

LI214 - STRUCTURES DISCRETES

Benjamin BARON

Table des matières

1	Récursion, induction	3
1	Récurrance sur \mathbb{N}	3
2	Induction structurelle	6
2	Ensembles / ordres	11
1	Quelques rappels	11
2	Ensembles ordonnés	13
3	Maj, borne sup, max	14
4	Ensemble bien fondé	14
3	Terminaison	16
1	La conjecture de Syracuse	16
2	Les points fixes	17
3	Ordres bien fondés	18
4	Exemples	18
4	Outils pour la logique	20
1	Termes	20
2	Algèbre de Boole	21
3	Fonctions booléennes	22
4	Formes normales	23
5	Fonctions duales	23
5	Calcul propositionnel	25
1	Syntaxe	25
2	Sémantique	26
3	Conséquence sémantique	28
4	Conséquence logique (ou déduction)	30
6	Logique du premier ordre	31
1	Introduction	31
2	Syntaxe	31
3	Variables libres et liées	32
4	Sémantique	33
5	Propriétés sémantiques	36
6	Déduction	37

7	Automates finis	38
1	Motivations et exemples	38
2	Définitions	40
3	Détermination	41
4	Opérations	42
5	Langages rationnels	43
6	Minimisation	45
8	Exercices	48
1	Induction	48
2	Ensembles et ordre	59
3	Logique	66
4	Automates	71

Chapitre 1

Récursion, induction

1 Récurrence sur \mathbb{N}

Remarque. Une induction sur \mathbb{N} est équivalente à une récurrence sur \mathbb{N} .

Une induction sur d'autres ensembles est équivalente à une induction structurale.

1.1 Récurrence (premier principe d'inductions sur \mathbb{N})

On veut montrer une propriété \mathcal{P} sur \mathbb{N} où $\mathcal{P}(n)$ soit vraie pour tout $n \in \mathbb{N}$.

Théorème. Si $\mathcal{P}(0)$ est vraie et si pour tout $n \in \mathbb{N}$, $n \geq 0$, on suppose que $\mathcal{P}(n)$ est vraie et on montre que $\mathcal{P}(n+1)$ est vraie, alors $\mathcal{P}(n)$ est vraie pour tout $n \in \mathbb{N}$.

$$\left\{ \begin{array}{l} \text{Base :} \quad \mathcal{P}(0) \text{ vraie} \\ \text{Induction :} \quad \forall n \in \mathbb{N}, \mathcal{P}(n) \Rightarrow \mathcal{P}(n+1) \end{array} \right\} \left| \forall n \in \mathbb{N}, \mathcal{P}(n) \text{ vraie} \right.$$

Exemple. Soit $S_n = 1 + 2 + \dots + n$ pour tout $n \geq 0$

$$\mathcal{P}(n) : S_n = \frac{n(n+1)}{2} \text{ pour tout } n \in \mathbb{N}$$

Montrons ce résultat par récurrence :

- (i) Base. $S_0 = 0$, donc $\mathcal{P}(0)$ est vraie.
- (ii) Induction. Supposons $\mathcal{P}(n)$ vraie à un certain rang $n \in \mathbb{N}$ (ie. $2 \cdot S_n = n(n+1)$)
Montrons alors que $\mathcal{P}(n+1)$ est vraie.

$$\begin{aligned} 2 \cdot S_{n+1} &= 2(1 + 2 + \dots + n + (n+1)) \\ &= 2 \cdot S_n + 2(n+1) \\ &= n(n+1) + 2(n+1) \text{ par hypothèse de récurrence} \\ 2 \cdot S_{n+1} &= (n+1)(n+2) \end{aligned}$$

- (iii) Conclusion. On a montré :

- Base : $\mathcal{P}(0)$ vraie.
- Induction : Pour un certain $n \in \mathbb{N}$, et en supposant $\mathcal{P}(n)$, on en a déduit $\mathcal{P}(n+1)$.

Donc \mathcal{P} est vraie pour tout $n \in \mathbb{N}$.

Théorème (Variante). Une variante du théorème précédent :

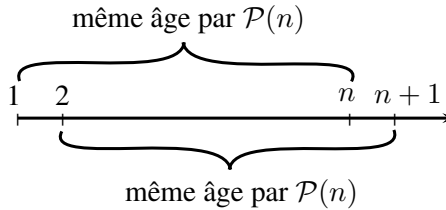
$$\left\{ \begin{array}{l} \text{Base :} \quad \mathcal{P}(n_0) \text{ vraie} \\ \text{Induction :} \quad \text{Soit } n \in \mathbb{N}, \forall n \geq n_0, \mathcal{P}(n) \Rightarrow \mathcal{P}(n+1) \end{array} \right\} \left| \forall n \geq n_0, \mathcal{P}(n) \text{ vraie} \right.$$

Exemple. Soit $\mathcal{P}(n)$: les n personnes de cette salle ont le même âge.

Base $n_0 = 1 \quad \mathcal{P}(1)$

Induction $\forall n \geq 2 \quad \mathcal{P}(n) \Rightarrow \mathcal{P}(n+1)$

Problème : $\mathcal{P}(1) \not\Rightarrow \mathcal{P}(2)$



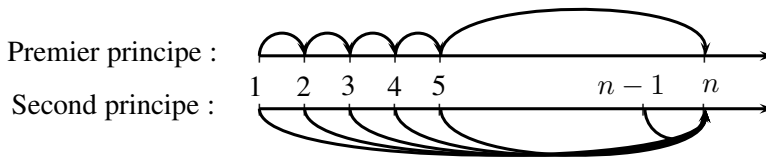
1.2 Récurrence complète sur \mathbb{N} (deuxième principe d'induction sur \mathbb{N})

Théorème. Soit $\mathcal{P}(n)$ une propriété sur \mathbb{N} . Si

$$(I') : \forall n \in \mathbb{N}, [(\forall k < n, \mathcal{P}(k) \text{ vraie}) \Rightarrow \mathcal{P}(n) \text{ vraie}]$$

Alors $\mathcal{P}(n)$ est vraie pour tout $n \in \mathbb{N}$.

Remarque. Pour résumer :



Théorème (Variante). Variante du théorème précédent pour montrer $\mathcal{P}(n)$ vraie pour tout $n \geq n_0$:

$$(I') : \forall n \geq n_0, n \in \mathbb{N}, [(\forall k \in \mathbb{N}, n_0 \leq k < n, \mathcal{P}(k)) \Rightarrow \mathcal{P}(n)]$$

Exemple. Soit $\mathcal{P}(n)$: n est décomposable en produit de facteurs premiers. Montrons que $\mathcal{P}(n)$ est vraie pour tout $n \in \mathbb{N}$.

$\forall n \geq 2$, n est décomposable en produit de facteurs premiers. Ainsi, $\mathcal{P}(2)$ est vraie.

(I'_2) : Si $\forall k \in \mathbb{N}, 2 \leq k < n$, k est décomposable en produit de facteurs premiers, alors n est décomposable en produit de facteurs premiers.

Soit $n \geq 2$. Il existe deux cas :

Cas 1. n est premier. Ainsi, $n = n$ est bien un produit d'un seul facteur premier (n).

Cas 2. n n'est pas premier. Ainsi, $n = n_1 \cdot n_2$ où $\begin{cases} n_1, n_2 > 1 \\ n_1, n_2 < n \end{cases}$

Par hypothèse d'induction, on a :

$$- 2 \leq n_1 < n \Rightarrow \mathcal{P}(n_1)$$

$$- 2 \leq n_2 < n \Rightarrow \mathcal{P}(n_2)$$

Or d'après la propriété \mathcal{P} , on a :

$$- \mathcal{P}(n_1) : n_1 = p_1 \cdots p_{i_1} \text{ où } p_{i_1} \text{ est premier}$$

$$- \mathcal{P}(n_2) : n_2 = p'_1 \cdots p'_{i_2} \text{ où } p'_{i_2} \text{ est premier}$$

De ce fait, $n = p_1 \cdots p_{i_1} \cdot p'_1 \cdots p'_{i_2}$ est décomposable en produit de facteurs premiers.

Ainsi, $\mathcal{P}(n_1)$ et $\mathcal{P}(n_2) \Rightarrow \mathcal{P}(n)$

Définition (Fonction définie par récurrence). Une fonction par récurrence sur \mathbb{N} est définie par :

$$\begin{cases} \text{Base :} & f(0) = a \\ \text{Induction :} & f(n+1) = h(n, f(n)) \end{cases}$$

Exemple. Soit la fonction f définie par récurrence par :

$$\begin{cases} \text{Base :} & f(0) = 1 \\ \text{Induction :} & f(n+1) = 2 \cdot f(n) \end{cases}$$

Prouvons par récurrence sur n que $f(n) = 2^n$.

- Base : $f(0) = 1 = 2^0$
- Induction : Supposons $f(n) = 2^n$
 $f(n+1) = 2 \cdot f(n) = 2 \cdot 2^n = 2^{n+1}$

Exemple. Donner l'écriture explicite de la fonction f définie par récurrence par :

$$\begin{cases} \text{Base :} & f(0) = 1 \\ \text{Induction :} & f(n+1) = 3 + f(n) \end{cases}$$

Montrons à l'aide d'une récurrence sur n que $f(n) = 3n + 1$:

- Base : $f(0) = 1 = 3 \cdot 0 + 1$
- Induction : Supposons $f(n) = 3n + 1$
 $f(n+1) = 3(n+1) + 1 = \underbrace{3n+1}_{f(n)} + 3 = f(n) + 3$

2 Induction structurelle

2.1 Ensembles définis par induction

2.1.1 Définition

Une définition inductive de l'ensemble X , $X \subset E$ est la donnée de :

- Base (B). B est un ensemble d'éléments tel que $B \subset X$
- Induction (I). Procédé pour construire des éléments nouveaux.
Soit K l'ensemble des opérations. $K = \{f_1, f_2, f_3, \dots\}$ où
 $f_1: E^{n_1} \rightarrow E$ $f_2: E^{n_2} \rightarrow E$ $f_3: E^{n_3} \rightarrow E$ où f_i est une opération d'arité n_i sur E
 (ie. n_i est le nombre d'arguments de la fonction f_i).
 $\forall x_1, x_2, \dots, x_n \in X, \forall f \in K, f$ d'arité $n_i, f(x_1, \dots, x_{n_i}) \in X$

2.1.2 Exemples sur les entiers

Soit $X \subset \mathbb{R}$ tel que :

$$\begin{cases} \text{Base :} & 0 \in X \\ \text{Induction :} & \forall x \in X, x + 1 \in X \end{cases}$$

$$E = \mathbb{R}, B = \{0\}$$

$$K = \{s\} \text{ avec } s: \mathbb{R} \rightarrow \mathbb{R} \\ x \mapsto x + 1$$

Remarque. $X = \mathbb{N}$ satisfait $(B) + (I)$.

Mais sans la condition $B = \{0\}$, $X = \mathbb{R}$, $X = \mathbb{Q}$ satisfont également $(B) + (I)$.

On ajoute alors une clause (qui sera sous-entendue par la suite) : X est le plus petit sous-ensemble de \mathbb{R} satisfaisant $(B) + (I)$.

Ainsi, $X = \mathbb{N}$.

Soit $X \subset \mathbb{N}$ tel que :

$$\begin{cases} \text{Base :} & 0 \in X \\ \text{Induction :} & \forall x \in X, x + 2 \in X \end{cases}$$

$$E = \mathbb{R}, B = \{0\}$$

$$K = \{s\} \text{ avec } s: \mathbb{R} \rightarrow \mathbb{R} \\ x \mapsto x + 2$$

Le plus petit X satisfaisant $(B) + (I)$ est l'ensemble $P = \{0, 2, 4, \dots\}$ des entiers pairs.

2.1.3 Exemple sur un alphabet

Soit $A = \{a_1, \dots, a_n\}$ un ensemble de lettres (alphabet)

A : alphabet et A^* : monoïdes libres engendrés par A .

A^* forment des suites finies de lettres de A .

L'opération sur A est la concaténation (notée $.$).

La suite vide (ie. qui n'a pas de lettres) est notée ε .

Exemple. Soit $A = \{a, b, \dots, y, z\}$, on a alors : $abc \in A^*$ et $abc.cb = abccb \in A^*$.

Soit $A = \{0, 1\}$, on a alors :

$$A^* = \{\varepsilon, 0, 1, \overbrace{00, 01, 10, 11}^{2^2 \text{ termes}}, \overbrace{000, 001, 010, 011, 100, 101, 110, 111, \dots}^{2^3 \text{ termes}}\}$$

A^* est infini, mais chaque élément de A^* est une suite finie.

$X \subset A^*$ tel que :

$$\begin{cases} \text{Base :} & \varepsilon \in X \\ \text{Induction :} & \forall a_i \in A, \forall a_j \in A, \forall x \in X, a_i a_j x \in X \end{cases}$$

$$E = A^*, B = \{\varepsilon\}$$

$$K = \{Pa_i a_j / a_i, a_j \in A \text{ et } Pa_i a_j \text{ est le préfixage par } a_i \text{ et } a_j\}$$

$$|K| = 2^{|A|} \text{ où } |A| \text{ est le cardinal de l'ensemble } A \text{ (ie. le nombre de lettres de } A).$$

X est l'ensemble des mots de longueur paire sur A .

Soit $A = \{a\}$, alors $X = \{\varepsilon, aa, aaaa, aaaaaa, \dots\}$

$$\text{Soit } A = \{a, b\}, \text{ alors } X = \{\varepsilon, \overbrace{aa, ab, ba, bb}^{2^2 \text{ termes}}, \overbrace{aaaa, aaab, aaba, abaa, baaa, aabb, abba, bbaa, abab, baba, baab, abbb, babb, bbab, bbba, bbbb, \dots}^{2^4 \text{ termes}}\}$$

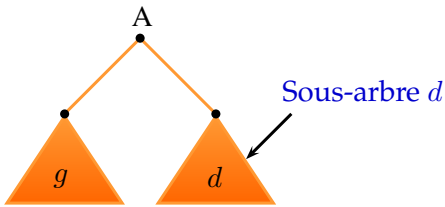
2.1.4 Exemple sur des arbres binaires

Soit $X: AB$, l'ensemble des arbres binaires sur l'alphabet A où $AB \subset [A \cup \{\emptyset, (,), , \}]^* = E$ tel que :

$$\begin{cases} \text{Base :} & \emptyset \in AB \\ \text{Induction :} & \forall g, d \in AB, \forall a \in A, (a, g, d) \in AB \end{cases}$$

$$E = AB, B = \emptyset$$

$$K = \{p_a, a \in A, p_a \text{ d'arité } 2/p_a(g, d) = (a, g, d)\}$$



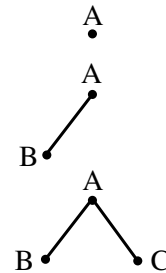
Exemple. Soit $A = \{a, b, c, d\}$. On a alors :

$$\emptyset$$

$$(a, \emptyset, \emptyset)$$

$$(a(b, \emptyset, \emptyset), \emptyset)$$

$$(a(b, \emptyset, \emptyset), (c, \emptyset, \emptyset))$$



2.2 Fonctions définies par induction structurelle

2.2.1 Définition

Sur X défini inductivement, on peut définir une fonction comme suit :

- Base. $g(b)$ est donné explicitement pour tout $b \in B$
- Induction. Un procédé pour calculer $g(f(x_1, x_2, \dots, x_{n_i}))$ en fonction de x_1, x_2, \dots, x_{n_i}

Exemple. Soit la fonction g définie par :

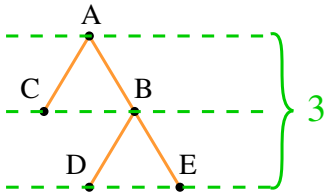
$$\begin{cases} \text{Base :} & g(0) = 1 \\ \text{Induction :} & g(n+1) = (n+1) \times g(n) \end{cases}$$

La fonction g est alors définie explicitement par : $g: \mathbb{N} \longrightarrow \mathbb{N}^*$
 $n \longmapsto n!$

2.2.2 Fonctions relatives à un arbre binaire

Soit $h(b)$ la hauteur d'un arbre binaire b définie par :

$$\begin{cases} \text{Base :} & h(\emptyset) \\ \text{Induction :} & h((a, g, d)) = 1 + \max(h(g), h(d)) \end{cases}$$



Parcours préfixe d'un arbre binaire. « De haut en bas et de gauche à droite »

Le parcours préfixe d'un arbre binaire est défini par :

$$\begin{cases} \text{Base :} & \text{pref}(\emptyset) = \varepsilon \\ \text{Induction :} & \text{pref}((a, g, d)) = a \text{ pref}(g) \text{ pref}(d) \end{cases}$$

Le parcours infixe de l'arbre ci-dessus est : A - C - B - D - E

Parcours suffixe (postfixe) d'un arbre binaire. « De gauche à droite et de bas en haut »

Le parcours suffixe d'un arbre binaire est défini par :

$$\begin{cases} \text{Base :} & \text{suf}(\emptyset) = \varepsilon \\ \text{Induction :} & \text{suf}((a, g, d)) = \text{suf}(g) \text{ suf}(d) a \end{cases}$$

Le parcours infixe de l'arbre ci-dessus est : C - D - E - B - A

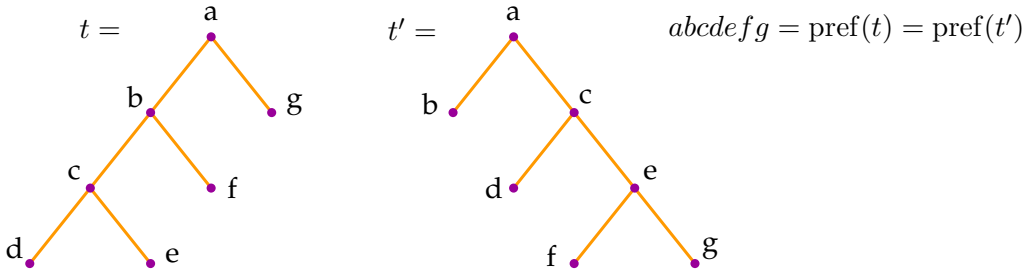
Parcours infixe d'un arbre binaire. « Parcourir le sous-arbre gauche, puis la racine, puis le sous-arbre droit »

Le parcours infixe d'un arbre binaire est défini par :

$$\begin{cases} \text{Base :} & \inf(\emptyset) = \varepsilon \\ \text{Induction :} & \inf((a, g, d)) = \inf(g) a \inf(d) \end{cases}$$

Le parcours infixe de l'arbre ci-dessus est : C - A - D - B - E

Remarque. Deux arbres binaires différents peuvent avoir le même parcours préfixe.



