Expressions régulières (https://en.wikipedia.org/wiki/Regular_expression)

et introduction aux transformations algébriques sur les langages

Feuille « 03 » (du 16 septembre 2024)

Rappel : la théorie des langages étudie des formalismes, en fait même des (méta)langages, pour définir des langages. Le plus simple de ces formalismes est celui des expressions régulières. Ce formalisme est en fait une famille de méta-langages (paramétrée par un vocabulaire de base). Il est fondamental en informatique : il a de très nombreuses propriétés mathématiques remarquables et est à la base de nombreux outils informatiques comme `grep`[1].

1 Introduction (motivation mathématique)

La syntaxe des expressions régulières remplace usuellement le (méta)symbole « $\cup$ » par « $+$ ». Ce remplacement est basé sur une analogie profonde entre certaines égalités (listées ci-dessous) sur les langages et celles sur les nombres. Dans cette analogie, l'union de langages (notée « $+$ ») ressemble à l'addition. Et la concaténation de langages (notée « $.$ ») ressemble à la multiplication.

Pour aller encore plus loin, on aurait pu remplacer « $\varnothing$ » par « 0 » et « $\{\varepsilon\}$ » par « 1 ». Mais cela aurait pu amener à des confusions car l'égalité « $1 + 1 = 2$ » n'a pas vraiment d'analogue sur les langages. On a plutôt « $\{\varepsilon\} + \{\varepsilon\} = \{\varepsilon\}$ » (cf. idempotence ci-dessous).

Égalités sur les langages analogues à celles sur les nombres[2] Soient L, L_1, L_2 et L_3 des langages quelconques sur un vocabulaire V. Les égalités suivantes sont valides :

associativé de « $+$ » et « $.$ »
$(L_1 + L_2) + L_3 = L_1 + (L_2 + L_3)$ $\qquad$ $(L_1.L_2).L_3 = L_1.(L_2.L_3)$

neutralité de $\varnothing$ vis-à-vis de $+$, et de ε vis-à-vis de $.$
$\varnothing + L = L + \varnothing = L$ $\qquad$ $\{\varepsilon\}.L = L.\{\varepsilon\} = L$

commutativité (uniquement) de « $+$ »
$L_1 + L_2 = L_2 + L_1$

distributivité de « $.$ » sur « $+$ »
$L_1.(L_2 + L_3) = (L_1.L_2) + (L_1.L_3)$ $\qquad$ $(L_2 + L_3).L_1 = (L_2.L_1) + (L_3.L_1)$

absorbance de $\varnothing$ sur « $.$ »
$\varnothing.L = L.\varnothing = \varnothing$

NB Une conséquence importante de cette analogie, c'est qu'une bonne partie des transformations sur **les systèmes d'équations** qu'on utilise pour résoudre des problèmes sur les *nombres* peuvent être réutilisées pour résoudre des problèmes sur les *langages*. Bien sûr, dans les systèmes d'équations sur les langages, on utilisera aussi des égalités plus spécifiques aux langages, notamment l'idempotence ci-dessous.

Idempotence de « $+$ » Pour tout langage L, on a : $L + L = L$.[3]

Préservation de la finitude Soient L_1 et L_2 deux langages finis (c-à-d. ayant un nombre fini de mots). Alors $L_1 + L_2$ et $L_1.L_2$ sont finis.
NB Pour fabriquer des langages infinis à partir de langages finis, on s'intéresse à l'opérateur d'itération de Kleene « * ». Celui-ci vérifie aussi l'égalité algébrique détaillée juste ci-dessous, qui très utile dans les techniques basées sur des systèmes d'équations.

1. https://en.wikipedia.org/wiki/Grep.
2. Une structure qui vérifie l'ensemble de ces égalités s'appelle un demi-anneau.
Voir https://fr.wikipedia.org/wiki/Demi-anneau
3. Sur les nombres (entiers, rationnels, réels ou complexes), une égalité $n + n = n$ implique $n = 0$.

Caractérisation algébrique de l'itération de Kleene Soit V un vocabulaire fixé. Soit L un langage sur V. Alors L^* est le *plus petit langage* (sur V) à vérifier l'égalité suivante

$$L^* = L.L^* + \{\varepsilon\}$$

Cela signifie que L^* vérifie l'égalité ci-dessus, et que si un langage X vérifie $X = L.X + \{\varepsilon\}$ alors $L^* \subseteq X$.

