition 1.1.8 (complémentaire)**.** Soit $L \subseteq \Sigma^*$. Le complémentaire de L est donné par

$$L^C = \{w \in \Sigma^* | w \notin L\}.$$

Définition 1.1.9 (union)**.** Soient $L, M \subseteq \Sigma^*$. L'union des langages L et M est donnée par

$$L \cup M = \{w \in \Sigma^* | w \in L \vee w \in M\}.$$

Définition 1.1.10 (intersection)**.** Soient $L, M \subseteq \Sigma^*$. L'intersection des langages L et M est donnée par

$$L \cap M = \{w \in \Sigma^* | w \in L \wedge w \in M\}.$$

Définition 1.1.11 (union disjointe)**.** Soient $L, M \subseteq \Sigma^*$. L'union disjointe des langages L et M est donnée par

$$\mathrm{Xor}(L, M) = \{w \in \Sigma^* | (w \in L \wedge w \notin L') \vee (w \notin L \wedge w \in L')\}.$$

On dispose également des opérations préfine, concaténation, miroir, étoile de Kleene, racine n-ème et quotient à droite.

Définition 1.1.12 (préfine)**.** Soit $L \subseteq \Sigma^*$. Le préfine de L est donné par

$$\mathrm{Prefin}(L) = \{w = uv \in \Sigma^* | u \in L, v \in \Sigma^*\}.$$

Définition 1.1.13 (concaténation)**.** Soient $L, M \subseteq \Sigma^*$ deux langages. La concaténation des langages L et M est le langage

$$L \cdot M = \{uv | u \in L, v \in M\}.$$

En particulier, on peut définir la puissance n-ième d'un langage L, $n > 0$, par

$$L^n = \{w_1 \cdots w_n | \forall i \in \{1, \ldots, n\}, w_i \in L\}$$

et on pose $L^0 = \{\varepsilon\}$.

Notation 1.1.1. Soit $n \geq 0$. L'ensemble des mots de longueur n sur Σ est Σ^n.

Définition 1.1.14 (étoile de Kleene)**.** Soit $L \subseteq \Sigma^*$. L'étoile de Kleene de L est donnée par

$$L^* = \bigcup_{i \geq 0} L^i.$$

Ainsi, les mots de Star(L) sont exactement les mots obtenus en concaténant un nombre arbitraire de mots de L.

Définition 1.1.15 (opération miroir)**.** L'opération miroir est définie par récurrence sur la longueur de w de la façon suivante : si $|w| = 0$, alors $w = \varepsilon$ et $w^R = \varepsilon$, sinon $|w| > 0$ et $w = \sigma u, \sigma \in \Sigma, u \in \Sigma^*$ et $w^R = u^R\sigma$.

Définition 1.1.16 (miroir)**.** Le miroir d'un langage L est

$$L^R = \{u^R | u \in L\}.$$

Définition 1.1.17 (racine)**.** Pour tout $n \in \mathbb{N}$, nous définissons la racine n-ème du langage L par

$$\sqrt[n]{L} = \{w \in \Sigma^* | w^n \in L\}.$$

Remarquons que $\sqrt[0]{L} = \Sigma^*$ si $\varepsilon \in L$ et $\emptyset$ sinon et $\sqrt[1]{L} = L$. Par convention, on écrit $\sqrt{\ }$ pour $\sqrt[2]{\ }$.

Définition 1.1.18 (quotient à droite)**.** Soient $L, M \subseteq \Sigma^*$. Le quotient à droite de L et M est donné par

$$L \cdot M^{-1} = \{u \in \Sigma^* | uv \in L \text{ pour un certain } v \in M\}.$$

Définition 1.1.19 (expression régulière)**.** Soit Σ un alphabet. Supposons que $0, e, +, \cdot, (,), *$ sont des symboles n'appartenant pas à Σ. L'ensemble R_Σ des expressions régulières sur Σ est défini récursivement par

- 0 et e appartiennent à R_Σ,
- $\forall \sigma \in \Sigma$, σ appartient à R_Σ,
- si ϕ et ψ appartiennent à R_Σ, alors
  - $(\phi + \psi)$ appartient à R_Σ,
  - $(\phi \cdot \psi)$ appartient à R_Σ ;

— ϕ^* appartient à R_Σ.

Notation 1.1.2. Soit Q un ensemble. On note 2^Q l'ensemble des parties de Q.

Remarque 1.1.1. Soit Σ un alphabet, 2^{Σ^*} est l'ensemble des langages sur Σ.

A une expression régulière, on associe un langage grâce à l'application

$$L : R_\Sigma \to 2^{\Sigma^*}$$

par

— $L(0) = \emptyset, L(e) = \{\varepsilon\}$,
— si $\sigma \in \Sigma$, alors $L(\sigma) = \{\sigma\}$,
— si ϕ et ψ sont des expressions régulières,
  — $L(\phi + \psi) = L(\phi) \cup L(\psi)$,
  — $L(\phi \cdot \psi) = L(\phi)L(\psi)$,
  — $L(\phi^*) = (L(\phi))^*$.

Définition 1.1.20 (langage régulier)**.** Un langage L sur Σ est régulier s'il existe une expression régulière $\phi \in R_\Sigma$ telle que

$$L = L(\phi).$$

Si ϕ et ψ sont deux expressions régulières telles que $L(\phi) = L(\psi)$, alors on dit que ϕ et ψ sont équivalentes.

1.2 Automates

Définition 1.2.1 (AFD)**.** Un automate fini déterministe (ou AFD) est la donnée d'un quintuple

$$A = (Q, q_0, F, \Sigma, \delta)$$

où

— Q est un ensemble fini dont les éléments sont les états de A,
— $q_0 \in Q$ est un état privilégié appelé état initial,
— $F \subseteq Q$ désigne l'ensemble des états finals,
— Σ est l'alphabet de l'automate,

— $\delta : Q \times \Sigma \to Q$ est la fonction de transition de A.

Notation 1.2.1. Pour tout AFD $A = (Q, i, F, \Sigma, \delta)$, tout état $q \in Q$ et tout lettre $a \in \Sigma$, $\delta^a(q)$ signifie $\delta(q, a)$.

Définition 1.2.2 (AFDC)**.** Si $A = (Q, q_0, F, \Sigma, \delta)$ est un AFD et si δ est une fonction totale, i.e., δ est défini pour tout couple $(q, \sigma) \in Q \times \Sigma$ alors on dit que A est un automate fini déterministe complet (AFDC).

Définition 1.2.3 (langage accepté)**.** Soit $A = (Q, q_0, F, \Sigma, \delta)$ un AFD. On étend naturellement la fonction de transition δ à $Q \times \Sigma^*$ de la manière suivante :

$$\delta(q, \varepsilon) = q$$

et

$$\delta(q, \sigma w) = \delta(\delta(q, \sigma), w), \sigma \in \Sigma, w \in \Sigma^*.$$

Le langage accepté par A est alors

$$L(A) = \{w \in \Sigma^* | \delta(q_0, w) \in F\}.$$

Si $w \in L(A)$, on dit encore que A accepte le mot w (ou que w est accepté par A).

Définition 1.2.4 (AFND)**.** Un automate fini non déterministe (AFND) est la donnée d'un quintuple

$$A = (Q, I, F, \Sigma, \Delta)$$

où

— Q est un ensemble fini dont les éléments sont les états de A,
— $I \subseteq Q$ est l'ensemble des états initiaux,
— $F \subseteq Q$ désigne l'ensemble des états finals,
— Σ est l'alphabet de l'automate,
— $\Delta \subset Q \times \Sigma^* \times Q$ est une relation de transition.

Définition 1.2.5 (langage accepté)**.** Un mot $w = w_1 \cdots w_k$ est accepté par un AFND $A = (Q, I, F, \Sigma, \Delta)$ s'il existe $q_0 \in I, l \in \mathbb{N} \setminus \{0\}, v_1, \ldots, v_l \in \Sigma^*, q_1, \ldots, q_l \in Q$ tels que

$$(q_0, v_1, q_1), (q_1, v_2, q_2), \ldots, (q_{l-1}, v_l, q_l) \in \Delta,$$

$$w = v_1 \cdots v_l \text{ et } q_l \in F.$$

Le langage accepté par un AFND A est l'ensemble des mots acceptés par A et se note encore $L(A)$. Enfin, deux AFND A et B sont dits équivalents si $L(A) = L(B)$.

Théorème 1.2.1 (Kleene)**.** Un langage est régulier si et seulement si il est accepté par un automate fini déterministe.

1.3 Automate minimal

Un automate fini déterministe est minimal s'il n'existe pas d'AFD équivalent avec moins d'états.

Définition 1.3.1. Soit $L \subseteq \Sigma^*$ un langage arbitraire. Si w est un mot sur Σ, on dénote par $w^{-1} \cdot L$ l'ensemble des mots qui, concaténés avec w, appartiennent à L, i.e.,

$$w^{-1} \cdot L = \{u \in \Sigma^* | wu \in L\}.$$

On définit une relation sur Σ^*, notée $\sim_L$, de la manière suivante. Pour tous $x, y \in \Sigma^*$,

$$x \sim_L y \Leftrightarrow x^{-1} \cdot L = y^{-1} \cdot L.$$

En d'autres termes, $x \sim_L y$ si et seulement si pour tout $w \in \Sigma^*$, $xw \in L \Leftrightarrow yw \in L$.

Remarque 1.3.1. On parle souvent pour $\sim_L$ de la congruence de Nérode. On note $[w]_L$ la classe d'équivalence du mot w pour la relation $\sim_L$,

$$[w]_L = \{u \in \Sigma^* | u \sim_L w\}.$$

Définition 1.3.2. Dans le cas d'un automate déterministe $A = (Q, q_0, F, \Sigma, \delta)$, par analogie avec la notation $w^{-1} \cdot L$, on utilise la notation suivante. Si $q \in Q$ est un état de A et si $G \subseteq Q$ est un sous-ensemble d'états, on note $q^{-1} \cdot G$, l'ensemble des mots qui sont labels des chemins débutant en q et aboutissant dans un état de G, i.e.,

$$q^{-1} \cdot G = \{w \in \Sigma^* | \delta(q, w) \in G\}$$

On définit sur Q une relation d'équivalence comme suit : si $p, q \in Q$, alors

$$p \sim_A q \Leftrightarrow p^{-1} \cdot F = q^{-1} \cdot F.$$

Remarque 1.3.2. Avec la notation que nous venons d'introduire, le langage accepté par l'automate déterministe $A = (Q, q_0, F, \Sigma, \delta)$ est simplement

$$q_0^{-1} \cdot F.$$

Définition 1.3.3 (automate minimal)**.** On définit l'automate minimal

$$A_L = (Q_L, q_{0,L}, F_L, \Sigma, \delta_L)$$

d'un langage $L \subseteq \Sigma^*$ comme suit :

— $Q_L = \{w^{-1} \cdot L | w \in \Sigma^*\}$,

— $q_{0,L} = \varepsilon^{-1} \cdot L = L$,

— $F_L = \{w^{-1} \cdot L | w \in L\} = \{q \in Q_L | \varepsilon \in q\}$,

— $\delta_L(q, \sigma) = \sigma^{-1} \cdot q$, pour tous $q \in Q_L, \sigma \in \Sigma$.

La fonction de transition de l'automate s'étend à $Q_L \times \Sigma^*$ par

$$\delta_L(q, w) = w^{-1} \cdot q, \forall q \in Q_L, w \in \Sigma^*.$$

Remarque 1.3.3. Au vu de la définition de $\sim_L$, il est clair que l'ensemble des états de A, $\{w^{-1} \cdot L | w \in \Sigma^*\}$, est en bijection avec l'ensemble quotient $\{[w]_L | w \in \Sigma^*\}$. En effet, à chaque classe d'équivalence $[w]_L$ pour $\sim_L$ correspond un état de $w^{-1} \cdot L$ de l'automate minimal A_L et réciproquement. C'est pour cette raison que, dans la littérature, on trouve également une définition de l'automate minimal en termes des classes d'équivalence de $\sim_L$. Ainsi, on aurait pu définir l'automate minimal comme suit :

— $Q_L = \{[w]_L | w \in \Sigma^*\}$

— $q_{0,L} = [\varepsilon]_L$

— $F_L = \{[w]_L | w \in L\}$

— $\delta_L([w]_L, \sigma) = [w\sigma]_L$.

Remarque 1.3.4. Pour tout AFD, il existe un unique automate minimal équivalent (à un renommage des états près).

Définition 1.3.4 (accessible)**.** Un automate déterministe $A = (Q, q_0, F, \Sigma, \delta)$ est accessible si pour tout état $q \in Q$, il existe un mot $w \in \Sigma^*$ tel que $\delta(q_0, w) = q$.

Définition 1.3.5 (réduit)**.** Un automate déterministe $A = (Q, q_0, F, \Sigma, \delta)$ est réduit si pour tous $p, q \in Q$

$$p^{-1} \cdot F = q^{-1} \cdot F \text{ entraîne } p = q.$$

En d'autres termes, un AFD est réduit, si les langages acceptés depuis deux états distincts sont distincts ou encore si chaque classe d'équivalence pour la relation $\sim_A$ sur Q est un singleton.

Définition 1.3.6 (distinguables, équivalents)**.** Par définition de la relation $\sim_A$ sur Q, l'automate est réduit si pour tout couple (p, q) d'états avec $p \neq q$,

$$p \not\sim_A q.$$

En particulier, $p \not\sim_A q$ s'il existe un mot $w \in \Sigma^*$ tel que

$$\delta(p, w) \in F \text{ et } \delta(q, w) \notin F$$

ou

$$\delta(p, w) \notin F \text{ et } \delta(q, w) \in F.$$

On dit alors que les états p et q sont distinguables. Si deux états ne sont pas distinguables, alors ils sont équivalents.

Proposition 1.3.1. *Soit $L \subseteq \Sigma^*$ un langage. Un automate est minimal si et seulement si il est accessible et réduit.*

1.4 Opérations sur des langages et complexité en états

Une opération k-aire sur des langages est une application envoyant tout k-uplet de langages définis sur le même alphabet sur un langage du même alphabet que sa préimage. Une opération k-aire est régulière si elle envoie chaque k-uplet de langages réguliers sur un langage régulier.

La complexité en états d'un langage régulier L, notée $sc(L)$, est le nombre d'états de son automate fini déterministe minimal. Cette notion s'étend aux opérations régulières. En effet, la complexité en états d'une opération unaire régulière $\otimes$ est la fonction $sc_\otimes$ telle que pour tout $n \in \mathbb{N}\setminus\{0\}$, $sc_\otimes(n)$ est le maximum de toutes les complexités d'état de $\otimes(L)$ où L est de complexité en états n, c'est-à-dire $sc_\otimes$(n)=max$\{sc(\otimes(L))|sc(L) = n\}$. Plus généralement, la complexité en états d'une opération k-aire $\otimes$ est la fonction k-aire $sc_\otimes$ qui associe à tout $(n_1, \ldots, n_k) \in (\mathbb{N}\setminus\{0\})^k$ l'entier

$$sc_\otimes(n_1, \ldots, n_k) = max\{sc(\otimes(L_1, \ldots, L_k))|\forall i \in \{1, \ldots, k\}, sc(L_i) = n_i\}.$$

Il s'agit de la complexité d'états dans le pire cas.

Par exemple, la complexité en états de l'union de deux langages réguliers qui ont des complexités en états égales respectivement à m et n vaut $m \cdot n$. Autrement dit, $sc_{\cup}(m, n) = m \cdot n$. Cela signifie que pour toute paire de langages réguliers avec les complexités en états m et n, leur union est acceptée par un AFD minimal possédant au plus $m \cdot n$ états.

Remarque 1.4.1. Pour information, un tableau reprenant les complexités en états d'opérations basiques sur des langages réguliers se trouve dans [6].

Un témoin pour $\otimes$ est une façon d'assigner à chaque $(n_1, \ldots, n_k)$, (supposé suffisamment grand) un k-uplet de langages $(L_1, \ldots, L_k)$ sur les mêmes alphabets avec $sc(L_i) = n_i \forall i \in \{1, \ldots, k\}$, tel que $sc_{\otimes}(n_1, \ldots, n_k) = sc(\otimes(L_1, \ldots, L_k))$.

Par exemple, un témoin pour l'union de deux langages réguliers est une paire de langages réguliers avec les complexités en états m et n tel que leur union est acceptée par un AFD minimal à exactement $m \cdot n$ états.

1.5 Morphismes

Définition 1.5.1. Soient Σ et Γ deux alphabets. Un morphisme est une fonction ϕ de Σ^* dans Γ^* telle que, pour tous $w, v \in \Sigma^*, \phi(wv) = \phi(w)\phi(v)$.

Remarquons que ϕ est complètement défini par sa valeur sur les lettres car $\phi(\varepsilon) = \varepsilon$.

Un morphisme ϕ de Σ^* dans Γ^* est dit 1-uniforme si $\phi(a) \in \Gamma \ \forall a \in \Sigma$. Ainsi, ϕ est 1-uniforme si l'image par ϕ de toute lettre est une lettre. En d'autres mots, un morphisme 1-uniforme est un renommage (pas nécessairement injectif) des lettres.

Proposition 1.5.1. *Soit L un langage régulier sur l'alphabet Σ accepté par l'AFD $A = (\Sigma, Q, i, F, \delta)$ et soit ϕ un morphisme 1-uniforme de Γ^* dans Σ^*. Alors $\phi^{-1}(L)$ est le langage régulier accepté par l'AFD $B = (\Gamma, Q, i, F, \delta')$ où, pour tous $a \in \Gamma$ et $q \in Q$, $\delta'(q, a) = \delta(q, \phi(a))$.*

Démonstration. Un mot w est accepté par B si et seulement si $\delta'^w(i) \in F$. On a $\delta'^w(q) = \delta(q, \phi(w))$ par induction. Ainsi, un mot w est accepté par B si et seulement si $\delta^{\phi(w)}(i) \in F$. Cependant, pour tout mot v, $\delta^v(i) \in F$ si et seulement si v est accepté par A. On obtient que le mot w est accepté par B si et seulement si $\phi(w)$ est accepté par A. Au final, $L(B) = \phi^{-1}(L(A))$.

□

Ainsi, remarquons qu'on a

Proposition 1.5.2. *Soit L un langage régulier et ϕ un morphisme 1-uniforme. On a*

$$sc(\phi^{-1}(L)) \leq sc(L).$$

1.6 Transformations finies

Notation 1.6.1. Pour tout entier n, nous écrivons $[\![n]\!]$ pour $\{0, \ldots, n-1\}$.

Définition 1.6.1 (transformation)**.** Soit n un entier. Une transformation t est un élément de $[\![n]\!]^{[\![n]\!]}$. Nous écrivons it l'image de i sous t. Une transformation de $[\![n]\!]$ peut être représentée par $t = [i_0, i_1, \ldots, i_{n-1}]$ avec $i_k = kt$ pour chaque $k \in [\![n]\!]$ et $i_k \in [\![n]\!]$.

Définition 1.6.2 (permutation)**.** Soit n un entier. Une permutation est une transformation bijective sur $[\![n]\!]$. La permutation identité est notée Id.

Définition 1.6.3 (cycle)**.** Soit n un entier. Un cycle de longueur $l \leq n$, noté $(i_0, i_1, \ldots, i_{l-1})$, est une permutation c sur un sous-ensemble $I = \{i_0, \ldots, i_{l-1}\}$ de $[\![n]\!]$ où $i_k c = i_{k+1}$ pour $0 \leq k < l-1$ et $i_{l-1}c = i_0$.

Définition 1.6.4 (transposition)**.** Soit n un entier. Une transposition $t = (i, j)$ est une permutation sur $[\![n]\!]$ où $it = j$ et $jt = i$ et pour tous les éléments $k \in [\![n]\!] \backslash \{i, j\}$, $kt = k$.

Définition 1.6.5 (contraction)**.** Soit n un entier. Une contraction $t = (ij)$ est une transformation où $it = j$ et pour tous les éléments $k \in [\![n]\!] \backslash \{i\}$, $kt = k$.

1.7 Notation pratique

Pour tout caractère X et tout entier k donné par le contexte, nous écrivons $\underline{X}$ pour $(X_1, \ldots, X_k)$.

Chapitre 2

Opérations 1-uniformes et modificateurs

Nous allons décrire une classe d'opérations régulières, appelées 1-uniformes qui sont intéressantes pour l'étude de la complexité en états. L'intérêt de ces opérations est que à chacune d'entre elles correspond une construction sur des automates finis déterministes. Ainsi, nous allons définir des algorithmes sur des AFDs appelés modificateurs, qui nous permettront de calculer la complexité en états d'opérations 1-uniformes. Pour finir, nous décrirons un sous-ensemble de ces modificateurs qui correspond à l'ensemble des opérations régulières 1-uniformes.

2.1 Opérations 1-uniformes

Nous allons définir les opérations 1-uniformes et en donner des exemples. Nous présenterons aussi un exemple d'opération qui n'est pas 1-uniforme. Enfin, nous verrons que ces opérations sont stables par composition. Le lecteur intéressé peut trouver les preuves de la 1-uniformité de beaucoup d'opérations telles que le miroir et la concaténation dans l'article [5].

2.1.1 Définition

Définition 2.1.1 (1-uniforme)**.** Une opération k-aire $\otimes$ est 1-uniforme si elle est régulière et si elle commute avec l'inverse de chaque morphisme 1-uniforme, c'est-à-dire, pour tout k-uplet de langages réguliers $(L_1, .., L_k)$ et tout morphisme 1-uniforme ϕ,

$$\otimes(\phi^{-1}(L_1), \ldots, \phi^{-1}(L_k)) = \phi^{-1}(\otimes(L_1, \ldots, L_k)).$$

Par exemple, l'étoile de Kleene et l'union sont des opérations 1-uniformes.

Proposition 2.1.1. *L'étoile de Kleene est 1-uniforme.*

Démonstration. Soient Σ et Γ deux alphabets. Soit L un langage régulier sur Σ, et soit ϕ un morphisme 1-uniforme de Γ^* dans Σ^*.

— Prouvons d'abord que $(\phi^{-1}(L))^* \subseteq \phi^{-1}(L^*)$. Soit v un mot dans $(\phi^{-1}(L))^*$. Il existe alors un entier n et n mots de $\phi^{-1}(L)$ $u_1, \ldots, u_n$ tels que $v = u_1 \cdots u_n$. De plus, il existe n mots de L, $t_1, \ldots, t_n$ tels que $\phi(u_i) = t_i$, pour tout $i \in \{1, \ldots, n\}$. On obtient $\phi(v) = w$ avec $w = t_1 \cdots t_n$ et donc $v \in \phi^{-1}(L^*)$.

— Maintenant, montrons que $\phi^{-1}(L^*) \subseteq (\phi^{-1}(L))^*$. Soit v un mot de $\phi^{-1}(L^*)$. Il existe un entier n et n mots de L $t_1, \ldots, t_n$ tels que $\phi(v) = w$, avec $w = t_1 \cdots t_n$. Comme ϕ est 1-uniforme, $\phi(v) = \phi(v_1) \cdots \phi(v_{|v|})$, et chaque $\phi(v_j)$ est une lettre de Σ. Ainsi, v et w ont la même longueur, et $\phi(v_j) = w_j$, pour tout $j \in \{1, \ldots, |v|\}$. Par conséquent, pour tout $i \in \{1, \ldots, n\}$, si $u_i = v_{|t_1|+|t_2|\cdots+|t_{i-1}|+1} \cdots v_{|t_1|+|t_2|\cdots+|t_i|}$, on a $\phi(u_i) = t_i$ et $v = u_1 \cdots u_n$. On a donc $v \in (\phi^{-1}(L))^*$.

□

Proposition 2.1.2. *L'union est 1-uniforme.*

Démonstration. Soient E et F deux ensembles. On sait que pour toute fonction ϕ de E dans F,

$$\phi^{-1}(X \cup Y) = \phi^{-1}(X) \cup \phi^{-1}(Y), \text{ pour tous } X, Y \subseteq F$$

En effet, on a

$$\begin{aligned} & x \in \phi^{-1}(X \cup Y) \\ \Leftrightarrow & \phi(x) \in X \cup Y \\ \Leftrightarrow & \phi(x) \in X \text{ ou } \phi(x) \in Y \\ \Leftrightarrow & x \in \phi^{-1}(X) \text{ ou } x \in \phi^{-1}(Y) \\ \Leftrightarrow & x \in \phi^{-1}(X) \cup \phi^{-1}(Y) \end{aligned}$$

Ainsi, il suffit d'appliquer ce résultat au cas particulier : $E = \Gamma^*$ et $F = \Sigma^*$ où Σ et Γ sont deux alphabets, et ϕ est un morphisme 1-uniforme de Γ^* dans Σ^*. □

2.1.2 Opérations non uniformes

Beaucoup d'autres opérations unaires régulières connues sont 1-uniformes. Par contre, le quotient à droite est un exemple d'opération régulière qui n'est pas 1-uniforme.

Exemple 2.1.1. Soient L_1, L_2 deux langages réguliers. L'opération $(L_1, L_2) = L_1 \cdot L_2^{-1}$ est régulière mais pas 1-uniforme.

Commençons par montrer qu'elle est régulière.
Soit A l'AFD acceptant L_1. Nous allons construire un AFD B qui accepte $L_1.L_2^{-1}$. Cet automate sera identique à A sauf qu'il aura des états finaux différents. Pour chaque état q_i de A, nous allons déterminer de la façon suivante s'il s'agit d'un état final de B :

— On construit C_i l'AFD égal à A sauf que l'état q_i est l'état initial.
— On construit l'AFD D_i qui accepte l'intersection de L_2 et du langage accepté par C_i.
— Si D_i reconnait un mot quelconque alors q_i doit être marqué comme final dans B.

Il faut reproduire ce processus pour chaque état de A.
Finalement, comme on a construit un AFD qui accepte $L_1 \cdot L_2^{-1}$, ce langage est donc régulier.

Maintenant, montrons que cette opération n'est pas 1-uniforme, soient
— $\Gamma = \Sigma = \{a, b\}$;
— ϕ le morphisme 1-uniforme de Γ^* dans Σ^* tel que $\phi(a) = \phi(b) = a$;
— $L_1 = \{ab\}$;
— $L_2 = \{b\}$.
On obtient $\phi^{-1}(L_1) = \phi^{-1}(L_2) = \emptyset$ et donc $\phi^{-1}(L_1) \cdot (\phi^{-1}(L_2))^{-1} = \emptyset$. Cependant, on a $L_1 \cdot L_2^{-1} = \{a\}$, et $\phi^{-1}(L_1 \cdot L_2^{-1}) = \phi^{-1}(\{a\}) = \{a, b\}$. On a alors
$\phi^{-1}(L_1) \cdot \phi^{-1}(L_2)^{-1} \neq \phi^{-1}(L_1 \cdot L_2^{-1})$, ainsi le quotient à droite n'est pas 1-uniforme.

2.1.3 Composition d'opérations 1-uniformes

Remarquons que la 1-uniformité est stable par composition.

Définition 2.1.2 (Composition d'opérations)**.** Soient $\otimes$ et $\oplus$ deux opérations 1-uniformes, respectivement j-aire et k-aire. Pour tout entier p tel que $1 \leq p \leq j$, la composition de ces opérations est définie par l'opérateur $(j + k - 1)$-aire

$$\otimes \circ_p \oplus(L_1, \ldots, L_{j+k-1}) = \otimes(L_1, \ldots, L_{p-1}, \oplus(L_p, \ldots, L_{p+k-1}), L_{p+k}, \ldots, L_{j+k-1})$$

Exemple 2.1.2. Soient $\otimes$ et $\oplus$ deux opérations respectivement 3-aire et 2-aire, on a

$$\otimes \circ_1 \oplus(L_1, \ldots, L_4) = \otimes(\oplus(L_1, L_2), L_3, L_4)$$
$$\otimes \circ_2 \oplus(L_1, \ldots, L_4) = \otimes(L_1, \oplus(L_2, L_3), L_4)$$
$$\otimes \circ_3 \oplus(L_1, \ldots, L_4) = \otimes(L_1, L_2, \oplus(L_3, L_4))$$

Proposition 2.1.3. *Soient $\otimes$ et $\oplus$ deux opérations 1-uniformes, respectivement j-aire et k-aire. Pour tout entier p tel que $1 \leq p \leq j$, l'opérateur $(j+k-1)$-aire*

$$\otimes \circ_p \oplus(L_1, \ldots, L_{j+k-1}) = \otimes(L_1, \ldots, L_{p-1}, \oplus(L_p, \ldots, L_{p+k-1}), L_{p+k}, \ldots, L_{j+k-1})$$

est 1-uniforme.