2.3 Preuves par induction structurelle

2.3.1 Définitions

Définition. Preuve par induction d'une propriété \mathcal{P} des éléments de X

- Base (B) : vérifier que $\mathcal{P}(b)$ est vraie pour tout $b \in B$
- Induction (I) : Montrer que si $\mathcal{P}(x_1)$ et $\mathcal{P}(x_n)$ sont vraies, alors $\mathcal{P}(f(x_1, \dots, x_n))$ sont vrais pour tous $f \in K$ d'arité n et $x_i \in X$.
- Conclusion : $B + I \Rightarrow \mathcal{P}(x)$ vraie pour tout $x \in X$

Définition. Ensemble X défini par induction structurelle

- Base (B) : $B \in X$
- Induction : $\forall f \in K, \forall x_1, \dots, x_n \in X, f(x_1, \dots, x_n) \in X$

2.3.2 Induction sur \mathbb{N} :

Exemple. L'ensemble \mathbb{N} est défini par :

- Base (B) : $0 \in \mathbb{N}$
- Induction (I) : $\forall n \in \mathbb{N}, n + 1 \in \mathbb{N}$
- $K = \{s\}$, s d'arité 1 (un argument) et $s(x) = x + 1$

Pour montrer une proposition $\mathcal{P}(n)$ vraie pour tout $n \in \mathbb{N}$:

- Base (B) : $\mathcal{P}(0)$ vraie
- Induction : Si $\mathcal{P}(n)$ vraie, alors $\mathcal{P}(n + 1)$ est vraie
- Conclusion : $\mathcal{P}(n)$ vraie pour tout $n \in \mathbb{N}$

2.3.3 Liste sur l'alphabet $A = L$

Soit l'alphabet $A = L$ défini par :

- Base (B) : $\forall a \in A, (a) \in L$ et $(\varepsilon) \in L$
- Induction (I) : $\forall x_1, x_2 \in L, x_1 x_2 \in L$ et $(x_1) \in L$ où (x_1) est la liste contenant l'élément x_1
- $K = \{f_1, f_2\}$ où :
 - f_1 d'arité 1 : $f_1(x) = (x)$
 - f_2 d'arité 2 : $f_2(x_1, x_2) = x_1 x_2$

Exemple. Soit l'alphabet $A = \{a, b\}$

$L = (\varepsilon) (a) (b) (\varepsilon) (a) (a) (b) (a) (a) ((a)) ((a) (b)) (a) ((a) (b)) ((a) ((a) (b)))$

Remarque. Les éléments $)a(, (\varepsilon) (a), (a b)$ ne sont pas des éléments de L

Proposition. $\forall x \in L, x$ a autant de parenthèses ouvrantes que de parenthèses fermantes.

Démonstration. Soit $|m|_a$ le nombre de a dans $m \in A$.

Soit $\mathcal{P}(x)$ la propriété : $|x|_(< = |x|_>$

- Base (B) : $\mathcal{P}((\varepsilon))$ vraie car $|(\varepsilon)|_(< = |(\varepsilon)|_> = 1$
 $\forall a \in A, \mathcal{P}((a))$ vraie car $|a|_(< = |a|_> = 1$
- Induction (I) : Il y a deux éléments à montrer :
 - Montrons : $\mathcal{P}(x) \Rightarrow \mathcal{P}((x))$ pour tout $x \in L$
 Supposons $\mathcal{P}(x)$.
 On a : $|x|_(< = 1 + |x|_(< = 1 + |x|_> = |(x)|_>$ par hypothèse d'induction : $\mathcal{P}(x) : |x|_(< = |x|_>$
 - Montrons : $\mathcal{P}(x_1)$ et $\mathcal{P}(x_2) \Rightarrow \mathcal{P}(x_1 x_2)$ pour tous $x_1, x_2 \in L$
 Supposons $\mathcal{P}(x_1)$ et $\mathcal{P}(x_2)$ (ie. $|x_1|_(< = |x_1|_>$ et $|x_2|_(< = |x_2|_>$).
 On a : $|x_1 x_2|_(< = |x_1|_(< + |x_2|_(< = |x_1 x_2|_> = |x_1|_> + |x_2|_>$
 Or par hypothèse d'induction, $|x_1 x_2|_(< = |x_1 x_2|_>$
- Conclusion : pour tout $x \in L, x$ a autant de parenthèses ouvrantes que de parenthèses fermantes.

□

Chapitre 2

Rappels sur les ensembles et les ordres

1 Quelques rappels

1.1 Fonctions

f est injective si $\forall x, y \in E ; f(x) = f(y) \Rightarrow x = y$

f est surjective si $\forall y \in F, \exists x \in E ; f(x) = y$

f est bijective si elle est injective et surjective : $\forall y \in F, \exists ! x \in E ; f(x) = y$

1.2 Groupes

Définition (Loi de composition interne). Une loi de composition interne \star sur E est une application telle que : $\star : E \longrightarrow E \times E$

\star est commutative si : $\forall (x, y) \in E^2, x \star y = y \star x$

Définition (Groupe). Un groupe (E, \star) est un ensemble E muni d'une loi de composition interne \star satisfaisant les propriétés suivantes :

- Associativité : $\forall (x, y, z) \in \mathbb{R}^3$, on a : $x \star (y \star z) = (x \star y) \star z$
- Existence d'un élément neutre : $\exists e \in E : \forall x \in E, x \star e = e \star x = x$
- Existence d'un symétrique : tout élément $x \in E$ admet un symétrique noté $x^{-1} \in E$
 $\forall x \in E, \exists x^{-1} \in E : x \star x^{-1} = x^{-1} \star x = e$

Si \star est commutative, alors le groupe est dit *abélien* ou *commutatif*.

Définition (Sous-groupe). Soit (G, \star) un groupe et H un sous-ensemble de G [$H \subset G$].

On dit que H est un sous-groupe de (G, \star) si :

- (i) $\forall (x, y) \in H^2, x \star y \in H$
- (ii) $e_G \in H$
- (iii) $\forall x \in H, x^{-1} \in H$

La loi de composition interne \star induit une application de $H \times H$ dans H :

$$\begin{aligned} \star & : H \times H \longrightarrow H \\ & (h, h') \longrightarrow h \star h' \end{aligned}$$

H est un groupe pour cette loi de composition interne. En effet, \star est associative dans G , donc la loi de composition interne qu'elle induit dans H est associative.

De plus, $e_G \in H$ est l'élément neutre dans H [et G].

1.3 Monoïdes

Définition (Monoïde). Un monoïde (E, \star, e) est une structure algébrique constituant en un ensemble E muni d'une loi de composition interne \star et d'un élément neutre e si :

- (i) Stabilité : $\forall (x, y) \in E^2, x \star y \in E$
- (ii) Associativité : $\forall (x, y, z) \in E^3, x \star (y \star z) = (x \star y) \star z$
- (iii) Élément neutre : $\exists e \in E, \forall x \in E, x \star e = e \star x = x$

Un monoïde est commutatif lorsque : $\forall (x, y) \in E^2, x \star y = y \star x$

Exemple. Les structures $(\mathbb{N}, +, 0)$ et $(\mathbb{N}, \cdot, 1)$ sont des monoïdes commutatifs.

La structure $(\mathbb{Z}, +)$ est un groupe commutatif, donc en particulier, un monoïde commutatif.

Soit A un alphabet.

La structure $A^* = \{\text{mots sur } A\}$ est un monoïde noté (A^*, \cdot) où \cdot est la concaténation.

Soit $A = \{a, b, c\}$. Montrons que A^* est un monoïde :

- (i) Stabilité : Soient $ab \in A^*$ et $aac \in A^*$. On a bien $ab.aac \in A^*$
- (ii) Associativité : Soient $ab \in A^*$ et $aac \in A^*$. On a bien $(ab).aac = ab.(aac)$
- (iii) Élément neutre. Soient $\varepsilon \in A^*$ le mot vide et $m \in A^*$. On a bien : $\varepsilon.m = m.\varepsilon = m$

1.4 Relations sur un ensemble

Définition (Relation binaire). Une relation binaire \mathcal{R} d'un ensemble E vers un ensemble F est définie par une partie \mathcal{G} de $E \times F$.

Si $(x, y) \in \mathcal{G}$, on dit que x est en relation avec y et on note $x\mathcal{R}y$.

- Dans le cas particulier où $E = F$, on dit que \mathcal{R} est une relation binaire définie sur E ou dans E .
- Dans le cas où $E = F \times F$, on parlera d'une relation ternaire interne sur F .
- Plus généralement, si $E = F^{n-1}$, on parlera de relation n -aire sur F .

Remarque. Dans la suite, on considérera une relation binaire \mathcal{R} définie sur un ensemble E

Ainsi, on aura : $\forall (x, y) \in E^2, x\mathcal{R}y \Leftrightarrow (x, y) \in \mathcal{R}$

Soit \mathcal{R} une relation binaire sur un ensemble E .

\mathcal{R} peut avoir les propriétés suivantes :

1.4.1 Propriétés liées à la réflexivité

Relation réflexive : La relation \mathcal{R} est réflexive si tout élément de E est en relation avec lui-même (ie. $\forall x \in E, x\mathcal{R}x$)

Relation irreflexive : La relation \mathcal{R} est irreflexive si aucun élément de E n'est en relation avec lui-même (ie. $\forall x \in E, x \not\mathcal{R}x$)

1.4.2 Propriétés liées à la symétrie

Relation symétrique : La relation \mathcal{R} est symétrique si, et seulement si, lorsqu'un premier élément de E est en relation avec un second élément de E , le second élément de E est en relation avec le premier (ie. $\forall (x, y) \in E^2, (x\mathcal{R}y) \Rightarrow (y\mathcal{R}x)$)

Relation antisymétrique : La relation \mathcal{R} est symétrique si, et seulement si, lorsque deux éléments de E sont en relation mutuelle, ils sont en fait confondus (ie. $\forall (x, y) \in E^2, [(x\mathcal{R}y) \wedge (y\mathcal{R}x)] \Rightarrow (x = y)$)

1.4.3 Autres propriétés

Relation transitive : La relation \mathcal{R} sur E est transitive si, et seulement si, lorsqu'un premier élément de E est en relation avec un deuxième élément lui-même en relation avec un troisième, le premier élément est aussi en relation avec le troisième (ie. $\forall (x, y, z) \in E^3, [(x\mathcal{R}y) \wedge (y\mathcal{R}z)] \Rightarrow (x\mathcal{R}y)$)

Relation totale : La relation \mathcal{R} sur E est totale ssi pour toute paire d'éléments de E , elle institue au moins un lien entre les deux éléments considérés (ie. $\forall (x, y) \in E^2, (x\mathcal{R}y) \vee (y\mathcal{R}x)$).

Relation d'équivalence : Une relation d'équivalence est une relation réflexive, transitive et symétrique.

L'exemple le plus simple est celui de l'égalité.

Relation d'ordre : Une relation d'ordre est une relation réflexive, transitive et antisymétrique.

- Si la relation est totale, on dit que l'ordre est *total* (cas de la relation \geq)
- Si tous les éléments de E ne sont pas comparables, on dit que l'ordre est *partiel*

Relation bien fondée : Soit E un ensemble non vide. On dit qu'une relation \mathcal{R} sur E est bien fondée si, et seulement si, elle vérifie l'une des deux conditions suivantes :

- Pour toute partie X non vide de E , il existe un élément $x \in X$ n'ayant aucun \mathcal{R} -antécédent dans X . Un \mathcal{R} -antécédent de x dans X est un élément $y \in X$ vérifiant $y\mathcal{R}x$.
- Il n'existe pas de suite infinie $(x_n)_{n \in \mathbb{N}}$ d'éléments de E telle que l'on ait $x_{n+1}\mathcal{R}x_n$ pour tout $n \in \mathbb{N}$

2 Ensembles ordonnés

Dans cette partie, on considèrera que E est un ensemble.

Définition (Ordre large). \leq est un ordre large sur E si, et seulement si, \leq est une relation binaire sur E qui est :

- Réflexive : $\forall x \in E, x \leq x$
- Antisymétrique : $\forall x, y \in E, (x \leq y \text{ et } y \leq x) \Rightarrow x = y$
- Transitive : $\forall x, y, z \in E, (x \leq y \text{ et } y \leq z) \Rightarrow x \leq z$

Définition (Ordre strict). $<$ est un ordre strict sur E si, et seulement si, $<$ est une relation binaire sur E qui est :

- Irréflexible : $\forall x \in E, x \not< x$
- Transitive : $\forall x, y, z \in E, (x < y \text{ et } y < z) \Rightarrow x < z$

Définition (Ensemble totalement ordonné). Soit $A \subset E$ un ensemble totalement ordonné et $a_1, \dots, a_n \in A$, alors $a_1 < a_2 < \dots < a_n$.

On dit alors que l'ordre est *total*.

Dans le cas contraire, si $a_1 \not\leq a_2 \not\leq \dots \not\leq a_n$, on dit que A est *partiel*.

Exemple. Quelques exemples d'ordre :

- $\mathbb{N} : \leq$ est un ordre large total
- $\mathbb{N} : <$ (ie. $x < y$ si, et seulement si, $x \leq y$ et $x \neq y$) est un ordre strict partiel
- $\mathbb{N} : \text{Ordre } x|y$ (ie. si x divise y , alors $x \leq y$) est un ordre large partiel
- $A^* : \text{Ordre préfixe } <_p$ (ie. $\forall m_1, m_2, m \in A^*, m_1 <_p m_2$ si, et seulement si, $m_2 = m_1 m$) n'est pas un ordre total : $\exists x, y \in A^*, x \not\leq_p y$ et $y \not\leq_p x$

Exemple. bon $<_p$ bonbon et bon $<_p$ bonjour

- $A : \text{Ordre lexicographique (ou alphabétique) est un ordre total.}$

Montrons que la relation $x|y$ est bien un ordre :

- Réflexive : $x|x$
- Transitive : si $x|y$ et $y|z$, on a $y = kx$ et $z = k'y$, et $z = kk'x$, donc $x|z$
- Antisymétrique : si $x|y$ et $y|x$, on a $y = kx$ et $x = k'y$, donc $y = kk'x$, donc $kk' = 1$.
Puisque k et k' sont inversibles dans \mathbb{N} , alors $k = k' = 1$, donc $x = y$

3 Majorants, borne sup, maximum

Remarque. Les minorants, borne inf, minimum sont obtenus en considérant les ordres opposés.

Soit E un ensemble ordonné total et E' un ensemble tel que $E' \subset E$

Définition (Majorant). Soit $M \in E$, M est un majorant de E' si, et seulement si, $\forall e \in E', e \leq M$.

On notera $\text{Maj}(E') = \{M \in E, \forall e \in E', e \leq M\}$ l'ensemble des majorants de E'

Définition (Maximum). Soit $M \in E$, M est le maximum de E' si, et seulement si, $M \in E'$ et $M \in \text{Maj}(E')$.

Définition (Élément maximal). Soit $N \in E'$, N est maximal dans E' si, et seulement si, $N \in E'$ et $\forall e \in E', (N \leq e \Rightarrow N = e)$.

Définition (Borne supérieure). Soit $B \in E$, B est la borne supérieure de E' si, et seulement si, B est le minimum de $\text{Maj}(E')$

4 Ensemble bien fondé

Définition (Ensemble bien fondé). (E, \geq) est un ensemble bien fondé s'il n'existe pas dans E une suite infinie $a_0 > a_1 > \dots > a_n > a_{n+1} > \dots$ strictement décroissante.

Exemple. L'ensemble \mathbb{N} est bien fondé.

L'ensemble \mathbb{Z} n'est pas bien fondé.

Proposition. L'ensemble E est bien fondé si, et seulement si, toute partie non vide de E a un élément minimal.