Exercice 1. (Vérification sur un exemple). Soit $L = \{\varepsilon\}$. Quels sont tous les langages X (sur V) qui vérifient $X = L.X + \{\varepsilon\}$? Que vaut L^* ?

Exercice 2. (avancé) Montrer la caractérisation de l'itération de Kleene donnée ci-dessus.

2 Définition des expressions régulières

Nous allons définir le formalisme des expressions régulières à niveau assez informel. Nous reviendrons sur le cadre mathématique nécessaire pour bien formaliser ce formalisme (c'est-à-dire sa propre méta-théorie) en période 2.

Syntaxe des expressions régulières Soit V un vocabulaire (fini) qui ne contient aucun des 7 méta-symboles suivant :

$$\varnothing \quad \epsilon \quad . \quad (\quad) \quad + \quad ^*$$

On définit $\mathbb{E}[V]$ comme le plus petit méta-langage sur le méta-vocabulaire $V \cup \{\varnothing, \epsilon, ., (,), +, ^*\}$ tel que :

- les 2 méta-symboles $\varnothing$ et ϵ sont des méta-mots de $\mathbb{E}[V]$;
- tout symbole de V est un méta-mot de $\mathbb{E}[V]$;
- pour tout $E_1 \in \mathbb{E}[V]$ et $E_2 \in \mathbb{E}[V]$, les deux méta-mots « (E_1+E_2) » et « $(E_1 . E_2)$ » appartiennent à $\mathbb{E}[V]$;
- pour tout $E \in \mathbb{E}[V]$, le méta-mot « E^* » est dans $\mathbb{E}[V]$.

Explication : cette définition donne un procédé pour *engendrer* l'ensemble des méta-mots de $\mathbb{E}[V]$; à partir des méta-mots de base (items 1 et 2), les items 3 et 4 indiquent comment les combiner pour fabriquer d'autres méta-mots.

Par définition, un méta-mot de $\mathbb{E}[V]$ s'appelle une expression régulière sur V.

Exercice 3. Quelles sont les méta-mots de $\mathbb{E}(\{\mathtt{a}, \mathtt{b}\})$ parmi les deux méta-mots suivants :

$$\mathtt{b+} \qquad \mathtt{(b+((a.b^*).a))^*}$$

Sémantique des expressions régulières On définit une fonction $\mathcal{L} \in \mathbb{E}[V] \to \mathcal{P}(V^*)$ qui associe à chaque expression régulière E de $\mathbb{E}[V]$, un langage noté $\mathcal{L}(E)$ sur V. La construction de $\mathcal{L}(E)$ est définie par *induction* (une généralisation de la récurrence) sur la construction de E :

- $\mathcal{L}(\varnothing) \stackrel{def}{=} \{\}$ et $\mathcal{L}(\epsilon) \stackrel{def}{=} \{\varepsilon\}$;
- pour tout $x \in V$, $\mathcal{L}(x) \stackrel{def}{=} \{x\}$;
- pour tout $E_1 \in \mathbb{E}[V]$ et $E_2 \in \mathbb{E}[V]$,
  $\mathcal{L}((E_1+E_2)) \stackrel{def}{=} \mathcal{L}(E_1) \cup \mathcal{L}(E_2)$ et $\mathcal{L}((E_1 . E_2)) \stackrel{def}{=} \mathcal{L}(E_1).\mathcal{L}(E_2)$;
- pour tout $E \in \mathbb{E}[V]$, $\mathcal{L}(E^*) \stackrel{def}{=} \mathcal{L}(E)^*$.

Exercice 4. Vérifier que $\mathcal{L}(\varnothing^*) = \{\varepsilon\}$ et $\mathcal{L}(\mathtt{((a+b).a)^*}) = \{\mathtt{aa}, \mathtt{ba}\}^*$.

NB. Il y a des conditions à vérifier sur la définition de $\mathbb{E}[V]$ pour garantir que la définition de $\mathcal{L}$ ci-dessus soit bien une définition correcte de fonction. On admet ici que c'est correct et on reviendra là-dessus en période 2. On utilisera parfois la notation classique $[\![E]\!]$ au lieu de $\mathcal{L}(E)$, l'utilisation de $\mathcal{L}$ rappelant juste que la sémantique produit un langage !