Démonstration. Soient $\otimes$ et $\oplus$ deux opérations 1-uniformes, respectivement j-aire et k-aire, p un entier tel que $1 \leq p \leq j$, $L_1, \ldots, L_{j+k-1}$ des langages réguliers et ϕ un morphisme 1-uniforme. On a

$$\begin{aligned}
&\otimes \circ_p \oplus (\phi^{-1}(L_1), \ldots, \phi^{-1}(L_{j+k-1})) \\
=&\otimes (\phi^{-1}(L_1), \ldots, \phi^{-1}(L_{p-1}), \oplus(\phi^{-1}(L_p), \ldots, \phi^{-1}(L_{p+k-1})), \phi^{-1}(L_{p+k}), \ldots, \phi^{-1}(L_{j+k-1})) \\
=&\otimes (\phi^{-1}(L_1), \ldots, \phi^{-1}(L_{p-1}), \phi^{-1}(\oplus(L_p, \ldots, L_{p+k-1})), \phi^{-1}(L_{p+k}), \ldots, \phi^{-1}(L_{j+k-1})) \\
&\text{car } \oplus \text{ est 1-uniforme} \\
=&\phi^{-1}(\otimes(L_1, \ldots, L_{p-1}, \oplus(L_p, \ldots, L_{p+k-1}), L_{p+k}, \ldots, L_{j+k-1})) \text{ car } \otimes \text{ est 1-uniforme} \\
=&\phi^{-1}(\otimes \circ_p \oplus(L_1, \ldots, L_{j+k-1}))
\end{aligned}$$

□

2.2 Modificateurs

La définition de la complexité en états d'opérations régulières découle directement de celle de la complexité en états de langage. De plus, la définition de la complexité en états de langage implique directement la notion d'automate fini déterministe minimal. Une façon facile pour calculer l'AFD minimal associé à un langage est de d'abord donner un AFD qui reconnait ce langage, et après minimiser cet automate. On peut donc calculer la complexité en états d'opérations régulières en faisant des calculs directement sur des AFDs. Ainsi, pour prouver des résultats sur la complexité en états pour les opérations 1-uniformes, il est pratique de lier ces opérations sur des langages avec des opérations agissant directement sur des AFDs appelées modificateurs. En fait, un k-modificateur est un algorithme prenant comme entrées k automates et en produisant un. Beaucoup d'opérations régulières peuvent être décrites par ce mécanisme. On les appelle des opérations descriptibles.

2.2.1 Définition

Définition 2.2.1 (configuration d'état)**.** La configuration d'état d'un AFD $A = (\Sigma, Q, i, F, \delta)$ est le triplet (Q, i, F).

Définition 2.2.2 (modificateur)**.** Un k-modificateur est une opération k-aire agissant sur un k-uplet d'automates finis déterministes $(A_1, \ldots, A_k)$ définis sur le même alphabet Σ et produisant un automate fini déterministe $\mathrm{m}(A_1, \ldots, A_k)$ tel que

— son alphabet est Σ

— sa configuration d'état dépend seulement des configurations d'état de $A_1, \ldots, A_k$

— pour tout $a \in \Sigma$, la fonction de transition de a dans $\mathrm{m}(A_1, \ldots, A_k)$ dépend seulement des configurations d'état de $A_1, \ldots, A_k$ et des fonctions de transition de a dans chacun des AFDs $A_1, \ldots, A_k$.

Plus formellement, tout k-modificateur m peut être vu comme un 4-uplet de relations $(\mathtt{Q},\ \mathtt{i},\ \mathtt{f},\ \rho)$ agissant sur k automates finis déterministes $\underline{\mathrm{A}}$ avec $A_j = (\Sigma, Q_j, i_j, F_j, \delta_j)$ afin de construire un automate fini déterministe $\mathrm{m}\underline{\mathrm{A}}=(\Sigma, Q, i, F, \delta)$ où $\mathrm{Q}=\mathtt{Q}\underline{\mathrm{Q}}$, $\mathrm{i}=\mathtt{i}(\underline{\mathrm{Q}},\underline{\mathrm{i}},\underline{\mathrm{F}})$, $\mathrm{F}=\mathtt{f}(\underline{\mathrm{Q}},\underline{\mathrm{i}},\underline{\mathrm{F}})$, et $\forall a \in \Sigma, \delta^a = \rho(\underline{i}, \underline{F}, \underline{\delta^a})$.

2.2.2 Composition de modificateurs

Nous allons définir la composition de modificateurs et nous allons montrer que la composition de deux modificateurs est encore un modificateur.

Définition 2.2.3 (Composition de modificateurs)**.** Soient m_1 un k_1-modificateur, m_2 un k_2-modificateur et $1 \leq j \leq k_1$. Leur composition est définie par

$$m_1 \circ_j m_2(A_1, \ldots, A_{k_1+k_2-1}) = m_1(A_1, \ldots, A_{j-1}, m_2(A_j, \ldots, A_{j+k_2-1}), A_{j+k_2}, \ldots, A_{k_1+k_2-1}).$$

Proposition 2.2.1. *Soient m_1 et m_2 deux modificateurs. La composition $m_1 \circ_j m_2$ est aussi un modificateur.*

Démonstration. Soient $m_1 = (\mathtt{Q}^{(1)}, \mathtt{i}^{(1)}, \mathtt{f}^{(1)}, \rho^{(1)})$ un modificateur k_1-aire et $m_2 = (\mathtt{Q}^{(2)}, \mathtt{i}^{(2)}, \mathtt{f}^{(2)}, \rho^{(2)})$ un modificateur k_2-aire. Posons
$\underline{Q^\star} = (Q_1, \ldots, Q_{j-1}, \mathtt{Q}^{(2)}(Q_j, \ldots, Q_{j+k_2-1}), Q_{j+k_2}, \ldots, Q_{k_1+k_2-1})$
$\underline{i^\star} = (i_1, \ldots, i_{j-1}, \mathtt{i}^{(2)}((Q_j, \ldots, Q_{j+k_2-1}), (i_j, \ldots, j_{j+k_2-1}), (F_j, \ldots, F_{j+k_2-1})), i_{j+k_2}, \ldots, i_{k_1+k_2-1})$
$\underline{F^\star} = (F_1, \ldots, F_{j-1}, \mathtt{f}^{(2)}((Q_j, \ldots, Q_{j+k_2-1}), (i_j, \ldots, j_{j+k_2-1}), (F_j, \ldots, F_{j+k_2-1})), F_{j+k_2}, \ldots, F_{k_1+k_2-1})$
$\underline{\delta^\star} = (\delta^a_1, \ldots, \delta^a_{j-1}, \rho^{(2)}((i_j, \ldots, i_{j+k_2-1}), (F_j, \ldots, F_{j+k_2-1}), (\delta^a_j, \ldots, \delta^a_{j+k_2-1})), \delta^a_{j+k_2}, \ldots, \delta^a_{k_1+k_2-1})$

On définit le modificateur $(k_1 + k_2 - 1)$-aire $(\mathtt{Q}, \mathtt{i}, \mathtt{f}, \rho)$ par
$\mathtt{Q}\underline{Q} = \mathtt{Q}^{(1)}\underline{Q^\star}$, $\mathtt{i}(\underline{Q}, \underline{i}, \underline{F}) = \mathtt{i}^{(1)}(\underline{Q^\star}, \underline{i^\star}, \underline{F^\star})$, $\mathtt{f}(\underline{Q}, \underline{i}, \underline{F}) = \mathtt{f}^{(1)}(\underline{Q^\star}, \underline{i^\star}, \underline{F^\star})$, $\rho(\underline{i}, \underline{F}, \underline{\delta^a}) = \rho^{(1)}(\underline{i^\star}, \underline{F^\star}, \underline{\delta^\star})$.

Il est clair que $(\mathtt{Q}, \mathtt{i}, \mathtt{f}, \rho)$ agit sur des automates comme $m_1 \circ_j m_2$. □

2.3 Modificateurs particuliers

Michel Rigo présente la stabilité des langages acceptés par automate dans "Théorie des automates et langages formels". Par exemple, il démontre que si L et M sont des langages acceptés par deux automates finis, alors $L \cup M$ est aussi accepté par un automate fini. Ici, nous allons présenter de manière concrète comment construire l'automate qui accepte $L \cup M$ en appliquant le modificateur union à L et M. Ainsi, nous allons présenter des constructions classiques : le complémentaire, l'union, l'intersection, le xor, la concaténation, l'étoile de Kleene, le préfine, le miroir et la racine.

Des exemples supplémentaires d'application de ces modificateurs sont donnés dans l'annexe A afin de mieux visualiser leur effet.

Définition 2.3.1 (Le modificateur complémentaire)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ un automate fini déterministe. Notons le modificateur complémentaire `Comp`=(`Q`,`i`,`f`,ρ) et définissons le par :

— $\texttt{Q}(Q_1) = Q_1$
— $\texttt{i}(Q_1, i_1, F_1) = i_1$
— $\texttt{f}(Q_1, i_1, F_1) = Q_1 \backslash F_1$
— $\rho(i_1, F_1, \delta_1^a) = \delta_1^a$

Exemple 2.3.1 (modificateur complémentaire)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ un AFD représenté à la Figure 2.1 qui est tel que $\Sigma = \{a, b\}, Q_1 = \{0, 1\}, i_1 = 0, F_1 = \{0\}$. Remarquons que le langage accepté par A_1 est $((a + b)b^*a)^*$.

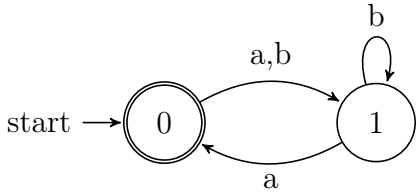


FIGURE 2.1 – Automate A_1

En appliquant le modificateur complémentaire à cet automate, nous obtenons l'automate `Comp`(A_1) représenté à la Figure 2.2. En effet, on a

— $\texttt{Q}(Q_1) = Q_1 = \{0, 1\}$
— $\texttt{i}(Q_1, i_1, F_1) = i_1 = 0$
— $\texttt{f}(Q_1, i_1, F_1) = Q_1 \backslash F_1 = \{1\}$

— $\rho(i_1, F_1, \delta_1^a) = \delta_1^a$

Le langage accepté par $\texttt{Comp}(A_1)$ est $(a+b)b^*(a(a+b)b^*)^*$.

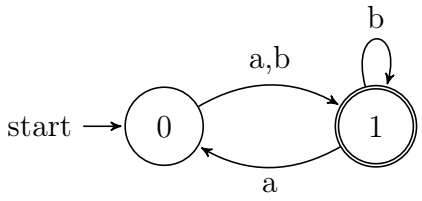


FIGURE 2.2 – Automate $\texttt{Comp}(A_1)$

Définition 2.3.2 (Le modificateur union)**.** Soient $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, i_2, F_2, \delta_2)$ deux automates finis déterministes. Notons le modificateur union $\texttt{Union}=(\texttt{Q},\texttt{i},\texttt{f},\rho)$ et définissons le par :

— $\texttt{Q}(Q_1, Q_2) = Q_1 \times Q_2$

— $\texttt{i}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = (i_1, i_2)$

— $\texttt{f}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = (F_1 \times Q_2) \cup (Q_1 \times F_2)$

— $\rho((i_1, i_2), (F_1, F_2), (\delta_1^a, \delta_2^a)) = (\delta_1^a, \delta_2^a)$

Exemple 2.3.2. Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, i_2, F_2, \delta_2)$ deux AFDs représentés aux Figures 2.3 et 2.4 qui sont tels que $\Sigma = \{a, b\}, Q_1 = \{0, 1\}, i_1 = 0, F_1 = \{1\}$ et $Q_2 = \{0, 1\}, i_2 = 0, F_2 = \{1\}$. Remarquons que le langage accepté par A_1 est $(a+b)b^*(a(a+b)b^*)^*$ et celui accepté par A_2 est $b^*aa^*(bb^*aa^*)^*$.

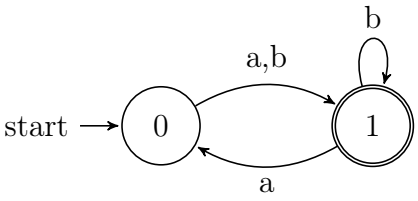


FIGURE 2.3 – AFD A_1

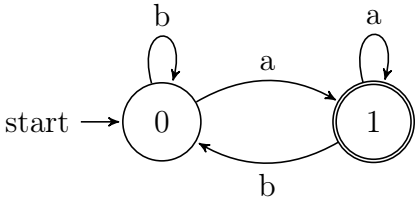


FIGURE 2.4 – AFD A_2

En appliquant le modificateur union à ces automates, nous obtenons l'automate $\texttt{Union}(A_1, A_2)$ représenté à la Figure 2.5. En effet, on a

— $\texttt{Q}(Q_1, Q_2) = Q_1 \times Q_2 = \{0, 1\} \times \{0, 1\} = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$

— $\texttt{i}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = (i_1, i_2) = (0, 0)$

— $\texttt{f}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = (F_1 \times Q_2) \cup (Q_1 \times F_2)$
$= (\{1\} \times \{0, 1\}) \cup (\{0, 1\} \times \{1\})$
$= \{(1, 0), (1, 1)\} \cup \{(0, 1), (1, 1)\}$
$= \{(0, 1), (1, 0), (1, 1)\}$

— $\rho((i_1, i_2), (F_1, F_2), (\delta_1^a, \delta_2^a)) = (\delta_1^a, \delta_2^a)$

Le langage accepté par $\texttt{Union}(A_1, A_2)$ est $(a + b)(a^*b^*)^*$.

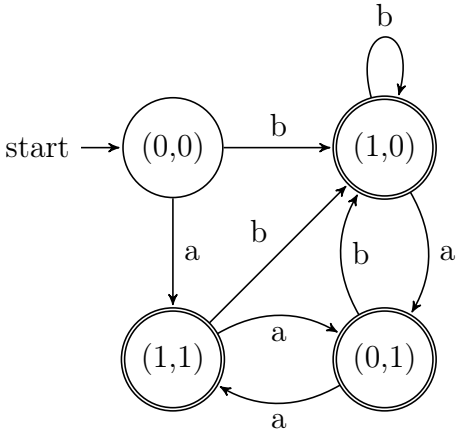


FIGURE 2.5 – Automate $\texttt{Union}(A_1, A_2)$

Définition 2.3.3 (Le modificateur intersection)**.** Soient $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, i_2, F_2, \delta_2)$ deux automates finis déterministes. Notons le modificateur intersection $\texttt{Inter}=(\texttt{Q},\texttt{i},\texttt{f},\rho)$ et définissons le par :

— $\texttt{Q}(Q_1, Q_2) = Q_1 \times Q_2$
— $\texttt{i}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = (i_1, i_2)$
— $\texttt{f}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = F_1 \times F_2$
— $\rho((i_1, i_2), (F_1, F_2), (\delta_1^a, \delta_2^a)) = (\delta_1^a, \delta_2^a)$

Définition 2.3.4 (Le modificateur xor)**.** Soient $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, i_2, F_2, \delta_2)$ deux automates finis déterministes. Notons le modificateur xor $\texttt{Xor}=(\texttt{Q},\texttt{i},\texttt{f},\rho)$ et définissons le par :

— $\texttt{Q}(Q_1, Q_2) = Q_1 \times Q_2$
— $\texttt{i}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = (i_1, i_2)$
— $\texttt{f}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = F_1 \times (Q_2 \backslash F_2) \cup (Q_1 \backslash F_1) \times F_2$
— $\rho((i_1, i_2), (F_1, F_2), (\delta_1^a, \delta_2^a)) = (\delta_1^a, \delta_2^a)$

Définition 2.3.5 (Le modificateur concaténation)**.** Soient $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, i_2, F_2, \delta_2)$ deux automates finis déterministes. Notons le modificateur concaténation `Conc`$=$(`Q,i,f`,ρ) et définissons le par :

- $\texttt{Q}(Q_1, Q_2) = Q_1 \times 2^{Q_2}$
- $\texttt{i}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = \begin{cases} (i_1, \emptyset) & \text{si } i_1 \notin F_1 \\ (i_1, \{i_2\}) & \text{si } i_1 \in F_1 \end{cases}$
- $\texttt{f}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = \{(q_1, E) \in Q_1 \times 2^{Q_2} | E \cap F_2 \neq \emptyset\}$
- pour tout $(q_1, E) \in Q_1 \times 2^{Q_2}$,
  $\rho((i_1, i_2), (F_1, F_2), (\delta_1^a, \delta_2^a))(q_1, E) = \begin{cases} (\delta_1^a(q_1), \delta_2^a(E)) & \text{si } \delta_1^a(q_1) \notin F_1 \\ (\delta_1^a(q_1), \delta_2^a(E) \cup \{i_2\}) & \text{si } \delta_1^a(q_1) \in F_1 \end{cases}$

Définition 2.3.6 (Le modificateur étoile)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ un automate fini déterministe. Notons le modificateur étoile `Star`$=$(`Q,i,f`,ρ) et définissons le par :

- $\texttt{Q}(Q_1) = 2^{Q_1}$
- $\texttt{i}(Q_1, i_1, F_1) = \emptyset$
- $\texttt{f}(Q_1, i_1, F_1) = \{E | E \cap F_1 \neq \emptyset\} \cup \{\emptyset\}$
- $\rho(i_1, F_1, \delta_1^a)(E) = \begin{cases} \{\delta_1^a(i)\} & \text{si } E = \emptyset \text{ et } \delta_1^a(i) \notin F \\ \{\delta_1^a(i), i\} & \text{si } E = \emptyset \text{ et } \delta_1^a(i) \in F \\ \delta_1^a(E) & \text{si } E \neq \emptyset \text{ et } \delta_1^a(E) \cap F = \emptyset \\ \delta_1^a(E) \cup \{i\} & \text{si } E \neq \emptyset \text{ et } \delta_1^a(E) \cap F \neq \emptyset \end{cases}$

Exemple 2.3.3. Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ un AFD représenté à la Figure 2.6 qui est tel que $\Sigma = \{a, b\}, Q_1 = \{0, 1\}, i_1 = 0, F_1 = \{1\}$. Remarquons que le langage accepté par cet automate est

$$b^*aa^*(bb^*aa^*)^*.$$

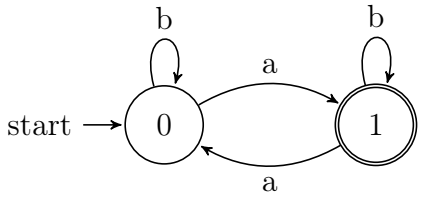


FIGURE 2.6 – Automate A_1

En appliquant le modificateur étoile à cet automate, nous obtenons l'automate `Star`(A_1) représenté à la Figure 2.7. En effet, on a

- $\texttt{Q}(Q_1) = 2^{Q_1} = 2^{\{0,1\}} = \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$

— $\mathtt{i}(Q_1, i_1, F_1) = \emptyset$
— $\mathtt{f}(Q_1, i_1, F_1) = \{E | E \cap F_1 \neq \emptyset\} \cup \emptyset = \{\{1\}, \{0,1\}\} \cup \emptyset = \{\emptyset, \{1\}, \{0,1\}\}$
— On a par exemple :
  — $\rho(i_1, F_1, \delta_1^a)(\emptyset) = \{\delta_1^a(0), 0\} = \{0,1\}$ car $\delta_1^a(0) \in F_1$
  — $\rho(i_1, F_1, \delta_1^b)(\emptyset) = \{\delta_1^b(0)\} = \{0\}$ car $\delta_1^b(0) \notin F_1$
  — $\rho(i_1, F_1, \delta_1^a)(\{1\}) = \delta_1^a(\{1\}) = \{0\}$ car $\delta_1^a(\{1\}) \cap F_1 = \emptyset$
  — ...

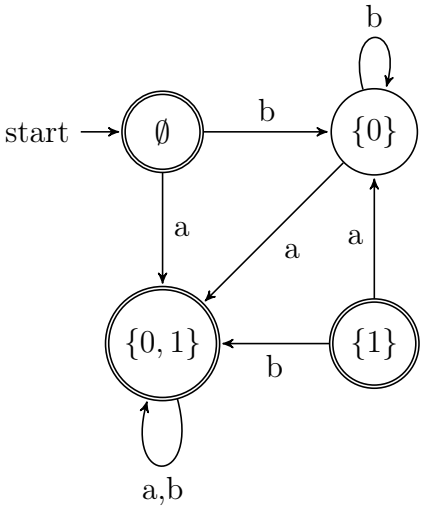


FIGURE 2.7 – Automate $\mathtt{Star}(A_1)$

En retirant l'état inutile, $\mathtt{Star}(A_1)$ se représente tel qu'à la Figure 2.8. Le langage accepté par cet automate est

$$\varepsilon + a(a+b)^* + bb^*a(a+b)^*.$$

Définition 2.3.7 (Le modificateur préfine)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ un automate fini déterministe. Notons le modificateur préfine $\mathtt{Prefin}{=}(\mathtt{Q},\mathtt{i},\mathtt{f},\rho)$ et définissons le par :
— $\mathtt{Q}(Q_1) = Q_1$
— $\mathtt{i}(Q_1, i_1, F_1) = i_1$
— $\mathtt{f}(Q_1, i_1, F_1) = F_1$

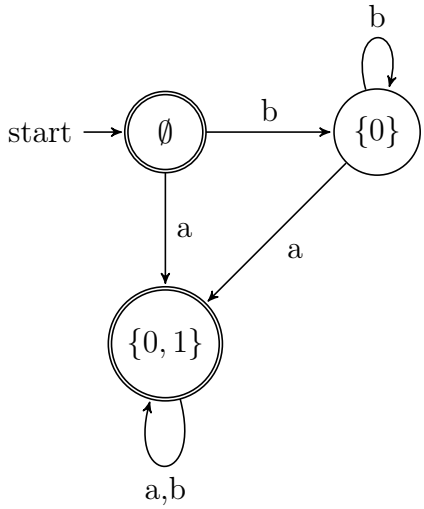


FIGURE 2.8 – Automate $\texttt{Star}(A_1)$ simplifié

— $\rho(i_1, F_1, \delta_1^a) = q \to \begin{cases} \delta_1^a(q) & \text{si } q \notin F_1 \\ q & \text{sinon.} \end{cases}$

Définition 2.3.8 (Le modificateur miroir)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ un automate fini déterministe. Notons le modificateur miroir $\texttt{Rev}=(\texttt{Q},\texttt{i},\texttt{f},\rho)$ et définissons le par :

— $\texttt{Q}(Q_1) = 2^{Q_1}$

— $\texttt{i}(Q_1, i_1, F_1) = F_1$

— $\texttt{f}(Q_1, i_1, F_1) = \{E \subset Q_1 | i_1 \in E\}$

— $\rho(i_1, F_1, \delta_1^a) = E \to \bigcup_{q \in E} \{q' | \delta_1^a(q') = q\}$

Exemple 2.3.4 (modificateur miroir)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ un AFD représenté à la Figure 2.9 qui est tel que $\Sigma = \{a, b, c, d\}, Q_1 = \{0, 1\}, i_1 = 0, F_1 = \{1\}$.

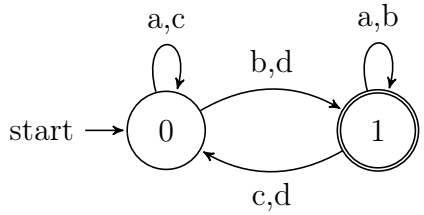


FIGURE 2.9 – Automate A_1

En appliquant le modificateur miroir à cet automate, nous obtenons l'automate $\texttt{Rev}(A_1)$ représenté à la Figure 2.10.

En effet, on a

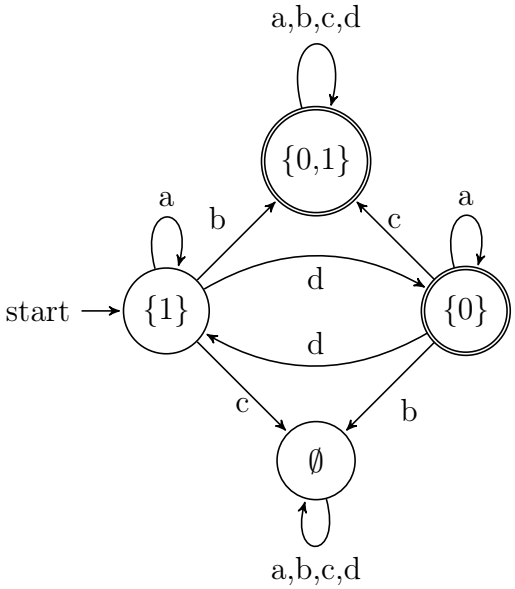


FIGURE 2.10 – Automate $\texttt{Rev}(A_1)$

— $\texttt{Q}(Q_1) = 2^{Q_1} = \{\emptyset, \{0\}, \{1\}, \{0,1\}\}$
— $\texttt{i}(Q_1, i_1, F_1) = F_1 = \{1\}$
— $\texttt{f}(Q_1, i_1, F_1) = \{E \subset Q_1 | i_1 \in E\} = \{E \subset Q_1 | 0 \in E\} = \{\{0\}, \{0,1\}\}$
— $\rho(i_1, F_1, \delta_1^a) = E \to \bigcup_{q \in E} \{q' | \delta_1^a(q') = q\}$ et par exemple, on a bien
  — $\rho(i_1, F_1, \delta_1^a)(\{1\}) = \bigcup_{q \in \{1\}} \{q' | \delta_1^a(q') = q\} = \{1\}$ car $\delta_1^a(1) = 1$
  — $\rho(i_1, F_1, \delta_1^b)(\{1\}) = \bigcup_{q \in \{1\}} \{q' | \delta_1^b(q') = q\} = \{0,1\}$ car $\delta_1^b(0) = 1$ et $\delta_1^b(1) = 1$
  — $\rho(i_1, F_1, \delta_1^c)(\{1\}) = \bigcup_{q \in \{1\}} \{q' | \delta_1^c(q') = q\} = \emptyset$ car $\nexists q' : \delta_1^c(q') = 1$
  — $\rho(i_1, F_1, \delta_1^d)(\{1\}) = \bigcup_{q \in \{1\}} \{q' | \delta_1^d(q') = q\} = \{0\}$ car $\delta_1^d(0) = 1$

Définition 2.3.9 (Le modificateur racine)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ un automate fini déterministe. Notons le modificateur racine $\texttt{SRoot}=(\texttt{Q},\texttt{i},\texttt{f},\rho)$ et définissons le par :

— $\texttt{Q}(Q_1) = Q_1^{Q_1}$
— $\texttt{i}(Q_1, i_1, F_1) = Id$
— $\texttt{f}(Q_1, i_1, F_1) = \{g | g^2(i_1) \in F_1\}$
— $\rho(i_1, F_1, \delta_1^a) = g \to (\delta_1^a \circ g)$

Définition 2.3.10 (Le modificateur racine k-ème)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ un automate fini déterministe. Notons le modificateur racine k-ème $\texttt{SRoot}_k=(\texttt{Q},\texttt{i},\texttt{f},\rho)$ et définissons le par :

— $\texttt{Q}(Q_1) = Q_1^{Q_1}$

— $\mathtt{i}(Q_1, i_1, F_1) = Id$
— $\mathtt{f}(Q_1, i_1, F_1) = \{g | g^k(i_1) \in F_1\}$
— $\rho(i_1, F_1, \delta_1^a) = g \to (\delta_1^a \circ g)$

2.4 Opérations descriptibles

Dans cette section, nous allons définir les opérations descriptibles qui sont en fait des opérations régulières décrites par un modificateur. Nous donnerons des exemples d'opérations descriptibles et de non descriptibles. Ensuite, on prouvera qu'il existe un modificateur pour la composition d'opérations descriptibles.

2.4.1 Définition

Définition 2.4.1 (opération descriptible)**.** Soit une opération $\otimes$ agissant sur des k-uplets de langages définis sur le même alphabet. L'opération $\otimes$ est dite descriptible (m-descriptible) s'il existe un k-modificateur m tel que pour tout k-uplet d'automates finis déterministes complets $\underline{A}$, on a $\otimes(L(A_1), \ldots, L(A_k)) = L(m\underline{A})$.

Exemple 2.4.1. L'opération racine k-ème est descriptible. En effet, il suffit de montrer que $\sqrt[k]{L(A)} = L(\mathtt{SRoot}_k(A))$.

Soient $A = (\Sigma, Q, i, F, \delta)$ un AFD, $\mathtt{SRoot}_k(A) = (\Sigma, Q', i', F', \delta')$ et w un mot de Σ^*. Le mot w^k est accepté par A si et seulement si $\delta^{w^k}(i) \in F$. Cependant, on a

$$\delta^{w^k} = \delta^w \circ \cdots \circ \delta^w = (\delta^w)^k$$

Par conséquent, $w^k \in L(A)$ si et seulement si $\delta^w \in F'$. Mais, $\delta'^w(i') = \delta^w \circ Id = \delta^w$. On obtient alors $w^k \in L(A)$ si et seulement si $\delta'^w(i') \in F'$. On a bien $\sqrt[k]{L(A)} = L(SRoot_k(A))$.

Exemple 2.4.2. L'opération intersection est descriptible. En effet, pour tous AFDs A_1, A_2, on a $L(A_1) \cap L(A_2) = L(\mathtt{Inter}(A_1, A_2))$.

Soient A_1, A_2 deux AFDs avec $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, i_2, F_2, \delta_2)$, soit $w = a_1 \cdots a_n$ un mot dans Σ^* et soit $\mathtt{Inter}(A_1, A_2)$=$(\Sigma, Q, i, F, \delta)$. Le mot w est dans $L(A_1) \cap L(A_2)$ si et seulement si $\delta_1^w(i_1) \in F_1 \wedge \delta_2^w(i_2) \in F_2$. Par induction, on a

$\delta^w(i_1, i_2) = (\delta_1^w(i_1), \delta_2^w(i_2))$ vu la définition de `Inter`. Ainsi, $w \in L(A_1) \cap L(A_2)$ si et seulement si $\delta^w(i_1, i_2) \in F$. Comme $(i_1, i_2) = i$, w est dans $L(A_1) \cap L(A_2)$ si et seulement si w est dans $L(\texttt{Inter}(A_1, A_2))$.

Remarque 2.4.1. Nous avons défini le modificateur de certaines opérations habituelles sur les langages telles que `Comp, Union, Star`, etc. Ainsi, toutes ces opérations sont descriptibles.