Corollaire. Soit (E, s) un ensemble bien fondé, \mathcal{P} une propriété des éléments de E . On a la proposition (I) vraie :

(I) : $\forall x \in E, [(\forall y < x, \mathcal{P}(y) \text{ vraie}) \Rightarrow \mathcal{P}(x) \text{ vraie}], \text{ alors } \forall x \in E, \mathcal{P}(x) \text{ vraie}.$

Démonstration par l'absurde. Soit $X = \{x \in E, \mathcal{P}(x) \text{ fausse}\}$. Supposons (I) vraie par \mathcal{P} et E . Montrons : $X = \emptyset$.

Supposons $X \neq \emptyset$.

Ainsi, par la proposition, il existe un élément minimal $x_0 \in X$ tel que :

- $x_0 \in X \Rightarrow \mathcal{P}(x_0) \text{ fausse}$
- x_0 minimal dans X , donc $\forall y < x_0, y \notin X \Rightarrow \mathcal{P}(y) \text{ vraie}.$

Mais par (I), $\mathcal{P}(x_0)$ est vraie ce qui entre en contradiction avec les hypothèses.

De ce fait, $X = \emptyset$

□

Chapitre 3

Terminaison

1 Exemple : la conjecture de Syracuse

La conjecture de Syracuse (ou encore conjecture Collatz) a été établie en 1928, par Lothar Collatz.

1.1 Version programme

Soit le programme suivant :

```
while (k > 1) {  
    if (k % 2)  
        k = 3 * k + 1; // Si k impair  
    else  
        k = k / 2; // Si k pair  
}
```

Exemple. On a :

$k = 4 \rightarrow 2 \rightarrow 1$

$k = 33 \rightarrow 100 \rightarrow 50 \rightarrow 25 \rightarrow 76 \rightarrow 38 \rightarrow 19 \rightarrow 58 \rightarrow 29 \rightarrow 88 \rightarrow 44 \rightarrow 22 \rightarrow 11$
 $\rightarrow 34 \rightarrow 17 \rightarrow 52 \rightarrow 26 \rightarrow 13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

La question est alors de savoir si le programme s'arrête pour tout $k \in \mathbb{N}$ à la valeur $k = 1$.

On a montré qu'il s'arrête pour tout $k < 2^{62}$

En fait, l'arrêt d'un programme est indécidable. Il n'existe pas d'algorithme qui, étant donné un programme, répond **true** si un programme s'arrête, ou **false** sinon.

Idée de la démonstration . Raison intuitive.

Supposons la fonction `termine(f)` qui retourne **true** si `f` se termine et **false** sinon.

On définit alors une fonction `g` par :

```
void g(void) {  
    while(termine(g)); // Programme récursif  
}
```

- Si la fonction `g` se termine, alors `termine(g)` renvoie **true** et on a une boucle infinie.
- Si la fonction `g` ne se termine pas, alors `termine(g)` renvoie **false** et `g` se termine.

On a donc montré par l'absurde qu'une telle fonction ne pouvait exister. □

1.2 Version suite

Soit $(u_n)_{n \in \mathbb{N}}$ la suite définie récursivement par :

$$\begin{aligned} u_0 &= k, k \in \mathbb{N}^* \\ u_{n+1} &= \begin{cases} u_n/2 & \text{si } u_n \text{ pair} \\ 3u_n + 1 & \text{si } u_n \text{ impair} \end{cases} \end{aligned}$$

La question est de savoir si il existe $n \in \mathbb{N}^*$ tel que $u_n = 1$ pour tout $k \in \mathbb{N}^*$.

Exemple. On a :

$$1 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

$$9 \rightarrow 28 \rightarrow 14 \rightarrow 7 \rightarrow 22 \rightarrow 11 \rightarrow 34 \rightarrow 17 \rightarrow 52 \rightarrow 26 \rightarrow 13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

1.3 Version fonction

Soit $f: \mathbb{N}^* \rightarrow \mathbb{N}$ une fonction définie par :

$$f(k) = \begin{cases} k/2 & \text{si } k \text{ pair} \\ 3k + 1 & \text{si } k \text{ impair} \\ 1 & \text{si } k = 1 \end{cases}$$

La question est de savoir s'il existe $n \in \mathbb{N}$ tel que $f^n(k) = \underbrace{f \circ f \circ \dots \circ f}_n(k) = 1$ pour tout $k \in \mathbb{N}^*$.

Il y a deux méthodes mathématiques pour montrer la terminaison d'une fonction :

- Les points fixes
- Les ordres bien fondés

2 Les points fixes

Définition (Point fixe). Soit $f: E \rightarrow E$ un endomorphisme.

L'élément $x \in E$ est un point fixe de f si, et seulement si, $f(x) = x$.

Méthode. Etude de $k, f(k), f^2(k), \dots, f^n(k)$

Un théorème de point fixe peut fournir une preuve de terminaison.

Théorème (Cas simple). Soit (E, \leq) un ensemble ordonné fini avec un minimum \perp .

On considère une application $f: E \rightarrow E$ monotone.

Alors il existe $p \leq \text{Card}(E)$ tel que $f^p(\perp)$ est le plus petit point fixe de f .

Démonstration. On considère la suite $(x_n)_{n \in \mathbb{N}}$ définie par $x_n = f^n(\perp)$

On a $\perp \leq f(\perp)$ car \perp est le plus petit élément de E .

En appliquant f , on a $f(\perp) \leq f^2(\perp)$ car f est monotone.

Et $x_n \leq x_{n+1}$, donc la suite $(x_n)_{n \in \mathbb{N}}$ est croissante.

Parmi les $\text{Card}(E) + 1$ premiers éléments de E , au moins deux éléments consécutifs sont égaux. Il existe alors $p \in \mathbb{N}$ tel que $f^p(\perp) = f^{p+1}(\perp) = f(f^p(\perp))$, donc $f^p(\perp)$ est un point fixe.

Montrons alors que $f^p(\perp)$ est le plus petit des points fixes de f .

Soit z un autre point fixe de f tel que $f(z) = z$.

On a $\perp \leq z$ car \perp est le plus petit élément de E

De plus, $f(\perp) \leq f(z) = z$ car f est monotone, et, par récurrence, $f^n(\perp) \leq z$ pour tout $n \in \mathbb{N}$.

Donc $f^n(\perp) \leq z$ et $f^p(\perp)$ est bien le plus petit point fixe de f . \square

3 Ordres bien fondés

Méthode. Associer une valeur au corps de la boucle telle que cette valeur décroît strictement tous les tours successifs de boucle.

Si les valeurs sont dans un ensemble ordonné (E, s) avec un ordre bien fondé, on obtient une terminaison.

Définition (Ensemble bien fondé). L'ensemble E est un ensemble avec un ordre bien fondé s'il n'existe pas de suite strictement décroissante (infinie).

Théorème. L'ensemble (E, \leq) est un ensemble avec un ordre bien fondé si, et seulement si, toute partie non vide a (au moins) un élément minimal.

Démonstration. On montre : il existe une suite strictement décroissante si, et seulement si, il existe une partie non vide sans élément minimal.

(\Rightarrow) Soit $(u_n)_{n \in \mathbb{N}}$ une suite strictement décroissante, alors $X = \{x_n, n \in \mathbb{N}\}$ n'a pas d'élément minimal.

(\Leftarrow) Soit $X \subset E$ une partie non vide de E sans élément minimal.

$\forall x \in X$, x est non minimal, donc il existe $y \in X$ tel que $y < x$.

Soit $f: X \rightarrow X$

$$x \mapsto f(x) = y < x$$

Soit $x_0 \in X$ (non vide).

On pose $x_n = f^n(x_0)$, $n \in \mathbb{N}$

Alors $x_{n+1} = f^{n+1}(x_0) = f(f^n(x_0)) = f(x_n) < x_n$ par définition de f

On a alors une suite strictement décroissante (infinie). \square

4 Exemples (et contre exemples)

4.1 Exemples types

L'ensemble \mathbb{N} est un ensemble bien fondé et admet 0 comme élément minimal.

L'ensemble \mathbb{Z} n'est pas un ensemble bien fondé. En effet, il existe une suite strictement décroissante $(x_n)_{n \in \mathbb{N}}$ définie par :

$$- x_0 = k, k \in \mathbb{Z}$$

$$- x_{n+1} = x_n - 1$$

L'ordre de divisibilité : $n \leq m$ si m divise n

La divisibilité est un ordre bien fondé sur \mathbb{N} et sur \mathbb{Z}

Exemple. $24 \geq 12 \geq 6 \geq 3 \geq 1$ où 1 est un point fixe de cet ordre.

$$\dots - 1 < 0 < 1 < 2 < \dots$$

$-n < -(n+1)$ et $-n < 0$ pour tout $n \in \mathbb{N}^*$ (à l'envers de l'ordre naturel)

4.2 Sur le monoïde libre A^* sur l'alphabet A

Soit $A = \{a, b\}$.

Ainsi, $A^* = \{\varepsilon, a, b, aa, ab, ba, bb, \dots\}$

Ordre lexicographique Par exemple avec $a < b$

Ce n'est pas un ordre bien fondé :

- $a \preceq a^2 \preceq a^3$,
- $(a^n b)_{n \in \mathbb{N}}$ est une suite strictement décroissante :
 $a^n b \preceq \dots \preceq a^2 b \preceq ab \preceq b$

Ordre préfixe $u \leq v$ si v commence par u (ie. $\exists w \in A^*, v = uw$)

C'est un ordre bien fondé à cause du mot vide ε .

4.3 Ordres produits (sur $\mathbb{N} \times \mathbb{N}$)

Il y a deux possibilités :

Produit Soit X tel que $(a, b) \leq (c, d)$ si $a \leq c$ et $b \leq d$

Alors c'est un ordre bien fondé.

Soient $X_1 = \{a / \exists b \text{ tq } (a, b) \in X\}$ d'élément minimal x_1 et $X_2 = \{b / (x, b) \in X\}$ d'élément minimal x_2

Ainsi, (x_1, x_2) est minimal pour X

Lexicographique $(a, b) < (c, d)$ si $a < c$ ou $(a = c \text{ et } b < d)$

Alors c'est un ordre bien fondé.

Chapitre 4

Outils pour la logique

1 Termes

Définition. Soit F un ensemble de symboles de fonctions.

Soit la fonction $a: F \rightarrow \mathbb{N}$ arité (ie. le nombre d'arguments de la fonction de F)

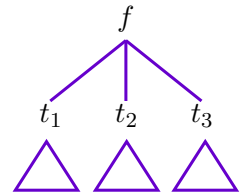
Soit A un alphabet défini par $A = F \cup \{(\cup\{\})\} \cup \{, \}$

L'ensemble des termes sur F , noté $T(F)$ est défini inductivement par :

- Base. Si $f \in F$, $a(f) = 0$, alors $f \in T(F)$
- Induction. Si $f \in F$, $a(f) = n$ et si $t_1, \dots, t_n \in T(F)$, alors $f(t_1, \dots, t_n) \in T(F)$

Remarque. Quelques remarques :

- Il est possible de représenter des termes par des arbres (voir ci-contre)
- On note aussi $ft_1t_2\dots t_n$ - histoire de ne pas utiliser $(,), ,$
- On note $F_n = \{f \in F / a(f) = n\}$ les fonctions à n arguments.
- Les termes définissent une *syntaxe*.
- On ajoute parfois des variables dans un ensemble X et on note $T(F, X)$ l'ensemble des termes correspondants.
- On ajoute dans la base (B) : « Si $x \in X$, alors $x \in T(F)$ » - Les variables sont d'arité 0



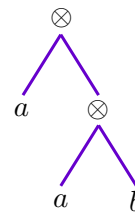
Exemple (Exercice 26 du poly de TD). Soient $F_0 = \{a\}$, $F_1 = \{s\}$ et on note $F = F_0 \cup F_1$.

$s(s(a))$ où $s^2(a) : s \rightarrow s \rightarrow a$. a est aussi un terme.

Dans ce cas, $T(F) = \{a, s(a), s^2(a), \dots\} = \{s^n(a), n \in \mathbb{N}\}$

Exemple. Soient $F_0 = \{a, b\}$ et $F_2 = \{\otimes\}$ et on note $F = F_0 \cup F_2$

L'ensemble des termes $T(F)$ sur F est représenté par l'arbre :



Notation : $\otimes(a, \otimes(a, b))$

Ou bien : $a \otimes (a \otimes b)$

Théorème (Interprétation des termes - sémantique). Soit F et $T(F)$ l'ensemble des termes sur F et soit V un ensemble (valeurs)

On se donne $h: F \rightarrow V$ et pour tout $f \in F$ d'arité n , $h_f: V^n \rightarrow V$

Alors il existe une unique application $h^*: T(F) \rightarrow V$ (valeur d'un terme) définie inductivement par :

- Base. Si $f \in F_0$, alors $h^*(t) = h(t)$
- Induction. Si $t = f(t_1, \dots, t_n)$, alors $h^*(t) = h_f(h^*(t_1), \dots, h^*(t_n))$

Exemple. Soit $F_0 = \{a\}$, $F_1 = \{s\}$ et $T(F) = \{s^n(a), n \in \mathbb{N}\}$

$$V = \mathbb{N} \quad h(a) = 0 \quad h_s: \mathbb{N} \rightarrow \mathbb{N} \\ n \mapsto n + 1$$

Montrer par récurrence sur n que $h^*(s^n(a)) = n$

- Base. Au rang $n = 1$, on a : $h^*(s(a)) = h_s(h^*(a)) = h(0) = 0 + 1 = 1$

- Induction. Soit $n \in \mathbb{N}$. Supposons la propriété vraie au rang n et montrons qu'elle est vraie au rang $n + 1$.

On a alors : $h^*(s^{n+1}(a)) = h^*(s(s^n(a))) = h_s(h^*(s^n(a))) = h_s(n)$ par hypothèse de récurrence.

Ainsi, on a $h^*(s^{n+1}(a)) = n + 1$

Exemple. Refaire l'exemple précédent avec $V = \mathbb{N}$, $h(a) = 1$ et $h_s(n) = n + 2$

Montrons par récurrence sur n que $h^*(s^n(a)) = 2n + 1$

- Base. Au rang $n = 1$, on a $h^*(s(a)) = h_s(h^*(a)) = h_s(1) = 1 + 2 = 3$
- Induction. Soit $n \in \mathbb{N}$. Supposons la propriété vraie au rang n et montrons qu'elle est vraie au rang $n + 1$.

On a alors : $h^*(s^{n+1}(a)) = h^*(s(s^n(a))) = h_s(h^*(s^n(a))) = h_s(2n + 1)$ par hypothèse de récurrence.

Ainsi, on a $h^*(s^{n+1}(a)) = 2n + 3$

Exemple. Soit $F_0 = \{a, b\}$, $F_1 = \{\otimes\}$

$$\begin{array}{llll} V = \mathbb{N} & h(a) = 1 & h_{\otimes}: \mathbb{N}^2 & \longrightarrow \mathbb{N} \\ & h(b) = 2 & (m, n) & \longmapsto m + n \end{array}$$

Calculer la valeur des termes $T(F)$ dans les deux cas.

$$t = (((a \otimes b) \otimes a) \otimes b) = (((1 + 2) + 1) + 2) = 1 + 2 + 1 + 2 = 6$$

On considère alors $t = a \otimes \dots \otimes a \otimes b \otimes \dots \otimes b$

On pose nb_a (resp. nb_b) le nombre de termes de a (resp. b).

Induction sur la structure. Montrons alors $h^*(t) = nb_a(t) + 2 \times nb_b(t)$

- Base. Pour $t = a$, on a $h^*(t) = h^*(a) = h_s(a) = h(a) = 1 = 1 \times nb_a(a)$
Pour $t = b$, on a $h^*(t) = h^*(b) = h_s(b) = h(b) = 2 = 2 \times nb_b(b)$
- Induction. Supposons que pour $h^*(t_1)$ soit vérifié.
Montrons alors que $t = t_1 \otimes a$, $h^*(t)$ est vérifié.

$$\begin{aligned} h^*(t) &= h_{\otimes}(h^*(t_1), h^*(a)) \text{ par définition de } h^* \\ &= h^*(t_1) + h(a) \text{ par définition de } h_{\otimes} \\ &= h^*(t_1) + 1 \\ &= (nb_a(t_1) + 1) + 2 \times nb_b(t_1) \\ h^*(t) &= nb_a(t) + 2 \times nb_b(t) \end{aligned}$$

Ainsi, $t = t_1 \otimes a$ est vérifié.

On montre de même que $t = t_1 \otimes b$, $t = a \otimes t_1$ et $t = b \otimes t_1$ sont vérifiés.