Soit L un langage (sur un certain vocabulaire V fixé), on dit que L est un **langage régulier** ssi il existe $E \in \mathbb{E}[V]$ tel que $L = \mathcal{L}(E)$.

Notations « informelles » sur les expressions régulières Les notations décrites ci-dessous permettent de simplifier la manipulation des expressions régulières, mais pourraient éventuellement introduire des confusions entre le niveau "méta" et le "méta-méta" (même si en général, c'est bien clair en fonction du contexte).

égalité vs identité Si $E_1 \in \mathbb{E}[V]$ et $E_2 \in \mathbb{E}[V]$, on utilisera « $E_1 = E_2$ » au lieu de $\mathcal{L}(E_1) = \mathcal{L}(E_2)$. On utilisera « $E_1 \equiv E_2$ » pour exprimer que E_1 et E_2 sont des *méta-mots identiques*. Par exemple $\varnothing^\star \not\equiv \epsilon$ mais $\varnothing^\star = \epsilon$.

abus de méta-méta-symboles On utilisera le méta-méta-symbole (par exemple « $\cup$ ») au lieu du méta-symbole correspondant (par exemple « `+` ») ou réciproquement (comme on avait déjà commencé à le faire en introduction). On pourra aussi omettre le méta-symbole « `.` ».

omission des parenthèses on pourra omettre les parenthèses inutiles en appliquant les conventions usuelles : « * » est prioritaire sur « . » qui est prioritaire sur « $+$ ». Ainsi, on pourra écrire `(b+((a.b*).a))*` $\equiv$ `(b` $+$ `ab*a)`*.

abréviations classiques On pourra utiliser $E^+ \equiv E.E^*$ (comme vu dans la feuille « 02 ») et $E^? \equiv E + \varepsilon$.

Dans toute la suite du cours, on utilisera systématiquement ces notations informelles, sauf en cas de définitions ou raisonnements par *induction* sur la syntaxe des expressions régulières. Auquel cas, on reviendra à la syntaxe formelle.

Exercices A faire en appliquant les notations ci-dessus bien sûr !

Exercice 5. Donner le langage L engendré par (`b` $+$ `ab`*`a`)* à l'aide d'un schéma par compréhension et exprimer ce schéma en français.

Exercice 6. Donner une expression régulière représentant chacun des langages suivants :

1. Les mots sur $\{0, 1\}$ contenant au moins un 0.
2. Les mots sur $\{0, 1\}$ de longueur paire.
3. Les mots sur $\{0, 1\}$ contenant deux 0 et/ou deux 1 consécutifs.
4. Les mots sur $\{0, 1\}$ où chaque 0 est suivi d'un 1.
5. Les mots sur $\{0, 1\}$ composés de 0 et de 1 alternés.

Exercice 7. Nous considérons une représentation des messages codés en Morse à l'aide du formalisme suivant. Les signaux qui peuvent être émis sont :

— le signal de début de phrase : D ;
— les signaux pour constituer des mots : L (signal long) et C (signal court)
— le signal de fin de phrase : F.

Un mot en Morse est une succession de trois signaux, longs ou courts. Une phrase en Morse est une séquence non vide de mots, précédée du signal de début de phrase, et terminée par le signal de fin de phrase. Un message en Morse est une séquence éventuellement vide de phrases.

Donner une expression régulière décrivant l'ensemble des messages valides en Morse.

Exercice 8. Caractériser (par une phrase en français) les langages représentés par les expressions régulières suivantes :

1. $0^*(10^*10^*10^*)^*$
2. $(1 + 01 + 001)^*(\varepsilon + 0 + 00)$
3. $1^*(0 + \varepsilon)1^*$

3 Théorèmes de base

Théorème des langages finis La classe des langages finis (sur un vocabulaire V fixé) correspond exactement à la classe des langages engendrés par les expressions régulières sans « * ».

Autrement dit, pour tout langage L avec un nombre fini de mots, il existe une expression régulière E ne contenant pas le méta-symbole « * » tel que $\mathcal{L}(E) = L$. Et, réciproquement, pour toute expression régulière E sans « * », le langage $\mathcal{L}(E)$ a un nombre fini de mots.

Exercice 9. (Vérification sur un exemple).
Sur le vocabulaire $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, soit $E \equiv (\varepsilon + \mathtt{a} + \mathtt{b})(\varepsilon + \mathtt{c}(\mathtt{a} + \varnothing))$.
Calculer $\mathcal{L}(E)$: quel est son nombre de mots ?