2.4.2 Opérations non descriptibles

Nous allons montrer qu'il existe des opérations non descriptibles.

Considérons trois alphabets X, X' et Y tels que $X \cap X' = \emptyset$, une bijection $\varphi : X \to Y$, étendue comme un isomorphisme de monoïdes de X^* dans Y^*, et

$$\eta : 2^{(X \cup X')^*} \to 2^{Y^*} : L \mapsto \eta(L) = \varphi(L \cap X^*).$$

Proposition 2.4.1. *Si $A = (X \cup X', Q, i, F, \delta)$ est un automate fini déterministe qui reconnait un langage L, alors $\eta(L)$ est un langage régulier accepté par (Y, Q, i, F, δ_1) où $\delta_1^y = \delta^{\varphi^{-1}(y)}$ pour tout $y \in Y$.*

Exemple 2.4.3. Supposons que $X = \{a, b\}, X' = \{c\}, Y = \{e, f\}$ et φ est tel que $\varphi(a) = e$, $\varphi(b) = f$. Soit $A = (\{a, b, c\}, Q, i, F, \delta)$ l'AFD représenté à la Figure 2.11 qui reconnait le langage

$$L = a^*(b + c)\{a, b, c\}^*,$$

alors

$$\begin{aligned} \eta(L) &= \varphi(L \cap X^*) \\ &= \varphi(a^*(b + c)\{a, b, c\}^* \cap \{a, b\}^*) \\ &= \varphi(a^*b\{a, b\}^*) \\ &= e^*f\{e, f\}^* \end{aligned}$$

est un langage régulier reconnu par $B = (\{e, f\}, Q, i, F, \delta_1)$ représenté à la Figure 2.12. En effet, on a

— $\delta_1^e(0) = \delta^{\varphi^{-1}(e)}(0) = \delta^a(0) = 0$
— $\delta_1^f(0) = \delta^{\varphi^{-1}(f)}(0) = \delta^b(0) = 1$

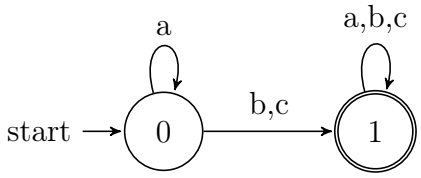


FIGURE 2.11 – Automate A

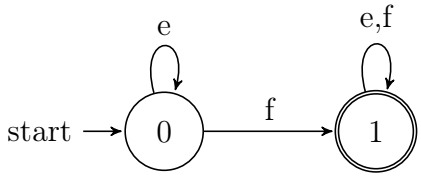


FIGURE 2.12 – Automate B

— $\delta_1^e(1) = \delta^{\varphi^{-1}(e)}(1) = \delta^a(1) = 1$
— $\delta_1^f(1) = \delta^{\varphi^{-1}(f)}(1) = \delta^b(1) = 1$

Proposition 2.4.2. *Soit $\otimes$ une opération k-aire descriptible. Pour tout $\underline{L} \in (2^{(X\cup X')^*})^k$ où les L_i sont réguliers, nous avons*

$$\otimes(\eta(L_1), \ldots, \eta(L_k)) = \eta(\otimes \underline{L}).$$

Démonstration. Soit $\underline{A}$, un k-uplet d'automates finis déterministes complets avec $A_j = (X \cup X', Q_j, i_j, F_j, \delta_j)$ tel que L(A_j)=L_j.

Vu que l'opération $\otimes$ est descriptible, il existe un modificateur $m = (\mathtt{Q},\mathtt{i},\mathtt{f},\rho)$ tel que $L(m\underline{A}) = \otimes\underline{L}$. On a $m\underline{A} = (X \cup X', \mathtt{Q}\underline{Q}, \mathtt{i}(\underline{Q}, \underline{i}, \underline{F}), \mathtt{f}(\underline{Q}, \underline{i}, \underline{F}), \delta)$ avec $\delta^a = \rho(\underline{i}, \underline{F}, \underline{\delta^a})$. Par la proposition 2.4.1, le langage $\eta(\otimes L)$ est reconnaissable par l'automate fini déterministe complet $A_{\underline{\Delta}} = (Y, \mathtt{Q}\underline{Q}, \mathtt{i}(\underline{Q}, \underline{i}, \underline{F}), \mathtt{f}(\underline{Q}, \underline{i}, \underline{F}), \delta_{\underline{\Delta}})$ avec $\delta^a_{\underline{\Delta}} = \delta^{\varphi^{-1}(a)} = \rho(\underline{i}, \underline{F}, \underline{\delta^{\varphi^{-1}(a)}})$.

Soit $\underline{A_\diamond}$ un k-uplet d'AFDCs tels que $A_{\diamond_j}$=(Y,Q_j,i_j,F_j,$\delta_{\diamond_j}$) avec $\delta^a_{\diamond_j}$=$\delta_j^{\varphi^{-1}(a)}$. On obtient que $A_{\diamond_j}$ reconnait $\eta(L_j)$ par la proposition 2.4.1. Vu que $\otimes$ est descriptible, $m\underline{A_\diamond}$=$(Y, \mathtt{Q}\underline{Q}, \mathtt{i}(\underline{Q}, \underline{i}, \underline{F}), \mathtt{f}(\underline{Q}, \underline{i}$
avec $\delta^a_\diamond = \rho(\underline{i}, \underline{F}, \underline{\delta^a_\diamond}) = \rho(\underline{i}, \underline{F}, \underline{\delta^{\varphi^{-1}(a)}}) = \delta^a_{\underline{\Delta}}$ ce qui termine la preuve.

□

Corollaire 2.4.1. *Soit $\otimes$ une opération k-aire descriptible et Y un alphabet. Soit $\underline{L}$ un k-uplet de langages réguliers sur Y. Alors*

— *Si $X \subset Y$ alors $\otimes(L_1 \cap X^*, \ldots, L_k \cap X^*) = \otimes\underline{L} \cap X^*$*

— *Pour toute bijection $\sigma : Y \to Y$ étendue comme un automorphisme de monoïdes, on $a \otimes(\sigma(L_1), \ldots, \sigma(L_k)) = \sigma(\otimes \underline{L})$.*

Grâce à ce résultat, nous pouvons construire des exemples d'opérations non descriptibles.

Exemple 2.4.4. Considérons l'opération quotient à droite

$$\otimes(L_1, L_2) = L_1 \cdot L_2^{-1} = \{u | uv \in L_1 \text{ pour un } v \in L_2\}.$$

Nous obtenons que cette opération n'est pas descriptible car elle ne respecte pas la première condition du Corollaire 2.4.1.
Par exemple, prenons $Y = \{a, b, c\}$, $L_1 = \{abc\}$ et $L_2 = \{c\}$. Nous avons

$$\otimes(L_1 \cap \{a, b\}^*, L_2 \cap \{a, b\}^*) = \otimes(\{abc\} \cap \{a, b\}^*, \{c\} \cap \{a, b\}^*) = \otimes(\emptyset, \emptyset) = \emptyset\emptyset^{-1} = \emptyset$$

alors que

$$\otimes(L_1, L_2) \cap \{a, b\}^* = \otimes(\{abc\}, \{c\}) \cap \{a, b\}^* = \{ab\} \cap \{a, b\}^* = \{ab\}.$$

Exemple 2.4.5. Considérons l'opération unaire définie par

$$\otimes(L) = \begin{cases} L \backslash \{a\} & \text{si les mots } a \text{ et } a^2 \text{ appartiennent à } L \\ L & \text{sinon.} \end{cases}$$

Cette opération satisfait la première condition du Corollaire 2.4.1 mais elle ne respecte pas la seconde.
Par exemple, si $Y = \{a, b\}$ et σ est tel que $\sigma(a) = b$ alors

$$\begin{aligned} \sigma(\otimes(\{a, a^2\})) &= \sigma(\{a, a^2\} \backslash \{a\}) \\ &= \sigma(\{a^2\}) \\ &= \{b^2\} \end{aligned}$$

car a et a^2 appartiennent à $\{a, a^2\}$. Par contre, on a

$$\begin{aligned} \otimes(\{\sigma(a), \sigma(a^2)\}) &= \otimes(\{b, b^2\}) \\ &= \{b, b^2\} \end{aligned}$$

car a et a^2 n'appartiennent pas à $\{b, b^2\}$. Donc elle n'est pas descriptible.

2.4.3 Composition d'opérations descriptibles

Nous avons montré dans la section 2.2.2 que la composition de deux modificateurs est un modificateur et nous allons utiliser cette propriété pour montrer qu'il existe un modificateur pour la composition d'opérations descriptibles.

Proposition 2.4.3. *Soit $\otimes$ une opération k_1-aire m_1-descriptible et $\oplus$ une opération k_2-aire m_2-descriptible. Alors pour tout $j \in \{1, \ldots, k_1\}$, le modificateur $m_1 \circ_j m_2$ décrit l'opération $\otimes \circ_j \oplus$ qui est $(k_1 + k_2 - 1)$-aire.*

Démonstration. Soient $L_1, \ldots, L_{k_1+k_2-1}$ des langages réguliers acceptés respectivement par les AFD $A_1, \ldots, A_{k_1+k_2-1}$. On a

$$\begin{aligned}\otimes \circ_j \oplus(L_1, \ldots, L_{k_1+k_2-1}) &= \otimes\,(L_1, \ldots, L_{j-1}, \oplus(L_j, \ldots, L_{j+k_2-1}), L_{j+k_2}, \ldots, L_{k_1+k_2-1}) \\ &\quad \text{par définition de } \circ_j \\ &= L(m_1(A_1, \ldots, A_{j-1}, m_2(A_j, \ldots, A_{j+k_2-1}), A_{j+k_2}, \ldots, A_{k_1+k_2-1})) \\ &\quad \text{car les opérations } \otimes \text{ et } \oplus \text{ sont descriptibles} \\ &= L(m_1 \circ_j m_2(A_1, \ldots, A_{k_1+k_2-1})) \text{ par définition de } \circ_j\end{aligned}$$

Vu la Proposition 2.2.1, $m_1 \circ_j m_2$ est un modificateur. Ainsi, $\otimes \circ_j \oplus$ est descriptible. □

Exemple 2.4.6 (Modificateur étoile de l'intersection)**.** Le 2-modificateur étoile de l'intersection est défini pour toute paire d'AFD $A_1 = (\Sigma, Q_1, i_1, f_1, \delta_1), A_2 = (\Sigma, Q_2, i_2, f_2, \delta_2)$ par

$\texttt{Star} \circ \texttt{Inter}(A_1, A_2) = (\Sigma, 2^{Q_1 \times Q_2}, \emptyset, \{E \in 2^{Q_1 \times Q_2} | E \cap (F_1 \times F_2) \neq \emptyset\} \cup \{\emptyset\}, \delta)$ tel que

pour tout $a \in \Sigma$

$$\delta^a(\emptyset) = \begin{cases} \{(\delta_1^a(i_1), \delta_2^a(i_2)), (i_1, i_2)\} & \text{si } \{(\delta_1^a(i_1), \delta_2^a(i_2))\} \in F_1 \times F_2 \\ \{(\delta_1^a(i_1), \delta_2^a(i_2))\} & \text{sinon.} \end{cases}$$

et pour tout $E \neq \emptyset$,

$$\delta^a(E) = \begin{cases} (\delta_1^a, \delta_2^a)(E) \cup \{(i_1, i_2)\} & \text{si } (\delta_1^a, \delta_2^a)(E) \cap F_1 \times F_2 \neq \emptyset \\ (\delta_1^a, \delta_2^a)(E) & \text{sinon.} \end{cases}$$

Par le Corollaire 2.4.3, pour toute paire de langages réguliers (L_1, L_2) définis sur le même alphabet, et toute paire d'automates déterministes complets $\underline{A} = (A_1, A_2)$ telle que $L_1 = L(A_1)$ et $L_2 = L(A_2)$, on obtient $(L_1 \cap L_2)^* = L((\texttt{Star} \circ \texttt{Inter})\underline{A})$.

2.5 Modificateurs 1-uniformes

Nous allons définir les modificateurs 1-uniformes qui sont des modificateurs qui peuvent être associés de manière naturelle avec une opération régulière. De plus, nous verrons dans le chapitre suivant un théorème qui permet de lier ce type de modificateurs avec les opérations 1-uniformes.

2.5.1 Définition

Définition 2.5.1 (modificateur 1-uniforme)**.** Un k-modificateur m est 1-uniforme si, pour chaque paire de k-uplet d'automates finis déterministes $(A_1, \ldots, A_k)$ et $(B_1, \ldots, B_k)$ telle que pour tout $j \in \{1, \ldots, k\}$ $L(A_j) = L(B_j)$, on a $L(m(A_1, \ldots, A_k)) = L(m(B_1, \ldots, B_k))$. Dans ce cas, il existe une opération régulière $\otimes_m$ telle que, pour tous les k-uplets d'automates finis déterministes $(A_1, \ldots, A_k)$, $\otimes_m(L(A_1), \ldots, L(A_k)) = L(m\underline{A})$. Par conséquent, l'opération $\otimes_m$ est descriptible et on dit que m décrit l'opération $\otimes_m$.

2.5.2 Modificateurs non uniformes

Il existe des k-modificateurs qui ne sont pas uniformes et donc qui ne peuvent pas être associés à des opérations.

Exemple 2.5.1. Considérons le modificateur $\texttt{Fto1}=(\texttt{Q},\texttt{i},\texttt{f},\rho)$ tel que

- $\texttt{Q}Q = Q$,
- $\texttt{i}(Q, i, F) = i$,
- $\texttt{f}(Q, i, F) = F$,
- $\rho(i, F, \delta_1^a)(q) = \begin{cases} \delta_1^a(q) \text{ si } q \notin F \\ \begin{cases} 1 \text{ si } 1 \in Q & \text{si } q \in F \\ \delta_1^a(q) \text{ sinon} \end{cases} \end{cases}$

Si A_1 et A_1' sont deux automates déterministes reconnaissant le même langage, alors nous avons en général $L(\texttt{Fto1}(A_1)) \neq L(\texttt{Fto1}(A_1'))$ parce que le langage accepté dépend des étiquettes des états de A_1 et A_1'.

Par exemple, considérons les automates $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1) = (\{a\}, \{0, 1, 2\}, 0, \{2\}, \delta_1)$ et $A_2 = (\Sigma, Q_2, i_2, F_2, \delta_2) = (\{a\}, \{0, 1, 2\}, 0, \{1\}, \delta_2)$ représentés aux Figures 2.13 et 2.14. Ces automates reconnaissent le même langage a^2a^*.

Cependant, si on applique $\texttt{Fto1}$ au premier, on obtient l'automate représenté à la Figure 2.15. En effet, on a

- $\texttt{Q}Q_1 = \{0, 1, 2\}$,

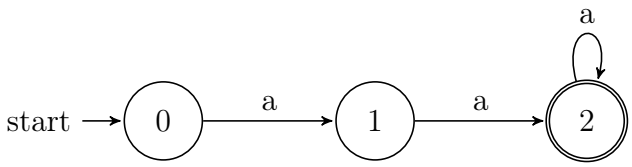


FIGURE 2.13 – Automate A_1

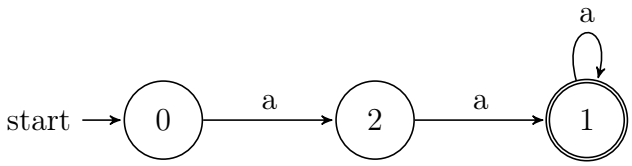


FIGURE 2.14 – Automate A_2

— $\mathtt{i}(Q_1, i_1, F_1) = 0$,
— $\mathtt{f}(Q_1, i_1, F_1) = \{2\}$,
—
  - $\rho(Q_1, F_1, \delta_1^a)(0) = 1$ car $0 \notin F_1$
  - $\rho(Q_1, F_1, \delta_1^a)(1) = 2$ car $1 \notin F_1$
  - $\rho(Q_1, F_1, \delta_1^a)(2) = 1$ car $2 \in F_1$ et $1 \in Q_1$

L'automate $\mathtt{Fto1}(A_1)$ reconnait le langage $(aa)^+$.

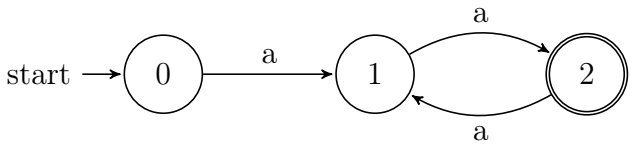


FIGURE 2.15 – Automate $\mathtt{Fto1}(A_1)$

Alors que `Fto1` laisse le second automate inchangé. En effet, on a
— $\mathtt{Q}Q_2 = \{0, 1, 2\}$,
— $\mathtt{i}(Q_2, i_2, F_2) = 0$,
— $\mathtt{f}(Q_2, i_2, F_2) = \{1\}$,
—
  - $\rho(Q_2, F_2, \delta_2^a)(0) = 2$ car $0 \notin F_2$,
  - $\rho(Q_2, F_2, \delta_2^a)(2) = 1$ car $2 \notin F_2$,
  - $\rho(Q_2, F_2, \delta_2^a)(2) = 1$ car $1 \in F_2$ et $1 \in Q_2$.

2.5.3 Composition de modificateurs 1-uniformes

Nous avons d'abord vu que la composition d'opérations 1-uniformes est une opération 1-uniforme. Ensuite, nous avons défini la composition de modificateurs dans la Section 2.2.2 et on a vu que la composition de deux modificateurs est un modificateur. Après ça, on a également vu que si deux opérations $\otimes$ et $\oplus$ sont m_1- et m_2- descriptibles alors $\otimes \circ_j \oplus$ est $m_1 \circ_j m_2$- descriptible. Maintenant, nous allons montrer que la 1-uniformité des modificateurs est stable par composition.

Proposition 2.5.1. *Si deux modificateurs m_1, m_2 sont 1-uniformes, alors la composition de m_1 et m_2, $m_1 \circ_j m_2$, est aussi 1-uniforme.*

Démonstration. Soient m_1 un modificateur k_1-aire 1-uniforme et m_2 un modificateur k_2-aire 1-uniforme. Soient $L_1, \ldots, L_{k_1+k_2-1}$ des langages réguliers acceptés respectivement par les AFDs $A_1, \ldots, A_{k_1+k_2-1}$.

On a

$$m_1 \circ_j m_2(A_1, \ldots, A_{k_1+k_2-1}) = m_1(A_1, \ldots, A_{j-1}, m_2(A_j, \ldots, A_{j+k_2-1}), A_{j+k_2}, \ldots, A_{k_1+k_2-1}).$$

Par conséquent, $m_1 \circ_j m_2$ est bien 1-uniforme si m_1 et m_2 le sont.

□

Chapitre 3

Les monstres

Nous allons définir des AFDs particuliers avec de grands alphabets appelés monstres. L'idée est de définir des k-uplets d'AFDs sur le même alphabet afin que cet alphabet soit aussi vaste que possible. Chaque k-uplet possible de fonctions de transition d'un monstre doit correspondre à une seule lettre de son alphabet. Ceci nous permettra d'avoir la plus grande flexibilité possible pour prouver des résultats sur l'accessibilité et la distinguabilité lorsqu'on minimisera l'AFD de sortie d'opérations 1-uniformes. Plus tard, nous verrons qu'une opération 1-uniforme a toujours un témoin qui est un monstre. Dans l'article [5], les monstres sont appelés témoins OLPA où OLPA signifie "one letter per action".

3.1 Définition

Dans le cas unaire, les monstres (1-monstres) de taille n sont des AFD minimaux ayant n^n lettres qui représentent chaque fonction de $[\![n]\!]$ dans $[\![n]\!]$. Il y a 2^n automates 1-monstre différents qui dépendent de l'ensemble de leurs états finaux. Un k-monstre est un k-uplet d'AFDs qui utilise l'ensemble des k-uplets de transformations comme alphabet. En effet, l'alphabet d'un monstre k-aire doit encoder toutes les transformations agissant sur chaque ensemble d'états indépendamment les uns des autres.

Définition 3.1.1 (Monstre)**.** Un k-monstre est un k-uplet d'automates $\underline{Mon}_{\underline{n},\underline{F}} = (M_1, \ldots, M_k)$ où chaque $M_j = (\Gamma_{\underline{n}}, [\![n_j]\!], 0, F_j, \delta_j)$ est défini par

- l'alphabet commun $\Gamma_{\underline{n}} = [\![n_1]\!]^{[\![n_1]\!]} \times [\![n_2]\!]^{[\![n_2]\!]} \times \cdots \times [\![n_k]\!]^{[\![n_k]\!]}$,
- l'ensemble d'états $[\![n_j]\!]$,
- l'état initial 0,
- l'ensemble des états finaux F_j,
- la fonction de transition δ_j définie $\forall q \in [\![n_j]\!], \forall \underline{g} = (g_1, \ldots, g_k) \in \Gamma_{\underline{n}}$ par $\delta_j(q, \underline{g}) = g_j(q)$, c'est-à-dire $\underline{\delta^{\underline{g}}} = \underline{g}$.

Un k-uplet de langages $(L_1, \ldots, L_k)$ est appelé k-langage monstre s'il existe un k-monstre $(M_1, \ldots, M_k)$ tel que $(L_1, \ldots, L_k) = (L(M_1), \ldots, L(M_k))$.

Notation 3.1.1. Notons $\underline{Mon}^n$ le k-monstre $\underline{Mon}_{\underline{n,F}}$ où $F_j = \{n_j - 1\} \forall j$.

Exemple 3.1.1. Le 1-monstre $Mon_{2,\{1\}}$ (ou $Mon^{(2)}$) est l'automate suivant avec a=[01],

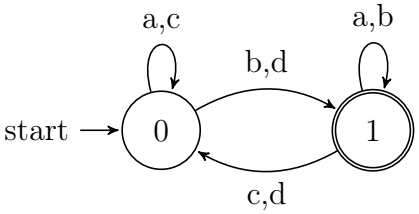


FIGURE 3.1 – 1-monstre

b=[11], c=[00], d=[10] où, pour tous i,j $\in \{0, 1\}$, l'étiquette [ij] représente la transformation qui envoie 0 sur i et 1 sur j.

Il s'agit bien du 1-monstre $Mon_{2,\{1\}}$ car on a :

— l'alphabet $\Gamma_2 = [\![2]\!]^{[\![2]\!]}$,

— l'ensemble d'états $[\![2]\!]$,

— l'état initial est 0,

— l'ensemble des états finaux $\{1\}$,

— si δ est la fonction de transition alors on a :

  — $\delta(0, a) = a(0) = [01](0) = 0$ car [01] envoie 0 sur 0,

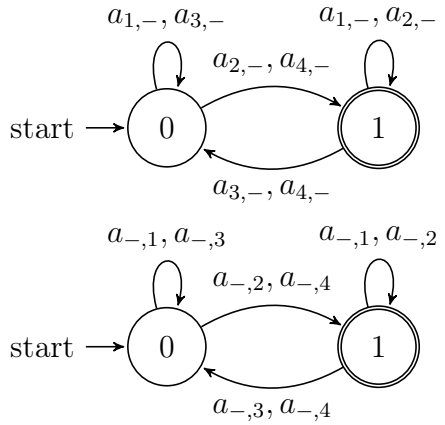
  — $\delta(0, b) = b(0) = [11](0) = 1$ car [11] envoie 0 sur 1,

  — $\delta(0, c) = c(0) = [00](0) = 0$ car [00] envoie 0 sur 0,

  — $\delta(0, d) = d(0) = [10](0) = 1$ car [10] envoie 0 sur 1.

  Il en va de même pour les transitions qui démarrent de l'état 1.

Exemple 3.1.2. Le 2-monstre $\underline{Mon}_{(2,2),(\{1\},\{1\})}$ (ou $\underline{Mon}^{(2,2)}$) est donné par la paire d'automates ci-dessous qui est définie sur un alphabet contenant $2^2 \times 2^2 = 16$ symboles où $a_{i,-}$ (respectivement $a_{-,j}$) représente l'ensemble des transitions $a_{i,x}$ (respectivement $a_{x,j}$) pour $x \in \{1, 2, 3, 4\}$.



Chaque symbole code une paire de fonctions. On a

$$\begin{array}{llll} a_{1,1} = [01, 01] & a_{1,2} = [01, 11] & a_{1,3} = [01, 00] & a_{1,4} = [01, 10] \\ a_{2,1} = [11, 01] & a_{2,2} = [11, 11] & a_{2,3} = [11, 00] & a_{2,4} = [11, 10] \\ a_{3,1} = [00, 01] & a_{3,2} = [00, 11] & a_{3,3} = [00, 00] & a_{3,4} = [00, 10] \\ a_{4,1} = [10, 01] & a_{4,2} = [10, 11] & a_{4,3} = [10, 00] & a_{4,4} = [10, 10] \end{array}$$

Ici, $a_{1,2} = [01, 11]$ signifie que le symbole $a_{1,2}$ étiquette une transition de 0 à 0 et une transition de 1 à 1 dans le premier automate et une transition de 0 à 1 et une transition de 1 à 1 dans le second automate. Il en va de même pour les autres symboles.

Remarquons que les monstres diffèrent les uns des autres seulement par leur taille et par les états finaux de leurs AFDs. Ainsi, lorsqu'on les utilisera comme témoins, nous aurons seulement besoin de discuter leurs états finaux.

3.2 Lien entre les opérations et les modificateurs 1-uniformes

Maintenant qu'on a défini les monstres, on va pouvoir les utiliser pour démontrer le théorème suivant. Celui-ci met en lumière la correspondance entre les opérations 1-uniformes et les modificateurs 1-uniformes. Un modificateur 1-uniforme décrit toujours une opération 1-uniforme, et chaque opération 1-uniforme est décrite par un modificateur 1-uniforme.

Lemme 3.2.1. *Soit $(L_1, \ldots, L_k)$ un k-uplet de langages réguliers sur le même alphabet, et soit $(A_1, \ldots, A_k)$ un k-uplet de AFDs sur le même alphabet, tel que A_j satisfait les propriétés suivantes pour tout $j \in \{1, \ldots, k\}$:*

— *A_j reconnait le langage L_j,*

— *l'ensemble des états de A_j est $[\![n_j]\!]$ pour un certain entier n_j,*
— *l'état initial de A_j est 0.*

Supposons que $(\Sigma, [\![n_j]\!], 0, F_j, \delta_j)$ dénote A_j et $(M_1, \ldots, M_k)$ dénote $Mon_{\underline{n},\underline{F}}$. De plus, supposons que ϕ dénote le morphisme 1-uniforme de Σ dans $\Gamma_{\underline{n}}$ tel que, pour tout $a \in \Sigma$, on a $\phi(a) = (\delta_1^a, \delta_2^a, \ldots, \delta_k^a)$. Pour tout $j \in \{1, \ldots, k\}$, le langage L_j est la préimage de M_j par le morphisme 1-uniforme ϕ, c'est-à-dire, on a

$$(L_1, \ldots, L_k) = (\phi^{-1}(L(M_1)), \ldots, \phi^{-1}(L(M_k))).$$

Démonstration. Soit j un entier de $\{1, \ldots, k\}$. Par la Définition 3.1.1, la fonction de transition ϵ_j de M_j satisfait $\epsilon_j^{(\delta_1^a, \ldots, \delta_k^a)} = \delta_j^a$. Ainsi, par définition de ϕ^{-1}, un mot est dans $\phi^{-1}(L(M_j))$ si et seulement si il est reconnu par l'AFD $B_j = (\Sigma, [\![n_j]\!], 0, F_j, \zeta_j)$, avec, pour tout $l \in [\![n_j]\!]$ et tout $a \in \Sigma$, on a

$$\zeta_j^a = \epsilon_j^{\phi(a)} = \epsilon_j^{(\delta_1^a, \ldots, \delta_k^a)} = \delta_j^a.$$

Pour conclure, $A_j = B_j$ et $L_j = \phi^{-1}(L(M_j))$, pour tout $j \in \{1, \ldots, k\}$.

□

Théorème 3.2.1. Une opération k-aire $\otimes$ est 1-uniforme si et seulement si il existe un k-modificateur m tel que $\otimes = \otimes_m$.

Démonstration. Soit $\otimes$ une opération unaire 1-uniforme. On définit un 1-modificateur m comme suit. Pour tout AFD $A = (\Sigma, Q_A, i_A, F_A, \delta_A)$ possédant n états, on peut renommer son ensemble d'états pour que A devienne l'AFD $D = (\Sigma, [\![n]\!], 0, F, \delta)$.

Notons $B = ([\![n]\!]^{[\![n]\!]}, Q', i', F', \delta')$ l'AFD minimal de $\otimes(L(Mon_{n,F}))$. On pose $m(A) = (\Sigma, Q', i', F', \delta'_1)$, avec $\delta'_1(q, a) = \delta'(q, \delta^a)$. Remarquons que m est bien un 1-modificateur :

— (Q', i', F') dépend seulement de (Q_A, i_1, F_A),
— $\delta_1'^a$ dépend seulement de δ^a et de δ', qui à son tour dépend seulement de (Q_A, i_A, F_A) et δ_1^a.

Par la Proposition 1.5.1, $L(m(A)) = \phi^{-1}(L(B))$, où ϕ est le morphisme 1-uniforme tel que $\phi(a) = \delta_D^a$ pour tout $a \in \Sigma$. Ainsi, on a $L(m(A)) = \phi^{-1}(\otimes(L(Mon_{n,F})))$. Et vu que $\otimes$ est 1-uniforme, on obtient $L(m(A)) = \otimes(\phi^{-1}(L(Mon_{n,F})) = \otimes(L)$ vu le Lemme 3.2.1.