- Conclusion. Ainsi, pour tout t , $h^*(t) = nb_a(t) + 2 \times nb_b(t)$

2 Algèbre de Boole

Définition (Algèbre de Boole). Une algèbre de Boole est un tuple $\mathcal{B} = (\varepsilon, \perp, \top, \vee, \wedge, \neg)$

E est un ensemble,

\perp et \top sont deux éléments de E distincts,

\vee et \wedge sont deux opérateurs binaires, \neg est une opération unaire avec les propriétés suivantes :

- *Associativité* : $\forall a, b, c \in E$
 $(a \vee b) \vee c = a \vee (b \vee c)$ et $(a \wedge b) \wedge c = a \wedge (b \wedge c)$

- *Commutativité* : $\forall a, b \in E$
 $a \vee b = b \vee a$ et $a \wedge b = b \wedge a$
- *Distributivité* d'une loi par rapport à l'autre : $\forall a, b, c \in E$
 $(a \vee b) \wedge c = (a \wedge c) \vee (b \wedge c)$ et $(a \wedge b) \vee c = (a \vee c) \wedge (b \vee c)$
- *Absorption* : $\forall a, b \in E$
 $a \wedge (a \vee b) = a$ et $a \vee (a \wedge b) = a$
- *Idempotence* : $\forall a \in E$
 $a \vee a = a$ et $a \wedge a = a$
- *Bornes* : $\forall a \in E$
 $a \wedge \perp = \perp$, $a \vee \perp = a$ et $a \wedge \top = a$, $a \vee \top = \top$
- *Complémentarité* : $\forall a \in E$
 a possède un complémentaire noté \bar{a} vérifiant : $a \wedge \bar{a} = \perp$ et $a \vee \bar{a} = \top$

Exemple. Soit $E = \mathcal{P}(A)$ pour un ensemble A non vide. On définit alors une algèbre de Boole avec :

\perp	\top	\vee	\wedge	$-$
\emptyset	A	\cup	\cap	complémentaire

Soit $\mathcal{B} = \{0, 1\}$. On définit alors une algèbre de Boole avec :

\perp	\top	\vee	\wedge	$-$
0 (faux)	1 (vrai)	et (conjonction \vee)	ou (disjonction \wedge)	négation

3 Fonctions booléennes

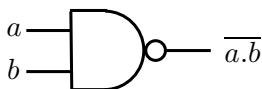
Définition (Fonction booléenne). Soit $n \in \mathbb{N}$. Une fonction booléenne d'arité n (à n arguments) est une application $f: \mathcal{B}^n \rightarrow \mathcal{B}$

Remarque. Si $n = 0$, il y a deux fonctions constantes : 0, 1

Si $n = 1$, il y a quatre fonctions : $x \rightarrow 0$, $x \rightarrow 1$, $x \rightarrow x$ et $x \rightarrow \bar{x}$

Il y a 2^{2^n} fonctions booléennes à n arguments.

Exemple. Table de vérité de la fonction NAND (x, y)



x	y	NAND(x, y)
0	0	1
0	1	1
1	0	1
1	1	0

Théorème. Toute fonction booléenne f^n s'écrit comme combinaison de ses arguments ($n \geq 1$) avec somme, produit, complémentaire.

Exemple. $\text{NAND}(x, y) = \bar{x} \cdot \bar{y} + \bar{x} \cdot y + x \cdot \bar{y}$

La démonstration est par récurrence sur n , à partir du lemme suivant :

Lemme. Soit f une fonction booléenne à n arguments.

Alors $f(x_1, \dots, x_n) = x_1 f(1, x_2, \dots, x_n) + \bar{x}_1 f(0, x_2, \dots, x_n)$

Démonstration. Posons $g(x_1, \dots, x_n) = x_1 f(1, x_2, \dots, x_n) + \bar{x}_1 f(0, x_2, \dots, x_n)$

- Si $x_1 = 1$, $g(x_1, \dots, x_n) = f(1, x_2, \dots, x_n) = f(x_1, \dots, x_n)$ par le lemme.
- Si $x_1 = 0$, $g(x_1, \dots, x_n) = f(0, x_2, \dots, x_n) = f(x_1, \dots, x_n)$ par le lemme.

□

4 Formes normales

Définition (Forme normale disjonctive). Somme de produits de x_i où \bar{x}_i

Soit f une fonction booléenne à n arguments, $\mathcal{D}_f = \{\vec{b} \in \mathcal{B}^n / f(\vec{b}) = 1\}$ où $\vec{b} = x_1, \dots, x_n \in \mathcal{B}^n$

Alors la forme normale disjonctive de f est :

$$f(x_1, \dots, x_n) = \sum_{\vec{b} \in \mathcal{D}_f} M_{\vec{b}}(x_1, \dots, x_n) \text{ où } M_{\vec{b}}(x_1, \dots, x_n) = x'_1 \cdots x'_n$$

$$x'_i = \begin{cases} x_i & \text{si } b_i = 1 \\ \bar{x}_i & \text{si } b_i = 0 \end{cases}$$

Exemple. Pour la fonction NAND, on a : $\text{NAND}(x, y) = \underbrace{\bar{x} \cdot \bar{y}}_{M_{(0,0)}(x,y)} + \underbrace{\bar{x} \cdot y}_{M_{(0,1)}(x,y)} + \underbrace{x \cdot \bar{y}}_{M_{(1,0)}(x,y)}$

Définition (Forme normale conjonctive). Produit de somme de x_i ou \bar{x}_i

Soit f une fonction booléenne à n arguments, $\mathcal{D}_f = \{\vec{b} \in \mathcal{B}^n / f(\vec{b}) = 1\}$

Alors la forme normale conjonctive de f est :

$$f(x_1, \dots, x_n) = \prod_{\vec{b} \in \mathcal{D}_f} S_{\vec{b}}(x_1, \dots, x_n) \text{ où } S_{\vec{b}}(x_1, \dots, x_n) = x'_1 + \cdots + x'_n$$

$$x'_i = \begin{cases} x_i & \text{si } b_i = 0 \\ \bar{x}_i & \text{si } b_i = 1 \end{cases}$$

Exemple. Pour la fonction NAND, on a : $\text{NAND}(x, y) = \underbrace{\bar{x} + \bar{y}}_{S_{(1,1)}(x,y)}$

5 Fonctions duales

Définition (Fonction duale). Soit f une fonction booléenne à n arguments.

Sa fonction duale, notée \tilde{f} est définie par :

$$\tilde{f}(x_1, \dots, x_n) = \overline{f(\bar{x}_1, \dots, \bar{x}_n)}$$

Exemple. Soit f une fonction définie par : $f(x, y) = x + y$

Sa fonction duale \tilde{f} est alors définie par : $\tilde{f}(x, y) = \overline{\bar{x} + \bar{y}} = \overline{\bar{x} \cdot \bar{y}} = x \cdot y$ d'après les lois de Morgan

Proposition. Soit f une fonction booléenne à n arguments. On a alors :

$$(i) \quad \tilde{\tilde{f}} = f$$

(ii) *Préservation par composition :*

$$\text{Si } g(x_1, \dots, x_n) = f(f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n))$$

$$\text{Alors } \tilde{g}(x_1, \dots, x_n) = \tilde{f}(\tilde{f}_1(x_1, \dots, x_n), \dots, \tilde{f}_n(x_1, \dots, x_n))$$

Exemple. On a :

$$\begin{aligned} \text{Si } g(x, y) &= g_1(x, y) + g_2(x, y) \\ &= f(g_1(x, y), g_2(x, y)) \end{aligned}$$

$$\begin{aligned} \text{Alors } \tilde{g}(x, y) &= \tilde{f}(\tilde{g}_1(x, y), \tilde{g}_2(x, y)) \\ &= f(\overline{\tilde{g}_1(x, y)}, \overline{\tilde{g}_2(x, y)}) \\ &= \overline{\tilde{g}_1(x, y)} + \overline{\tilde{g}_2(x, y)} \\ \tilde{g}(x, y) &= \tilde{g}_1(x, y) \cdot \tilde{g}_2(x, y) \text{ d'après les lois de Morgan} \end{aligned}$$

Chapitre 5

Calcul propositionnel

1 Syntaxe

Définition. Soit \mathcal{P} un ensemble de symboles propositionnels (ou variables) et l'alphabet $A = \mathcal{P} \cup \{ (,), \neg, \rightarrow \}$

Où $\neg x$ est la négation de x et \rightarrow représente l'implication.

Les formules de calcul propositionnel sont définies inductivement par :

- Base. Si $p \in \mathcal{P}$, alors p est une formule.
- Induction. Si F est une formule, alors $\neg F$ est une formule
- Si F_1 et F_2 sont deux formules, alors $(F_1 \rightarrow F_2)$ est une formule.

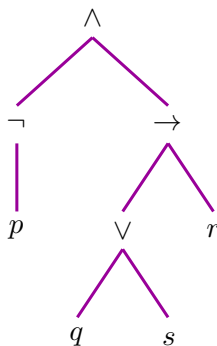
Remarque. Il est également possible d'utiliser le symbole \supset au lieu de \rightarrow pour l'implication.

Exemple. $F = (p \rightarrow (p \rightarrow q))$ est une formule sur $\{p, q\}$

Définition. On définit \wedge, \vee et \equiv des symboles propositionnels par :

- $F_1 \vee F_2 = (\neg F_1 \rightarrow F_2)$
- $F_1 \wedge F_2 = \neg (F_1 \rightarrow \neg F_2)$
- $F_1 \equiv F_2 = (F_1 \rightarrow F_2) \wedge (F_2 \rightarrow F_1)$

Exemple. $F = \neg p \wedge ((q \vee r) \rightarrow s)$ est une formule sur $\{p, q, r, s\} \subset \mathcal{P}$



Représentation de l'arbre de la formule $F = \neg p \wedge ((q \vee r) \rightarrow s)$

2 Sémantique - interprétation des formules

Exemple. On reprend le dernier exemple.

$F = \neg p \wedge ((q \vee r) \rightarrow s)$ est une formule sur $\{p, q, r, s\} \subset \mathcal{P}$

Si l'on considère $p = 0, q = 0, r = 1, s = 1$, alors F vaut 1.

Définition. Soit $I: \mathcal{P} \longrightarrow \mathcal{B} = \{0, 1\}$ une interprétation de symboles de \mathcal{P} .

On définit le prolongement de I aux formules de calcul propositionnel, noté I (et non I^*) inductivement par :

- Base. Si $F = p$, alors $I(F) = I(p)$
- Induction. Si $F = \neg G$, alors $I(F) = \overline{I(G)}$
Si $F = F_1 \rightarrow F_2$, alors $I(F) = \underbrace{\overline{I(F_1)} + I(F_2)}_{\text{ou}}$

Proposition. Soit I une interprétation. Alors :

- $I(F_1 \vee F_2) = I(F_1) + I(F_2)$
- $I(F_1 \wedge F_2) = I(F_1) . I(F_2)$
- $I(F_1 \equiv F_2) = I(F_1) . I(F_2) + \overline{I(F_1)} . \overline{I(F_2)}$

Démonstration. Montrons les trois points de la proposition précédente.

- $I(F_1 \vee F_2) = I(F_1) + I(F_2)$

$$\begin{aligned} I(F_1 \vee F_2) &= I(\neg F_1 \rightarrow F_2) \text{ par définition de } F_1 \vee F_2 \\ &= \overline{I(\neg F_1)} + I(F_2) \text{ par définition de } I(X \rightarrow Y) \\ &= \overline{\overline{I(F_1)}} + I(F_2) \text{ par définition de } I(\neg X) \\ I(F_1 \vee F_2) &= I(F_1) + I(F_2) \end{aligned}$$

- $I(F_1 \wedge F_2) = I(F_1) . I(F_2)$

$$\begin{aligned} I(F_1 \wedge F_2) &= I(\neg(F_1 \rightarrow \neg F_2)) \text{ par définition de } F_1 \wedge F_2 \\ &= \overline{I(F_1 \rightarrow \neg F_2)} \text{ par définition de } I(\neg X) \\ &= \overline{\overline{I(F_1)} + I(\neg F_2)} \text{ par définition de } I(X \rightarrow Y) \\ &= \overline{\overline{I(F_1)} + \overline{I(F_2)}} \text{ par définition de } I(\neg X) \\ &= \overline{I(F_1)} . \overline{I(F_2)} \text{ par les lois de Morgan} \\ I(F_1 \wedge F_2) &= I(F_1) . I(F_2) \end{aligned}$$

- $I(F_1 \equiv F_2) = I(F_1) . I(F_2) + \overline{I(F_1)} . \overline{I(F_2)}$

$$\begin{aligned} I(F_1 \equiv F_2) &= I((F_1 \rightarrow F_2) \wedge (F_2 \rightarrow F_1)) \text{ par définition de } F_1 \equiv F_2 \\ &= I(F_1 \rightarrow F_2) . I(F_2 \rightarrow F_1) \text{ par définition de } I(X \wedge Y) \\ &= (\overline{I(F_1)} + I(F_2)) . (\overline{I(F_2)} + I(F_1)) \text{ par définition de } I(X \rightarrow Y) \\ &= \overline{I(F_1)} . \overline{I(F_2)} + \overline{I(F_1)} . I(F_1) + I(F_2) . \overline{I(F_2)} + I(F_2) . I(F_1) \text{ en développant} \\ I(F_1 \equiv F_2) &= \overline{I(F_1)} . \overline{I(F_2)} + I(F_2) . I(F_1) \end{aligned}$$

□

Exemple. Soit $F = \neg p \wedge ((q \vee r) \rightarrow s)$

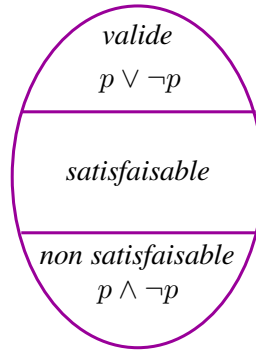
$$\begin{aligned}
 I(F) &= \overline{I(p)}.I((q \vee r) \rightarrow s) \\
 &= \overline{I(p)}.(\overline{I(q \vee r)} + I(s)) \\
 &= \overline{I(p)}.(\overline{I(q) + I(r)} + I(s)) \\
 I(F) &= \overline{I(p)}.(\overline{I(q)}. \overline{I(r)} + I(s)) \text{ d'après les lois de Morgan}
 \end{aligned}$$

Pour $I(p) = 0$, $I(r) = 0$, $I(q) = 0$ et $I(s) = 1$, alors $I(F) = 1$

Lorsqu'une interprétation I est telle que $I(F) = 1$, on note $I \models F$: « I satisfait F » ou « F est vraie pour I ».

Définition. Une formule F est :

- Valide (ou une tautologie) si pour toute interprétation I , $I(F) = 1$
- Satisfaisable s'il existe une interprétation I telle que $I(F) = 1$
- Non satisfaisable si pour toute interprétation I , $I(F) = 0$



Remarque. Une formule F est non satisfaisable si $\neg F$ est valide.

Définition. Deux formules F et G sont équivalentes, noté $F \sim G$, si pour toute interprétation I , $I(F) = I(G)$

Un ensemble de formules \mathcal{F} est satisfaisable s'il existe une interprétation I qui satisfait toutes les formules F de \mathcal{F} .

Exemple. Les formules p et $\neg(\neg p)$ sont équivalentes, et en général :

- $F \sim \neg(\neg F)$
- $F \vee G \sim G \vee F$ (car le $+$ est commutatif dans \mathcal{B})
- $\neg(F \vee G) \sim (\neg F \wedge \neg G)$

Remarque. On obtient associativité, idempotence, absorption, distributivité... mais avec \sim au lieu de $=$

Donc le quotient calcul propositionnel / \sim est une algèbre de Boole.

Exemple. Soit $\mathcal{F} = \{F_1, \dots, F_n\}$

\mathcal{F} est satisfaisable si $\bigwedge_{i=1}^n F_i$ est satisfaisable (ie. $I\left(\bigwedge_{i=1}^n F_i\right)$ doit être égal à 1).

$$\mathcal{F}_1 = \{p, p \rightarrow q\}$$

$$\text{Or } I(p \wedge (p \rightarrow q)) = I(p) \cdot (\overline{I(p)} + I(q)) = I(p) \cdot I(q)$$

Donc \mathcal{F} est satisfaisable si $I(p) = I(q) = 1$

$$\mathcal{F}_2 = \{p, p \rightarrow q, \neg q\} \text{ n'est pas satisfaisable}$$

En effet, si \mathcal{F}_2 avait été satisfaisable, on aurait eu $I(p) = I(q) = 1$ et $I(q) = 0$

Il y a alors contradiction.

Remarque. Soit $\mathcal{P} = \{p_1, \dots, p_n\}$.

Une interprétation $I: \mathcal{P} \rightarrow \mathcal{B}$ est décrite par : $(I(p_1), \dots, I(p_n)) \in \mathcal{B}$

$$\text{Soit } F \text{ une formule sur } \mathcal{P}. \text{ On lui associe : } \begin{array}{ccc} g_f: & \mathcal{B}^n & \longrightarrow \mathcal{B} \\ & I & \longrightarrow I(F) \end{array}$$

Proposition. *Pour tout fonction booléenne d'arité 1 (de un argument), il existe une formule (unique à équivalence près) qui lui est associée.*

$$\text{Exemple. Soit } f: \begin{array}{ccc} \mathcal{B}^2 & \longrightarrow & \mathcal{B} \\ (x, y) & \longrightarrow & x + y \end{array}$$

La formule de f qui lui est associée est $p_1 \vee p_2$ sur $\{p_1, p_2\}$

Cette correspondance permet d'associer à une formule une forme normale conjonctive (FNC) et une forme normale disjonctive (FND) équivalente.

Définition. Une clause est une formule de la forme $c_1 \vee c_2 \vee \dots \vee c_n$ où soit $c_i = p \in \mathcal{P}$, soit $c = \neg p \in \mathcal{P}$.