Exercice 10. Existe-t-il une expression régulière E contenant « * » et engendrant un langage fini ?

Exercice 11. Prouver le théorème.

Exercice 12. En s'inspirant de la preuve de l'exo précédent, construire une fonction $C \in \mathbb{E}[V] \to \mathbb{N}$ telle que si $E \in \mathbf{E}[V]$ est sans « * », alors $C(E)$ majore le nombre de mots dans $\mathcal{L}(E)$, c-à-d. $C(E) \geq \text{card}(\mathcal{L}(E))$. On construira C par une simple induction, sur le même modèle que $\mathcal{L}$.

Théorème de l'intersection (finie) de langages réguliers Soit E_1 et E_2 dans $\mathbb{E}[V]$. On peut **calculer** une expression régulière E dans $\mathbb{E}[V]$ telle que

$$\mathcal{L}(E) = \mathcal{L}(E_1) \cap \mathcal{L}(E_2)$$

Exercice 13. (Vérification sur des exemples). Sur $V = \{\mathtt{a}, \mathtt{b}\}$, vérifier avec

1. $E_1 = \mathtt{a}^*\mathtt{b}$ et $E_2 = \mathtt{b}\mathtt{a}^*$
2. $E_1 = (\mathtt{b} + \mathtt{a}\mathtt{b}^*\mathtt{a})^*$ et $E_2 = \mathtt{a}^*\mathtt{b}^*$ (sur $V = \{\mathtt{a}, \mathtt{b}\}$)

Théorème du complémentaire de langage régulier Soit E dans $\mathbb{E}[V]$. On peut **calculer** une expression régulière E' dans $\mathbb{E}[V]$ telle que

$$\mathcal{L}(E') = \overline{\mathcal{L}(E)}$$

où $\overline{\mathcal{L}(E)}$ est le complémentaire de $\mathcal{L}(E)$ défini par $\overline{\mathcal{L}(E)} \stackrel{def}{=} \mathcal{P}(V^*) \backslash \mathcal{L}(E)$.

Exercice 14. (Vérification sur un exemple). Vérifier avec $E = (\mathtt{b} + \mathtt{a}\mathtt{b}^*\mathtt{a})^*$ (sur $V = \{\mathtt{a}, \mathtt{b}\}$).

Discussion sur les preuves de ces deux derniers théorèmes Sur les exemples précédents, on arrive à faire le calcul indiqué par le théorème en passant par la sémantique. De même, on pourrait « facilement » faire le calcul de l'intersection pour les cas où E_1 et E_2 sont sans « * », en développant chaque E_i comme une somme finie de mots et en écrivant E comme la somme finie des mots communs aux deux sommes. Mais, généraliser cette méthode pour des expressions régulières avec « * » ne semble pas simple. Pour le complémentaire, la situation est encore plus compliquée, car si L est un langage fini alors $\overline{L}$ est infini (dès que $V \neq \varnothing$) et on devra nécessairement faire apparaître des « * ».

4 Vers la mise en équation des expressions régulières

Ces preuves sont en fait à priori « non triviales » et la suite du cours consiste justement à introduire des techniques et des formalismes pour les réaliser plus facilement. Au final, les outils qu'on va ainsi introduire sont tellement puissants et généraux qu'ils vont permettre de résoudre bien d'autres questions sur les langages réguliers.

4.1 Introduction au calcul de l'intersection d'expressions régulières

On cherche à résoudre l'équation $E = E_1 \cap E_2$ où E_1 et E_2 vérifient $E_1 = \mathtt{a}^*\mathtt{b}$ et $E_2 = \mathtt{ba}^*$. L'objectif est d'essayer de procéder par pure *transformations algébriques* (c-à-d. substitutions dans des équations sur les langages) **sans passer par un calcul dans la sémantique** !

Pour réaliser le calcul de l'intersection, les opérations « * » apparaissent comme assez problématiques : on commence donc par « les éliminer » à l'aide d'équations. Sur l'exemple, on remplace la sous-expression $\mathtt{a}^*$ en introduisant une *nouvelle* variable et *nouvelle* équation $X = \mathtt{a}\cdot X + \varepsilon$ (dans ce cas particulier, $\mathtt{a}^*$ est l'unique solution de cette équation, comme démontré en exercice plus loin).