Le cas k-aire avec $k > 1$ se montre de manière analogue. □

3.3 Calcul de la complexité en états

Le résultat suivant est majeur, il permettra de concevoir une méthode pour calculer la complexité en états des opérations descriptibles en utilisant les monstres.

Si une opération est descriptible, il est suffisant d'étudier le comportement de son modificateur sur des monstres pour calculer sa complexité en états.

Théorème 3.3.1. Soit m un modificateur et $\otimes$ une opération m-descriptible. Nous avons $sc_{\otimes}(\underline{n}) = max\{\#_{Min}(m\underline{Mon}_{\underline{n},\underline{F}})|\underline{F} \subset [\![n_1]\!] \times \cdots \times [\![n_k]\!]\}$

Démonstration. Soit $\underline{A}$ un k-uplet d'automates ayant $\underline{n}$ états et pour tout ensemble d'états finaux $\underline{F}$ reconnaissant un k-uplet de langages $\underline{L}$ sur un alphabet Σ. Nous pouvons supposer que $A_i = (\Sigma, [\![n_i]\!], 0, F_i, \delta_i)$ pour $i \in \{1, \ldots, k\}$ quitte à renommer les états.

Soit δ_A la fonction de transition de $m\underline{A}$, et δ_M la fonction de transition de $m\underline{Mon}_{\underline{n},\underline{F}}$. Par définition d'un modificateur, les états de $m\underline{A}$ et de $m\underline{Mon}_{\underline{n},\underline{F}}$ sont les mêmes. Pour toute lettre a et tout état q de $m\underline{A}$, nous avons

$$\delta_A^a(q) = \rho((0, \ldots, 0), \underline{F}, \underline{\delta}^a)(q) = \rho((0, .., 0), \underline{F}, \underline{\delta_M^{\delta^a}})(q) = \delta_M^{\delta^a}(q).$$

Et donc, pour tout mot w sur l'alphabet Σ :

$$\delta_A^w(q) = \delta_M^{\delta^w}(q).$$

Ainsi, tous les états accessibles dans $m\underline{A}$ sont aussi accessibles dans $m\underline{Mon}_{\underline{n},\underline{F}}$, et pour tout mot w sur l'alphabet Σ, $\delta_A^w(q) \in \mathtt{f}(([\![n_1]\!], \ldots, [n_k]\!]), (0, \ldots, 0), \underline{F})$ si et seulement si $\delta_M^{\delta^w}(q) \in \mathtt{f}(([\![n_1]\!], \ldots, [n_k]\!]), (0, \ldots, 0), \underline{F})$, ce qui implique que toutes les paires d'états distinguables dans $m\underline{A}$ sont aussi distinguables dans $m\underline{Mon}_{\underline{n},\underline{F}}$.

Par conséquent, $\#_{Min} m\underline{A} \leq \#_{Min} m\underline{Mon}_{\underline{n},\underline{F}}$. □

Exemple 3.3.1 (Modificateur miroir d'un 1-monstre)**.** Remarquons que dans l'Exemple 2.3.4, l'automate A_1 était en fait le 1-monstre Mon^2 et nous avons donc obtenu $\mathtt{Rev}(\mathrm{Mon}^2)$.

Montrons que l'automate $\mathtt{Rev}(Mon^{n_1})$ est minimal lorsque $n_1 > 1$. En effet,

— Chaque état est accessible.
Soit g_E le symbole qui envoie chaque élément d'un ensemble $E \subset [\![n_1]\!]$ sur $n_1 - 1$ et les éléments de $[\![n_1]\!] \backslash E$ sur 0. Alors, on a $\delta^{g_E}(n_1 - 1) = g_E^{-1}(n_1 - 1) = E$ (cela fonctionne aussi pour $E = \emptyset$).

— Les états ne sont pas deux à deux équivalents.
Soient E et E' deux états distincts de $\texttt{Rev}(Mon^{n_1})$ et supposons qu'il existe $i \subset E \backslash E'$. Soit g le symbole qui envoie 0 vers i et les autres états vers $j \neq i$. L'état $\delta^g(E)$ est final car $\{0\} = g^{-1}(i) \subset \delta^g(E)$ alors que $\delta^g(E')$ n'est pas final car $\delta^g(E') \subset [\![n_1]\!] \backslash \{0\}$.

Chapitre 4

Applications

Si une opération est descriptible, on peut obtenir sa complexité en états en étudiant le comportement de son modificateur sur les monstres. En effet, l'algorithme à suivre est

1. Décrire l'opération en utilisant un modificateur dont les états sont représentés par des objets combinatoires ;
2. Appliquer ce modificateur à des k-monstres bien choisis et discuter les états finaux ;
3. Minimiser l'automate obtenu et estimer sa taille.

Il ne faut pas voir cette méthode comme une règle compliquée et rapide mais plutôt comme un point de départ de recherche. Nous allons appliquer notre algorithme pour déterminer la complexité en états de quatre opérations : l'étoile, la concaténation, l'étoile d'intersection et la racine carrée. Les deux premières applications sont issues de [9] et les deux suivantes de [2]. Pour d'autres exemples, nous renvoyons le lecteur à l'article [1] dans lequel le calcul de la complexité en états de l'étoile de l'union disjointe est réalisé en utilisant également les modificateurs et les monstres. Pour des exemples n'utilisant pas ces deux outils, le lecteur peut consulter [7], [11] et [8].

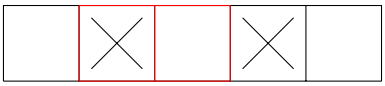
4.1 L'étoile

Soient un naturel $n \geq 2$, G un sous-ensemble de $[\![n]\!]$, et A=$(\Sigma, Q, i, F, \delta)$=$\texttt{Star}(Mon_{n,G})$. Par la Définition 2.3.6, on a $\Sigma = \Gamma_n = [\![n]\!]^{[\![n]\!]}, Q = 2^{[\![n]\!]}, i = \emptyset, F = \{E \in Q | E \cap G \neq \emptyset\} \bigcup \{\emptyset\}$ et, pour tout $E \in Q$ et tout $\phi \in \Sigma$,

$$\delta^{\phi}(E) = \begin{cases} \{\phi(0)\} & \text{si } E = \emptyset \text{ et } \phi(0) \notin G \\ \{\phi(0), 0\} & \text{si } E = \emptyset \text{ et } \phi(0) \in G \\ \phi(E) & \text{si } E \neq \emptyset \text{ et } \phi(E) \cap G = \emptyset \\ \phi(E) \cup \{0\} & \text{si } E \neq \emptyset \text{ et } \phi(E) \cap G \neq \emptyset \end{cases}$$

On peut représenter un sous-ensemble E de $[\![n]\!]$ comme une "ligne" de carrés, qui peuvent être vides ou remplis par une croix. Un carré vide en position i signifie que $i \notin E$

alors qu'un carré rempli avec une croix en position i signifie que $i \in E$. Tous les carrés représentant une position qui est dans G sont en rouge. Si $n = 5$, et $G = \{1, 2\}$, le sous-ensemble $\{1, 3\}$ de $[\![5]\!]$ est représenté avec la ligne suivante



Exemple 4.1.1. Si on utilise cette représentation sur les Figures 2.6 et 2.7, on obtient les Figures 4.1 et 4.2. Dans la Figure 4.1, les états 0 et 1 sont identifiés avec les sous-ensembles de $\{0, 1\}$: $\{0\}$ et $\{1\}$.

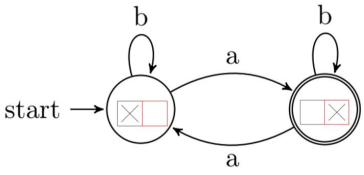


FIGURE 4.1 – AFD A

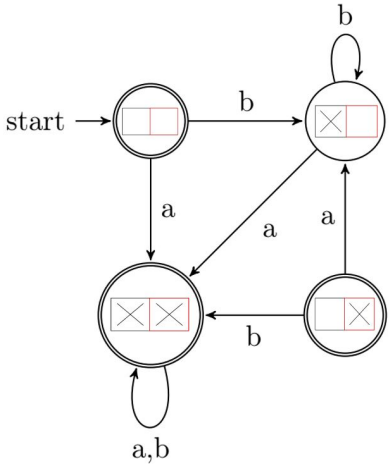


FIGURE 4.2 – AFD `Star`(A)

Cette représentation nous donne une interprétation de la fonction de transition de A. En effet, dans A, pour aller de la représentation d'un sous-ensemble E ($E \neq \emptyset$) à la représentation d'un sous-ensemble E' en lisant la lettre ϕ, il suffit de changer la position des croix de E en leurs appliquant la fonction ϕ et d'ajouter une croix au début de la ligne si et seulement si il y a une croix dans un carré rouge. Par exemple, si $n = 4, G = \{1, 2\}, \phi(1) = 2, \phi(2) = 3$, on a $\delta^{\phi}(\{1, 2\}) = \{0, 2, 3\}$, ce qui est représenté à la figure ci-dessous.

4.1.1 Une borne supérieure

Nous allons commencer par établir une borne supérieure pour la complexité en états de l'étoile de Kleene. Le raisonnement est basé sur la remarque suivante :

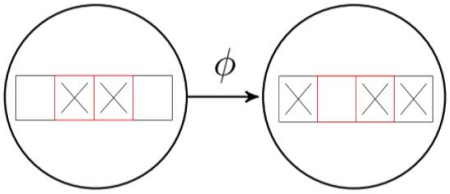


FIGURE 4.3 – Une transition $\texttt{Star}(\text{Mon}_{4,\{1,2\}})$

Remarque 4.1.1. Si lire une lettre ϕ à partir de tout état de A mène à un état avec une croix dans un carré rouge, alors cet état a aussi une croix dans le carré le plus à gauche. Plus formellement, si E est un élément de Q et ϕ une lettre de Γ_n, si $G \cap \phi(E) \neq \emptyset$, alors $0 \in \delta^{\phi}(E)$. Ainsi, tout état E de A tel que $E \cap G \neq \emptyset$ et $0 \notin E$ n'est pas accessible dans A. Par exemple, l'état à gauche dans la Figure 4.3 n'est pas accessible dans l'AFD $\texttt{Star}(Mon_{4,\{2,3\}})$.

Lemme 4.1.1. *Pour tout entier $n \geq 2$, la complexité en états $sc_{\texttt{Star}}$ de l'étoile de Kleene satisfait $sc_{\texttt{Star}}(n) \leq 2^{n-1} + 2^{n-2}$.*

Démonstration. On va distinguer plusieurs cas. En dessous de chaque cas se trouvera un exemple de représentation pour n=4 afin de mieux visualiser.

Premièrement, supposons que $G = \emptyset$. Alors A n'a pas d'états finaux et la taille de l'AFD minimal associé à A est 1.

Maintenant, montrons que dans chaque autre cas, le nombre d'états accessibles dans A est plus petit ou égal à $2^{n-1} + 2^{n-2}$.

Deuxièmement, supposons que $G = \{0\}$. Soit E un état de A qui est un singleton $\{j\}$.

— Si $\phi(j) = 0$, alors $\phi(E) \cap G \neq \emptyset$, et $\delta^{\phi}(E) = \{0\} \cup \{0\} = \{0\}$.

— Si $\phi(j) = l \neq 0$, alors $\phi(E) \cap G = \emptyset$ et $\delta^{\phi}(E) = \{\phi(j)\} = \{l\}$.

Dans les deux cas, $\delta^{\phi}(E)$ est un singleton. Ceci peut être remarqué en utilisant notre représentation. En effet, dans ce cas, ajouter une croix à une ligne avec seulement une croix via une transition est impossible. On obtient que chaque état accessible de A est un singleton. Par conséquent, le nombre d'états accessibles de A est au plus $n+1 \leq 2^{n-1}+2^{n-2}$.

Finalement, supposons que $G \notin \{\emptyset, \{0\}\}$. On va utiliser la Remarque 4.1.1 pour déduire une borne supérieure sur le nombre d'états accessibles de A. On distingue deux cas :

— Supposons que $0 \notin G$. Déterminons le nombre d'états qui ne sont pas accessibles dans A. Autrement dit, déterminons le nombre d'états E de A tels que $E \cap G \neq \emptyset$ et $0 \notin E$. Étant donné que 0 ne peut appartenir ni à E, ni à G et que E doit contenir au moins un élément de G, le nombre d'états E vaut le nombre de parties d'un ensemble de $n-1$ éléments auquel on retire le nombre de parties d'un ensemble de $n-1-\#G$ éléments, ainsi on obtient $2^{n-1} - 2^{n-1-\#G}$ états. Par conséquent, le nombre d'états accessibles de A est au plus

$$\begin{aligned} 2^n - (2^{n-1} - 2^{n-1-\#G}) =& (2^n - 2^{n-1}) + 2^{n-1-\#G} \\ =& 2^{n-1} + 2^{n-1-\#G} \\ \leq& 2^{n-1} + 2^{n-2} \text{ car } \#G \geq 1. \end{aligned}$$

— Supposons que $0 \in G$. Le nombre d'états non accessibles dans A est le nombre d'états E de A tels que $E \cap G \neq \emptyset$ et $0 \notin E$. Il vaut $2^{n-1} - 2^{n-\#G}$. Ainsi, le nombre d'états accessibles de A est au plus

$$2^n - (2^{n-1} - 2^{n-\#G}) = 2^{n-1} + 2^{n-\#G}$$

Cependant, comme $G \notin \{\emptyset, \{0\}\}$ et $0 \in G$, on a $\#G \geq 2$. Par conséquent, le nombre d'états accessibles de A est au plus

$$2^{n-1} + 2^{n-2}.$$

Ainsi, on a prouvé dans chaque cas que la taille de l'AFD minimal associé à A est plus petit ou égal à $2^{n-1} + 2^{n-2}$. Donc, par le Théorème 3.3.1, $sc_{\texttt{Star}}(n) \leq 2^{n-1} + 2^{n-2}$. □

4.1.2 Une borne inférieure

Nous allons maintenant prouver que le langage accepté par Mon^n est un témoin pour l'étoile de Kleene et que la borne supérieure est atteinte.

Lemme 4.1.2. *Soit un naturel $n \geq 2$. Si $G = \{n-1\}$, alors la taille de l'AFD minimal associé à A est $2^{n-1} + 2^{n-2}$.*

Démonstration. Soit $G = \{n-1\}$, on a $A = \texttt{Star}(\text{Mon}^n)$. Supposons que S est l'ensemble de tous les états E de A qui sont tels que si $n-1 \in E$ alors $0 \in E$.

Montrons que chaque état E de S est accessible dans A par induction sur le nombre d'éléments de E. On suit l'intuition donnée à la Figure 4.4.

L'ensemble vide est initial dans A. Chaque singleton d'éléments de $[\![n]\!]$ est dans S, à part le singleton $\{n-1\}$. De plus, si $j \in [\![n-1]\!]$, alors $\{j\}$ est accessible à partir de l'ensemble vide en lisant toute lettre ϕ telle que $\phi(0) = j$. Ainsi, tout élément E de S tel que $\#E \leq 1$ est accessible dans A.

Maintenant, soit $j \in \{1, \ldots, n-1\}$, et supposons que tout élément E de S tel que $\#E \leq j$ est accessible dans A. Soit E' un élément de S tel que $\#E' = j+1$. Si $E' = \{0, n-1\}$ alors il est accessible à partir de $\{0\}$ en lisant la lettre $(0, n-1)$. Sinon, soient l et l' deux éléments distincts de E' tels que $l \neq n-1$, et soit $E = (0,l) \circ (l', n-1)(E')$. On a $0 \in E, n-1 \in E$ et $\#E = \#E'$. De plus, E est accessible à partir de l'ensemble $E'' = E \backslash \{n-1\}$ en lisant la lettre $(0, n-1)$. Par conséquent, E' est accessible à partir de E'' en lisant la lettre $(0, n-1)$ suivie de la lettre $(0, l)$ et puis la lettre $(l', n-1)$. De plus, $\#E'' = j$ et $n-1 \notin E''$, ce qui implique que $E'' \in S$. Ainsi, E' est accessible dans A.

Donc, on a montré que chaque élément de S est accessible dans A.

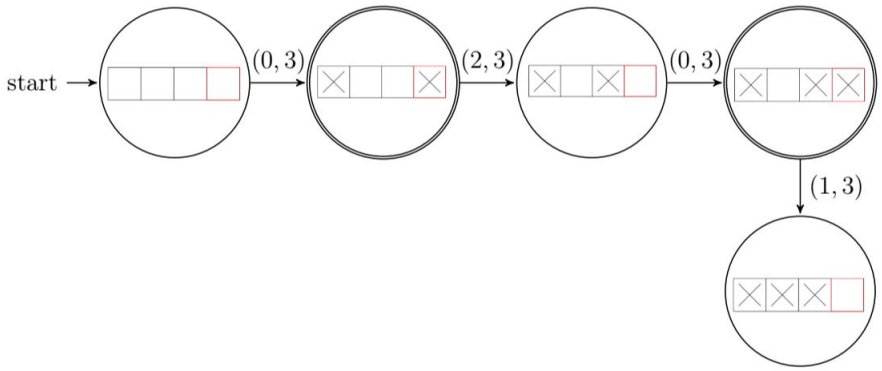


FIGURE 4.4 – Une exécution dans $\texttt{Star}(Mon_{4,\{3\}})$ à partir de l'état initial $\emptyset$ jusqu'à l'état $\{0, 1, 2\}$

Vu la Remarque 4.1.1 , les états accessibles de A sont exactement les états dans S. Nous allons montrer que les états de S sont deux à deux distinguables dans A. On suit l'intuition de la Figure 4.5.

Soient E et E' deux éléments non vides différents de S. Il existe un entier j tel que, soit

$j \in E$ et $j \notin E'$, soit $j \notin E$ et $j \in E'$. Vu que les deux cas sont symétriques, on suppose que j est un entier tel que $j \in E$ et $j \notin E'$. Soit ϕ la lettre de Γ_n telle que $\phi(j) = n-1$ et telle que, pour tout $l \in [\![n]\!]$ qui n'est pas égal à j, $\phi(l) = 0$. Lire la lettre ϕ à partir de l'état E mène à l'état $\{0, n-1\}$. De plus, lire la lettre ϕ à partir de l'état E' mène à l'état $\{0\}$. Cependant, $\{0, n-1\}$ est final dans A, alors que $\{0\}$ n'est pas final dans A. Ainsi, E et E' sont distinguables dans A.

De plus, lire le mot vide à partir de l'état $\emptyset$ dans A mène à $\emptyset$, qui est final, mais lire le mot vide à partir de l'état $\{0\}$ dans A mène à $\{0\}$, qui n'est pas final. Par conséquent, $\emptyset$ et $\{0\}$ sont distinguables dans A. Ainsi, toute paire d'états distincts de A est distinguable.

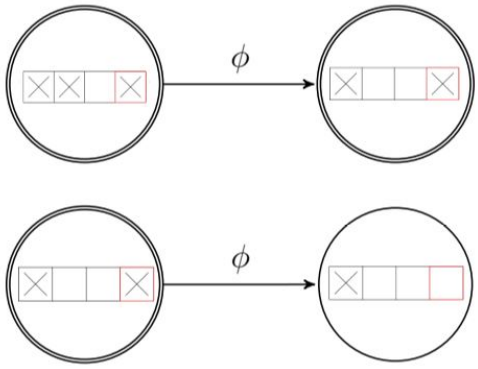


FIGURE 4.5 – Comment distinguer deux états de $\texttt{Star}(Mon_{4,\{3\}})$, avec la lettre ϕ telle que $\phi(1) = 3$, et $\phi(0) = \phi(2) = \phi(3) = 0$ (le "j" de la preuve vaut 1 dans cet exemple).

Ainsi, la taille de l'AFD minimal associé à A est de la même taille que S, qui est $2^{n-1} + 2^{n-2}$. □

Par le Lemme 4.1.1 et le Lemme 4.1.2, Mon^n est un témoin pour l'étoile de Kleene, et la borne supérieure du Lemme 4.1.1 est atteinte.

Proposition 4.1.1. *Pour tout naturel $n \geq 2$, la complexité en états $sc_{\texttt{Star}}$ de l'étoile de Kleene satisfait $sc_{\texttt{Star}} = 2^{n-1} + 2^{n-2}$.*

4.2 La concaténation

Soit (n_1, n_2) une paire de naturels plus grands ou égaux à 2, soit (F_1, F_2) une paire d'ensembles finis tels que $F_1 \subseteq [\![n_1]\!]$ et $F_2 \subseteq [\![n_2]\!]$, soit $Mon_{(n_1,n_2),(F_1,F_2)} = (M_1, M_2)$ et soit $A = (\Sigma, Q, i, F, \delta) = \texttt{Conc}(M_1, M_2)$. Par la Définition 2.3.5, on a

— $\Sigma = \Gamma_{n_1,n_2} = [\![n_1]\!]^{[\![n_1]\!]} \times [\![n_2]\!]^{[\![n_2]\!]}$,

— $Q = [\![n_1]\!] \times 2^{[\![n_2]\!]}$,

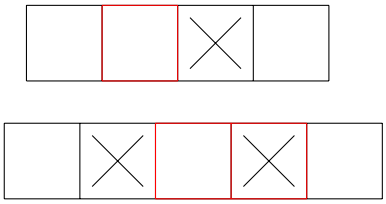
— $i = \begin{cases} (0, \emptyset) & \text{si } 0 \notin F_1 \\ (0, \{0\}) & \text{si } 0 \in F_1 \end{cases}$

— $F = \{(q_1, E) \in [\![n_1]\!] \times 2^{[\![n_2]\!]} | E \cap F_2 \neq \emptyset\}$,

— pour tout $(q_1, E) \in [\![n_1]\!] \times 2^{[\![n_2]\!]}$ et tout $(\phi_1, \phi_2) \in \Gamma_{n_1,n_2}$,

$$\delta^{(\phi_1,\phi_2)}(q_1, E) = \begin{cases} (\phi_1(q_1), \phi_2(E)) & \text{si } \phi_1(q_1) \notin F_1 \\ (\phi_1(q_1), \phi_2(E) \cup \{0\}) & \text{si } \phi_1(q_1) \in F_1. \end{cases}$$

Pour représenter les états de A, on va utiliser une représentation similaire au cas de l'étoile de Kleene. Cependant, cette fois-ci, un état (q, E) de A est représenté par deux "lignes" de carrés. La première représente q, et donc a exactement une croix. La seconde représente E, et donc a autant de croix que la taille de E. Par exemple, si $n_1 = 4, n_2 = 5, F_1 = 1$ et $F_2 = \{2, 3\}$, l'état $\{2, \{1, 3\}\}$ de A est représenté par la paire de lignes ci-dessous :



Cette représentation nous donne une interprétation de la fonction de transition de A. En effet, dans A, pour aller de l'état (q, E) à l'état (q', E') en lisant une lettre (ϕ_1, ϕ_2) (dans le cas où $E \neq \emptyset$), on peut changer les positions des croix des deux lignes représentant (q, E) en appliquant ϕ_1 à la croix de la première ligne et ϕ_2 aux croix de la seconde ligne et après ça il ne reste plus qu'à ajouter une croix au début de la seconde ligne si et seulement si la croix de la première ligne est dans un carré rouge. Par exemple, si $n_1 = 3, n_2 = 4, F_1 = \{2\}, F_2 = \{1\}, \phi_1(1) = 2, \phi_2(1) = 2, \phi_2(2) = 3$, on a $\delta^{\phi}((1, \{1, 2\})) = (2, \{0, 2, 3\})$, ce qui est représenté à la figure ci-dessous.

4.2.1 Une borne supérieure

On commence par établir une borne supérieure pour la complexité en états de la concaténation. Notre raisonnement sera basé sur la remarque suivante :

Remarque 4.2.1. Si lire une lettre à partir d'un état de A mène à un état dont la croix de la première ligne se trouve dans un carré rouge, alors la seconde ligne de cet état a une croix dans le carré le plus à gauche. Plus formellement, si (q, E) est un élément de Q, si

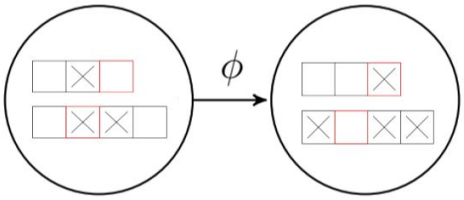


FIGURE 4.6 – Une transition dans $\texttt{Conc}(\text{Mon}_{(3,4),(\{2\},\{1\})})$

(ϕ_1, ϕ_2) est une lettre de Γ_{n_1,n_2}, et si $\phi_1(q) \in F_1$, alors en notant (q', E') l'état $\delta^{(\phi_1,\phi_2)}(q, E)$, on a $0 \in E'$. Ainsi, tout état (q, E) de A tel que $q \in F_1$ et $0 \notin E$ n'est pas accessible dans A.

Lemme 4.2.1. *La complexité en états $sc_{\texttt{Conc}}$ de la concaténation satisfait*

$$sc_{\texttt{Conc}}(n_1, n_2) \leq (n_1 - 1)2^{n_2} + 2^{n_2-1}.$$

Démonstration. Si $F_1 = \emptyset$, tout état (q, E) de A tel que $E \neq \emptyset$ n'est pas accessible. Ainsi, le nombre d'états accessibles de A est plus petit ou égal à n_1.

Supposons que $F_1 \neq \emptyset$. Le nombre d'états accessibles dans A est le nombre des états (q, E) de A tels que, si $q \in F_1$ alors $0 \in E$. Ce nombre vaut

$$\begin{aligned}
(n_1 - \#F_1)2^{n_2} + \#F_1 \cdot 2^{n_2-1} &= n_1 \cdot 2^{n_2} - \#F_1(2^{n_2} - 2^{n_2-2}) \\
&= n_1 \cdot 2^{n_2} - \#F_1 \cdot 2^{n_2-1} \\
&= (n_1 - 1)2^{n_2} + 2^{n_2} - \#F_1 \cdot 2^{n_2-1} \\
&= (n_1 - 1)2^{n_2} + 2^{n_2-1}(2 - \#F_1) \\
&\leq (n_1 - 1)2^{n_2} + 2^{n_2-1} \text{ car } (2 - \#F_1) \leq 1
\end{aligned}$$

Dans tous les cas, le nombre d'états accessibles de A est plus petit ou égal à

$$(n_1 - 1)2^{n_2} + 2^{n_2-1}.$$

Ainsi, par le Théorème 3.3.1, on a $sc_{\texttt{Conc}}(n_1, n_2) \leq (n_1 - 1)2^{n_2} + 2^{n_2-1}$.

□

4.2.2 Une borne inférieure

On va maintenant prouver que $Mon_{(n_1,n_2),(\{n_1-1\},\{n_2-1\})}$ est un témoin pour la concaténation, et que la borne supérieure ci-dessus est atteinte.

Lemme 4.2.2. *Si $F_1 = \{n_1 - 1\}$ et $F_2 = \{n_2 - 1\}$, alors la taille de l'AFD minimal associé à A est $(n_1 - 1)2^{n_2} + 2^{n_2 - 1}$.*

Démonstration. Rappelons que n_1 et n_2 sont tous les deux plus grands ou égaux à 2. Soient $F_1 = \{n_1 - 1\}$ et $F_2 = \{n_2 - 1\}$, et donc $A = \texttt{Conc}(Mon_{(n_1,n_2),(\{n_1-1\},\{n_2-1\})})$. Soit S l'ensemble de tous les états (j, E) de A tels que, si $j = n_1 - 1$, alors $0 \in E$. Montrons que chaque état (j, E) de S est accessible dans A par induction sur le nombre d'éléments de E. On va suivre l'intuition donnée à la Figure 4.7.

L'état $(0, \emptyset)$ est initial dans A. Un état $(j, \emptyset)$ est dans S si et seulement si $j \in [\![n_1 - 1]\!]$. Cependant, si $j \in [\![n_1 - 1]\!]$, $(j, \emptyset)$ est accessible à partir de $(0, \emptyset)$ en lisant la lettre $((0, j), Id)$. Ainsi, tout élément $(j, \emptyset)$ de S est accessible dans A.

L'état $(n_1 - 1, \{0\})$ est accessible dans A à partir de $(0, \emptyset)$ en lisant la lettre $((0, n_1 - 1), Id)$. De plus, tout état $(j, \{m\})$ de S, où $j \in [\![n_1 - 1]\!]$ et $m \in [\![n_2]\!]$ est atteint à partir de l'état $(n_1 - 1, \{0\})$ en lisant la lettre $((n_1 - 1, j), (0, m))$. Donc, tout état (j, E) de S avec $\#E \leq 1$ est accessible dans A.

Maintenant, soit $l \in \{1, \ldots, n_2 - 1\}$, et supposons que chaque élément (j, E) de S tel que $\#E \leq l$ est accessible dans A. Soit (j', E') un éléments de S tel que $\#E' = l + 1$. Soit r un élément de E', soit r' un élément non nul de $(0, r)(E')$, et soit $E = (0, r)(E')\backslash\{r'\}$. L'état (j', E') est atteint dans A à partir de l'état $(n_1 - 1, E)$ en lisant la lettre $(Id_{[\![n_1-1]\!]}, (0, r'))$ et après la lettre $((n_1 - 1, j'), (0, r))$. De plus, $0 \in E$, ce qui implique que $(n_1 - 1, E)$ est dans S, et $\#E = l$. Par conséquent, (j', E') est accessible dans A. Ainsi, on a montré par induction que chaque élément de S est accessible dans A.