Proposition. *Toute formule est équivalente à une conjonction de clause (FNC).*

3 Conséquence sémantique

Définition. On dit que G est conséquence de F , noté $F \models G$ si pour toute interprétation I , si $I(F) = 1$, alors $I(G) = 1$

Proposition. *F satisfait G si, et seulement si, $(F \rightarrow G)$ est valide.*

Démonstration. Si F satisfait G , alors G est conséquence logique de F et $F \models G$, ainsi, pour toute interprétation I , si $I(F) = 1$, alors $I(G) = 1$ par définition. Ainsi, pour toute interprétation I , on a $I(F \rightarrow G) = \overline{I(F)} + I(G) = 1$, donc $(F \rightarrow G)$ est valide.

Si $(F \rightarrow G)$ est valide, alors pour toute interprétation I , $I(F \rightarrow G) = 1$

On a alors $I(F \rightarrow G) = \overline{I(F)} + I(G) = 1$.

Ainsi, si $I(F) = 1$, alors $I(G) = 1$, donc G est conséquence logique de F et F satisfait G .

Conclusion. F satisfait G si, et seulement si, $(F \rightarrow G)$ est valide. □

Proposition. *Montrer que $F \sim G$ si, et seulement si, $F \equiv G$ est valide.*

Démonstration. Si $F \sim G$, alors pour toute interprétation I , $I(F) = I(G)$

Or par définition, $I(F \equiv G) = I(F) \cdot I(G) + \overline{I(F)} \cdot \overline{I(G)} = I(F) + \overline{I(F)}$

Ainsi, pour toute interprétation I , $I(F \equiv G) = 1$ et $F \equiv G$ est valide.

Si $F \equiv G$ est valide, alors $I(F \equiv G) = I(F) \cdot I(G) + \overline{I(F)} \cdot \overline{I(G)} = 1$

Supposons $I(F) \neq I(G)$. On a alors :

- Si $I(F) = 1$, alors $I(G) = 0$ et $I(F \equiv G) = 0$

- Si $I(F) = 0$, alors $I(G) = 1$ et $I(F \equiv G) = 0$

Ainsi, $I(F) = I(G)$ pour toute interprétation I .

Alors $F \sim G$. □

Définition. Soit $\mathcal{F} = \{F_1, \dots, F_n\}$ un ensemble fini de formules et \mathcal{G} une formule.

On note $\mathcal{F} \models \mathcal{G}$ si $\bigwedge_{i=1}^n F_i \models \mathcal{G}$ (ie. pour toute interprétation I , si pour tout $F \in \mathcal{F}$, $I(F) = 1$, alors $I(\mathcal{G}) = 1$).

Définition. Un séquent est une paire $(\mathcal{F}, \mathcal{G})$ où \mathcal{F} est un ensemble de formules et \mathcal{G} une formule. Le séquent $(\mathcal{F}, \mathcal{G})$ est valide si $\mathcal{F} \models \mathcal{G}$.

Exemple. Soient $\mathcal{F} = \{p, p \rightarrow q\}$ et $\mathcal{G} = q$

On vérifie que $(\mathcal{F}, \mathcal{G})$ est un séquent valide.

Montrons alors que pour toute interprétation I , si pour toute $F \in \mathcal{F}$, $I(F) = 1$, alors $I(\mathcal{G}) = 1$

Ceci revient à montrer $(p) \wedge (p \rightarrow q) \models q$

Supposons $I(p) = I(p \rightarrow q) = 1$

$I(p \rightarrow q) = \overline{I(p)} + I(q) = 1$

Or $I(p) = 1$, donc $\overline{I(p)} = 0$ et $I(q) = 1$

Ainsi, $(\mathcal{F}, \mathcal{G})$ est un séquent valide.

Remarque. Si l'on remplace dans \mathcal{F} un des F_i par F'_i tel que $F_i \sim F'_i$, alors $(\mathcal{F}, \mathcal{G})$ est valide si, et seulement si, $(\mathcal{F}', \mathcal{G})$ est valide.

Proposition. On a les propositions suivantes :

(i) $\mathcal{F} \models G$ si, et seulement si, $\mathcal{F} \cup \{\neg G\}$ est non satisfaisable.

(ii) $\mathcal{F} \cup \{F\} \models G$ si, et seulement si, $\mathcal{F} \models (F \rightarrow G)$

Démonstration. On démontre les contraposées.

(i) Montrons les deux sens de l'équivalence.

- Supposons $\mathcal{F} \cup \{\neg G\}$ satisfaisable

Alors il existe une interprétation I telle que $I(F_i) = 1$ pour toute formule F_i de \mathcal{F} .

Et $I(\neg F) = 1$, donc $I(G) = 0$

Donc I ne satisfait pas $\mathcal{F} \models G$.

- Supposons qu'il existe une interprétation I ne satisfaisant pas $\mathcal{F} \models G$.

Donc $I(F_i) = 1$ pour toute F_i de \mathcal{F} et $I(G) = 0$.

Alors I satisfait $\mathcal{F} \cup \{\neg G\}$.

(ii) Montrons cette proposition à l'aide d'équivalences.

$$\begin{aligned} \mathcal{F} \cup \{F\} \models G & \text{ ssi } \mathcal{F} \cup \{F, \neg G\} \text{ non satisfaisable (1)} \\ & \text{ ssi } \mathcal{F} \cup \{\neg(F \rightarrow G)\} \text{ non satisfaisable} \\ \mathcal{F} \cup \{F\} \models G & \text{ ssi } \mathcal{F} \models (F \rightarrow G) \text{ par (1)} \end{aligned}$$

□

4 Conséquence logique (ou déduction)

Définition. Un séquent (\mathcal{F}, G) est dit prouvable, noté $\mathcal{F} \vdash G$, s'il est obtenu après un nombre fini d'applications des six règles suivantes :

- a) Utilisation d'une hypothèse :
Si $F \in G$, alors $\mathcal{F} \vdash F$
- b) Argumentation d'hypothèse :
Si $G \notin \mathcal{F}$ et $\mathcal{F} \vdash F$, alors $\mathcal{F} \cup \{F\} \vdash F$
- c) Modus ponens :
Si $\mathcal{F} \vdash (F \rightarrow G)$ et $\mathcal{F} \vdash F$, alors $\mathcal{F} \vdash G$
- d) Retrait d'hypothèse (synthèse) :
Si $\mathcal{F} \cup \{F\} \vdash G$, alors $\mathcal{F} \vdash (F \rightarrow G)$
- e) Double négation :
 $\mathcal{F} \vdash F$ si, et seulement si, $\mathcal{F} \vdash \neg\neg F$
- f) Absurde :
Si $\mathcal{F} \cup \{F\} \vdash G$ et $\mathcal{F} \cup \{F\} \vdash \neg G$, alors $\mathcal{F} \vdash \neg F$

Exemple. Preuve de la démonstration par contraposée :

On veut prouver : $p \rightarrow q \vdash (\neg q \rightarrow \neg p)$

- (i) $\{p \rightarrow q, \neg q, p\} \vdash p$ d'après a)
- (ii) $\{p \rightarrow q, \neg q, p\} \vdash \neg q$ d'après a)
- (iii) $\{p \rightarrow q, \neg q, p\} \vdash p \rightarrow q$ d'après a)
- (iv) $\{p \rightarrow q, \neg q, p\} \vdash q$ d'après c) appliquée à 1 et 3
- (v) $\{p \rightarrow q, \neg q\} \vdash \neg p$ d'après f) sur 2 et 4
- (vi) $\{p \rightarrow q\} \vdash (\neg q \rightarrow \neg p)$ d'après d)

Théorème. Un séquent (\mathcal{F}, G) est valide si, et seulement si, il est prouvable.

Remarque. Signification des deux sens de l'équivalence :

- Sens \Rightarrow : Complétude - *Ce qui est vrai peut être prouvé.*
- Sens \Leftarrow : Correction/adéquation : *Ce qui peut être prouvé est vrai.*

Principe de la démonstration. Par induction sur la longueur de la preuve.

A partir d'un séquent valide, en appliquant une des six règles a), ..., f) on obtient un nouveau séquent valide.

Par exemple avec a) :

On suppose que $\mathcal{F} \models G$ est obtenu par la règle a), donc on a $G \in \mathcal{F}$.

Si I est une interprétation telle que $I(F) = 1$ pour toute formule F de \mathcal{F} , alors $I(G) = 1$ puisque $G \in \mathcal{F}$, donc $\mathcal{F} \models G$. \square

Chapitre 6

Logique du premier ordre (ou calcul des prédicats)

1 Introduction

Rappel. Le calcul propositionnel est une logique à l'ordre 0.

Les objets d'ordre 1 sont les variables et les constantes.

Les objets d'ordre 2 sont les relations et les fonctions.

Exemple. Soient :

- H : Homme - $H(x)$: x est un homme
- G : Grecs - $M(x)$: x est mortel
- M : Mortels - $G(x)$: x est grec

Alors

$$\forall x (H(x) \rightarrow M(x)) \wedge (\exists x G(x) \wedge H(x)) \rightarrow (\exists x G(x) \wedge M(x))$$

est une formule logique du premier ordre.

2 Syntaxe

Variables + fonctions \rightarrow termes

Termes + relation \rightarrow formules

Soit \mathcal{G} un ensemble de symboles de fonctions.

Soit \mathcal{R} un ensemble de symboles de relations.

On définit $a: \mathcal{R} \cup \mathcal{G} \rightarrow \mathbb{N}$ l'arité (ie. nombre d'arguments)

$C = \mathcal{G}_0 = \{f \in \mathcal{G}, a(f) = 0\}$ symboles de constantes.

$P = \mathcal{R}_0 = \{R \in \mathcal{R}, a(R) = 0\}$ symboles de propositions.

Soit X un ensemble de variables.

Définition. Les termes de $T(\mathcal{G}, X)$ sont définis inductivement par :

- Base. Les constantes de C et les variables X sont des termes.
- Induction. Si $f \in \mathcal{G}$ d'arité n et t_1, \dots, t_n des termes, alors $f(t_1, \dots, t_n)$ est un terme.

Remarque. Un terme sans variable est un terme clos.

Définition. Les formules du premier ordre sur \mathcal{G} et \mathcal{R} sont définies inductivement par :

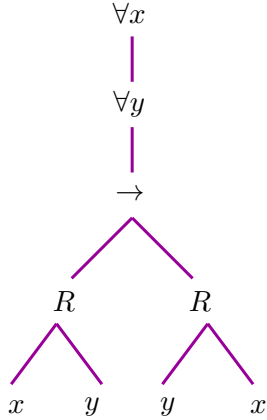
- Base. Si R est un symbole de relation d'arité n et t_1, \dots, t_n sont des termes, alors $R(t_1, \dots, t_n)$ est une formule dite atomique.
- Induction. Si F et G sont deux formules, alors $\neg F$, $F \vee G$, $F \wedge G$, $F \rightarrow G$ sont des formules.
Si F est une formule, alors $\forall x F$ et $\exists x F$ sont des formules et x est une variable.

Exemple. Soit $F = \forall x \forall y (R(x, y) \rightarrow R(y, x))$

L'arbre représentant F est :

Remarque. Attention à l'ordre des quantificateurs.

« Tous les enfants ont une mère »



Soit Mère(x, y) où x est la mère de y

$\exists x \forall y \text{Mère}(x, y) \neq \forall y \exists x \text{Mère}(x, y)$

Termes sur \mathcal{G} et $X \longrightarrow T(\mathcal{G}, X)$

Formules sur \mathcal{R} : $R(t_1, \dots, t_n)$ - formule atomique R d'arité n avec t_1, \dots, t_n termes

$\wedge, \vee, \rightarrow, \neg$

$\forall x F$ et $\exists x F$ (ie. il existe x tel que f), $x \in X$

3 Variables libres et variables liées

Définition (Variables d'un terme et d'une formule). (i) Termes :

- $\text{Var}(x) = \{x\}$, $\text{Var}(c) = \emptyset$ si $c \in C$ constante
- $\text{Var}(f(t_1, \dots, t_n)) = \bigcup_{i=1}^n \text{Var}(t_i)$ pour un terme $f(t_1, \dots, t_n)$

(ii) Formules :

- $\text{Var}(R(t_1, \dots, t_n)) = \bigcup_{i=1}^n \text{Var}(t_i)$
- $\text{Var}(\neg F) = \text{Var}(F)$, $\text{Var}(F \star G) = \text{Var}(F) \cup \text{Var}(G)$ pour $\star \in \{\wedge, \vee, \rightarrow\}$
- $\text{Var}(\exists x F) = \text{Var}(\forall x F) = \text{Var}(F) \cup \{x\}$

Exemple. Soit un terme $t = h(f(x), g(x, y))$

Ainsi, $\text{Var}(t) = \{x, y\}$

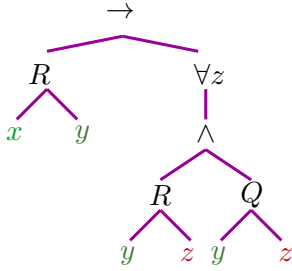
Soit la formule $F: \forall y(R(y, c) \rightarrow \exists(R(t, c) \wedge R(c, t)))$, où $c \in C$

Ainsi, $\text{Var}(F) = \{x, y\}$

Remarque. Dans l'arbre associé à une formule, les variables sont toujours en position de feuilles (elles n'ont pas de descendance).

Définition (Variables libres). Dans l'arbre d'une formule F , une feuille d'étiquette $x \in X$ est une occurrence *libre* de x s'il n'y a aucun quantificateur : $\forall x$ ou $\exists x$ dans les ascendants de cette feuille. Sinon, l'occurrence est dite *liée*.

Exemple. Soit la formule $F: R(x, y) \rightarrow \forall z (R(y, z) \wedge Q(y, z))$



Occurrences libres (en vert) - (ie. celles qui n'ont pas de quantificateur)

Occurrences liées (en rouge) - la variable z car $\forall z$

Définition (Variable libre et liée). Une variable est *libre* dans une formule F si elle a au moins une occurrence libre dans cette formule.

Une variable est *liée* dans une formule si elle n'est pas libre dans cette formule.

On note $L(F)$ l'ensemble des variables libres dans F et $B(F) = \text{Var}(F) \times L(F)$ les variables liées dans F (B pour *bonding* en anglais).

Exemple. On considère l'exemple précédent.

On a alors : $L(F) = \{x, y, z\}$ et $B(F) = \emptyset$

Définition (Formule close). Une formule est dite *close* si elle n'a aucune variable libre.

Proposition. Les variables libres d'une formule sont définies inductivement par :

- Base. $L(R(t_1, \dots, t_n)) = \bigcup_{i=1}^n \text{Var}(t_i)$
- Induction. Soit F et G deux formules. On a alors :
 - $L(\neg F) = L(F)$, $L(F \star G) = L(F) \cup L(G)$ pour $\star \in \{\wedge, \vee, \rightarrow\}$
 - $L(\forall x F) = L(\exists x F) = L(F) \setminus \{x\}$

Proposition. Les variables liées d'une formule sont définies inductivement par :

- Base. $B(R(t_1, \dots, t_n)) = \emptyset$
- Induction. Soient F et G deux formules. On a alors :
 - $B(\neg F) = B(F)$, $B(F \star G) = B(F) \cup B(G)$ pour $\star \in \{\wedge, \vee, \rightarrow\}$
 - $B(\forall x F) = B(\exists x F) = B(F) \cup \{x\}$

4 Sémantique

On va considérer une structure \mathcal{M} , donnée par :

- Un domaine \mathcal{D}
- Pour toute fonction f de \mathcal{G} d'arité n , une fonction $f_{\mathcal{D}}: \mathcal{D}^n \rightarrow \mathcal{D}$
- Pour toute relation R de \mathcal{R} d'arité n , une relation $R_{\mathcal{D}} \subset \mathcal{D}^n$

En particulier, les fonctions d'arité 0 (les constantes) ont des éléments $a_{\mathcal{D}}$ de \mathcal{D} et les relations d'arité 0 (les propositions) sont des éléments de $\{0, 1\} = \mathcal{B}$

On note $\mathcal{M} = (\mathcal{D}, (f_{\mathcal{D}})_{f \in \mathcal{G}}, (R_{\mathcal{D}})_{R \in \mathcal{R}})$

Exemple. $F: \exists x R(((x \odot x) \oplus x) \oplus c_1, c_0)$

Domaine $\mathcal{D} = \mathbb{R}$ ensemble des nombres réels, $c_0 = 0$, $c_1 = 1$

R s'interprète comme l'égalité, \odot comme la multiplication et \oplus comme l'addition.

$\exists x, x^2 + x + 1 = 0$

Exemple. \mathcal{M} est une base de données, les formules sont des requêtes.