On commence par éliminer « * » dans E_1. On obtient les équations

$$\begin{array}{rcl} E_1 & = & X\mathtt{b} \\ X & = & \mathtt{a}X + \varepsilon \end{array}$$

On fait de même dans E_2. On obtient l'équation $E_2 = \mathtt{b}X$ (en réutilisant ici le X précédent).

Il faut maintenant résoudre $E = E_1 \cap E_2$.
On obtient alors une équation $E = X\mathtt{b} \cap \mathtt{b}X$. Comment continuer ?
Ce serait quand même plus simple si on avait par exemple des termes de la forme « $u.X' \cap \mathtt{b}.X$ » avec $u \in V^+$ car on pourrait continuer le calcul en factorisant le *préfixe commun* entre u et $\mathtt{b}$.

Autrement dit, il faudrait mieux que le membre droit de l'équation de E_1 se *termine* par une variable plutôt que de *commencer* par une variable.
On « retravaille » donc l'équation de E_1 :

$$\begin{array}{lll} E_1 = X\mathtt{b} & & \\ \quad = (\mathtt{a}X + \varepsilon)\mathtt{b} & \text{en substituant } X \text{ par son membre droit} \\ \quad = \mathtt{a}(X\mathtt{b}) + \varepsilon\mathtt{b} & \text{par distributivité de . sur } + \\ \quad = \mathtt{a}E_1 + \mathtt{b} & \text{en substituant } X\mathtt{b} \text{ par } E_1 \end{array}$$

On peut alors continuer les calculs dans l'équation de E :

$$\begin{array}{lll} E = E_1 \cap E_2 & & \\ \quad = (\mathtt{a}E_1 + \mathtt{b}) \cap \mathtt{b}X & \text{en substituant } E_1 \text{ et } E_2 \text{ par leur membre droit} \\ \quad = (\mathtt{a}E_1 \cap \mathtt{b}X) + (\mathtt{b} \cap \mathtt{b}X) & \text{par distributivité de } \cap \text{ sur } + \\ \quad = \varnothing + \mathtt{b}(\varepsilon \cap \mathtt{b}X) & \text{par simplification et factorisation de préfixe de . sur } \cap \\ \quad = \mathtt{b}(\varepsilon \cap (\mathtt{a}X + \varepsilon)) & \text{en substituant } X \text{ et par son membre droit} \\ \quad = \mathtt{b}((\varepsilon \cap \mathtt{a}X) + (\varepsilon \cap \varepsilon)) & \text{par distributivité de } \cap \text{ sur } + \\ \quad = \mathtt{b}(\varnothing + \varepsilon) & \text{par simplification de préfixe distinct et idempotence} \\ \quad = \mathtt{b} & \end{array}$$

On a donc bien trouvé le résultat attendu $E = \mathtt{b}$ uniquement par des substitutions dans des systèmes d'équations sur les langages. Les égalités utilisées pour $\cap$ sont listées au paragraphe suivant. Elles permettent dans le cas général d'aboutir à un système d'équations exprimant l'intersection des deux langages, modulo le fait qu'il reste alors à éliminer les variables restantes dans les équations obtenues en réintroduisant des « * » (donc par le processus exactement « inverse » d'élimination des « * »).

De plus, dans le cas général, la justification des substitutions dans les équations est *plus subtile* que ci-dessus : on ne résonne en effet pas forcément sur *l'unique* solution des équations, mais la *plus petite solution*. Et, on va voir qu'il faut restreindre un peu les substitutions que nous employons pour garantir la préservation de cette plus petite solution.

L'algorithme général pour calculer une telle intersection sera donnée en feuille « 04 ». Dans la suite, nous étudions déjà comment généraliser notre méthode de calcul d'un système d'équations d'une expression régulière (et réciproquement).