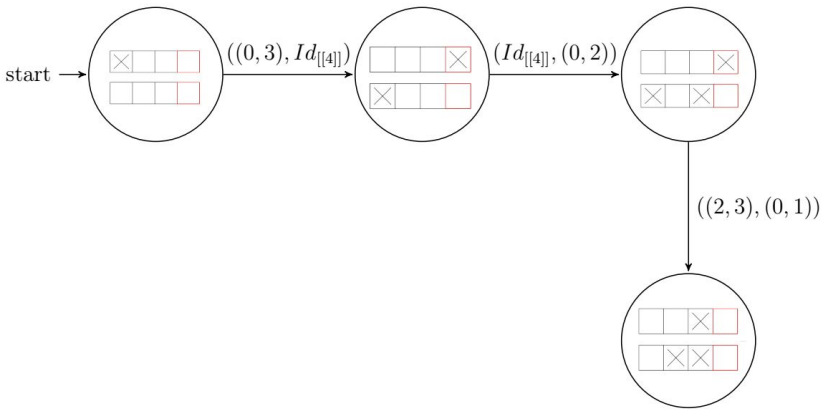


FIGURE 4.7 – Une exécution dans $\texttt{Conc}(Mon_{(4,4),(\{3\},\{3\})})$ à partir de l'état initial $(0, \emptyset)$ jusqu'à l'état $(2, \{1, 2\})$

Vu la Remarque 4.3.1, les états accessibles de A sont exactement les états de S.

Maintenant, montrons que les états sont deux à deux distincts dans A.
Soient (j, E) et (j', E') deux éléments différents de S. On distingue deux cas et on suit l'intuition donnée aux Figures 4.8 et 4.9.

— Premièrement, supposons que $j \neq j'$. Soit ϕ_1 une fonction de $[\![n_1]\!]^{[\![n_1]\!]}$ telle que $\phi_1(j) = 0$ et $\phi_1(j') = n_1 - 1$ et soit ϕ_2 la fonction de $[\![n_2]\!]^{[\![n_2]\!]}$ telle que $\phi_2(l) = n_2 - 1$, pour tout $l \in [\![n_2]\!]$. Lire la lettre (ϕ_1, ϕ_2) à partir de l'état (j, E) mène à l'état $(0, \{n_2 - 1\})$. De plus, lire la lettre (ϕ_1, ϕ_2) à partir de l'état (j', E') mène à l'état $(n_1 - 1, \{0, n_2 - 1\})$. Ainsi, si $\phi_3 = Id_{[\![n_1]\!]}$ et $\phi_4 = (0, n_2 - 1)$, on a $\delta^{(\phi_1,\phi_2)(\phi_3,\phi_4)}(j, E) = (0, \{0\})$ et $\delta^{(\phi_1,\phi_2)(\phi_3,\phi_4)}(j', E') = (n_1 - 1, \{0, n_2 - 1\})$.
Cependant, $(n_1 - 1, \{0, n_2 - 1\})$ est final dans A alors que $(0, \{0\})$ ne l'est pas vu que $n_2 \geq 2$. Ainsi, (j, E) et (j', E') sont distinguables dans A.

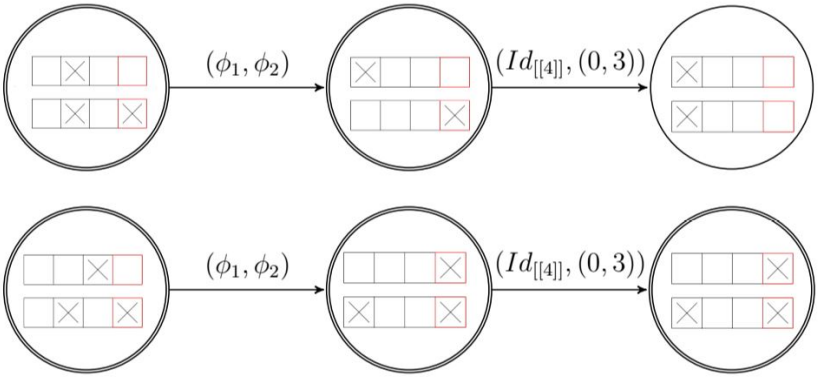


FIGURE 4.8 – Comment distinguer les deux états $(1, \{1, 3\})$ et $(2, \{1, 3\})$ de $\texttt{Conc}(Mon_{(4,4),(\{3\},\{3\})})$ où $j \neq j'$, avec $\phi_1(2) = 3$, $\phi_1(1) = 0$ et $\phi_2(1) = \phi_2(3) = 3$.

— Supposons que $E \neq E'$. Il existe un entier j tel que, soit $j \in E$ et $j \notin E'$, soit $j \notin E$ et $j \in E'$. Comme les deux cas sont symétriques, on suppose que j est un entier tel que $j \in E$ et $j \notin E'$. Soit ϕ_1 la fonction de $[\![n_1]\!]^{[\![n_1]\!]}$ telle que $\phi_1(l) = 0$ pour tout $l \in [\![n_1]\!]$. Soit ϕ_2 la fonction de $[\![n_2]\!]^{[\![n_2]\!]}$ telle que $\phi_2(j) = n_2 - 1$, et telle que pour tout $l \in [\![n_2]\!]$ qui n'est pas égal à j, $\phi_2(l) = 0$. Lire la lettre (ϕ_1, ϕ_2) depuis l'état (j, E) mène à l'état $(0, \{n_2 - 1\})$ ou à l'état $(0, \{0, n_2 - 1\})$. De plus, lire la lettre (ϕ_1, ϕ_2) à partir de l'état (j', E') mène à l'état $(0, \{0\})$. Cependant, $(0, \{n_2 - 1\})$ et $(0, \{0, n_2 - 1\})$ sont finaux dans A, alors que $(0, \{0\})$ n'est pas final dans A. Ainsi, (j, E) et (j', E') sont distinguables dans A.

On a montré que les états de S sont distinguables deux à deux dans A. Ainsi, la taille de l'AFD minimal associé à A est la taille de S, qui est $(n_1 - 1)2^{n_2} + 2^{n_2-1}$. □

Par conséquent, par le Lemme 4.2.1 et le Lemme 4.2.2, $Mon_{n_1,n_2}^{\{n_1-1\},\{n_2-1\}}$ est un témoin

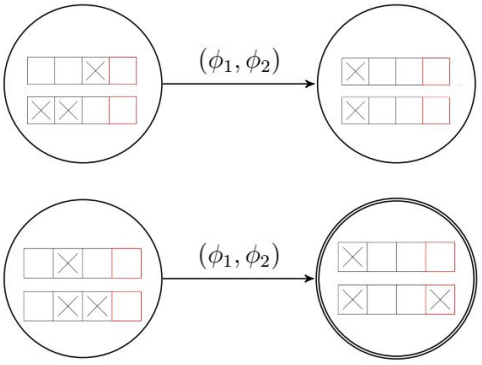


FIGURE 4.9 – Comment distinguer deux états (j, E) et (j', E') de $\texttt{Conc}(Mon_{4,4}^{\{3\},\{3\}})$ quand $E \neq E'$, avec $\phi_1(1) = \phi_1(2) = 0$, $\phi_2(1) = \phi_2(0) = 0$ et $\phi_2(2) = 3$.

pour la concaténation et la borne supérieure du Lemme 4.2.1 est atteinte.

Proposition 4.2.1. *Pour toute paire d'entiers positifs (n_1, n_2) avec $n_2 \geq 2$, la complexité en états de la concaténation $sc_{\texttt{Conc}}$ satisfait $sc_{\texttt{Conc}}(n_1, n_2) = (n_1 - 1)2^{n_2} + 2^{n_2 - 1}$.*

4.3 L'étoile de l'intersection

Nous allons appliquer notre méthode à l'opération étoile de l'intersection.

Les éléments de $2^{[\![n_1]\!]\times[\![n_2]\!]}$ vont être vus comme des matrices booléennes de taille $n_1 \times n_2$. Une matrice de ce type est appelée tableau. On écrit $\mathrm{T}_{x,y}$ pour la valeur du tableau T à la ligne x et à la colonne y. Et #T représente le nombre de 1 dans le tableau.

Soient (n_1, n_2) une paire de naturels, soit (F_1, F_2) un sous-ensemble de $[\![n_1]\!] \times [\![n_2]\!]$, soit $Mon_{(n_1,n_2),(F_1,F_2)} = (M_1, M_2)$ et soit $A = (\Sigma, Q, i, F, \delta) = (\texttt{Star} \circ \texttt{Inter})(M_1, M_2)$. En utilisant les Définitions 2.3.3, 2.3.6, on a

— $\Sigma = \Gamma_{n_1,n_2} = [\![n_1]\!]^{[\![n_1]\!]} \times [\![n_1]\!]^{[\![n_1]\!]}$,

— $Q = 2^{[\![n_1]\!]\times[\![n_2]\!]}$

— $i = \emptyset$

— $F = \{E \in 2^{[\![n_1]\!]\times[\![n_2]\!]} | E \cap (F_1 \times F_2) \neq \emptyset\} \cup \{\emptyset\}$

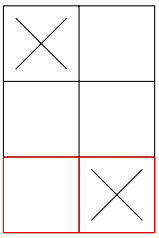
— pour tout $a \in \Sigma$

$$\delta^a(\emptyset) = \begin{cases} \{(\delta_1^a(0), \delta_2^a(0)), (0,0)\} & \text{si } \{(\delta_1^a(0), \delta_2^a(0))\} \in F_1 \times F_2 \\ \{(\delta_1^a(0), \delta_2^a(0))\} & \text{sinon.} \end{cases}$$

et pour tout $E \neq \emptyset$,

$$\delta^a(E) = \begin{cases} (\delta_1^a, \delta_2^a)(E) \cup \{(0,0)\} & \text{si } (\delta_1^a, \delta_2^a)(E) \cap F_1 \times F_2 \neq \emptyset \\ (\delta_1^a, \delta_2^a)(E) & \text{sinon.} \end{cases}$$

Nous allons représenter un sous-ensemble E de $[\![n_1]\!] \times [\![n_2]\!]$ par un tableau où chaque carré est soit vide, soit rempli d'une croix. Un carré vide en position (i, j) signifie que $(i, j) \notin E$ (i.e. $T_{i,j} = 0$), si par contre ce carré est rempli alors $(i, j) \in E$ (i.e. $T_{i,j} = 1$). Tous les carrés qui représentent une position qui est dans $F_1 \times F_2$ sont en rouge. Par exemple, si $n_1 = 3, n_2 = 2, F_1 \times F_2 = \{(2,0), (2,1)\}$, le sous ensemble $\{(0,0), (2,1)\}$ est représenté par le tableau ci-dessous :



On obtient une interprétation de la fonction de transition de A. Dans A, pour aller de l'état E à l'état E' en lisant (ϕ_1, ϕ_2), on peut changer les positions des croix représentant

E en appliquant ϕ_1 sur les lignes de E et ϕ_2 sur ses colonnes et après ça il suffit d'ajouter une croix dans la case le plus en haut à gauche du tableau (la case $T_{0,0}$) si et seulement si il y a une croix dans un carré rouge. Par exemple, si $n_1 = 3, n_2 = 2, F_1 \times F_2 = \{(2,0),(2,1)\}$, $\phi_1(0) = 2, \phi_1(1) = 1, \phi_2(1) = 1$, on a $\delta^{\phi}(\{(0,1),(1,1)\}) = \{(2,1),(1,1),(0,0)\}$, ce qui est représenté à la figure ci-dessous.

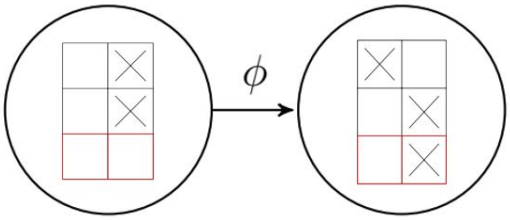


FIGURE 4.10 – Une transition dans $\texttt{Star} \circ \texttt{Inter}(Mon_{(3,2),\{(2,0),(2,1)\}})$

4.3.1 Une borne supérieure

On commence par établir une borne supérieure pour la complexité en états de l'étoile de l'intersection. Notre raisonnement sera basé sur la remarque suivante :

Remarque 4.3.1. Si lire une lettre à partir d'un état de A mène à un état dont au moins une croix se trouve dans un carré rouge, il y a une croix dans le carré le plus en haut à gauche ($T_{0,0} = 1$). Plus formellement, si E est un élément de Q, si (ϕ_1, ϕ_2) est une lettre de Γ_{n_1,n_2}, et si $(\phi_1, \phi_2)(E) \cap (F_1 \times F_2) \neq \emptyset$, alors en notant E' l'état $\delta^{(\phi_1,\phi_2)}(E)$, on a $(0,0) \in E'$. Ainsi, tout état E de A tel que $E \cap (F_1 \times F_2) \neq \emptyset$ et $(0,0) \notin E$ n'est pas accessible dans A.

Notation 4.3.1. Notons B_{F_1,F_2} l'automate déduit de $(\texttt{Star} \circ \texttt{Inter})\underline{Mon}_{\underline{n},(F_1,F_2)}$ en retirant les tableaux avec une croix dans (x,y)$\in F_1 \times F_2$ mais pas de croix dans (0,0).

Ainsi, nous allons obtenir une borne supérieure pour la complexité en états de la composition des opérations `Star` et `Inter` en maximisant le nombre d'états de B_{F_1,F_2}.

Remarque 4.3.2. L'état initial de $\texttt{Inter}\underline{Mon}_{\underline{n},(0,0)}$ est le seul état final. Ainsi, $L((\texttt{Star} \circ \texttt{Inter})\underline{Mon}_{\underline{n},(0,0)}) = L(\texttt{Inter}\underline{Mon}_{\underline{n},(0,0)})^* = L(\texttt{Inter}\underline{Mon}_{\underline{n},(0,0)})$ et ceci implique que $\#_{Min}(B_{0,0}) \leq \#_{Min}(\texttt{Inter}\underline{Mon}_{\underline{n},(0,0)}) \leq n_1 n_2$.

Lemme 4.3.1. *Le nombre maximal d'états de B_{F_1,F_2} avec $F_1 \times F_2 \notin \{\{(0,0)\}, \emptyset\}$ est atteint lorsque $\#(F_1 \times F_2) = 1$.*

Démonstration. Le nombre d'états de B_{F_1,F_2} est égal au nombre d'états de A auquel on retire le nombre de tableaux avec une croix dans $(x,y) \in F_1 \times F_2$ et dont la case $(0,0)$ n'est pas remplie. Ainsi, on a

$$\begin{aligned}
\#B =& \#2^{[\![n_1]\!]\times[\![n_2]\!]} - \#\{T \in 2^{[\![n_1]\!]\times[\![n_2]\!]} | (\exists (x,y) \in F_1 \times F_2 \text{ tel que } T_{x,y} = 1 \wedge T_{0,0} = 0\} \\
=& 2^{n_1 n_2} - (\#\{T \in 2^{[\![n_1]\!]\times[\![n_2]\!]} | T_{0,0} = 0\} - \#\{T \in 2^{[\![n_1]\!]\times[\![n_2]\!]} | \forall (x,y) \in F_1 \times F_2, \\
& T_{x,y} = 0 \wedge T_{0,0} = 0\}) \\
=& 2^{n_1 n_2} - (2^{n_1 n_2 - 1} - \#\{T \in 2^{[\![n_1]\!]\times[\![n_2]\!]} | \forall (x,y) \in F_1 \times F_2, T_{x,y} = 0 \wedge T_{0,0} = 0\}) \\
=& \begin{cases} 2^{n_1 n_2} - (2^{n_1 n_2 - 1} - 2^{n_1 n_2 - \#F_1 \#F_2 - 1}) \text{ si } (0,0) \notin F_1 \times F_2 \\ 2^{n_1 n_2} - (2^{n_1 n_2 - 1} - 2^{n_1 n_2 - \#F_1 \#F_2}) \text{ sinon} \end{cases}
\end{aligned}$$

Vu que qu'on est dans le cas où $(0,0) \notin F_1 \times F_2$, on obtient

$$\#B = 2^{n_1 n_2} - (2^{n_1 n_2 - 1} - 2^{n_1 n_2 - \#F_1 \#F_2 - 1}).$$

De plus, vu que $F_1 \times F_2 \neq \emptyset$, pour maximiser le nombre d'états de B_{F_1,F_2} il faut que $\#F_1 \#F_2$ soit égal à 1, c'est-à-dire $F_1 \times F_2 = 1$.

□

Corollaire 4.3.1. $\#_{Min}((\mathit{Star} \circ \mathit{Inter})\underline{M}_{\underline{n},(F_1,F_2)}) \leq \frac{3}{4} 2^{n_1 n_2}$

Démonstration. En utilisant le Lemme 4.3.1, on maximise le nombre de tableaux lorsque $\#F_1 \times \#F_2 = 1$. Dans ce cas, on a

$$\begin{aligned}
\#B =& 2^{n_1 n_2} - (2^{n_1 n_2 - 1} - 2^{n_1 n_2 - 1 - 1}) \\
=& 2^{n_1 n_2} - (2^{n_1 n_2 - 1} - 2^{n_1 n_2 - 2}) \\
=& 2^{n_1 n_2} - 2^{n_1 n_2 - 2} \\
=& 2^{n_1 n_2} - \frac{1}{4} 2^{n_1 n_2} \\
=& \frac{3}{4} 2^{n_1 n_2}
\end{aligned}$$

Ainsi, la borne supérieure est $\frac{3}{4} 2^{n_1 n_2}$.

□

4.3.2 Une borne inférieure

Lemme 4.3.2. *Tous les états de B sont accessibles.*

Démonstration. Soit T un état de B.
Définissons un ordre $<$ sur des tableaux comme T<T' si et seulement si

1. $\#(T) < \#(T')$ ou

2. $(\#(T) = \#(T'))$ et $T_{n_1-1,n_2-1} = 1$ et $T'_{n_1-1,n_2-1} = 0)$ ou
3. $(\#(T) = \#(T')$ et $T_{n_1-1,n_2-1} = T'_{n_1-1,n_2-1}$ et $T_{0,0} = 1$ et $T'_{0,0} = 0)$

Prouvons l'assertion par induction sur des tableaux non vides de B pour l'ordre partiel $<$.

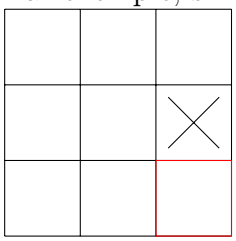
Cas de base
Remarquons d'abord que le tableau vide est l'état initial de B et donc il est bien accessible. Pour les tableaux non vides de B et l'ordre $<$, le seul tableau minimal est le tableau avec seulement un 1 dans (0,0). Il est accessible à partir de l'état initial $\emptyset$ en lisant la lettre (Id, Id). Remarquons que chaque lettre est une paire de fonctions de $[\![n_1]\!]^{[\![n_1]\!]} \times [\![n_2]\!]^{[\![n_2]\!]}$.

Induction Prenons un tableau T' et trouvons un tableau T tel que $T < T'$ et T' est accessible depuis T. Nous distinguons les cas ci-dessous selon certaines propriétés de T'. Pour chaque cas, on va définir un tableau T et une lettre (f, g). De plus, pour tous les cas sauf le dernier, on vérifie facilement que

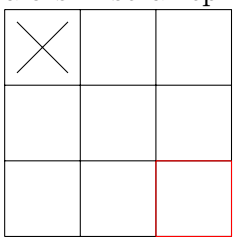
- $T_{0,0} = 1$ (ce qui implique que T est un état de B),
- $\delta^{(f,g)}(T) = (f, g)(T) = T'$ (où $(f, g)(T) = \{(f(i), g(j))|(i, j) \in T\}$) et
- $T < T'$

Cas 1 : $T'_{n_1-1,n_2-1} = 0$

- $T'_{0,0} = 0$. Soit (i, j) l'indexe d'un 1 dans T'. Définissons (f, g) par $((0, i), (0, j))$ où $(0, i)$ et $(0, j)$ sont des transpositions, et $T = (f, g)(T')$. De plus, on a bien $T < T'$ par le point 3 de la définition de cet ordre.
  Par exemple, si T' est représenté par le tableau ci-dessous :

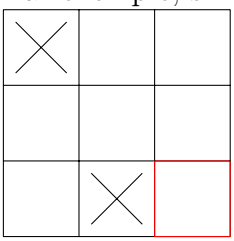


  alors T sera représenté par :

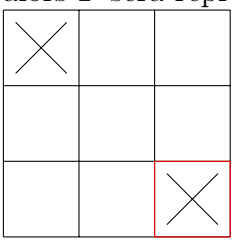


- $T'_{0,0} = 1$

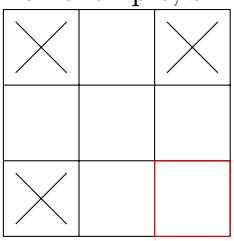
— Il existe $(i,j) \in \{1,2,\ldots,n_1-1\} \times \{1,2,\ldots,n_2-1\}$ tels que $T'_{i,j} = 1$. Définissons (f,g) comme $((n_1-1,i),(n_2-1,j))$, alors $T = (f,g)(T')$. De plus, on a bien $T < T'$ par le point 2 de la définition de cet ordre.
Par exemple, si T' est représenté par le tableau ci-dessous :



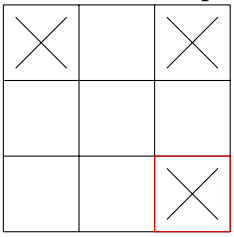
alors T sera représenté par :



— Pour tout $(i,j) \in \{1,2,\ldots,n_1-1\} \times \{1,2,\ldots,n_2-1\}, T'_{i,j} = 0, T'_{0,n_2-1} = 1$ et $T'_{n_1-1,0} = 1$. Dans ce cas, définissons (f,g) comme $(Id,(n_2-1,0))$, et T comme $(f,g)(T')$. De plus, on a bien $T < T'$ par le point 2 de la définition de cet ordre.
Par exemple, si T' est représenté par le tableau ci-dessous :



alors T sera représenté par :

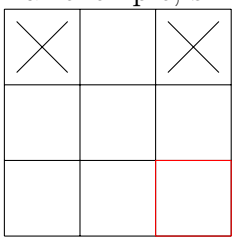


— Pour tout $(i,j) \in \{1,2,\ldots,n_1-1\} \times \{1,2,\ldots,n_2-1\}, T'_{i,j} = 0, T'_{0,n_2-1} = 1$ et

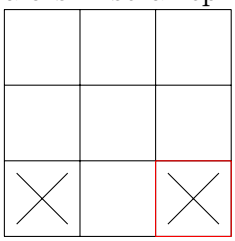
$T'_{n_1-1,0} = 0$. Définissons (f, g) comme $((n_1 - 1, 0), Id)$. Alors T est défini comme

$$\begin{cases} T_{0,n_2-1} & = 0 \\ T_{n_1-1,n_2-1} & = 1 \\ T_{i,j} & = T'_{i,j} \text{ si } (i,j) \notin \{(0, n_2 - 1), (n_1 - 1, n_2 - 1)\} \end{cases}$$

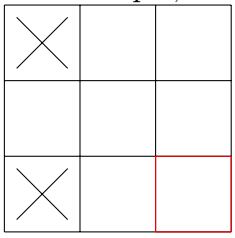
De plus, on a bien $T < T'$ par le point 2 de la définition de cet ordre.
Par exemple, si T' est représenté par le tableau ci-dessous :



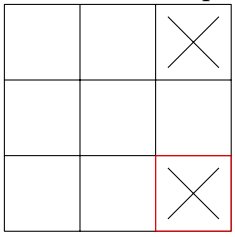
alors T sera représenté par :



— Pour tout $(i, j) \in \{1, 2, \ldots, n_1 - 1\} \times \{1, 2, \ldots, n_2 - 1\}, T'_{i,j} = 0, T'_{0,n_2-1} = 0$ et $T'_{n_1-1,0} = 1$. Définissons (f, g) comme $(Id, (n_2 - 1, 0))$. De plus, on a bien $T < T'$ par le point 2 de la définition de cet ordre.
Par exemple, si T' est représenté par le tableau ci-dessous :



alors T sera représenté par :



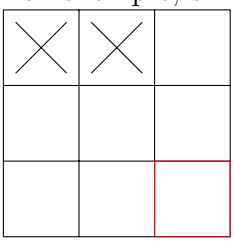
— Pour tout $(i,j) \in \{1,2,\dots,n_1-1\} \times \{1,2,\dots,n_2-1\}, T'_{i,j} = 0, T'_{0,n_2-1} = 0$ et $T'_{n_1-1,0} = 0$. Soit $(i,j) \neq (0,0)$ un 1 dans T'. Définissons

$$(f,g) = ((n_1-1,i),(n_2-1,j))$$

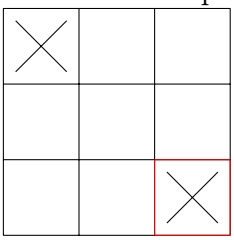
et définissons T comme suit

$$\begin{cases} T_{i,j} & = 0 \\ T_{n_1-1,n_2-1} & = 1 \\ T_{i',j'} & = T'_{i',j'} \text{ si } (i',j') \notin \{(i,j),(n_1-1,n_2-1)\} \end{cases}$$

De plus, on a bien $T < T'$ par le point 2 de la définition de cet ordre.
Par exemple, si T' est représenté par le tableau ci-dessous :



alors T sera représenté par :



Cas 2 : $T'_{0,0} = 1$ et $T'_{n_1-1,n_2-1} = 1$
Soit $(f,g) = ((n_1-1,0),(n_2-1,0))$. Soit T'' la matrice obtenue depuis T' en remplaçant 1 par 0 dans $(0,0)$. Soit $T = (f,g)(T'')$. Comme (f,g) est une bijection sur $[\![n_1]\!] \times [\![n_2]\!]$, nous avons $T_{0,0} = ((f,g)(T''))_{0,0} = T''_{n_1-1,n_2-1} = 1$, ce qui signifie que T est un état de B et $(f,g)(T) = (f,g)(f,g)(T'') = T''$. Comme $T''_{n_1-1,n_2-1} = 1$, nous avons $\delta^{(f,g)}(T) = T'$ dans B. De plus, $\#T < \#T'$ implique que $T < T'$.

□

Lemme 4.3.3. *Tous les états de B sont distinguables.*

Démonstration. Soient T et T' deux états différents de B. Il existe $(i,j) \in [\![n_1]\!] \times [\![n_2]\!]$ tels que $T_{i,j} \neq T'_{i,j}$. Supposons par exemple que $T_{i,j} = 1$ et $T'_{i,j} = 0$.
Soit $(f,g) \in [\![n_1]\!]^{[\![n_1]\!]} \times [\![n_2]\!]^{[\![n_2]\!]}$ tel que

$$f(x) = \begin{cases} n_1 - 1 & \text{si } x = i \\ 0 & \text{sinon} \end{cases}$$

$$g(x) = \begin{cases} n_2 - 1 & \text{si } x = j \\ 0 & \text{sinon} \end{cases}$$

Nous avons $\delta^{(f,g)}(T)_{n_1-1,n_2-1} = T_{i,j} = 1$ et $\delta^{(f,g)}(T')_{n_1-1,n_2-1} = T'_{i,j} = 0$. Ainsi T et T' sont distinguables dans B.

□

Théorème 4.3.1. La complexité en états de l'étoile de l'intersection est $\frac{3}{4}2^{n_1 n_2}$.

4.4 La racine carrée

Nous allons nous intéresser à la racine carrée du langage L qui est définie par $\sqrt{L} = \{x | xx \in L\}$. Nous verrons dans le Chapitre 4 une construction grâce à laquelle on obtiendra une borne supérieure de la complexité en états de la racine carrée égale à n^n. Pour l'instant admettons le et calculons la valeur exacte de sa complexité en états.

4.4.1 Une borne supérieure

Considérons l'automate SRoot($Mon_{n,F}$). On peut remarquer que tous les états dans SRoot($Mon_{n,F}$) sont accessibles. En effet, l'état étiqueté par la fonction g est atteint depuis Id en lisant la lettre g.

Pour la distinguabilité, on considère un état $g_{a,b}$ défini par : Soit $a \neq b \in [\![n]\!]$

— $\mathrm{g}_{a,b}(x) = a$ si $x \in F$

— $\mathrm{g}_{a,b}(x) = b$ sinon

Lemme 4.4.1. *Pour chaque pair $a, b \in [\![n]\!]$ telle que $a \neq b$, les deux états $g_{a,b}$ et $g_{b,a}$ ne sont pas distinguables dans SRoot($Mon_{n,F}$).*

Démonstration. Prouvons que pour tout h, les fonctions $h \circ g_{a,b}$ et $h \circ g_{b,a}$ sont soit toutes les deux finales soit toutes les deux non finales. En fait, nous avons seulement deux valeurs de h à analyser : $h(a)$ et $h(b)$. Si $h(a), h(b) \in F$ ou $h(a), h(b) \notin F$ alors les deux fonctions $h \circ g_{a,b}$ et $h \circ g_{b,a}$ sont évidemment toutes les deux finales ou toutes les deux non finales. Sans perte de généralité, on peut supposer que $h(a) \in F$ (et donc $h(b) \notin F$).