Définition. Pour une structure \mathcal{M} , une valuation est une application $v: X \longrightarrow \mathcal{D}$

Proposition. Etant donné une structure \mathcal{M} associée à $\mathcal{G} \cup \mathcal{R}$ et une valuation $v: X \longrightarrow \mathcal{D}$, on définit la valeur d'un terme $v^*(t) \in \mathcal{D}$ inductivement par :

- Base. $v^*(a) = a_0$ pour une constante a
 $v^*(x) = v(x)$ pour $x \in X$ (ie. x une variable)
- Induction. Si $t = f(t_1, \dots, t_n)$ avec f d'arité n et t_1, \dots, t_n des termes, alors $v^*(t) = f_{\mathcal{D}}(v^*(t_1), \dots, v^*(t_n))$

Remarque. On a :
$$\begin{array}{ccc} v^*: T(\mathcal{G}, X) & \longrightarrow & \mathcal{D} \\ t & \longmapsto & v^*(t) \end{array}$$

Proposition. Soit $v: X \longrightarrow \mathcal{D}$ une valuation d'une structure \mathcal{M} , $x \in X$ une variable et $a_{\mathcal{D}} \in \mathcal{D}$

On définit la valuation $v[x \mapsto a_{\mathcal{D}}]: X \longrightarrow \mathcal{D}$ par $(v[x \mapsto a_{\mathcal{D}}])(g) = \begin{cases} v(g) & \text{si } y \neq x \\ a_{\mathcal{D}} & \text{si } y = x \end{cases}$

Définition. On définit la valuation (ou valeur de vérité) d'une formule F , notée $\hat{v}(F)$, inductivement par :

- Base. Si $F = R(t_1, \dots, t_n)$, alors $\hat{v}(F) = 1$ si $(v^*(t_1), \dots, v^*(t_n)) \in R_{\mathcal{D}}$
- Induction. On a :
 - Si $F = \neg G$, alors $\hat{v}(F) = \overline{\hat{v}(G)}$
 - Si $F = G_1 \wedge G_2$, alors $\hat{v}(F) = \hat{v}(G_1) \hat{v}(G_2)$
 - Si $F = G_1 \vee G_2$, alors $\hat{v}(F) = \hat{v}(G_1) + \hat{v}(G_2)$
 - Si $F = G_1 \rightarrow G_2$, alors $\hat{v}(F) = \overline{\hat{v}(G_1)} + \hat{v}(G_2)$
 - Si $F = \forall x G$, alors $\hat{v}(F) = 1$ si, et seulement si, pour toute $a_{\mathcal{D}} \in \mathcal{D}$, $v[\widehat{x \mapsto a_{\mathcal{D}}}] (G) = 1$
 - Si $F = \exists x G$, alors $\hat{v}(F) = 1$ si, et seulement si, il existe $a_{\mathcal{D}} \in \mathcal{D}$ tel que $v[\widehat{x \mapsto a_{\mathcal{D}}}] (G) = 1$

Exemple. Soit $F = \exists x R(((x \odot x) \oplus x) \oplus c_1, c_0)$

$v(x) = 2$

$\hat{v}(F) = 0$

Cherchons la valeur de $v[\widehat{x \mapsto a}] (G)$

$G: R(((x \odot x) \oplus x) \oplus c_1, c_0)$

Or $R(a^2 + a + 1, 0)$, et $a^2 + a + 1 = 0$

Donc $v[\widehat{x \mapsto a}] (G) = 0$ et $\hat{v}(F) = 0$

Exemple. Soit $F = R(f(x), g(y))$ et $\mathcal{M} = (\mathbb{N}, \leq, f_{\mathbb{N}}, g_{\mathbb{N}})$

On a : $f_{\mathbb{N}}(n) = n + 1$ $g_{\mathbb{N}}(n) = n + 3$

Soit v une valuation définie par :
$$v \begin{cases} x \mapsto 4 \\ y \mapsto 3 \end{cases}$$

- $v^*(f(x)) = f_{\mathbb{N}}(v(x)) = f_{\mathbb{N}}(4) = 5$
- $v^*(g(y)) = g_{\mathbb{N}}(v(y)) = G$

Ainsi, $\hat{v}(F) = 1$ si, et seulement si, $5 \leq 6$

Définition. On a :

- (i) Etant données une formule F et une structure \mathcal{M} ,
 - a) F est satisfaisable pour \mathcal{M} , s'il existe une valuation v telle que $\hat{v}(F) = 1$
Problème des bases de données.
 - b) F est valide pour \mathcal{M} , si pour toute valuation v , $\hat{v}(F) = 1$
On dit que \mathcal{M} est un modèle de F (noté $\mathcal{M} \models F$)
- (ii) Etant donnée une formule F :
 - a) F est satisfaisable s'il existe une structure \mathcal{M} telle que F est satisfaisable pour \mathcal{M}
 - b) F est valide (ou universellement valide), si pour toute structure \mathcal{M} , F est valide pour \mathcal{M} .

Remarque. Le problème 2.a est indécidable. Le problème 1.a est décidable pour des structures finies.

Rappel. Soit X une variable

La valuation est une application de X dans \mathcal{D} (domaine où les variables ont leurs valeurs).

Exemple. $X = \{x, y, z\}$ et $\mathcal{D} = \mathbb{N}$

Soient v_1, v'_1 et v_2, v'_2 quatre valuations définies par :

$$\begin{array}{l}
 - v_1 : \left| \begin{array}{l} x \rightarrow 2 \\ y \rightarrow 8 \\ z \rightarrow 7 \\ w \rightarrow 14 \end{array} \right| \quad v'_1 = v_1[z \rightarrow 3] \left| \begin{array}{l} x \rightarrow 2 \\ y \rightarrow 8 \\ z \rightarrow 3 \\ w \rightarrow 14 \end{array} \right| \quad v'_1[w \rightarrow 1] \\
 - v_2 : \left| \begin{array}{l} x \rightarrow 3 \\ y \rightarrow 5 \\ z \rightarrow 3 \\ w \rightarrow 14 \end{array} \right| \quad v'_2 = v_2[x \rightarrow 3] \left| \begin{array}{l} x \rightarrow 8 \\ y \rightarrow 5 \\ z \rightarrow 3 \\ w \rightarrow 14 \end{array} \right|
 \end{array}$$

Proposition. Soient \mathcal{M} une structure finie, et F une formule.

Il existe un algorithme qui décide s'il existe une valuation v telle que $\hat{v}(F) = 1$ (c'est la satisfaction d'une requête sur une base de données).

Cas de structures infinies :

$$- \mathcal{M}_1 = (\underbrace{\mathbb{N}}_{\mathcal{D}}, \underbrace{+, \times, \exp}_{\mathcal{G}}, \underbrace{=}_{\mathcal{R}})$$

La satisfaisabilité (d'une formule du premier ordre) est indécidable.

Dommage... : $\exists n \exists x \exists y \exists z (x^n + y^n = z^n)$ pour $n \geq 3$

$$- \mathcal{M}_2 = (\mathbb{N}, +, \times, =)$$

Indécidable également.

Dommage... : $\exists x P(x) = 0$ où P est un polynôme

$$- \mathcal{M}_3 = (\mathbb{R}, +, \times, <)$$

Décidable !

5 Propriétés sémantiques

Définition. Soit \mathcal{M} une structure.

- Soient F et G deux formules.
 $F \models G$ si : si $\hat{v}(F) = 1$, alors $\hat{v}(G) = 1$ pour toute valuation v de \mathcal{M}
- Soit $\mathcal{F} = \{F_1, \dots, F_n\}$ un ensemble fini de formules et G une formule.
 $\mathcal{F} \models G$ si $\bigwedge_{i=1}^n F_i = F_1 \wedge \dots \wedge F_n \models G$
 (ie. pour toute valuation v , si $\hat{v}(F_i) = 1$, pour toute $F_i \in \mathcal{F}$, alors $\hat{v}(G) = 1$)
- Un séquent est une paire (\mathcal{F}, G) où \mathcal{F} est un ensemble fini de formules et G une formule.
 Le séquent (\mathcal{F}, G) est valide dans \mathcal{M} si $\mathcal{F} \models G$ dans \mathcal{M} .
 Il est universellement valide s'il est valide dans toute structure.
- Deux formules F et G sont équivalentes, noté $F \sim G$ si pour toute structure \mathcal{M} et pour toute valuation v , $\hat{v}(F) = \hat{v}(G)$

Remarque. Quelques remarques :

- $F \sim G$ si pour toute structure \mathcal{M} , $F \models G$ et $G \models F$.
- On a toutes les équivalences du calcul propositionnel : $\neg(F \vee G) \sim \neg F \wedge \neg G$, $\neg\neg F \sim F$, ...
- On voudrait des équivalences plus « vides », qui impliquent les variables.

Proposition. Soit F une formule.

On a l'équivalence suivante

$$\neg\forall x F \sim \exists x \neg F$$

Démonstration ()*. Soit \mathcal{M} une structure et v une valuation.

On a :

$$\begin{aligned} \hat{v}(\neg\forall x F) = 1 & \text{ ssi } \hat{v}(\forall x F) = 0 \\ & \text{ssi il existe } a \in \mathcal{D} \text{ tq } v[\widehat{x \rightarrow a}](F) = 0 \\ & \text{ssi il existe } a \in \mathcal{D} \text{ tq } v[\widehat{x \rightarrow a}](\neg F) = 1 \\ & \text{ssi } \hat{v}(\exists x \neg F) = 1 \end{aligned}$$

□

Lemme. Soit \mathcal{M} une structure, v_1 et v_2 deux valuations.

- (i) Si t est un terme tel que $v_1|_{\text{Var}(t)} = v_2|_{\text{Var}(t)}$ (v_1 et v_2 coïncident sur $\text{Var}(t)$)
 Alors $v_1^*(t) = v_2^*(t)$
- (ii) Si F est une formule telle que $v_1|_{L(t)} = v_2|_{L(t)}$ (v_1 et v_2 coïncident sur les variables libres de F).
 Alors $\hat{v}_1(F) = \hat{v}_2(F)$

Remarque. La valeur de vérité d'une formule ne dépend que des valeurs de ses variables libres (e.g. $F: \forall x R(x, y)$ où y est une variable libre).

Exemple. Soit $t = g(x, y)$ et $v_1(x) = v_2(x)$
 $v_1(y) = v_2(y)$

$$v_1^*(t) = g_{\mathcal{D}}(v_1(x), v_1(y)) = g_{\mathcal{D}}(v_2(x), v_2(y)) = v_2^*(t)$$

Corollaire (Conséquences). Si F est une formule close (ie. $L(F) \neq \emptyset$), alors $\hat{v}(F)$ est constante, indépendante de v .

F et $\forall xF$ sont équivalentes si x n'est pas libre dans F .

De même, $F \sim \exists xF$ si $x \notin L(F)$.

Exemple. Soit $F: \forall x \exists y M(x, y)$

$\mathcal{D} = \{\text{tout le monde}\}$

$M(x, y) : y$ est mère de x

Pour tout v , $\hat{v}(F) = 1$

Rappel (Définition inductive de $L(F)$). On a :

$L(\forall xF) = L(F) \setminus \{x\} = L(F)$ si x n'est pas libre dans F .

Remarque. Soit v une valuation.

$\hat{v}(\forall xF) = 1$ si pour tout $a \in \mathcal{D}$, $v[\widehat{x \rightarrow a}](F) = 1$

Mais v et $v[x \rightarrow a]$ coïncident sauf sur x , donc elles coïncident sur $L(F)$.

D'après le lemme : $\hat{v}(F) = v[\widehat{x \rightarrow a}](F) = 1$

Donc $\hat{v}(F) = \hat{v}(\forall xF)$

La formule $\forall xF$ est une *clôture universelle* de F .

Proposition. Si $\mathcal{F} \models G$ dans \mathcal{M} alors $\mathcal{F} \models \forall xG$ dans \mathcal{M} si x n'est libre dans aucune formule de \mathcal{F}

Proposition (Substitution). Soit F une formule, t un terme et x une variable.

On veut remplacer toutes les occurrences libres de x par t dans $F \rightarrow F[x \mapsto t]$ est la formule obtenue.

(i) Cette opération est possible si aucune variable de t ne se trouve liée dans $F(x \mapsto t)$.

(ii) On a alors pour toute structure \mathcal{M} et toute valuation v , $\hat{v}(F[x \mapsto t]) = v[\widehat{x \mapsto v^*(t)}](F)$

Exemple. Soit $F: \exists y (g(y, y) = x)$ interprétée dans \mathbb{N} avec $g \rightarrow$ addition

F dit que x est pair.

On change la signification de la formule si on remplace x par $y : \exists y (y + y = y) \rightarrow y = 0$ marche dans \mathbb{N} .

6 Dédution

Définition. Un séquent (\mathcal{F}, G) est dit prouvable s'il est obtenu par application des règles a), b), c), d), e) et f) du calcul propositionnel auxquelles on ajoute :

g) Instantiation : si $\mathcal{F} \vdash \forall xF$, alors $\mathcal{F} \vdash F[x \mapsto t]$ (pour une substitution autorisée)

h) Clôture universelle : si $\mathcal{F} \vdash F$ et si x n'est pas libre dans \mathcal{F} , alors $\mathcal{F} \vdash \forall xF$

i) $\mathcal{F} \vdash \exists xF$ si, et seulement si, $\mathcal{F} \vdash \neg \forall x \neg F$

Théorème. Un séquent est valide (universellement) si, et seulement si, il est prouvable.

Exemple. On a :

(i) Prouver d'abord : $\mathcal{F}, F \vdash G$ si, et seulement si, $\mathcal{F}, F \vdash \neg \neg F$

(ii) Prouver ensuite : Si $\mathcal{F}, F \vdash G$ et si x n'est pas libre dans \mathcal{F} et G , alors $\mathcal{F}, \exists xF \vdash G$

(iii) Prouver alors que : $\exists x \forall y F \vdash \forall y \exists x F$

Chapitre 7

Automates finis

1 Motivations et exemples

Objectif : formaliser un problème et trouver un programme qui le résoud.

Le problème est formalisé comme un langage sur un alphabet (ie. un ensemble de mots).

Soit un alphabet A (un ensemble fini) et le monoïde A^* des mots sur cet alphabet pour la concaténation, avec élément neutre le mot vide noté ε .

Rappel. Un langage est un sous-ensemble $L \subset A^*$

Un mot w est solution d'un problème P s'il appartient à un langage L_P (où L_P est le langage résolvant le problème P).

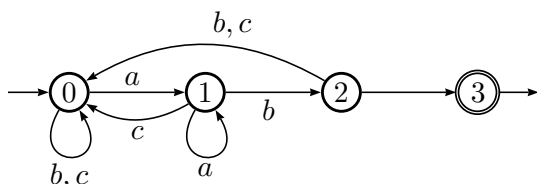
Exemple. Le digicode avec 3 boutons a, b, c a pour code d'ouverture aba [aba est mot sur l'alphabet $\{a, b, c\}^*$].

Les solutions du problème : ensemble des mots sur $\{a, b, c\}^*$ qui se terminent par une ouverture exactement une fois de la porte (ie. se termine exactement par aba).

Quelques exemples :

- $abaaba$: ouvre deux fois la porte \rightarrow pas OK
- $babba \underbrace{aba}$: ouvre une fois exactement la porte \rightarrow OK

Un automate fini pour le digicode :



Exemple. Soit F une formule du calcul propositionnel sur l'ensemble des propositions $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$

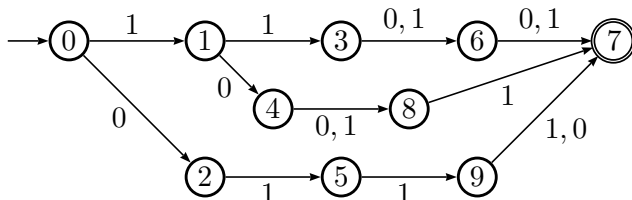
Problème. Trouver les interprétations $I: \mathcal{P} \rightarrow \mathcal{B}$ qui satisfont F .

On considère l'alphabet $A = \{0, 1\}$.

Une interprétation est décrite par un mot de longueur 4

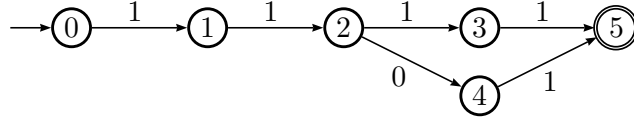
$w = a_1 a_2 a_3 a_4 \in A^*$ où a_i est une valeur de p_i

- Soit $F_1: (p_1 \wedge p_2) \vee (p_2 \wedge p_3) \vee (p_1 \wedge p_4)$
 $L_1 = \{1100, 1101, 1110, 1111, 0110, 0111, 1001, 1011\}$
 Schéma de l'automate \mathcal{A}_1 :



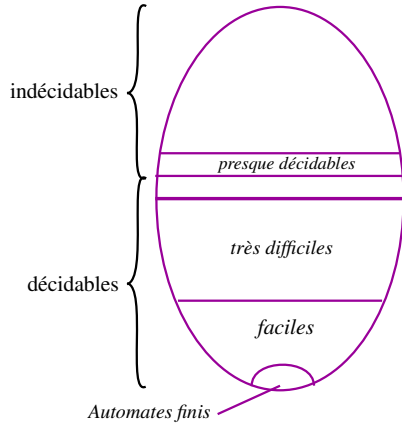
- Soit $F_2 = p_1 \wedge p_2 \wedge p_4$
 $L_2 = \{1101, 1111\}$

Schéma de l'automate \mathcal{A}_2 :

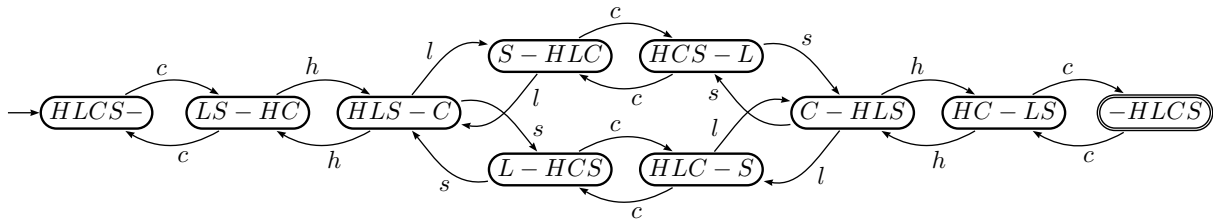


Méthode. On part du problème I , on définit le langage L_I des solutions de I , on construit l'automate \mathcal{A}_I tel que w est « passé » si, et seulement si, w est solution de I .