Égalités sur $\cap$ utilisées dans le calcul général Soient L, L_1, L_2 et L_3 des langages quelconques sur un vocabulaire V. Soit w un mot de V^*. Les égalités suivantes sont valides :

distributivité de « $\cap$ » sur « $+$ »
$L_1 \cap (L_2 + L_3) = (L_1 \cap L_2) + (L_1 \cap L_3) \qquad (L_2 + L_3) \cap L_1 = (L_2 \cap L_1) + (L_3 \cap L_1)$

factorisation de préfixe commun de . sur $\cap$
$(w.L_1) \cap (w.L_2) = w.(L_1 \cap L_2)$

simplification de préfixe distinct de . sur $\cap$ pour $x, y \in V$ tels que $x \neq y$
$(w.x.L_1) \cap (w.y.L_2) = \varnothing \qquad \varepsilon \cap (x.L) = \varnothing \qquad (x.L) \cap \varepsilon = \varnothing$

idempotence $L \cap L = L$

4.2 Préservation de la plus petite (ou unique) solution des équations

Dans le calcul d'intersection, on a vu l'intérêt que les variables se trouvent à droite dans les termes des sommes dans les équations. Cela va aussi se vérifier pour d'autres traitements sur les systèmes d'équations. Cela influe sur notre algorithme pour éliminer les « * » dans les expressions régulières. En effet, sur un terme de la forme $B.A^*$ le remplacement de A^* par l'équation $X = A.X + \varepsilon$ introduit bien une variable à droite dans le terme d'origine $B.X$. Par contre, on a vu que ça ne fonctionnait pas directement pour un terme de la forme $A^*.B$. On a ici besoin du lemme d'Arden suivant qui est en fait une généralisation de la caractérisation de L^* en tant que plus petite solution de $X = L.X + \varepsilon$: cette caractérisation en effet un cas particulier du lemme d'Arden quand $A \stackrel{def}{=} L$ et $B \stackrel{def}{=} \varepsilon$.

Avant de regarder le lemme d'Arden, il est bien de comprendre pourquoi on en a besoin. En effet, dans le calcul d'intersection particulier ci-dessous, on ne l'a pas utilisé (on s'est plutôt appuyé sur une propriété d'unicité de solution dans $\mathtt{a}\cdot X + \varepsilon$ dont une preuve générale est donnée dans le corrigé d'un exercice ci-dessous). Remarquons d'abord, que si $\varepsilon \in A$, alors V^* est solution de l'équation « $X = A.X + B$ », car $A.V^* = V^*$. Donc, il y a *potentiellement* plusieurs solutions. Ce cas peut naturellement arriver en cas d'étoiles imbriquées. On l'a dit : la plupart des transformations qu'on va effectuer sur les systèmes d'équations préservent la plus petite (ou unique) solution. Il est donc important d'expliciter les **cas « pathologiques »** qu'il s'agit d'éviter !

Exemple de transformation qui ne préserve pas la plus petite solution On considère le système de 3 équations sur le langage sur $\{\mathtt{a}, \mathtt{b}\}$.

$$\begin{cases} X = Y + Z \\ Y = a + Y \\ Z = b \end{cases}$$

On effectue les 3 transformations suivantes sur l'équation de X :

$$\begin{array}{ll} X = Y + Z & \\ \quad = (a + Y) + Z & \text{remplacement de } Y \text{ par son membre droit} \\ \quad = a + (Y + Z) & \text{associativité de } + \\ \quad = a + X & \text{remplacement du membre droit initial de } X \text{ par } X \end{array}$$

Finalement, on obtient le système :

$$\begin{cases} X = a + X \\ Y = a + Y \\ Z = b \end{cases}$$

Par définition, la plus petite solution d'un tel système est un triplet de langages (X, Y, Z) qui est solution du système et tel que pour tout triplet solution, (X', Y', Z') on ait $X \subseteq X'$ et $Y \subseteq Y'$ et $Z \subseteq Z'$.

Exercice 15. Répondre aux questions suivantes :

1. Pour chacun des deux systèmes, quelle est sa plus petite solution ?
2. En quoi, les deux systèmes n'ont-ils pas la même plus petite solution ?
3. Quelles sont les transformations parmi les 3 ci-dessus qui ne préservent pas la plus petite solution de départ ?

Lemme d'Arden Soit V un vocabulaire, soient A et B deux langages sur V^*.
L'équation « $X = A.X + B$ » admet $A^*.B$ comme plus petite solution.

En particulier, on vérifie facilement les cas ci-dessous.

— $B = \varnothing$: l'équation $X = A.X$ admet $X = \varnothing$ comme plus petite solution ;
— $B = \varepsilon$: l'équation $X = A.X + \varepsilon$ admet $X = A^*$ comme plus petite solution ;
— $A = \varnothing$: l'équation $X = B$ admet $X = B$ comme plus petite solution ;
— $A = \varepsilon$: l'équation $X = X + B$ admet $X = B$ comme plus petite solution.