On doit examiner deux cas :

1. Si $0 \in F$:
   On a alors $h(g_{a,b}(0)) = h(a) \in F$. Donc, $g_{a,b}(h(g_{a,b}(0))) = a$ et $h(g_{a,b}(h(g_{a,b}(0)))) = h(a) \in F$. Mais $h(g_{b,a}(0)) = h(b) \notin F$. D'où, $g_{b,a}(h(g_{b,a}(0))) = a$ et donc $h(g_{b,a}(h(g_{b,a}(0)))) \in F$. Ce qui implique que les deux états sont finaux.
2. Si $0 \notin F$:
   On a alors $h(g_{a,b}(0)) = h(b) \notin F$. Alors $g_{a,b}(h(g_{a,b}(0))) = b$ et $h(g_{a,b}(h(g_{a,b}(0)))) = h(b) \notin F$. Mais on a aussi $h(g_{b,a}(0)) = h(a) \in F$. D'où, $g_{b,a}(h(g_{b,a}(0))) = b$ et donc $h(g_{b,a}(h(g_{b,a}(0)))) \notin F$. Ce qui implique que les deux états ne sont pas finaux

On en déduit que les deux états ne sont pas distinguables. □

Remarque 4.4.1. Le nombre de transformation $g_{a,b}$ est égal à $2\binom{n}{2}$.

Corollaire 4.4.1. *On a*

$$sc_{\surd}(n) \leq n^n - \binom{n}{2}$$

4.4.2 Une borne inférieure

Lemme 4.4.2. *Soient F={n-1}, P={$(g, g')|g \neq g'$ et $\forall a, b \in [\![n]\!], (g, g') \neq (g_{a,b}, g_{b,a})$}. Pour toute paire d'états distincts $(g, g') \in P$, g et g' sont distinguables dans SRoot($Mon_{n,F}$).*

Démonstration. Nous allons considérer trois cas

— Supposons que $g(0) = g'(0)$.
Alors il existe $x \in [\![n]\!] \backslash \{0\}$ tel que $g(x) \neq g'(x)$. Posons $h(g(0)) = x$. D'où, $h(g(h(g(0)))) = h(g(x))$ et $h(g'(h(g'(0)))) = h(g'(x))$. Mais vu que $g(x) \neq g'(x)$, il est toujours possible de choisir h tel que $h(g(x)) = n - 1$ alors que $h(g'(x)) \neq n - 1$. Donc, $h \circ g$ est un état final alors que $h \circ g'$ ne l'est pas.

— Supposons que $g(0) \neq g'(0)$ et que $\#(Im(g) \bigcup Im(g')) > 2$.
Sans perte de généralité, supposons qu'il existe $x \in Im(g)$ tel que $x \notin \{g(0), g'(0)\}$. Donc les valeurs $h(g(0)), h(g'(0))$ et $h(x)$ peuvent être choisies indépendamment les unes des autres. Posons $h(g(0)) = y$ avec $g(y) = x$, $h(g'(0)) = 0$ et $h(x) = n - 1$. On obtient que $h \circ g$ est un état final alors que $h \circ g'$ ne l'est pas.

— Supposons que $g(0) \neq g'(0)$ et que $\#(Im(g) \bigcup Im(g')) = 2$.
Supposons que pour tout état non final x, nous avons $g(x) \neq g(n-1)$ et $g'(x) \neq g'(n-1)$. Étant donné que x n'est pas final et que $\#(Im(g) \bigcup Im(g')) = 2$,

on a $g(x) = g(0)$ et $g'(x) = g'(0)$ (rappelons que 0 n'est pas final). Ainsi, comme $g(0) \neq g'(0)$ ça implique $g(n-1) \neq g'(n-1)$. Autrement dit, $g = g_{a,b}$ et $g' = g_{b,a}$ pour certains a, b. Par contraposition, si $(g, g') \neq (g_{a,b}, g_{b,a}) \forall a, b$ alors il existe $x \neq n-1$ tel que $g(x) = g(n-1)$ ou $g'(x) = g'(n-1)$. Notons m l'élément minimal de $[\![n]\!]$ ayant cette propriété et sans perte de généralité, supposons que $g(m) = g(n-1)$ (en particulier, ça signifie que pour tout $p < m, g'(p) \neq g'(n-1)$). On doit considérer deux cas :

— Si $m = 0$ alors on fixe $h(g(0)) = n-1$ et $h(g'(0)) = 0$. On a alors $h(g'(h(g'(0)))) = 0$. D'autre part, $h(g(h(g(0)))) = h(g(n-1)) = h(g(0)) = n-1$. D'où, $h \circ g$ est final alors que $h \circ g'$ ne l'est pas.

— Si $m > 0$ alors on a $g(m) = g'(0)$ car il y a exactement deux valeurs dans l'image de g et g'. De plus, $g'(n-1) \neq g'(0)$ et donc $g'(n-1) = g(0)$. Posons $h(g(0)) = m$ et $h(g'(0)) = n-1$. On a $h(g(h(g(0)))) = h(g(m)) = h(g'(0)) = n-1$. D'autre part, $h(g'(h(g'(0)))) = h(g'(n-1)) = h(g(0)) = m \neq n-1$. Il en découle que $h \circ g$ est final alors que $h \circ g'$ n'est pas final.

□

Le théorème qui suit est une consuite directe du Corollaire 4.4.1 et du Lemme 4.4.2.

Théorème 4.4.1. On a

$$sc_{\surd}(n) = n^n - \binom{n}{2}.$$

Chapitre 5

Modificateurs amicaux

Dans cette section, nous allons présenter les modificateurs amicaux. Nous montrerons qu'on peut changer chaque modificateur amical en une forme standard tout en conservant l'opération qu'il décrit. Nous définirons les opérations amicales comme la composition d'opérations booléennes et de racines. Nous verrons que chaque opération amicale est décrite par un modificateur amical standard unique et que l'inverse est également vrai. Dans le chapitre suivant, la complexité en états maximale d'opérations amicales sera déterminée en fonction de leur arité. Nous invitons le lecteur désireux d'en savoir plus à propos des modificateurs amicaux à lire la thèse [9]. A titre informatif, il existe une autre classe particulière de modificateurs, appelés modificateurs produits. Le lecteur intéressé pourra consulter [4] et [9].

5.1 Définition

Définition 5.1.1 (modificateur amical)**.** Un k-modificateur m=(`Q,i,f`,ρ) est amical si, pour tout k-uplet d'ensembles finis $\underline{\mathrm{Q}}$, tout $\underline{\mathrm{F}}$ tel que $F_j \subseteq Q_j \forall j, \forall \underline{i} \in Q_1 \times \cdots \times Q_k, \forall \underline{\phi}, \underline{\psi} \in Q_1^{Q_1} \times \cdots \times Q_k^{Q_k}$,

$$\rho(\underline{i}, \underline{F}, (\phi_1 \circ \psi_1, \ldots, \phi_k \circ \psi_k)) = \rho(\underline{i}, \underline{F}, \underline{\phi}) \circ \rho(\underline{i}, \underline{F}, \underline{\psi})$$

L'idée de cette définition est que ρ est un morphisme de monoïdes par rapport à sa troisième coordonnée.

Exemple 5.1.1. Le modificateur `Comp` est amical. En effet, soient Q un ensemble fini, F⊆Q, i∈Q, $\phi, \psi \in Q^Q$. On a

$$\rho(i, F, (\phi \circ \psi)) = \phi \circ \psi = \rho(i, F, \phi) \circ \rho(i, F, \psi)$$

Exemple 5.1.2. Le modificateur `Xor` est amical. En effet, soient $\underline{Q}=(Q_1, Q_2)$ un 2-uple d'ensembles finis, $\underline{F}$ tel que $F_j \subseteq Q_j \forall j$, $\underline{i} \in \underline{Q}$, $\underline{\phi}, \underline{\psi} \in Q_1^{Q_1} \times Q_2^{Q_2}$. On a

$$\rho(\underline{i}, \underline{F}, (\phi_1 \circ \psi_1, \phi_2 \circ \psi_2)) = (\phi_1 \circ \psi_1, \phi_2 \circ \psi_2)$$

et

$$\rho(\underline{i}, \underline{F}, \underline{\phi}) \circ \rho(\underline{i}, \underline{F}, \underline{\psi}) = \underline{\phi} \circ \underline{\psi} = (\phi_1 \circ \psi_1, \phi_2 \circ \psi_2)$$

par définition de $\circ$. Ainsi, on a bien

$$\rho(\underline{i}, \underline{F}, (\phi_1 \circ \psi_1, \phi_2 \circ \psi_2)) = \rho(\underline{i}, \underline{F}, \underline{\phi}) \circ \rho(\underline{i}, \underline{F}, \underline{\psi})$$

Exemple 5.1.3. Nous allons montrer que le modificateur `SRoot` est amical. Soient Q un ensemble fini, F$\subseteq$Q, i$\in$Q, $\phi, \psi \in Q^Q$. On a

$$\rho(i, F, (\phi \circ \psi)) = g \to (\phi \circ \psi) \circ g$$

et on a

$$\rho(i, F, \phi) \circ \rho(i, F, \psi) = (g \to \phi \circ g) \circ (g \to \psi \circ g) = g \to (\phi \circ \psi) \circ g.$$

Ce qui permet de conclure.

Proposition 5.1.1. *Les modificateurs amicaux sont stables par composition.*

5.2 Modificateurs amicaux standards

A tout modificateur 1-uniforme amical est associé un modificateur amical standard qui est un autre modificateur 1-uniforme décrivant la même opération. Nous allons voir que toute opération décrite par un modificateur amical est également décrite par un unique modificateur amical standard. Ainsi, un modificateur amical standard est un modificateur de forme canonique associé à un modificateur amical 1-uniforme et décrivant la même opération.

Définition 5.2.1 (modificateur amical standard)**.** Un k-modificateur m=(`Q,i,f`,ρ) est amical standard si

— $\mathtt{Q}(\underline{Q}) = Q_1^{Q_1} \times \cdots \times Q_k^{Q_k}$

— $\mathtt{i}(\underline{Q}, \underline{i}, \underline{F}) = (Id_{Q_1}, \ldots, Id_{Q_k})$

— $\rho(\underline{i},\underline{F},\underline{\phi})(\underline{\psi}) = (\phi_1 \circ \psi_1, \ldots, \phi_k \circ \psi_k)$.

Il suit de cette définition qu'un modificateur amical standard est bien amical. Remarquons qu'un k-modificateur amical standard est entièrement défini par sa troisième coordonnée `f`. On peut aisément associer un modificateur standard à tout modificateur amical.

Notation 5.2.1. Soit $m = (\mathtt{Q},\mathtt{i},\mathtt{f},\rho)$ un k-modificateur amical. Nous dénotons par m_{sf} le k-modificateur amical standard tel que

$$\mathtt{f}_{sf}(\underline{Q},\underline{i},\underline{F}) = \{\underline{\phi} | \rho(\underline{i},\underline{F},\underline{\phi})(\mathtt{i}(\underline{Q},\underline{i},\underline{F})) \in \mathtt{f}(\underline{Q},\underline{i},\underline{F})\}.$$

Exemple 5.2.1. Voici un rappel de l'effet du modificateur complémentaire `Comp` sur un automate fini déterministe A ainsi que l'effet du modificateur complémentaire standard $\mathtt{Comp}_{sf}$ sur A. Dans la troisième image, [ij] représente la fonction ϕ telle que $\phi(0) = i$ et $\phi(1) = j$.

La Figure 5.3 représente bien un modificateur amical standard car on a

- $\mathtt{Q}(\{0,1\})$=$\{0,1\}^{\{0,1\}}$,
- $\mathtt{i}(\{0,1\},0,\{1\})$=$\mathrm{Id}_{\{0,1\}}$=[01],
- si δ est la fonction de transition de A alors
  - $\rho((0,\{1\},\delta^a)([01]) = \delta^a \circ [01] = [10]$ car $\delta^a(0) = 1$ et $\delta^a(1) = 0$,
  - $\rho((0,\{1\},\delta^a)([10]) = \delta^a \circ [10] = [01]$,
  - $\rho((0,\{1\},\delta^a)([11]) = \delta^a \circ [11] = [00]$,
  - $\rho((0,\{1\},\delta^a)([00]) = \delta^a \circ [00] = [11]$.

  Il en va de même pour b.

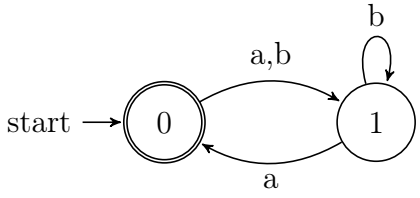


FIGURE 5.1 – Automate A

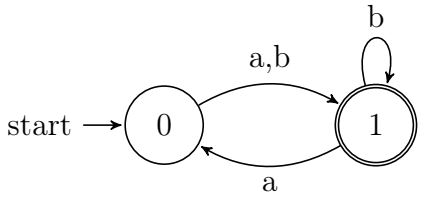


FIGURE 5.2 – Automate $\texttt{Comp}(A)$

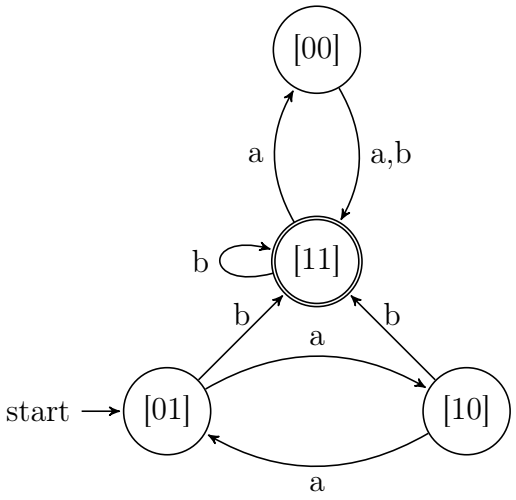


FIGURE 5.3 – Automate $\texttt{Comp}_{sf}(A)$

Lemme 5.2.1. *Pour tout modificateur amical 1-uniforme m, le modificateur amical standard m_{sf} décrit la même opération que m.*

Démonstration. Soit $m = (\texttt{Q},\texttt{i},\texttt{f},\rho)$ un modificateur amical 1-uniforme et soit $\underline{A}$ un k-uplet d'AFDs tel que $A_j{=}(\Sigma, Q_j, i_j, F_j, \delta_j)$.

Un mot $a_1a_2\cdots a_l$ est dans $L(m\underline{A})$
$\Leftrightarrow \rho(\underline{i},\underline{F},\underline{\delta}^{a_1a_2\cdots a_l})(\texttt{i}(\underline{Q},\underline{i},\underline{F})) \in \texttt{f}(\underline{Q},\underline{i},\underline{F})$
$\Leftrightarrow (\rho(\underline{i},\underline{F},\underline{\delta}^{a_l}) \circ \rho(\underline{i},\underline{F},\underline{\delta}^{a_{l-1}}) \circ \cdots \circ \rho(\underline{i},\underline{F},\underline{\delta}^{a_1}))(\texttt{i}(\underline{Q},\underline{i},\underline{F})) \in \texttt{f}(\underline{Q},\underline{i},\underline{F})$

De façon équivalente,
$(\rho_{sf}(\underline{i},\underline{F},\underline{\delta}^{a_l}) \circ \rho_{sf}(\underline{i},\underline{F},\underline{\delta}^{a_{l-1}}) \circ \cdots \circ \rho_{sf}(\underline{i},\underline{F},\underline{\delta}^{a_1}))(Id_{Q_1},\ldots,Id_{Q_k}) \in \texttt{f}_{sf}(\underline{Q},\underline{i},\underline{F})$
$\Leftrightarrow \underline{\delta}^{a_1a_2\cdots a_l} \in \texttt{f}_{sf}(\underline{Q},\underline{i},\underline{F})$

Ce dernier énoncé est équivalent à $a_1a_2\cdots a_l \in L(m_{sf}\underline{A})$. Donc, $L(m_{sf}\underline{A}) = L(m\underline{A})$.

□

Notation 5.2.2. Nous notons $\mathtt{M}_k$ l'ensemble des k-modificateurs amicaux standards 1-uniformes.

5.3 Suites caractéristiques

Un modificateur amical standard $(\mathtt{Q},\mathtt{i},\mathtt{f},\rho)$ est entièrement caractérisé par la relation $\mathtt{f}$. Nous allons montrer une propriété de régularité sur les états finaux d'un modificateur amical standard 1-uniforme. Dans ce but, nous associons à chaque état de l'automate de sortie une fonction caractéristique qui est telle que si deux états sont associés à la même fonction caractéristique alors ils ont la même finalité. Ces fonctions caractéristiques sont représentées par des k-uplets de suites ultimement périodiques ayant des valeurs dans {0,1}.

Définition 5.3.1 (ultimement périodique)**.** Une suite $(u_j)_{j\in\mathbb{N}}$ qui a ses valeurs dans un ensemble E est ultimement périodique si et seulement si il existe deux nombres naturels p et N tels que pour tout $n \geq N$, on a $u_{n+p} = u_n$.

Notation 5.3.1. Soit $\mathtt{U}_k$ l'ensemble de tous les k-uplets $\underline{u}$ où chaque u_j est une suite ultimement périodique à valeurs dans {0,1}. De plus, notons $\mathtt{U}$ l'ensemble $\bigcup_{k\in\mathbb{N}} \mathtt{U}_k$. Pour simplifier les notations, pour tout (j,p)$\in \{1,\ldots,k\} \times \mathbb{N}$, nous identifions $(u_j)_p$ à $u_{j,p}$.

Définition 5.3.2 (suite caractéristique)**.** Soit $\underline{\phi} \in Q_1^{Q_1} \times \cdots \times Q_k^{Q_k}$. Nous notons $\chi^{\underline{\phi}}_{\underline{i},\underline{F}}$ le k-uplet de suites $\underline{u} \in \mathtt{U}_k$ où pour tout p$\in \mathbb{N}$ et tout $j \in \{1,\ldots,k\}$,

$$u_{j,p} = \begin{cases} 1 & \text{si } \phi_j^p(i_j) \in F_j \\ 0 & \text{sinon} \end{cases}$$

avec la notation $\phi_j^p = \phi_j \circ \cdots \circ \phi_j$ (p fois).
On appelle $\chi^{\underline{\phi}}_{\underline{i},\underline{F}}$ la suite caractéristique de ϕ dans la configuration d'état $(\underline{Q},\underline{i},\underline{F})$.

Dans cette définition, $\phi_j^k(i_j)$ est ultimement périodique car ϕ_j est une fonction entre deux ensembles finis et donc on a bien $\underline{u} \in \mathtt{U}_k$.

Exemple 5.3.1. Comme illustré dans les deux figures ci-dessous, posons
$(Q_1,Q_2) = (\{0,1\},\{0,1\}), (i_1,i_2) = (0,0), (F_1,F_2) = (\{1\},\{0\}), \phi_1(0) = 1, \phi_1(1) = 0,$
$\phi_1 = \phi_2$ et $u = \chi^{\phi_1,\phi_2}_{(i_1,i_2),(F_1,F_2)}$.

Nous avons, pour tout $(j,p) \in \{1,2\} \times \mathbb{N}, u_{j,p} = 1$ si $\phi_j^p(i_j) \in F_j$ c'est-à-dire si $\phi_1^p(0) \in \{1\}$ et $\phi_2^p(0) \in \{0\}$. Ainsi, $u_{j,p} = 1$ si et seulement si $p + j$ est pair. En effet, on a bien $u_{1,2k+1} = 1 \forall k \in \mathbb{N}$ car $\phi_1^{2k+1}(i_1) \in F_1$. De même, on a $u_{2,2k} = 1 \forall k \in \mathbb{N}$ car $\phi_2^{2k+2}(i_2) \in F_2$.

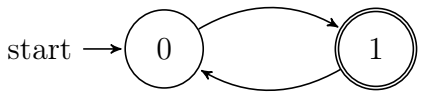


FIGURE 5.4 – Représentation de (Q_1, i_1, F_1) associé à la fonction ϕ_1

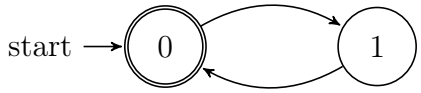


FIGURE 5.5 – Représentation de (Q_2, i_2, F_2) associé à la fonction ϕ_2

La proposition suivante exprime le fait que les états avec la même suite caractéristique ont la même finalité.

Proposition 5.3.1. *Soit $m = (\mathtt{Q}, \mathtt{i}, \mathtt{f}, \rho)$ un k-modificateur amical standard 1-uniforme. Soient $(\underline{Q}, \underline{i}, \underline{F})$ et $(\underline{Q'}, \underline{i'}, \underline{F'})$ deux configurations d'état, $\underline{\phi} \in Q_1^{Q_1} \times \cdots \times Q_k^{Q_k}$ et $\underline{\phi}' \in {Q'}_1^{Q'_1} \times \cdots \times {Q'}_k^{Q'_k}$. Si $\chi^{\underline{\phi}}_{\underline{i},\underline{F}} = \chi^{\underline{\phi'}}_{\underline{i'},\underline{F'}}$ alors $\underline{\phi} \in \mathtt{f}(\underline{Q}, \underline{i}, \underline{F})$ si et seulement si $\underline{\phi'} \in \mathtt{f}(\underline{Q'}, \underline{i'}, \underline{F'})$.*

Démonstration. Supposons que $\chi^{\underline{\phi}}_{\underline{i},\underline{F}} = \chi^{\underline{\phi'}}_{\underline{i'},\underline{F'}}$. Vu la symétrie entre ϕ et ϕ', il suffit de montrer que si $\underline{\phi} \in \mathtt{f}(\underline{Q}, \underline{i}, \underline{F})$ alors $\underline{\phi'} \in \mathtt{f}(\underline{Q'}, \underline{i'}, \underline{F'})$. Ainsi, supposons que $\underline{\phi} \in \mathtt{f}(\underline{Q}, \underline{i}, \underline{F})$.

Soient $\underline{A}$ et $\underline{A'}$ deux k-uplets d'automates finis déterministes avec $\forall l \in \{1, \ldots, k\}$ $A_l = (\{a\}, Q_l, i_l, F_l, \alpha_l)$ et $A'_l = (\{a\}, {Q'}_l, {i'}_l, {F'}_l, {\alpha'}_l)$ tels que $\alpha_l^a = \phi_l$ et ${\alpha'}_l^a = \phi'_l$. Vu que $\chi^{\underline{\phi}}_{\underline{i},\underline{F}} = \chi^{\underline{\phi'}}_{\underline{i'},\underline{F'}}$, pour tout $l \in \{1, \ldots, k\}$ $\phi_l^p(i_l) \in F_l$ si et seulement si ${\phi'}_l^p({i'}_l) \in {F'}_l$. De plus, $\alpha_l^{a^p} = \phi_l^p$ et ${\alpha'}_l^{a^p} = {\phi'}_l^p$.

Par conséquent, pour tout $l \in \{1, \ldots, k\}$, $L(A_l) = L(A'_l)$. Vu que $\rho(\underline{i}, \underline{F}, \underline{\alpha}^a)(Id_{Q_1}, \ldots, Id_{Q_k}) = \underline{\phi}$ et $\underline{\phi} \in \mathtt{f}(\underline{Q}, \underline{i}, \underline{F})$, on a $a \in L(m\underline{A})$. De plus, m est 1-uniforme, et donc on obtient $a \in L(m\underline{A'})$. Ceci implique que $\underline{\phi'} = \rho(\underline{i'}, \underline{F'}, \underline{\alpha'}^a)(Id_{Q_1}, \ldots, Id_{Q_k}) \in \mathtt{f}(\underline{Q'}, \underline{i'}, \underline{F'})$, ce qui permet de conclure. □

Grâce à ce résultat, la troisième coordonnée d'un modificateur amical standard $\mathtt{f}$ peut être représentée par un ensemble de fonctions caractéristiques. Commençons par définir une application $\mathtt{mod}$ qui nous permet de calculer un k-modificateur amical standard à partir de

tout sous-ensemble de $\mathtt{U}_k$. Nous allons voir qu'il y a en fait une correspondance entre les modificateurs amicaux standards et les sous-ensembles de $\mathtt{U}_k$.

Définition 5.3.3 (mod)**.** Pour tout naturel k, pour tout E$\subseteq\mathtt{U}_k$, nous désignons par $\mathtt{mod}$(E) le modificateur amical standard $(\mathtt{Q},\mathtt{i},\mathtt{f},\rho)$ tel que pour toutes les configurations d'état $(\underline{\mathrm{Q}},\underline{\mathrm{i}},\underline{\mathrm{F}})$, $\mathrm{f}(\underline{\mathrm{Q}},\underline{\mathrm{i}},\underline{\mathrm{F}})=\{\underline{\phi}\in Q_1^{Q_1}\times\cdots\times Q_k^{Q_k} | \chi^{\underline{\phi}}_{\underline{i},\underline{F}}\in E\}$.

Le corollaire suivant nous dit que tout k-modificateur amical standard 1-uniforme peut être construit de cette façon à partir d'un sous-ensemble de $\mathtt{U}_k$.

Corollaire 5.3.1. *L'ensemble des k-modificateurs amicaux standards 1-uniformes $\mathtt{M}_k$ est un sous-ensemble de l'image de $\mathtt{mod}$.*

Démonstration. Soit m=$(\mathtt{Q},\mathtt{i},\mathtt{f},\rho)$ un k-modificateur amical standard 1-uniforme. Soit E l'ensemble de toutes les suites u$\in\mathtt{U}_k$ telles qu'il existe une configuration d'état $(\underline{\mathrm{Q}},\underline{\mathrm{i}},\underline{\mathrm{F}})$ et $\underline{\phi}\in\mathtt{f}(\underline{\mathrm{Q}},\underline{\mathrm{i}},\underline{\mathrm{F}})$ avec $\chi^{\underline{\phi}}_{\underline{i},\underline{F}}=u$. Pour toute configuration d'état $(\underline{\mathrm{Q}},\underline{\mathrm{i}},\underline{\mathrm{F}})$, si $\underline{\phi}\in Q_1^{Q_1}\times\cdots\times Q_k^{Q_k}$ et $\chi^{\underline{\phi}}_{\underline{i},\underline{F}}\in E$, alors par la Proposition 5.3.1, $\phi\in\mathtt{f}(\underline{\mathrm{Q}},\underline{\mathrm{i}},\underline{\mathrm{F}})=\{\underline{\phi}\in Q_1^{Q_1}\times\cdots\times Q_k^{Q_k} | \chi^{\underline{\phi}}_{\underline{i},\underline{F}}\in E\}$. Ainsi, m=$\mathtt{mod}$(E).

□

5.4 Opérations amicales

Les exemples 5.1.2 et 5.1.3 montrent que la différence symétrique et la racine carrée peuvent être décrits par des modificateurs amicaux et donc par des modificateurs amicaux standards. Ces constructions s'étendent à toute opération de racine k-ème et toute opération booléenne k-aire. Ainsi, par la Proposition 5.1.1, on obtient que toute composition d'une opération booléenne k-aire avec des racines de langages est décrite par un modificateur amical standard. Nous allons étendre la notion d'opération booléenne à une arité infinie afin de décrire par un modificateur amical d'autres opérations telle que l'opération Racine infinie qui est définie par $\mathrm{Racine}_\infty=\bigcup_{p=1}^{+\infty}\sqrt[p]{L}$.

Définition 5.4.1 (opération booléenne)**.** Une fonction booléenne est une fonction de $\{0,1\}^{\mathbb{N}}$ dans $\{0,1\}$. Chaque fonction booléenne b définit une opération booléenne $\boxtimes_b$ produisant un langage lorsqu'il agit sur des suites de langages de la façon suivante : pour toute suite de langages $(L_p)_{p\in\mathbb{N}}$, un mot w est dans $\boxtimes_b((L_p)_{p\in\mathbb{N}})$ si et seulement si il existe une suite v dans $\{0,1\}^{\mathbb{N}}$ avec $b(v)=1$ telle que, pour tout $p\in\mathbb{N}$, $w\in L_p$ si et seulement si $v_p=1$.

Exemple 5.4.1. Soit la fonction booléenne b définie par : pour toute suite v dans $\{0,1\}^{\mathbb{N}}$, $b(v)=1$ si et seulement si soit pour tout $p\in\mathbb{N}\, v_p=1$, soit pour tout $p\in\mathbb{N}\, v_p=0$. Nous

avons pour toute suite de langages réguliers $(L_p)_{p\in\mathbb{N}}$, $w \in \boxtimes((L_p)_{p\in\mathbb{N}})$ si et seulement si soit pour tout $p \in \mathbb{N}$, $w \in L_p$, soit pour tout $p \in \mathbb{N}$, $w \notin L_p$. Cette assertion se traduit en l'équation suivante :

$$\boxtimes((L_p)_{p\in\mathbb{N}}) = \bigcap_{p=0}^{+\inf} L_p \cup \bigcap_{p=0}^{+\inf} L_p^C$$

Maintenant, nous pouvons définir les opérations amicales comme la composition d'une opération booléenne avec certaines racines de langages.

Définition 5.4.2 (opération amicale)**.** Une opération k-aire sur des langages réguliers $\otimes$ est amicale s'il existe une opération booléenne $\boxtimes$ telle que, pour tout k-uplet de langages réguliers $\underline{L} = (L_1, \ldots, L_k)$ définis sur le même alphabet,

$$\otimes(\underline{L}) = \boxtimes(\sqrt[0]{L_1}, \sqrt[0]{L_2}, \ldots, \sqrt[0]{L_k}, \sqrt[1]{L_1}, \sqrt[1]{L_2}, \ldots, \sqrt[1]{L_k}, \ldots, \sqrt[p]{L_1}, \sqrt[p]{L_2}, \ldots, \sqrt[p]{L_k}, \ldots).$$

Nous allons montrer qu'il y a une correspondance entre les opérations amicales, les modificateurs amicaux standards 1-uniformes et les sous-ensembles de $\mathtt{U}_k$.