Catégories de problèmes :



Exemple. Modéliser le problème « homme (h), loup (l), chèvre (c), salade (s) » par un automate fini.



Exemple. Modéliser le problème du barman aveugle (avec des gants de boxe)

But. Avoir la combinaison $(0, 0, 0, 0)$ (appelée G) gagnante.

Mouvements du barman :

- Retourner un seul verre (R)
- Retourner deux verres adjacents (A)
- Retourner deux verres disposés en diagonale (D)

Les différentes combinaisons :

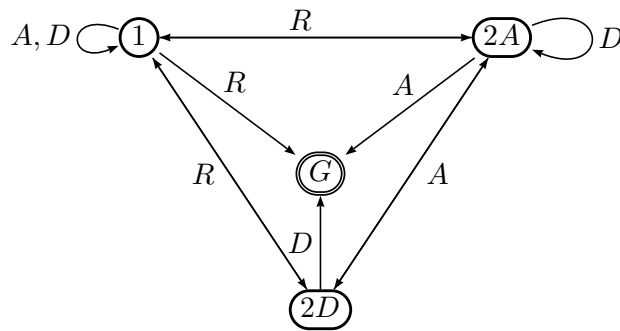
- 1 : $(0, 0, 0, 1)$ ou $(1, 1, 1, 0)$
- 2A : $(0, 0, 1, 1)$
- 2D : $(1, 0, 1, 0)$
- G : $(0, 0, 0, 0)$ ou $(1, 1, 1, 1)$

Question. Le barman a-t-il une stratégie gagnante ?

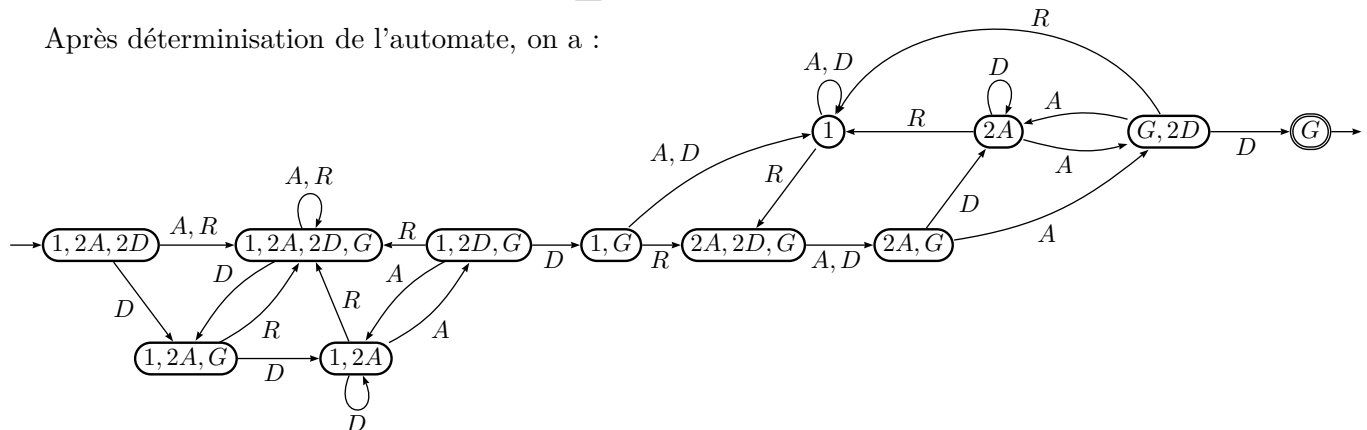
Si oui, quelle est la stratégie minimale ?

On obtiendra une stratégie (ou pas) pour le barman en :

- Déterminant cet automate
- Cherchant un chemin (s'il existe) minimal de l'état initial $\{1, 2D, 2A\}$



Après détermination de l'automate, on a :



Rappel. Si L et M sont deux langages sur A^* :

- $L.M = \{uv/u \in L, v \in M\}$
- $L^* = \bigcup_{i \geq 0} L^i \longrightarrow \begin{cases} L^0 = \{\varepsilon\} \\ L^{i+1} = L.L^i \end{cases}$

Example. On $a : a^* = \{a^n, n \in \mathbb{N}\} = \{\varepsilon, a, a^2, a^3, \dots\}$
Ainsi, $a^3 = aaa$

2 Définitions

Définition. Un système de transitions sur un alphabet A est un triplet $\mathcal{T} = (S, T, S_0)$ où :

- S est l'ensemble des configurations
- $S_0 \subset S$ est le sous-ensemble des configurations initiales.
- $T \subset S \times A \times S$ est l'ensemble des transitions

Une transition de T est un triplet (s, a, s') noté $s \xrightarrow{a} s'$ équivalent à

Une exécution de \mathcal{T} est un chemin dans ce graphe étiqueté, c'est-à-dire une suite de transitions $(s_0, a_1, s_1)(s_1, a_2, s_2) \cdots (s_{n-1}, a_n, s_n)$ noté : $\textcircled{s_0} \xrightarrow{a_1} \textcircled{s_1} \xrightarrow{a_2} \cdots \xrightarrow{a_n} \textcircled{s_n}$ partant d'une configuration initiale $(s_0 \in S_0)$.

Pour une transition $(s) \xrightarrow{a} (s')$, a est l'étiquette.

Pour un chemin $(s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} s_n)$, son étiquette est le mot $a_1 a_2 \dots a_n \in A^*$

On note $\textcircled{s} \xrightarrow{a} \textcircled{s'}$ s'il existe un chemin d'étiquette a de s à s' avec pour convention $\textcircled{s} \xrightarrow{E} \textcircled{s'}$ si $s = s'$

Exemple. Modéliser un tampon (buffer) à l'aide d'un système de transitions.

Etat : \mathbb{N} et $A = \{p, c\}$

p : production d'un objet ajouté au tampon, c consommation d'un objet retiré du tampon.

Définition. Un automate fini est un système de transitions :

- dont l'ensemble des configurations S est fini (états)
- auquel on adjoint un sous-ensemble $F \subset S$ d'états finals

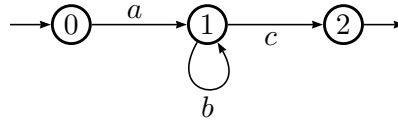
Remarque. On note $\mathcal{A} = (S, T, S_0, F)$ pour un automate fini.

Un mot $w \in A^*$ est accepté par \mathcal{A} s'il existe un chemin d'étiquette $w : \textcircled{s_0} \xrightarrow{w} \textcircled{s_n}$ où $s_0 \in S_0$ et $s_n \in F$

$L(\mathcal{A})$ est l'ensemble des mots acceptés par \mathcal{A} .

Définition. Un automate fini $\mathcal{A} = (S, T, S_0, F)$ est complet sur l'alphabet A si $\forall s \in S, \forall a \in A$, il existe une transition d'étiquette a sortant de s

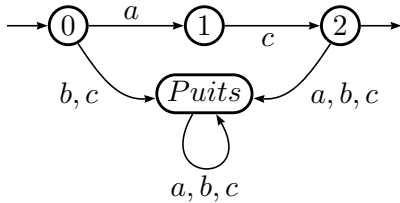
Exemple. Soit l'automate \mathcal{A} suivant :



Alphabet $A = \{a, b, c\}$

$L(\mathcal{A}) = ab^*c$

Cet automate n'est pas complet. On le complète *sans changer le langage accepté* :



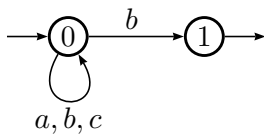
3 Détermination

Définition. Un automate fini $\mathcal{A} = (S, T, S_0, F)$ est déterministe si :

- Il y a exactement un état initial $S_0 = \{s_0\}$
- Pour tout état s et pour toute lettre a , il y a au plus une transition sortant de s avec l'étiquette a .

(ie. $\textcircled{s} \xrightarrow{a} \textcircled{s_1}$ et $\textcircled{s} \xrightarrow{a} \textcircled{s_2}$ alors $s_1 = s_2$)

Automate pour les mots se terminant par b sur l'alphabet $\{a, b, c\}$



$L(\mathcal{A}) = A^*b$

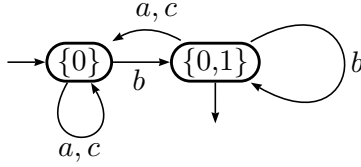
\mathcal{A} est non déterministe car : $\textcircled{0} \xrightarrow{b} \textcircled{1}$ et $\textcircled{0} \xrightarrow{b} \textcircled{0}$

Théorème. Soit \mathcal{A} un automate fini. Il existe un automate fini déterministe \mathcal{D} tel que $L(\mathcal{A}) = L(\mathcal{D})$

Remarque. Construction : états de $\mathcal{D} : \mathcal{P}(S)$

$\mathcal{A} = (S, T, S_0, F)$ Etat initial : $I = \{S_0\}$

Un état $P \subset S$ est final si $P \cap F \neq \emptyset$



$P \subset S, a \in A$

$(P) \xrightarrow{a} (P')$ dans \mathcal{D} avec $P' = \{s' \in S / \exists s \in P \text{ tq } s \xrightarrow{a} s'\}$

Exemple. $(\{0\}) \xrightarrow{b} (\{0,1\})$ parce que $(0) \xrightarrow{b} (0)$ et $(0) \xrightarrow{b} (1)$ dans \mathcal{A}

Rappel de la construction. On part de $\mathcal{A} = (S, T, S_0, F)$

On construit $\mathcal{D} = (S', T', \{s_0\}, F')$ avec :

$S' = \mathcal{R}(S), s_0 = S_0$

$F' = \{P \subset S / P \cap F \neq \emptyset\}$ si $P, P' \subset S$

$P \xrightarrow{a} P'$ si $P' = \{s' \in S / \exists s \in P \text{ tq } s \xrightarrow{a} s'\}$

Lemme. Soit $u \in A^*$

$P \xrightarrow{u} P'$ est un chemin d'étiquette u dans \mathcal{D} si, et seulement si,

$$P' = \left\{ s' \in S / \exists s \in P \text{ tq } s \xrightarrow{u} s' \text{ est un chemin d'étiquette } u \text{ dans } \mathcal{A} \right\}$$

Démonstration par récurrence sur $|u|$. On a :

- Si $|u| = 0$, alors $u = \varepsilon$ et $P \xrightarrow{\varepsilon} P'$ si, et seulement si, $P' = P$ et $\forall s \in S, s \xrightarrow{\varepsilon} s$
- Supposons $u = av, v \in A^*$ et $P \xrightarrow{u} P'$ donc $\exists P''$ tel que $P \xrightarrow{a} P'' \xrightarrow{v} P'$
Or, par construction $P'' = \{s'' \in S / \exists s \in P \text{ tq } s \xrightarrow{a} s''\}$
 $P'' \xrightarrow{v} P'$ avec $|v| < |u|$ par hypothèse de récurrence $P' = \{s' \in S / \exists s'' \in P'' \text{ tq } s'' \xrightarrow{v} s'\}$
Ainsi, $P' = \{s' \in S / \exists s \in P \text{ tq } s \xrightarrow{av} s'\}$

On termine la démonstration en vérifiant sur $\mathcal{L}(\mathcal{A}) \subset \mathcal{L}(\mathcal{D})$ et $\mathcal{L}(\mathcal{D}) \subset \mathcal{L}(\mathcal{A})$ □

4 Opérations

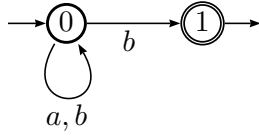
Définition. Soit A un alphabet et $L \subset A^*$. Le langage L est reconnaissable s'il existe un automate fini \mathcal{A} tel que $L = L(\mathcal{A})$ (ie. L est l'ensemble de mots acceptés par \mathcal{A}).

Proposition. L'ensemble des langages reconnaissables sur un alphabet A est formé par union, intersection, complémentaire.

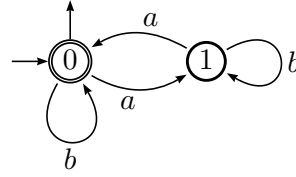
- (i) Soit L reconnaissable, soit \mathcal{D} un automate déterministe complet tel que $L = L(\mathcal{D})$
Si $\mathcal{D} = (S, T, \{s_0\}, F)$, alors $A^* \setminus L$ (complémentaire de L) est accepté par $\bar{\mathcal{D}} = (S, T, \{s_0\}, S \setminus F)$
- (ii) Soit L_1, L_2 reconnaissables, $L_1 = L(\mathcal{A}_1), L_2 = L(\mathcal{A}_2)$, on cherche un automate \mathcal{B} tel que $L(\mathcal{B}) = L_1 \cup L_2$. On suppose que $\mathcal{A}_1 = (S_1, T_1, S_{01}, F_1)$ et $\mathcal{A}_2 = (S_2, T_2, S_{02}, F_2)$ avec $S_1 \cap S_2 = \emptyset$, alors $\mathcal{B} = (S_1 \cup S_2, T_1 \cup T_2, S_{01} \cup S_{02}, F_1 \cup F_2)$
- (iii) Soit $L_1 = L(\mathcal{A}_1), L_2 = L(\mathcal{A}_2)$; on cherche \mathcal{C} tel que $L(\mathcal{C}) = L_1 \cap L_2$
 $\mathcal{C} = (S_1 \times S_2, \hat{T}, S_{01} \times S_{02},)$
 \hat{T} contient $(s_1, s_2) \xrightarrow{a} (s'_1, s'_2)$ si, et seulement si, $s_1 \xrightarrow{a} s'_1$ dans \mathcal{A}_1 et $s_2 \xrightarrow{a} s'_2$ dans \mathcal{A}_2

Exemple. Soit $A = \{a, b\}$ et les langages :

– $L_1 = \{\text{mots qui se terminent par } b\}$

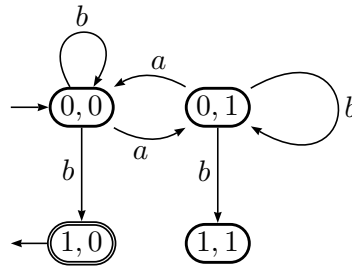


– $L_2 = \{\text{mots qui ont un nombre pair de } a\}$



Remarque. A condition que \mathcal{A}_1 et \mathcal{A}_2 soient complets, et en changeant les états finaux en $F_1 \times S_2 \cup S_2 \times F_2$.

Cet automate acceptera $L_1 \cup L_2$.



5 Langages rationnels

Définition. Soit A un alphabet. Un langage L de A^* est rationnel si :

- Soit $L = \emptyset$ ou $L = \{a\}$, $a \in A$
- Soit
$$\left. \begin{array}{l} L = L_1 \cup L_2 \\ \text{ou } L = L_1.L_2 \end{array} \right| \text{ où } L_1 \text{ et } L_2 \text{ sont des langages rationnels.}$$
- Soit $L = M^*$ où M est un langage rationnel

Théorème (Théorème de Kleene). *Un langage vide est rationnel si, et seulement si, il est reconnaissable :*

Proposition. *Tout langage rationnel est reconnaissable*

Démonstration par induction structurelle. On a :

- Base. $L = \emptyset$: $\rightarrow (\text{circles}) \rightarrow$ $L = \{a\}$: $\rightarrow (s_0) \xrightarrow{a} (s_1) \rightarrow$
- Induction.

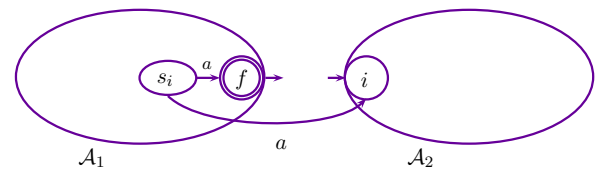
— $L_1 \cup L_2$: cf paragraphe précédent.