Exercice 16. (avancé) Montrer le lemme d'Arden.

Unicité de la solution dans l'équation d'Arden Dans l'équation d'Arden, si $\varepsilon \notin A$, alors $A^*.B$ est l'unique solution de l'équation.

Exercice 17. (très avancé) Montrer cette propriété d'unicité.
Indication : Soit $S = \{w \in V^* \mid \forall\, n \in \mathbb{N}, w \in A^n\}$ (qu'on note aussi $S = \bigcap_{n\in\mathbb{N}} A^n$).
Montrer que toute solution X à l'équation d'Arden vérifie $X \subseteq S.V^* + A^*B$ (sans supposer $\varepsilon \notin A$).
Conclure en montrant que si $\varepsilon \notin A$ et $A \neq \varnothing$, alors $S = \varnothing$.
Pour cette dernière partie, on calculera $\mu(S)$, où la fonction $\mu \in \mathcal{P}(V^*) \to \mathbb{N} \cup \{+\infty\}$ est telle que $\mu(L)$ retourne la longueur minimale d'un mot de L si $L \neq \varnothing$ ou une valeur $+\infty$ (considérée plus grande que tous les entiers) sinon.

Notion de fonction croissante Considérons une équation $X = f(X)$ avec $f \in \mathcal{P}(V^*) \to \mathcal{P}(V^*)$.
On illustrera en période 2 que dans le cas général, une telle équation n'a pas *forcément* une plus petite solution. Intuitivement, pour que cette équation ait une « signification », il faut que quand on « ajoute » des éléments à X en membre gauche, alors on en ajoute dans le membre droit « $f(X)$ ».
Ce qui, en toute généralité, peut être exprimé sous la forme

$$X \subseteq Y \Rightarrow f(X) \subseteq f(Y)$$

. On dit alors que f est croissante.

Exercice 18. Quelle est la fonction associée à l'équation d'Arden $X = A.X + B$? Vérifier qu'elle est croissante.

Le théorème de Knaster-Tarski (énoncé en toute généralité au paragraphe suivant) implique que toute équation « $X = f(X)$ », où f est croissante, a bien une plus petite solution. Notons que ce théorème s'applique à des fonctions transformant des ensembles de parties quelconques et pas seulement des langages. En période 2, c'est ce qui nous permettra notamment de justifier que les raisonnements faits ci-dessous sur **une seule** équation définissant **un seul** langage, se transposent sans difficulté à un n-uplet d'équations définissant un n-uplet de langages. Pour l'instant, on laisse cette question sous le tapis en admettant que le passage de 1 équation à n équations ne pose pas de problème. [4]

Dans **nos** équations sur les langages, la fonction associée à l'équation seront toujours croissante, car :

4. Pour les plus curieux et les plus curieuses, on utilisera une technique « assez haut niveau » basée sur le fait que

$$\mathcal{P}(E_1) \times \ldots \times \mathcal{P}(E_n) \simeq \mathcal{P}\left(\{1\}\times E_1 \cup \ldots \cup \{n\}\times E_n\right)$$

Voir les compléments de la feuille « 02 » sur la relation $\simeq$ entre ensembles. En résumé, toute construction qui s'applique à l'ensemble de droite peut être « transportée » sur l'ensemble de gauche (et réciproquement). Il suffit alors de vérifier que les effets de ce « transport » donne les propriétés attendues.

— les opérateurs binaires $+$, $\cap$ et $.$ sont croissants sur chacun de leurs arguments (par exemple, $X_1 \subseteq Y_1$ et $X_2 \subseteq Y_2$ implique $X_1 + X_2 \subseteq Y_1 + Y_2$) ;
— la fonction $X \mapsto X^*$ est elle aussi croissante ;
— la composée de deux fonctions croissantes est aussi une fonction croissante.

Autrement dit, nous n'utiliserons que des opérateurs dont les combinaisons produisent des fonctions croissantes.

Notons toutefois que la fonction $X \mapsto \overline{X}$ n'est pas croissante, mais décroissante. On ne pourra donc pas *uniquement* utiliser nos transformations algébriques sur les systèmes d'équations pour calculer le complémentaire d'une expression régulière. Mais, on y arrivera en les complétant avec une autre transformation simple du système d'équations lorsque celui-ci est mis dans une forme très particulière.