Définition 5.4.3. Soit $\underline{u}$ un k-uplet de suites avec des valeurs dans {0,1}. Pour tout k-uplet de langages réguliers $\underline{L}$, soit $< \underline{u}, \underline{L} >$ le langage $\bigcap_{(j,p)\in\{1,\ldots,k\}\times\mathbb{N}} E_{j,p}$, où

$$E_{j,p} = \begin{cases} \sqrt[p]{L_j} & \text{si } u_{j,p} = 1 \\ \sqrt[p]{L_j}^C & \text{si } u_{j,p} = 0. \end{cases}$$

De plus, posons $< \underline{u}, . >$ l'opération k-aire sur des langages réguliers telle que, pour tout k-uplet de langages réguliers $\underline{L}$, on a

$$< \underline{u}, . > (\underline{L}) =< \underline{u}, \underline{L} >$$

Exemple 5.4.2. Soit $(u_1, u_2) \in \mathtt{U}_2$ tel que $u_{j,p} = 1$ si et seulement si $p + j$ est pair. Alors, pour tous langages L_1 et L_2, $< (u_1, u_2), (L_1, L_2) >$ est égal à

$$(\sqrt[0]{L_1}^C \cap \sqrt[1]{L_1} \cap \sqrt[2]{L_1}^C \cap \sqrt[3]{L_1} \cap \sqrt[4]{L_1}^C \cap \cdots) \bigcap (\sqrt[0]{L_2} \cap \sqrt[1]{L_2}^C \cap \sqrt[2]{L_2} \cap \sqrt[3]{L_2}^C \cap \sqrt[4]{L_2} \cap \cdots)$$

Notation 5.4.1. Pour tout ensemble E et tout élément g d'un ensemble quelconque, on note

$$[g \in E] = \begin{cases} 1 & \text{si } g \in E \\ 0 & \text{sinon.} \end{cases}$$

Remarque 5.4.1. On peut reformuler la Définition 5.4.3 par : pour tout naturel k, pour tout k-uplet de langages $\underline{L}$, et pour tout k-uplet de suites $\underline{u}$ avec des valeurs dans {0,1}, un mot w est dans $< \underline{u}, \underline{L} >$ si et seulement si , pour tout $(j,p) \in \{1,\ldots,k\} \times \mathbb{N}$, $[w^p \in L_j] = u_{j,p}$.

Le lemme suivant prouve que, $< (u_1,\ldots,u_k), . >$ est l'opération constante ayant l'ensemble vide comme sortie si $(u_1,\ldots,u_k)$ n'est pas dans $\mathtt{U}_k$.

Lemme 5.4.1. *Pour tout naturel k, si $\underline{L}$ est un k-uplet de langages réguliers, et si $\underline{u}$ est un k-uplet de suites avec des valeurs dans {0,1} tel que $< \underline{u}, \underline{L} > \neq \emptyset$, alors on a $\underline{u} \in \mathtt{U}_k$.*

Démonstration. Soit $\underline{A}$ un k-uplet d'AFD avec $A_j = (\Sigma, Q_j, i_j, \delta_j)$ tel que, pour tout $j \in \{1,\ldots,k\}, L(A_j) = L_j$. On a $w^p \in L_j$ si et seulement si $(\delta_j^w)^p(i_j) \in F_j$. Ainsi, s'il existe un mot w et un k-uplet de suites $\underline{v}$ avec des valeurs dans {0,1} tel que, pour tout $(j,p) \in \{1,\ldots,k\} \times \mathbb{N}$, on a $[w^p \in L_j] = v_{j,p}$, alors $[(\delta_j^w)^p \in F_j] = v_{j,p}$ ce qui implique que $(v_{j,p})_{p\in\mathbb{N}}$ est ultimement périodique.

Pour résumer, si

$$\{w \in \Sigma^* | \forall (j,p) \in \{1,\ldots,k\} \times \mathbb{N}, [w^p \in L_j] = v_{j,p}\} \neq \emptyset,$$

alors $\underline{v} \in \mathtt{U}_k$. On conclut par la Remarque 5.4.1

□

Notation 5.4.2. Nous désignons par $\mathtt{O}_k$ l'ensemble des opérations amicales k-aires.

Définition 5.4.4 (op)**.** Soit $\mathtt{op}$ l'application de $2^{\mathtt{U}_k}$ dans $\mathtt{O}_k$ telle que, pour tout $E \subseteq \mathtt{U}_k$, $\mathtt{op}(E)$ représente l'opération amicale k-aire $\bigcup_{u\in E} < u, . >$.

Proposition 5.4.1. *Toute composition finie de racines, unions, intersections et compléments agit sur les langages réguliers comme un opérateur $\mathtt{op}(E)$ pour un $E \in 2^{\mathtt{U}_k}$.*

Démonstration. Soit L un langage régulier. Nous définissons pour tout mot w, la suite $u(w) = (u_i(w))_{i\in\mathbb{N}}$ telle que $u_i(w) = 1$ si $w^i \in L$ et 0 sinon. Vu que l'ensemble des quotients $(w^i)^{-1}L$ est fini, la suite $((w^i)^{-1}L)_{i\in\mathbb{N}}$ est ultimement périodique et donc la suite $u(w)$ est aussi ultimement périodique.

Par conséquent, $\mathtt{op}(\{u \in \mathtt{U}_1 | u_j = 1\})(L) = \sqrt[j]{L}$ pour tout langage régulier L. En effet, si $w \in \sqrt[j]{L}$ alors $u(w)$ est ultimement périodique et $u_j(w) = 1$. De plus, vu la construction, $w \in < u(w), L >$.

De la même façon,

$$\mathtt{op}(\{u \in \mathtt{U}_1 | u_1 = 0\})(L) = L^C, \mathtt{op}(\{(u,v) \in \mathtt{U}_2 | u_1 = 1 \text{ et } v_1 = 1\})(L_1, L_2) = L_1 \cap L_2$$

et

$$\mathtt{op}(\{(u,v) \in \mathtt{U}_2 | u_1 = 1 \text{ ou } v_1 = 1\})(L_1, L_2) = L_1 \cup L_2.$$

En itérant ces constructions, tout opérateur k-aire qui est une combinaison de $\sqrt[j]{}$, compléments, union, intersections peut être simulé sur des langages réguliers par l'action d'un opérateur $\mathtt{op}(E)$ pour $E \in 2^{\mathtt{U}_k}$.

□

Exemple 5.4.3. Si L_1, L_2 et L_3 sont des langages réguliers alors

$$(\sqrt[i]{L_1} \cup L_2) \cap {L_3}^C = \mathtt{op}((u,v,w) \in \mathtt{U}_3 | (u_i = 1 \text{ ou } v_1 = 1 \text{ et } w_1 = 0\})(L_1, L_2, L_3).$$

Remarque 5.4.2. Lorsque l'opérateur $\mathtt{op}(\{u \in U_1 | u_j = 1\})$ agit sur 2^{Σ^*}, il est distinct de $\sqrt[j]{L}$. Par contre, les deux opérateurs coïncident lorsqu'ils agissent sur des langages réguliers.

Le lemme qui suit prouve qu'il y a une correspondance entre les sous-ensembles de $\mathtt{U}_k$ et les opérations amicales k-aires.

Lemme 5.4.2. *L'application* $\mathtt{op}$ *est bijective*

Démonstration. Commençons par montrer que l'application $\mathtt{op}$ est surjective.
Soit $\mathtt{V}_k = (\{0,1\}^{\mathbb{N}})^k$ l'ensemble de tous les k-uplets de suites avec des valeurs dans $\{0,1\}$. Soit $\otimes$ une opération k-aire amicale et $\boxtimes_b$ une opération booléenne telle que, pour tout k-uplet de langages réguliers $\underline{L}$,

$$\otimes = \boxtimes_b(\sqrt[0]{L_1}, \ldots, \sqrt[0]{L_k}, \sqrt[1]{L_1}, \ldots, \sqrt[1]{L_1}, \ldots, \sqrt[p]{L_1}, \ldots, \sqrt[p]{L_k}, \ldots).$$

Soient $E = \{\underline{u} \in \mathtt{U}_k | b(u_{1,0}, \ldots, u_{k,0}, u_{1,1}, \ldots, u_{k,1}, \ldots, u_{1,p}, \ldots, u_{k,p}, \ldots) = 1\}$ et $E' = \{\underline{v} \in \mathtt{V}_k | b(v_{1,0}, \ldots, v_{k,0}, v_{1,1}, \ldots, v_{k,1}, \ldots, v_{1,p}, \ldots, v_{k,p}, \ldots) = 1\}$.

Vérifions que $\otimes(\underline{L}) = (\mathtt{op}(E))(\underline{L})$. Pour tout k-uplet de langages réguliers $\underline{L}$, nous avons

$$\otimes(\underline{L}) = \bigcup_{\underline{v} \in E'} \{w \in \Sigma^* | \forall (j,p) \in \{1, \ldots, k\} \times \mathbb{N}, w \in \sqrt[p]{L_j} \Leftrightarrow v_{j,p} = 1\}$$

Montrons que l'union ci-dessus ne fait intervenir que des suites ultimement périodiques. Si $\underline{A}$ est un k-uplet d'AFD avec $A_j = (\Sigma, Q_j, i_j, F_j, \delta_j)$ tel que, pour tout $j \in \{1, \ldots, k\}$, $L(A_j) = L_j$, alors $w \in \sqrt[p]{L_j}$ si et seulement si $(\delta_j^w)^p(i_j) \in F_j$. Ainsi, s'il existe un mot w et un k-uplet de suites $\underline{v} \in \mathtt{V}_k$ tels que, pour tout $(j,p) \in \{1, \ldots, k\} \times \mathbb{N}, w \in \sqrt[p]{L_j}$ si et seulement si $v_{j,p} = 1$, alors $(\delta_j^w)^p(i_j) \in F_j$ si et seulement si $v_{j,p} = 1$, ce qui implique que $(v_{j,p})_{p \in \mathbb{N}}$ est ultimement périodique.

Pour résumer, si $\{w \in \Sigma^* | \forall (j,p) \in \{1, \ldots, k\} \times \mathbb{N}, w \in \sqrt[p]{L_j} \Leftrightarrow v_{j,p} = 1\} \neq \emptyset$, alors $\underline{v} \in \mathtt{U}_k$.
Nous avons donc

$$\begin{aligned}
\otimes(\underline{L}) &= \bigcup_{\underline{u} \in E} \{w \in \Sigma^* | \forall (j,p) \in \{1, \ldots, k\} \times \mathbb{N}, w \in \sqrt[p]{L_j} \Leftrightarrow u_{j,p} = 1\} \\
&= \bigcup_{\underline{u} \in E} < \underline{u}, \underline{L} > \\
&= (\mathtt{op}(E))(\underline{L})
\end{aligned}$$

Maintenant, montrons que $\mathtt{op}$ est injectif.
Soient $E, E' \subseteq \mathtt{U}_k$ et $\underline{u} \in \mathtt{U}_k$ tel que $\underline{u} \in E$ et $\underline{u} \notin E'$. Vu que, pour tout $j \in \{1, \ldots, k\} (u_{j,l})_{l \in \mathbb{N}}$ est ultimement périodique, les langages $L_j = \{a^p | p \in \mathbb{N} \wedge u_{j,p} = 1\}$ sont réguliers.

Nous avons $a \in \sqrt[p]{L_j}$ si et seulement si $u_{j,p} = 1$. Ainsi, par la Définition 5.4.3, pour tout $\underline{u'} \in \mathtt{U}_k$, $a \in < \underline{u'}, \underline{L} >$ si et seulement si $\underline{u'} = \underline{u}$. Il suit que si $\otimes = \mathtt{op}(E)$ et $\otimes' = \mathtt{op}(E')$, alors $a \in \otimes \underline{L}$ parce que $\underline{u} \in E \backslash E'$. Par conséquent, $\otimes \neq \otimes'$ et $\mathtt{op}$ est injectif.

□

Exemple 5.4.4. Pour tout langage régulier L, nous avons

$$Root(L) = \mathtt{op}(\{u \in \mathtt{U}_1 | \exists i > 0 \text{ tel que } u_i = 1\})(L) = \bigcup_{i \geq 1} \sqrt[i]{L}.$$

Le lemme suivant montre que toute opération qui est décrite par un modificateur amical est amicale.

Lemme 5.4.3. *Soit $E \subseteq \mathtt{U}_k$, $\mathtt{mod}(E)$ décrit $\mathtt{op}(E)$.*

Démonstration. Soit $m = \mathtt{mod}(E)$ avec $m = (\mathtt{Q}, \mathtt{i}, \mathtt{f}, \rho)$ et soit $\otimes$ l'opération décrite par m. Soit $\underline{A}$ un k-uplet d'AFD avec $A_j = (\Sigma, \underline{Q}, \underline{i}, \underline{F}, \delta)$. Un mot $a_1 \cdots a_n$ est dans $L(m\underline{A})$ si et seulement si

$$\underline{\delta}^{a_1 \cdots a_n} = (\rho(\underline{i}, \underline{F}, \underline{\delta}^{a_n}) \circ \rho(\underline{i}, \underline{F}, \underline{\delta}^{a_{n-1}}) \circ \cdots \circ \rho(\underline{i}, \underline{F}, \underline{\delta}^{a_1})(Id_{Q_1}, \ldots, Id_{Q_K}) \in \mathtt{f}(\underline{Q}, \underline{i}, \underline{F}).$$

De façon équivalente, par la Définition 5.3.3, $\chi_{\underline{i},\underline{F}}^{\underline{\delta}^{a_1 \cdots a_n}} \in E$.

Cependant, par la Définition 5.3.2, $\chi_{\underline{i},\underline{F}}^{\underline{\delta}^{a_1 \cdots a_n}}$ est la seule fonction $\underline{u}$ dans E telle que, pour tout $(p, j) \in \mathbb{N} \times \{1, \ldots, k\}$, $(\delta_j^{a_1 \cdots a_n})^p(i_j) \in F_j$ si et seulement si $u_{(p,j)} = 1$. Ainsi, par la Définition 5.4.3, $a_1 \cdots a_n \in L(m(A_1, \ldots, A_k))$ si et seulement si il existe $\underline{u}$ dans E tel que $a_1 \cdots a_n \in < \underline{u}, \underline{L} >$. Nous avons donc

$$\otimes(L(A_1), \ldots, L(A_k)) = \bigcup_{u \in E} < \underline{u}, (L(A_1), \ldots, L(A_k)) > \text{ et } \otimes = \mathtt{op}(E).$$

□

Notation 5.4.3. Pour tout modificateur m 1-uniforme k-aire, notons $\mathtt{desc}$ l'application de $\mathtt{M}_k$ dans $\mathtt{O}_k$ telle que $\mathtt{desc}(m)$ est l'opération 1-uniforme régulière décrite par m.

Dans la prochaine proposition, nous allons obtenir le résultat important suivant : Toutes les applications représentées sur la figure ci-dessous sont des bijections et ce diagramme est commutatif.

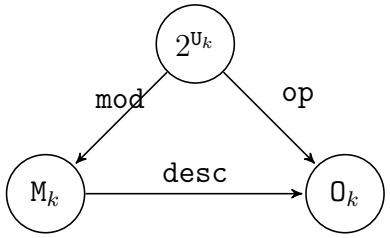


FIGURE 5.6 – Diagramme commutatif pour op, mod et desc

Proposition 5.4.2. *L'application $\mathtt{mod}$ est une bijection de $2^{\mathtt{U}_k}$ dans $\mathtt{M}_k$, et $\mathtt{op}$ et $\mathtt{desc}$ sont bijectifs. De plus, $\mathtt{desc} \circ \mathtt{mod} = \mathtt{op}$.*

Démonstration. Nous savons déjà que l'opération op est une bijection par le Lemme 5.4.2. De plus, le Lemme 5.4.3 montre qu'un k-modificateur amical standard dans l'image de $2^{\mathtt{U}_k}$ par mod est 1-uniforme. Ainsi, par le Corollaire 5.3.1, mod est une surjection de $2^{\mathtt{U}_k}$

dans l'ensemble des k-modificateurs amicaux standards 1-uniformes. Par le Lemme 5.4.3, l'image de $\mathtt{M}_k$ par `desc` est un sous-ensemble de $\mathtt{O}_k$. Le Lemme 5.4.3 prouve aussi que $\mathtt{desc} \circ \mathtt{mod} = \mathtt{op}$. Par conséquent, $\mathtt{desc} \circ \mathtt{mod}$ est une bijection, et le fait que `mod` est une surjection implique que `desc` et `mod` sont des bijections. □

Théorème 5.4.1. Chaque opération k-aire amicale est décrite par un k-modificateur amical standard 1-uniforme unique. Inversément, tout k-modificateur amical 1-uniforme décrit une opération k-aire amicale.

Chapitre 6

Complexité en états d'opérations amicales

Notre construction de modificateurs amicaux standards (Définition 5.2.1) nous donne une borne supérieure qui est telle que $sc_{\otimes}(n) \leq n^n$, pour toute opération amicale unaire $\otimes$. On a utilisé cette information dans la Section 4.4 et on a vu que la complexité en états de l'opération racine carrée est $sc_{\surd}(n) = n^n - \binom{n}{2}$, et qu'elle est égale à la complexité en états de l'opération Racine. Ceci nous amène à nous demander si la complexité en états d'une opération amicale unaire atteint la borne n^n et sinon, si on peut donner une borne serrée explicite. Des questions similaires découlent pour le cas général d'opérations amicales k-aires avec la borne supérieure $sc_{\otimes}(n_1, \ldots, n_k) \leq \prod_{j=1}^{k} n_j^{n_j}$ déduite de la Définition 5.2.1. Les preuves de ce chapitre proviennent de [9].

6.1 Le cas unaire

Nous allons montrer que la borne n^n n'est pas serrée pour la complexité en états d'opérations amicales unaires et nous donnerons une borne serrée explicite. Commençons par montrer que cette complexité en états est au plus $n^n - n + 1$.

Proposition 6.1.1. *Pour tout naturel n et toute opération amicale unaire $\otimes$, on a $sc_{\otimes}(n) \leq n^n - n + 1$.*

Démonstration. Considérons un sous-ensemble $E \subseteq \mathtt{U}_1$. Soient $\otimes = \mathtt{op}(E)$ et $m = \mathtt{mod}(E)$. Soit $\mathrm{A} = (\Sigma, Q, i, F, \alpha)$ un AFD de taille $n \in \mathbb{N} \setminus \{0\}$. Pour tous $s, t \in Q$, posons $g_{s,t}$ la fonction de Q^Q telle que

$$g_{s,t}(j) = \begin{cases} s & \text{si } j \in F \\ t & \text{sinon.} \end{cases}$$

et par G l'ensemble de toutes les fonctions $g_{s,t}$, pour $s, t \in Q$. De plus, soient

— $\mathtt{0}$ la suite $(0, 0, \ldots)$ de $\mathtt{U}_1$

— $\mathtt{0}^1$ la suite $(0, 1, 1, \ldots, 1, \ldots)$
— `impair` la suite $(0, 1, 0, 1, \ldots,$n $\mod 2, \ldots)$
— `pair` la suite $(1, 0, 1, 0, \ldots,$(n+1) $\mod 2, \ldots)$.

Il suit de la définition de $g_{s,t}$, que pour tout $\zeta \in Q^Q$ et pour tout $g_{s,t} \in G$, on a $\zeta \circ g_{s,t} = g_{\zeta(s),\zeta(t)}$ et ainsi nous disons que G est stable par composition externe.

Nous allons montrer que $sc(L(m(A)))$ est au plus $n^n - n + 1$ en étudiant la relation d'équivalence de Nérode induite par A. On va distinguer deux cas principaux, $i \in F$ et $i \notin F$.

Supposons que $i \notin F$. On a
— Si $t, s \in F$, alors on a
  — $g^0_{s,t}(i) = i \notin F$
  — $g_{s,t}(i) = t \in F$
  — $g^2_{s,t}(i) = g_{s,t}(t) = s \in F$
  — $\ldots$

  D'où, $\chi^{g_{s,t}}_{i,F} = \mathtt{0}^1$.

— Si $t \in F, s \notin F$, alors on a
  — $g^0_{s,t}(i) = i \notin F$
  — $g_{s,t}(i) = t \in F$
  — $g^2_{s,t}(i) = g_{s,t}(t) = s \notin F$
  — $\ldots$

  D'où, $\chi^{g_{s,t}}_{i,F} =$ `impair`.

— Si $t, s \notin F$, alors on a
  — $g^0_{s,t}(i) = i \notin F$
  — $g_{s,t}(i) = t \notin F$
  — $g^2_{s,t}(i) = g_{s,t}(t) = t \notin F$
  — $\ldots$

  D'où, $\chi^{g_{s,t}}_{i,F} = \mathtt{0}$.

En résumé, on a
— si $t \in F$
  — si $s \in F$, alors on a $\chi^{g_{s,t}}_{i,F} = \mathtt{0}^1$

  - sinon on a $\chi_{i,F}^{g_{s,t}} = \texttt{impair}$
- sinon $\chi_{i,F}^{g_{s,t}} = \texttt{0}$

Soit $E_1 = \{\texttt{0}, \texttt{0}^1, \texttt{impair}\} \cap E$ et $E_2 = \{\texttt{0}, \texttt{0}^1, \texttt{impair}\} \backslash E_1$.

On distingue les cas suivants :

- Si $\#E_1 = 0$, alors pour tous $s, t \in Q$, vu que $\chi_{i,F}^{g_{s,t}} \notin E$, l'état $g_{s,t}$ n'est pas final dans $m(\mathrm{A})$. Vu que G est stable par composition externe, tous les états dans G sont dans la même classe d'équivalence de Nérode. Par conséquent,

$$sc(L(m(A))) \leq n^n - n^2 + 1 \leq n^n - n + 1.$$

- Si $\#E_1 = 3$, alors pour tous $s, t \in Q$,vu que $\chi_{i,F}^{g_{s,t}} \in E$, l'état $g_{s,t}$ est final dans $m(\mathrm{A})$. Vu que G est stable par composition externe, tous les états dans G sont dans la même classe d'équivalence de Nérode. Par conséquent,

$$sc(L(m(A))) \leq n^n - n^2 + 1 \leq n^n - n + 1.$$

- Sinon si $\#E_1 = 1$ (respectivement $\#E_1 = 2$), alors nous notons u l'unique élément de $\#E_1$ (respectivement un élément de $\#E_2$).
  - Supposons que $u = \texttt{impair}$. Pour tout entier positif p et tous états $s, q \in Q$, $g_{s,s}^p(q) = g_{s,s}(q)$. Donc, pour tout $s \in Q$, on a $\chi_{i,F}^{g_{s,s}} \in \{\texttt{0}, \texttt{0}^1\}$. Ainsi, la stabilité de G par composition externe implique que deux états $g_{s,s}$ et $g_{s',s'}$ avec $s, s' \in Q$, ne sont pas distinguables dans $m(\mathrm{A})$, pour tous $s, s' \in Q$. Par conséquent,

$$sc(L(m(A))) \leq n^n - n + 1.$$

  - Supposons que $u = \texttt{0}^1$ et soient s, t deux éléments de Q. Si $\chi_{i,F}^{g_{s,t}} = \texttt{0}^1$, alors $t, s \in F$, ce qui implique que $\chi_{i,F}^{g_{t,s}} = \texttt{0}^1$. De même, si $\chi_{i,F}^{g_{t,s}} = \texttt{0}^1$, alors $\chi_{i,F}^{g_{s,t}} = \texttt{0}^1$. Ainsi, $\chi_{i,F}^{g_{s,t}} = \texttt{0}^1$ si et seulement si $\chi_{i,F}^{g_{t,s}} = \texttt{0}^1$. Donc, la stabilité de G par composition externe implique que les deux états $g_{s,t}$ et $g_{t,s}$ ne sont pas distinguables pour tous $s, t \in Q$. Par conséquent, on a

$$sc(L(m(A))) \leq n^n - \frac{1}{2}n(n-1) \leq n^n - n + 1.$$

— Finalement, supposons que $u = \mathtt{0}$, et soient s, s', t trois éléments de Q. Si $\chi_{i,F}^{g_{s,t}} = \mathtt{0}$, alors $t \notin F$, ce qui implique que $\chi_{i,F}^{g_{s',t}} = \mathtt{0}$. De même, $\chi_{i,F}^{g_{s',t}} = \mathtt{0}$ implique que $\chi_{i,F}^{g_{s,t}} = \mathtt{0}$. Ainsi, $\chi_{i,F}^{g_{s,t}} = \mathtt{0}$ si et seulement si $\chi_{i,F}^{g_{s',t}} = \mathtt{0}$. Donc, la stabilité de G par composition externe implique que les deux états $g_{s,t}$ et $g_{s',t}$ ne sont pas distinguables dans $m(\mathrm{A})$, pour tous $s, s' \in Q$. Par conséquent, on a

$$sc(L(m(A))) \leq n^n - n(n-1) \leq n^n - n + 1.$$

Le cas $i \in F$ est symétrique au cas $i \notin F$ de la façon suivante : on remplace dans la preuve toutes les occurrences de

— $s \in F$ par $t \notin F$,

— de $s \notin F$ par $t \in F$,

— de $t \in F$ par $s \notin F$,

— de $t \notin F$ par $s \in F$,

— de $\mathtt{0}$ par $(1, 1, \ldots)$,

— de $\mathtt{0}^1$ par $(1, 0, 0, \ldots, 0, \ldots)$,

— de $\mathtt{impair}$ par $(1, 0, 1, 0, \ldots, (n+1) \mod 2, \ldots)$.

De plus, dans le cas de $u = (1, 1, \ldots)$, c'est la finalité de $g_{s,t}$ et $g_{s,t'}$ qui est la même.

Pour résumer, dans tous les cas, on a $sc(L(m(A))) \leq n^n - n + 1$. Ainsi, $sc_{\otimes}(n) \leq n^n - n + 1$ pour toute opération amicale unaire $\otimes$.

□

Nous allons montrer que cette borne est serrée pour $\circledast_1 = \mathtt{op}(\{\mathtt{0}, \mathtt{0}^1\})$, où $\mathtt{0} = (0, 0, \ldots)$ et $\mathtt{0}^1 = (0, 1, 1, \ldots, 1, \ldots)$. Remarquons que pour tout langage régulier L sur un alphabet Σ, si $\varepsilon \notin L$, alors on a

$$\circledast_1(L) = \{w \in \Sigma^* | w \notin \sqrt[k]{L} \text{ pour tout } k > 0\} \cup \{w \in \Sigma^* | w \in \sqrt[k]{L}, \text{ pour tout } k > 0\}.$$

En effet, on a

$$\begin{aligned}\mathtt{op}(\{\mathtt{0}, \mathtt{0}^1\})(L) &= <\mathtt{0}, L> \cup <\mathtt{0}^1, L> \\ &= (\bigcap_{p>0} (\sqrt[p]{L})^C) \cup (\bigcap_{p>0} \sqrt[p]{L}) \\ &= \{w \in \Sigma^* | w \notin \sqrt[k]{L} \text{ pour tout } k > 0\} \cup \{w \in \Sigma^* | w \in \sqrt[k]{L}, \text{ pour tout } k > 0\}.\end{aligned}$$

Par contre, si $\varepsilon \in L$, alors on a

$$\circledast_1(L) = \emptyset.$$

En effet, on a

$$\begin{aligned}\mathtt{op}(\{\mathtt{0},\mathtt{0}^1\})(L) &= <\mathtt{0}, L> \cup <\mathtt{0}^1, L> \\ &= (\bigcap_{p\in\mathbb{N}} (\sqrt[p]{L})^C) \cup ((\sqrt[0]{L})^C \cap (\bigcap_{p\geq 1} \sqrt[p]{L})) \\ &= ((\sqrt[0]{L})^C \cap (\bigcap_{p\geq 1} \sqrt[p]{L})^C) \cup ((\sqrt[0]{L})^C \cap (\bigcap_{p\geq 1} \sqrt[p]{L})) \\ &= (\emptyset \cap (\bigcap_{p\geq 1} \sqrt[p]{L})^C) \cup (\emptyset \cap (\bigcap_{p\geq 1} \sqrt[p]{L})) \\ &= \emptyset.\end{aligned}$$

Notation 6.1.1. Soient $\mathtt{w}_1$=$\mathtt{mod}(\{\mathtt{0},\mathtt{0}^1\})$ et A_n l'AFD $\mathtt{w}_1(Mon^n)$, pour tout entier n.

On détermine une borne inférieure pour la complexité en états de $\circledast_1$ en calculant l'AFD minimal équivalent à A_n. Rappelons, par les Définition 3.1.1, Définition 5.2.1 et Définition 5.3.3, que l'alphabet de A_n est Γ_n=$[\![n]\!]^{[\![n]\!]}$, que son ensemble d'états est aussi $[\![n]\!]^{[\![n]\!]}$, et que chaque état ϕ de A_n est accessible à partir de son état initiale $Id_{[\![n]\!]}$ en lisant la lettre ϕ.

Notation 6.1.2. Pour toute fonction $\phi \in [\![n]\!]^{[\![n]\!]}$, posons $\kappa(\phi)$ la suite caractéristique $\chi^{\phi}_{0,\{n-1\}}$.

Afin de calculer l'équivalence de Nérode induite par A_n, nous montrons le résultat suivant.

Lemme 6.1.1. *Pour tout naturel n, et pour toutes fonctions distinctes $\phi, \psi \in [\![n]\!]^{[\![n]\!]}$ telles que ψ n'est pas constante, il existe $\zeta \in [\![n]\!]^{[\![n]\!]}$ tel que $\kappa(\zeta \circ \phi) \in \{\mathtt{0}, \mathtt{0}^1\}$ si et seulement si $\kappa(\zeta \circ \psi) \notin \{\mathtt{0}, \mathtt{0}^1\}$.*

Démonstration. On va considérer deux cas principaux : $\phi(0) \neq \psi(0)$ et $\phi(0) = \psi(0)$.

— Supposons que $\phi(0) \neq \psi(0)$. Si $\psi(0) \neq \psi(n-1)$, alors on fixe

$$\zeta(\phi(0)) = \zeta(\psi(n-1)) = 0 \text{ et } \zeta(\psi(0)) = n-1,$$

et ceci implique $\kappa(\zeta\circ\phi) = \mathtt{0}$ et $\kappa(\zeta\circ\psi) = (0,1,0,\ldots) \notin \{\mathtt{0},\mathtt{0}^1\}$. De façon symétrique, si $\phi(0) \neq \phi(n-1)$ alors on obtient le même résultat en permutant le rôle de ψ et ϕ dans le cas précédent. Maintenant, supposons que $\phi(0) = \phi(n-1)$ et $\psi(0) = \psi(n-1)$. Vu que ψ n'est pas constant, il existe $i \geq 1$ tel que $\psi(n-1) \neq \psi(i)$. Nous fixons $\zeta(\phi(0)) = \zeta(\phi(i)) = n-1$ et $\zeta(\psi(0)) = i$, ce qui implique que $\kappa(\zeta \circ \phi) = \mathtt{0}^1$ et $\kappa(\zeta \circ \psi) = (0,0,1,\ldots) \notin \{\mathtt{0},\mathtt{0}^1\}$.

— Supposons que $\phi(0) = \psi(0)$. Alors il existe $j > 0$ tel que $\phi(j) \neq \psi(j)$. On a $\phi(j) \neq \phi(0)$ ou $\psi(j) \neq \psi(0)$. Supposons que $\phi(j) \neq \phi(0)$ (l'autre cas peut être traité de la même manière). Si $j < n-1$, alors on fixe

$$\zeta(\phi(0)) = \zeta(\psi(j)) = j \text{ et } \zeta(\phi(j)) = n-1.$$

Dans ce cas, on a $\kappa(\zeta \circ \phi) = (0, 0, 1, \ldots) \notin \{\mathtt{0}, \mathtt{0}^1\}$ et $\kappa(\zeta \circ \psi) = \mathtt{0}$. Sinon si $j = n-1$, alors nous posons $\zeta(\phi(0)) = \zeta(\psi(n-1)) = n-1$ et $\zeta(\phi(n-1)) = 0$, ce qui implique que $\kappa(\zeta \circ \phi) = (0, 1, 0, \ldots) \notin \{\mathtt{0}, \mathtt{0}^1\}$ et $\kappa(\zeta \circ \phi) = \mathtt{0}^1$. Ce qui permet de conclure la preuve.

□

Par la Définition 5.3.3, le lemme ci-dessus, implique que deux états distincts de A_n, tels qu'au moins un d'eux est non constant, sont distinguables. Ainsi, tout état non constant est distinguable de tout autre état et la taille de l'ADF minimal équivalent à A_n est au moins égale à la cardinalité de l'ensemble de toutes les fonctions sur $[\![n]\!]$ qui ne sont pas constantes. Donc, pour tout $n \in \mathbb{N}_0$, la taille du AFD minimal équivalent à A_n est au moins $n^n - n + 1$. Ainsi, on a $sc_{\circledast_1}(n) \geq n^n - n + 1$, pour tout naturel n. Par conséquent, vu la Proposition 6.1.1, on a $sc_{\circledast_1}(n) = n^n - n + 1$ pour tout naturel n. On a donc prouvé le théorème suivant.

Théorème 6.1.1. Pour tout naturel n et pour toute opération amicale $\otimes$, $sc_{\otimes}(n) \leq n^n - n + 1$, et la borne est serrée pour $\circledast_1$, c'est-à-dire $sc_{\circledast_1}(n) = n^n - n + 1$. De plus, $\mathrm{L}(Mon^n)$ est un témoin de $\circledast_1$.

6.2 Le cas général

Contrairement au cas unaire, il y a des opérations amicales dont la complexité en états atteint la borne supérieure $\prod_{j=1}^{k} n_j^{n_j}$. On suppose que $k \geq 2$, et soit $\circledast_k$ l'opération k-aire $\mathtt{op}(E_k)$, où $E_k = \{\mathtt{0}, \mathtt{0}^1\} \backslash \{(\mathtt{0}, \ldots, \mathtt{0})\}$. Pour tout k-uplet de naturels $\underline{n}$, posons $A_{\underline{n}}$ l'AFD $\circledast_k(Mon_{\underline{n},\{n_1-1\},\ldots,\{n_k-1\}})$. De plus, pour tout k-uplet de fonctions $\underline{\phi}$, avec $\phi_j \in [\![n_j]\!]^{[\![n_j]\!]}$ pour tout $j \in \{1, \ldots, k\}$, on pose $\kappa(\underline{\phi})$ la suite caractéristique

$$\chi^{(\phi_1,\ldots,\phi_k)}_{(0,\ldots,0),(\{n_1-1\},\ldots,\{n_k-1\})}.$$

Théorème 6.2.1. Pour tout naturel $k \geq 2$ et pour tout k-uplet de naturels $\underline{n}$, on a $sc_{\circledast_k}(\underline{n}) = \prod_{j=1}^{k} n_j^{n_j}$. De plus, $sc_{\circledast_k}(L(A_{\underline{n}})) = \prod_{j=1}^{k} n_j^{n_j}$, c'est-à-dire $(L(M_1), \ldots, L(M_k))$ est un témoin pour $\circledast_k$, où $(M_1, \ldots, M_k) = Mon_{\underline{n},(\{n_1-1\},\ldots,\{n_k-1\})}$.

Démonstration. Cette preuve est inspirée du cas unaire. Soit k un naturel tel que $k \geq 2$, et soit $\underline{n}$ un k-uplet de naturels. De plus, soit $\underline{\phi}$ et $\underline{\psi}$ deux k-uplets de fonctions, avec $\phi_j, \psi_j \in [\![n_j]\!]^{[\![n_j]\!]}$ pour tout $j \in \{1, \ldots, k\}$, tels que $\underline{\phi} \neq \underline{\psi}$. Nous allons montrer qu'il existe $(\zeta_1, \ldots, \zeta_k)$, avec $\zeta_j \in [\![n_j]\!]^{[\![n_j]\!]}$ pour tout $j \in \{1, \ldots, k\}$, tel que $\kappa(\zeta_1 \circ \phi_1, \ldots, \zeta_k \circ \phi_k) \notin E_k$.

Soit l tel que $\phi_l \neq \psi_l$. On considère deux cas.

— Si ϕ_l et ψ_l sont des fonctions constantes, alors posons ζ_l toute fonction sur $[\![n_l]\!]$ telle que $\zeta_l(\phi(0)) = 0$ et $\zeta_l(\psi(0)) = n_l - 1$. De plus, pour tous $j \neq l$, nous posons ζ_j la fonction constante envoyant tout élément sur 0. On a

$$\kappa(\zeta_1 \circ \phi_1, \ldots, \zeta_k \circ \phi_k) = (\mathsf{0}, \ldots, \mathsf{0}) \notin E_k,$$

et

$$\kappa(\zeta_1 \circ \psi_1, \ldots, \zeta_k \circ \psi_k) = (u_1, \ldots, u_k) \in E_k, \text{ où } u_j = \begin{cases} \mathsf{0} & \text{si } j \neq l \\ \mathsf{0}^1 & \text{si} j = l \end{cases}$$

— Si une des fonctions ϕ_l et ψ_l n'est pas constante, alors on peut supposer que ψ_l n'est pas constante (l'autre cas étant symétrique). Ainsi, par le Lemme 6.1.1, il existe une fonction ζ_l sur $[\![n_l]\!]$ telle que $\kappa(\zeta_l \circ \phi_l) \in \{\mathsf{0}, \mathsf{0}^1\}$ si et seulement si $\kappa(\zeta_l \circ \psi_l) \notin \{\mathsf{0}, \mathsf{0}^1\}$. On suppose que $\kappa(\zeta_l \circ \phi_l) \in \{\mathsf{0}, \mathsf{0}^1\}$ (l'autre cas étant symétrique). De plus, pour tout $j \in \{1, \ldots, k\}$ avec $j \neq l$, nous posons ζ_j la fonction constante envoyant tout élément sur $n_j - 1$. On a

$$\kappa(\zeta_1 \circ \phi_1, \ldots, \zeta_k \circ \phi_k) = (\mathsf{0}^1, \ldots, \mathsf{0}^1, \kappa(\zeta_l \circ \psi_l), \mathsf{0}^1, \ldots, \mathsf{0}^1) \notin E_k,$$

vu que $\kappa(\zeta_l \circ \psi_l) \notin \{\mathsf{0}, \mathsf{0}^1\}$. De plus, on a

$$\kappa(\zeta_1 \circ \phi_1, \ldots, \zeta_k \circ \phi_k) = (\mathsf{0}^1, \ldots, \mathsf{0}^1, \kappa(\zeta_l \circ \phi_l), \mathsf{0}^1, \ldots, \mathsf{0}^1) \in E_k,$$

vu que $\kappa(\zeta_l \circ \phi_l) \in \{\mathsf{0}, \mathsf{0}^1\}$.

Par conséquent, dans les deux cas, il existe $(\zeta_1, \ldots, \zeta_k)$, avec $\zeta_j \in [\![n_j]\!]^{[\![n_j]\!]}$ pour tout $j \in \{1, \ldots, k\}$, tel que $\kappa(\zeta_1 \circ \phi_1, \ldots, \zeta_k \circ \phi_k) \in E_k$ si et seulement si $\kappa(\zeta_1 \circ \psi_1, \ldots, \zeta_k \circ \psi_k) \notin E_k$. On conclut grâce aux définitions 3.1.1, 5.2.1 et 5.3.3. □

Annexe

Annexe A

Exemples d'applications de modificateurs

Exemple A.0.1 (modificateur intersection)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, i_2, F_2, \delta_2)$ deux AFDs représentés aux Figures A.1 et A.2 qui sont tels que $\Sigma = \{a, b\}, Q_1 = \{0, 1\}, i_1 = 0, F_1 = \{1\}$ et $Q_2 = \{0, 1\}, i_2 = 0, F_2 = \{1\}$. Remarquons que le langage accepté par A_1 est $(a + b)b^*(a(a + b)b^*)^*$ et celui accepté par A_2 est $b^*aa^*(bb^*aa^*)^*$.

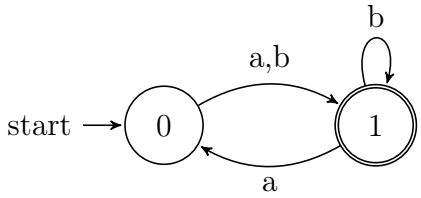


FIGURE A.1 – AFD A_1

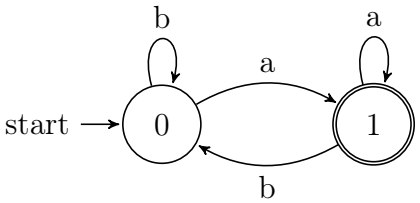


FIGURE A.2 – AFD A_2

En appliquant le modificateur intersection à ces automates, nous obtenons l'automate $\texttt{Inter}(A_1, A_2)$ représenté à la Figure A.3. En effet, on a

- $\texttt{Q}(Q_1, Q_2) = Q_1 \times Q_2 = \{0, 1\} \times \{0, 1\} = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$
- $\texttt{i}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = (i_1, i_2) = (0, 0)$
- $\texttt{f}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = F_1 \times F_2 = \{1\} \times \{1\} = (1, 1)$
- $\rho((i_1, i_2), (F_1, F_2), (\delta_1^a, \delta_2^a)) = (\delta_1^a, \delta_2^a)$

Le langage accepté par $\texttt{Inter}(A_1, A_2)$ est $(a + bb^*a(ba)^*a)((bb^*a(ba)^*a)^*(aa)^*)^*$.

Exemple A.0.2 (modificateur xor)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, i_2, F_2, \delta_2)$ les deux automates définis précédemment et représentés aux Figures A.1 et A.2.

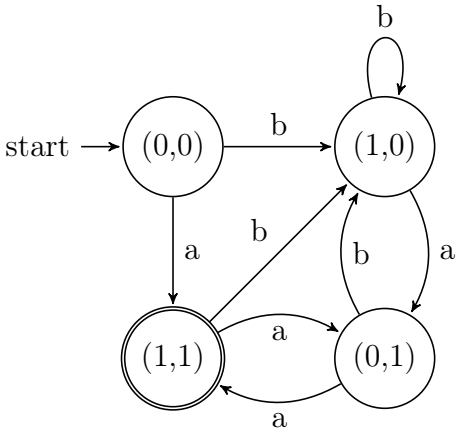


FIGURE A.3 – Automate $\texttt{Inter}(A_1, A_2)$

En appliquant le modificateur xor à ces automates, nous obtenons l'automate $\texttt{Xor}(A_1, A_2)$ représenté à la Figure A.4. En effet, on a

— $\texttt{Q}(Q_1, Q_2) = Q_1 \times Q_2 = \{0,1\} \times \{0,1\} = \{(0,0),(0,1),(1,0),(1,1)\}$

— $\texttt{i}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = (i_1, i_2) = (0,0)$

— $\texttt{f}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = F_1 \times (Q_2 \backslash F_2) \cup (Q_1 \backslash F_1) \times F_2$
$= \{1\} \times (\{0,1\} \backslash \{1\}) \cup (\{0,1\} \backslash \{1\}) \times \{1\}$
$= \{1\} \times \{0\} \cup \{0\} \times \{1\}$
$= \{(1,0),(0,1)\}$

— $\rho((i_1, i_2), (F_1, F_2), (\delta_1^a, \delta_2^a)) = (\delta_1^a, \delta_2^a)$

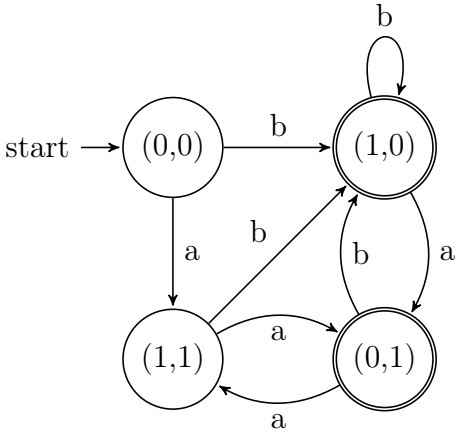


FIGURE A.4 – Automate $\texttt{Xor}(A_1, A_2)$

Exemple A.0.3 (modificateur concaténation)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, i_2, F_2, \delta_2)$ les deux automates définis précédemment et représentés aux Figures A.1 et A.2.

En appliquant le modificateur concaténation à ces automates, nous obtenons l'automate $\texttt{Conc}(A_1, A_2)$ représenté à la Figure A.5. En effet, on a

— $\texttt{Q}(Q_1, Q_2) = Q_1 \times 2^{Q_2}$
$= \{0,1\} \times 2^{\{0,1\}}$
$= \{0,1\} \times \{\emptyset, \{0\}, \{1\}, \{0,1\}\}$
$= \{(0,\emptyset), (1,\emptyset), (0,\{0\}), (1,\{0\}), (0,\{1\}), (1,\{1\}), (0,\{0,1\}), (1,\{0,1\})\}$

— $\texttt{i}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = (i_1, \emptyset) = (0, \emptyset)$ car $i_1 \notin F_1$

— $\texttt{f}((Q_1, Q_2), (i_1, i_2), (F_1, F_2)) = \{(q_1, E) \in Q_1 \times 2^{Q_2} | E \cap F_2 \neq \emptyset\}$
$= \{(q_1, E) \in \{0,1\} \times 2^{\{0,1\}} | E \cap \{1\} \neq \emptyset\}$
$= \{(0,\{1\}), (1,\{1\}), (0,\{0,1\}), (1,\{0,1\})\}$

— On a par example :
  — $\rho((i_1, i_2), (F_1, F_2), (\delta_1^a, \delta_2^a))(0,\emptyset) = (1, \{\emptyset, 0\}) = (1,\{0\})$ car $\delta_1(0) \in F_1$
  — $\rho((i_1, i_2), (F_1, F_2), (\delta_1^a, \delta_2^a))(1,\{1\}) = (0,\{1\})$ car $\delta_1(1) \notin F_1$
  — ...

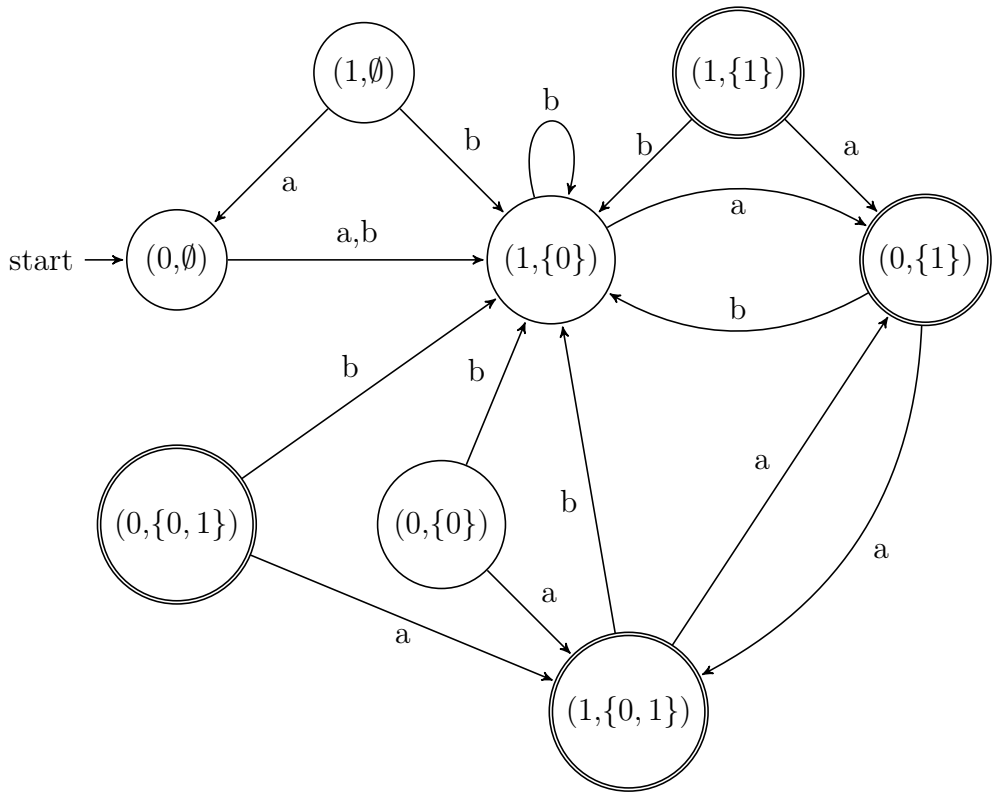


FIGURE A.5 – Automate $\texttt{Conc}(A_1, A_2)$

Si on retire les états inutiles, l'automate $\texttt{Conc}(A_1, A_2)$ se représente tel qu'à la Figure A.6. On peut vérifier que le langage accepté par cet automate est bien

$$(a+b)b^*(a(a+b)b^*)^*b^*aa^*(bb^*aa^*)^*.$$

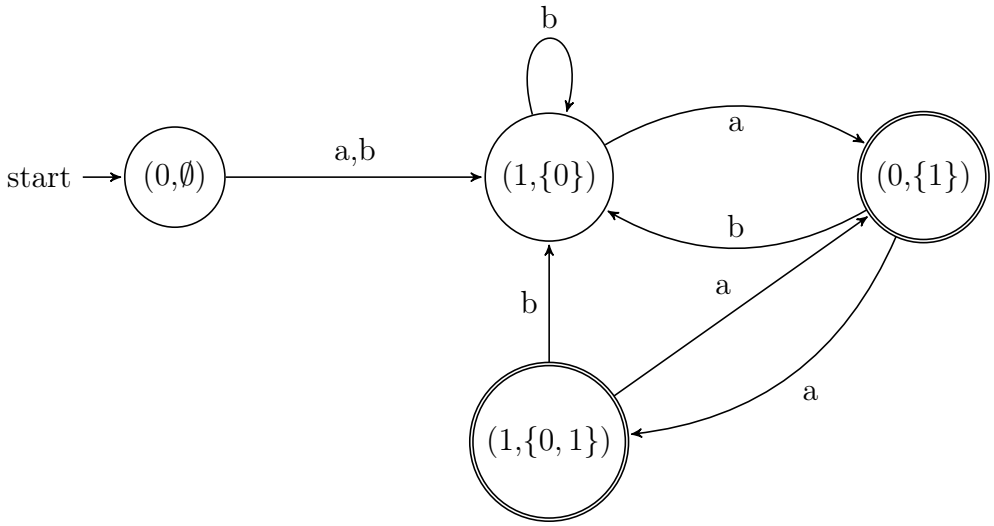


FIGURE A.6 – Automate $\texttt{Conc}(A_1, A_2)$ simplifié

Exemple A.0.4 (modificateur préfine)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ un AFD représenté à la Figure A.7 qui est tel que $\Sigma = \{a, b\}, Q_1 = \{0, 1, 2\}, i_1 = 0, F_1 = \{1, 2\}$. Remarquons que le langage accepté par cet automate est

$$b^*aa^*((b(a+b)b^*a)^* + (b(a+b)b^*a)^*b).$$

En appliquant le modificateur préfine à cet automate, nous obtenons l'automate $\texttt{Prefin}(A_1)$ représenté à la Figure A.8. En effet, on a

- $\texttt{Q}(Q_1) = Q_1 = \{0, 1, 2\}$
- $\texttt{i}(Q_1, i_1, F_1) = i_1 = 0$
- $\texttt{f}(Q_1, i_1, F_1) = F_1 = \{1, 2\}$
- On a par exemple :
  - $\rho^a(0) = \delta_1^a(0) = 1$ car $0 \notin F_1$
  - $\rho^a(1) = 1$ car $1 \in F_1$

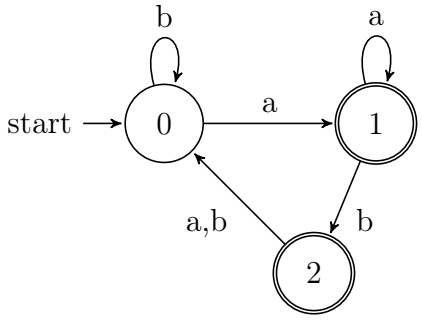


FIGURE A.7 – Automate A_1

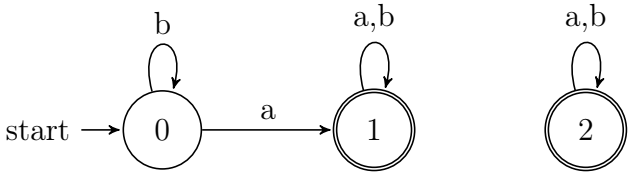


FIGURE A.8 – Automate $\texttt{Prefin}(A_1)$

— ...

En retirant l'état inutile, $\texttt{Prefin}(A_1)$ est tel que représenté par la Figure A.9. Le langage accepté par cet automate est

$$b^*a(a+b)^*.$$

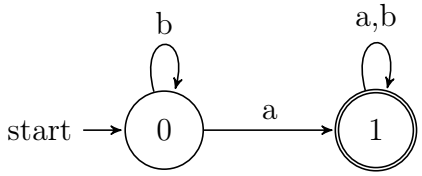


FIGURE A.9 – Automate $\texttt{Prefin}(A_1)$ simplifié

Exemple A.0.5 (modificateur racine)**.** Soit $A_1 = (\Sigma, Q_1, i_1, F_1, \delta_1)$ un AFD représenté à la Figure A.10 qui est tel que $\Sigma = \{a, b\}, Q_1 = \{0, 1\}, i_1 = 0, F_1 = \{1\}$.

En appliquant le modificateur racine à cet automate, nous obtenons l'automate $\text{Root}(A_1)$ représenté à la Figure A.11. Supposons que [ij] représente la fonction ϕ telle que $\phi(0) = i$ et $\phi(1) = j$. On a

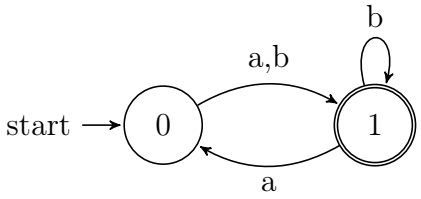


FIGURE A.10 – Automate A_1

— $\mathtt{Q}(Q_1) = Q_1^{Q_1} = \{0,1\}^{\{0,1\}}$
— $\mathtt{i}(Q_1, i_1, F_1) = Id = [01]$
— $\mathtt{f}(Q_1, i_1, F_1) = \{g | g^2(i_1) \in F_1\} = [11]$
— $\rho(i_1, F_1, \delta_1^a) = g \to (\delta_1^a \circ g)$
étant donné que $[10]$ représente $\phi(0) = 1$ et $\phi(1) = 0$ et que
  — $\delta_1^a(\phi(0)) = \delta_1^a(1) = 0$
  — $\delta_1^a(\phi(1)) = \delta_1^a(0) = 1$

  alors on a $\rho^a([10]) = [01]$.
  On a aussi
  — $\delta_1^b(\phi(0)) = \delta_1^b(1) = 1$
  — $\delta_1^b(\phi(1)) = \delta_1^b(0) = 1$

  d'où, $\rho^b([10]) = [11]$.
  Il en va de même pour les autres transitions.

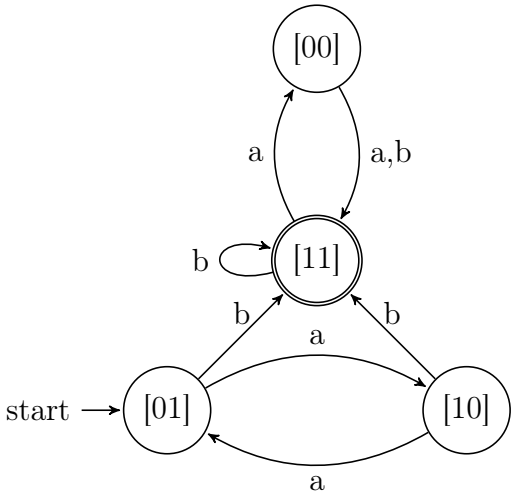


FIGURE A.11 – Automate $\mathtt{Root}(A_1)$

Bibliographie

[1] Pascal Caron, Edwin Hamel-de le Court et Jean-Gabriel Luque : Algebraic and combinatorial tools for state complexity : Application to the star-xor problem. *Electronic proceedings in theoretical computer science*, 305(Proc. GandALF 2019):154–168, 2019.

[2] Pascal Caron, Edwin Hamel-De le court, Jean-Gabriel Luque et Bruno Patrou : New tools for state complexity. 2018.

[3] Pascal Caron, Edwin Hamel-de-le court et Jean-Gabriel Luque : The state complexity of a class of operations involving roots and boolean operations. 2020.

[4] Pascal Caron, Edwin Hamel-de-le court et Jean-Gabriel Luque : A study of a simple class of modifiers : Product modifiers. *In Developments in Language Theory*, Lecture Notes in Computer Science, pages 110–121. Springer International Publishing, Cham, 2020.

[5] Sylvie Davies : A general approach to state complexity of operations : Formalization and limitations. *In Developments in Language Theory*, Lecture Notes in Computer Science, pages 256–268. Springer International Publishing, Cham, 2018.

[6] Yuan Gao, Nelma Moreira, Rogerio Reis et Sheng Yu : A survey on operational state complexity. 2015.

[7] Yuan Gao et Sheng Yu : State complexity of four combined operations composed of union, intersection, star and reversal. *In Descriptional Complexity of Formal Systems*, Lecture Notes in Computer Science, pages 158–171. Springer Berlin Heidelberg, Berlin, Heidelberg.

[8] Yuan Gao et Sheng Yu : State complexity of union and intersection combined with star and reversal. 2010.

[9] Edwin Hamel-De le court : *An algebraic theory for state complexity.* Thèse de doctorat, Normandie Université, 2020.

[10] Jozef Jirasek, Galina Jiraskova et Alexander Szabari : State complexity of concatenation and complementation of regular languages. *In Implementation and Application of Automata*, Lecture Notes in Computer Science, pages 178–189. Springer Berlin Heidelberg, Berlin, Heidelberg.

[11] Galina Jiraskova et Alexander Okhotin : On the state complexity of star of union and star of intersection. *Fundamenta informaticae*, 109(2):161–178, 2011.

[12] A. N. Maslov : Estimates of the number of states of finite automata. *Sov. Math., Dokl.*, 11:1373–1375, 1970.

[13] Michel Rigo : *Théorie des automates et langages formels*. Université de Liège, n édition, 2009-2010.

[14] Sheng Yu : State complexity of regular languages. 2000.

[15] Sheng Yu, Qingyu Zhuang et Kai Salomaa : The state complexities of some basic operations on regular languages. *Theoretical computer science*, 125(2):315–328, 1994.