Théorème de Knaster-Tarski Soit E un ensemble. Une fonction $f : \mathcal{P}(E) \to \mathcal{P}(E)$ est dite croissante, ssi pour toute partie X et Y de E, $X \subseteq Y \Rightarrow f(X) \subseteq f(Y)$.

Soit $f : \mathcal{P}(E) \to \mathcal{P}(E)$ croissante. Il existe $S \in \mathcal{P}(E)$ tel que

1. pour tout $X \in \mathcal{P}(E)$, si $f(X) \subseteq X$ alors $S \subseteq X$;
2. $S = f(S)$.

NB : ce S est alors la *plus petite solution* de l'équation « $X = f(X)$ » (S est aussi appelé *plus petit point-fixe* de f).

Exercice 19. (avancé) Montrer le théorème de Knaster-Tarski.
Indication : Soit $P = \{X \in \mathcal{P}(E) \mid f(X) \subseteq X\}$ et $S = \{e \in E \mid \forall X \in P, e \in X\}$ (on note aussi $S = \bigcap P$).
Montrer que le S ainsi défini a les propriétés attendues.

Transformations préservant la petite solution Le théorème de Knaster-Tarski nous dit que pour toute fonction croissante f, la plus petite solution de « $X = f(X)$ » est la plus petite solution de « $X \supseteq f(X)$ » (et réciproquement). Quand on est sur un système qui n'a pas une solution unique, il est en fait préférable de travailler avec de raisonner avec les « $\supseteq$ » plus tôt que des « $=$ ». Cela explicite en effet mieux comment préserver cette plus petite solution.

Techniquement, pour prouver que les deux systèmes d'inégalités « $X \supseteq f(X)$ » à « $X \supseteq f'(X)$ » ont bien la même plus petite solution :

1. on peut faire une preuve adhoc du type *preuve du lemme d'Arden généralisé* donnée ci-dessous ;
2. ou, on peut invoquer un tel lemme (pour passer de f à f') ;
3. ou, on peut invoquer des transformations qui ne modifient pas l'ensemble des solutions, autrement dit telles que « $X \supseteq f(X) \Leftrightarrow X \subseteq f'(X)$ » ;
4. ou, on peut prouver qu'un des deux systèmes a une unique solution et repasser aux équations : on pourra alors utiliser l'ensemble des transformations algébriques sur les équations.

Reprenons l'exemple précédent qui ne préservait pas la plus petite solution. A partir du système initial (écrit avec des inégalités) :

$$\begin{cases} X \supseteq Y + Z \\ Y \supseteq a + Y \\ Z \supseteq b \end{cases}$$

On essaye alors d'appliquer la suite de transformations précédentes :

$$\begin{array}{lll} X \supseteq Y + Z & & \\ \quad \supseteq (a + Y) + Z & \text{remplacement de } Y \text{ par son membre droit} & \\ \quad \supseteq a + (Y + Z) & \text{associativité de } + & \\ \quad \supseteq a + \ldots & \longleftarrow & \text{en essayant de remplacer } Y + Z \text{ par } X, \\ & & \text{on « voit » que } X \supseteq Y + Z \text{ n'est pas dans le bon sens !} \end{array}$$

La subtilité c'est que même si l'absence de solution unique ne vient initialement que d'une équation (celle de Y), c'est bien tout le système initial qui est « contaminé » : on ne peut pas garder $X = Y + Z$ comme une égalité, car ça autoriserait la transformation incorrecte.

Lemme d'Arden généralisé Soit V un vocabulaire et soient A et B deux fonctions croissantes de $\mathcal{P}(V^*) \to \mathcal{P}(V^*)$. Les équations « $X \supseteq A(X).X + B(X)$ » et « $X \supseteq A(X)^*.B(X)$ » admettent la même plus petite solution.

Exercice 20. (avancé) Montrer le lemme d'Arden généralisé.

Exercice 21. Sur le vocabulaire $V = \{\mathtt{a}, \mathtt{b}\}$, on considère le système

$$\begin{cases} X \supseteq X + \mathtt{a}X + \mathtt{b}Y + \varepsilon \\ Y \supseteq \mathtt{b}X + \mathtt{a}Y \end{cases}$$

En utilisant le lemme d'Arden généralisé, trouver une expression régulière E telle que la plus petite solution de ce système d'inéquations vérifie $X = E$.