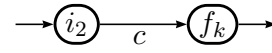
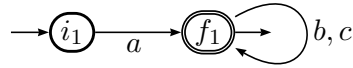
— $L_1.L_2$: Soit $\mathcal{A}_1 = (S_1, T_1, S_{01}, F_1)$ et $\mathcal{A}_2 = (S_2, T_2, S_{02}, F_2)$ et $\mathcal{B} = L_1.L_2$
 $\mathcal{B} = (S_1 \cup S_2, T_1 \cup T_2 \cup \hat{T}, \hat{S}_0, F_2)$

$$\hat{S}_0 = \begin{cases} S_{01} & \text{si } S_{01} \cap F_1 = \emptyset \\ S_{02} & \text{sinon (cas où } \varepsilon \in L_1) \end{cases}$$

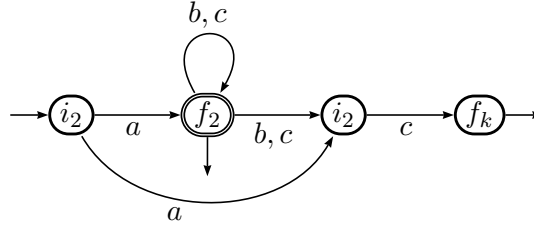
$$\hat{T} = \left\{ s \xrightarrow{a} i / i \in S_{02}, \exists f \in F_1 \text{ tq } s \xrightarrow{a} f \in T_1 \right\}$$

Exemple. On considère les deux automates \mathcal{A}_1 et \mathcal{A}_2 suivants :





On a finalement l'automate produit \mathcal{A} suivant :



— $L = M^*$ où $M^* = M^+ \cup \{\varepsilon\}$

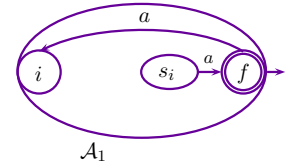
On construit :

— \mathcal{A} pour M

— \mathcal{C} pour M^+ : $M^* = \bigcup_{n \geq 0} M^n$ et $M^+ = \bigcup_{n \geq 1} M^n = M^* \setminus \{\varepsilon\}$

$\mathcal{A} = (S, T, S_0, F)$ pour M

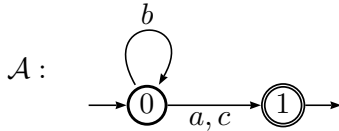
$\mathcal{C} = (S, T \cup \hat{T}, S_0, F)$ accepte M^+ avec $\hat{T} = \{s \xrightarrow{a} i / i \in S_0 \text{ et } \exists f \in F \text{ tq } s \xrightarrow{a} f \text{ dans } \mathcal{A}\}$



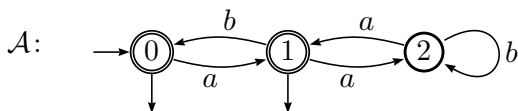
□

Proposition. *Tout langage reconnaissable est rationnel.*

Méthode. On calcule l'expression rationnelle sur un automate.



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



— $L_0 = L(\mathcal{A})$

— $L_1 = L(\mathcal{A}_1)$ où 1 est l'état initial

— L_2 : mots acceptés en partant de l'état 2

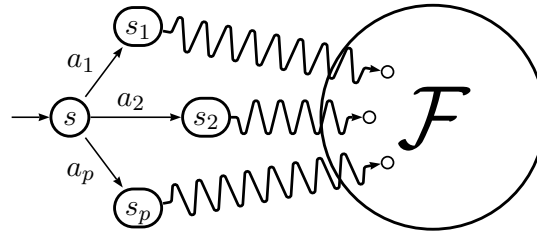
Soit $\mathcal{A} = (S, T, S_0, F)$ un automate fini.

On lui associe une famille d'automates $\mathcal{A}_s = (S, T, \{s\}, F) \longrightarrow L_s = L(\mathcal{A}_s)$

$$L = L(\mathcal{A}) = \bigcup_{s \in S_0} L_s$$

Soit $s \in S$ et les transitions sortant de s :

Tout chemin de s vers un état final passe par un des s_i (ie. commence par un des a_i).



$$L_s = \begin{cases} a_1 L_{s_1} + a_2 L_{s_2} + \dots + a_p L_{s_p} & \text{si } s \notin F \\ a_1 L_{s_1} + a_2 L_{s_2} + \dots + a_p L_{s_p} + \varepsilon & \text{si } s \in F \end{cases}$$

En reprenant l'exemple...

$$\begin{cases} L_0 = aL_1 + \varepsilon & (1) \\ L_1 = aL_2 + bL_0 + \varepsilon & (2) \\ L_2 = bL_2 + aL_1 & (3) \end{cases}$$

Lemme (Arden). Soit K un langage de A^* ne contenant pas ε et M un langage de A^* .

L'équation $X = KX + M$ a pour unique solution $X = K^*M$ (ie. K^*M est l'unique point fixe de $g: \mathcal{P}(A^*) \rightarrow \mathcal{P}(A^*)$)

$$X \mapsto KX + M$$

Exemple. On reprend l'exemple précédent.

D'après (3), on a $L_2 = KL_2 + M$ où $K = \{b\}$, $M = aL_1$

Donc $L_2 = b^*aL_1$

On remplace dans (2)

$L_1 = ab^*aL_1 + bL_0 + \varepsilon = KL_1 + M$ avec $K = ab^*a$ et $M = bL_0 + \varepsilon$

Donc $L_1 = (ab^*a)^*(bL_0 + \varepsilon)$

On remplace dans (1)

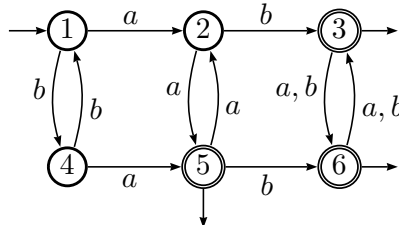
$$\begin{aligned} L_0 &= a(ab^*a)^*(bL_0 + \varepsilon) \\ &= \underbrace{a(ab^*a)^*}_K . bL_0 + \underbrace{a(ab^*a)^*}_M + \varepsilon \end{aligned}$$

Donc finalement, on a $L_0 = (a(ab^*a)^*b)^*[a(ab^*a)^* + \varepsilon]$

6 Minimisation

Objectif. Pour un langage L reconnaissable, trouver l'automate déterministe qui l'accepte avec un nombre minimal d'états.

Exemple. $A = \{a, b\}$



On considère dans ce paragraphe des automates finis déterministes complets et monogènes (ie. Tout état $s \in S$ est accessible depuis l'état initial).

Pour un automate déterministe et complet, $\mathcal{A} = (S, T, \{s_0\}, F)$ et tout état $s \in S$, tout mot $u \in A^*$, il existe un unique s' tel que $s \xrightarrow{u} s'$ qui sera noté $s' = s.u$

Définition. Soit $\mathcal{A} = (S, T, \{s_0\}, F)$ un automate fini déterministe complet et monogène. Deux états s_1 et s_2 sont dits inséparables si $\forall u \in A^*, s_1.u \in F$ si, et seulement si, $s_2.u \in F$

ie. $s_1 \xrightarrow{u} s'_1$ et $s_2 \xrightarrow{u} s'_2$ alors $\begin{cases} s'_1, s'_2 \in F \\ s'_1 \notin F \text{ et } s'_2 \notin F \end{cases}$

Ils sont séparables dans le cas contraire.

Proposition. La relation sur S définie par $s_1 \sim s_2$ si s_1 et s_2 sont inséparables est une relation d'équivalence (équivalence de Nérode).

De plus, si $s_1 \sim s_2$, alors $\forall u \in A^*, s_1.u \sim s_2.u$

Remarque. Si $s_1 \sim s_2$ ou bien $s_1 \in F$ et $s_2 \in F$ ou bien aucun des deux.

Exemple. En reprenant l'exemple précédent, vérifier que les classes pour \sim sont : $\{1\}, \{2\}, \{5\}, \{4\}, \{3, 6\}$.

On part de $\mathcal{A} = (S, T, \{s_0\}, F)$ et \sim l'équivalence de Nérode associée.

On note $\tilde{S} = S / \sim$ le quotient.

On note $[S]$ la classe d'un état s pour \sim

Si $s_1 \sim s$, alors $\forall a \in A, s_1.a \sim s.a$

On peut alors poser $[s].a = [s.a]$

On pose $\tilde{F} = \{[s], s \in F\}$

$\tilde{T} = \{[s] \xrightarrow{a} [s'] / s \xrightarrow{a} s' \text{ dans } \mathcal{A}\}$

L'automate quotient $\tilde{\mathcal{A}}$ est défini par $\tilde{\mathcal{A}} = (\tilde{S}, \tilde{T}, [s_0], \tilde{F})$

Théorème. Soit \mathcal{A} un automate fini déterministe complet et monogène et $L = L(\mathcal{A})$. Alors $\tilde{\mathcal{A}}$ est l'unique (à isomorphisme près) automate minimal de \mathcal{A} .

Méthode (Moore - Algorithme de calcul de \sim). On définit une suite d'équivalences :

- \sim_0 : la relation initiale à deux classes F et $S \setminus F$
- \sim_{k+1} : la relation définie par : $s_1 \sim_{k+1} s_2$ si $s_1 \sim_k s_2$ et $\forall a \in A, s_1.a \sim_k s_2.a$

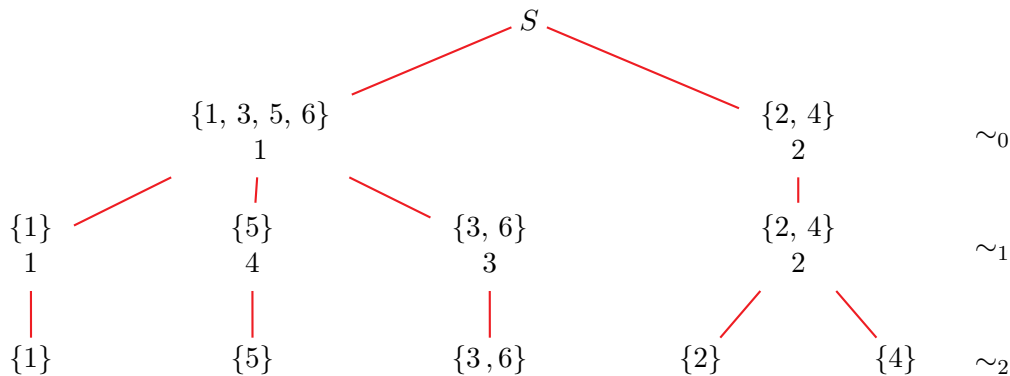
Proposition. L'algorithme se termine si $\sim_p = \sim_{p+1}$, alors $\sim_p = \sim$

Exemple. En reprenant l'exemple précédent, on a :

s	1	2	3	4	5	6
$\text{cl}(s)$	1	2	1	2	1	1
$\text{cl}(s.a)$	2	1	1	1	2	1
$\text{cl}(s.b)$	2	1	1	1	1	1

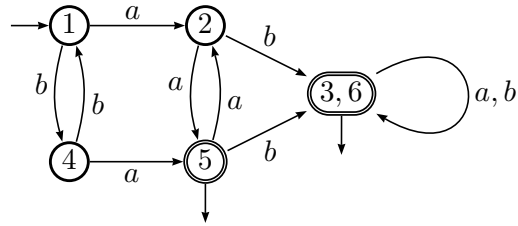
s	1	2	3	4	5	6
$\text{cl}(s)$	1	2	3	2	4	3
$\text{cl}(s.a)$	2	4	1	4	4	1
$\text{cl}(s.b)$	2	3	1	1	1	1

On a l'arbre suivant :



Conclusion. On a $\sim = \sim_2$ et $\text{cl}(s) \xrightarrow{a} \text{cl}(s')$

Ainsi, l'automate quotient minimal $\tilde{\mathcal{A}}$ est :



Chapitre 8

Exercices

1 Induction

Exercice (Lois de Morgan généralisées). Montrer que, pour $n \geq 2$, $\overline{\bigcup_{i=1}^n A_i} = \bigcap_{i=1}^n \overline{A_i}$ et $\overline{\bigcap_{i=1}^n A_i} = \bigcup_{i=1}^n \overline{A_i}$.

Correction. Soit $n \in \mathbb{N}$. On note $\mathcal{P}(n)$ la propriété : $\overline{\bigcup_{i=1}^n A_i} = \bigcap_{i=1}^n \overline{A_i}$ et $\overline{\bigcap_{i=1}^n A_i} = \bigcup_{i=1}^n \overline{A_i}$.

- Base. On a : $\overline{A_1 \cup A_2} = \overline{A_1} \cap \overline{A_2}$ et $\overline{A_1 \cap A_2} = \overline{A_1} \cup \overline{A_2}$ d'après les lois de Morgan.
- Induction. Soit $n \in \mathbb{N}$. Supposons $\mathcal{P}(n)$ et montrons $\mathcal{P}(n+1)$.

$$\begin{aligned} \overline{\bigcup_{i=1}^{n+1} A_i} &= \overline{\bigcup_{i=1}^n A_i \cup A_{n+1}} = \overline{\bigcup_{i=1}^n A_i \cap \overline{A_{n+1}}} \text{ d'après les lois de Morgan} \\ &= \bigcap_{i=1}^n \overline{A_i \cap \overline{A_{n+1}}} \text{ par } \mathcal{P}(n) \\ \overline{\bigcup_{i=1}^{n+1} A_i} &= \bigcap_{i=1}^{n+1} \overline{A_i} \\ \overline{\bigcap_{i=1}^n A_i} &= \overline{\bigcap_{i=1}^n A_i \cap A_{n+1}} = \overline{\bigcap_{i=1}^n A_i \cup \overline{A_{n+1}}} \text{ d'après les lois de Morgan} \\ &= \bigcup_{i=1}^n \overline{A_i \cup \overline{A_{n+1}}} \text{ par } \mathcal{P}(n) \\ \overline{\bigcap_{i=1}^n A_i} &= \bigcup_{i=1}^{n+1} \overline{A_i} \end{aligned}$$

Ainsi, $\mathcal{P}(n+1)$ est vraie.

- Conclusion. Pour tout $n \in \mathbb{N}$, $\mathcal{P}(n)$ est vraie.

Exercice . Donner une définition inductive de $f(n) = a^{2^n}$.

Correction. Pour tout $n \in \mathbb{N}$, on a $a^{2^{n+1}} = (a^{2^n})^2$

Ainsi, on définit la fonction $f: \mathbb{N} \rightarrow \mathbb{N}$ inductivement par :

$$n \mapsto a^{2^n}$$

- Base. $f(0) = a$
- Induction. $f(n+1) = f(n)^2$ pour tout $n \in \mathbb{N}$

Exercice (Nombres de Fermat). On rappelle que les nombres de Fibonacci sont définis par

$$\begin{cases} F_n = F_{n-1} + F_{n-2} & \text{si } n > 1 \\ F_0 = 0 \text{ et } F_1 = 1 \end{cases}$$

Montrer pour tout $n > 0$:

- (i) $F_1 + F_3 + \cdots + F_{2n-1} = F_{2n}$
- (ii) $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$
- (iii) $F_1^2 + F_2^2 + \cdots + F_n^2 = F_n F_{n+1}$
- (iv) $\varphi^{n-2} \leq F_n \leq \varphi^{n-1}$ où $\varphi = \frac{1+\sqrt{5}}{2}$ est la racine du polynôme $x^2 - x - 1$

Correction. On a :

$$F_1 + F_3 + \cdots + F_{2n-1} = F_{2n}$$

Soit $n \in \mathbb{N}^*$. On note $\mathcal{P}(n)$ la proposition : $F_1 + F_3 + \cdots + F_{2n-1} = F_{2n}$

- Base. Pour $n = 1$, on a $F_1 = 1$ et $F_2 = F_0 + F_1 = 0 + 1 = 1$
- Induction. Soit $n \in \mathbb{N}^*$. Supposons $\mathcal{P}(n)$ et montrons $\mathcal{P}(n+1)$

$$\begin{aligned} F_1 + F_3 + \cdots + F_{2(n+1)-1} &= F_1 + F_3 + \cdots + F_{2n+2-1} \\ &= F_1 + F_3 + \cdots + F_{2n-1} + F_{2n+1} \\ &= F_{2n} + F_{2n+1} \text{ par } \mathcal{P}(n) \\ &= F_{2n+2} \text{ par définition de } F_{2n+2} \end{aligned}$$

- Conclusion. $\forall n \in \mathbb{N}^*$, $F_1 + F_3 + \cdots + F_{2n-1} = F_{2n}$

$$F_{n+1}F_{n-1} - F_n^2 = (-1)^n$$

Soit $n \in \mathbb{N}^*$. On note $\mathcal{P}(n)$ la proposition : $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$

- Base. Pour $n = 1$, on a $F_2F_0 - F_1^2 = 0 - 1^2 = -1 = (-1)^1$
- Induction. Soit $n \in \mathbb{N}^*$. Supposons $\mathcal{P}(n)$ et montrons $\mathcal{P}(n+1)$.

$$\begin{aligned} F_{(n+1)+1}F_{(n+1)-1} - F_{(n+1)}^2 &= F_{n+2}F_n - F_{n+1}^2 = (F_{n+1} + F_n)F_n - F_{n+1}^2 \text{ par définition de } F_{n+2} \\ &= F_{n+1}F_n + F_n^2 - F_{n+1}^2 \\ &= F_{n+1}F_{n-1} - (-1)^n + F_{n+1}F_n - F_{n+1}^2 \text{ par } \mathcal{P}(n) \\ &= F_{n+1}(F_{n-1} + F_n) - F_{n+1}^2 - (-1)^n \\ &= F_{n+1}^2 - F_{n+1}^2 - (-1)^n \text{ par définition de } F_{n+1} \\ F_{(n+1)+1}F_{(n+1)-1} - F_{(n+1)}^2 &= (-1)^{n+1} \end{aligned}$$

- Conclusion. $\forall n \in \mathbb{N}^*$, $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$

$$F_1^2 + F_2^2 + \cdots + F_n^2 = F_n F_{n+1}$$

Soit $n \in \mathbb{N}^*$. On note $\mathcal{P}(n)$ la proposition : $F_1^2 + F_2^2 + \cdots + F_n^2 = F_n F_{n+1}$

- Base. Pour $n = 1$, on a $F_1F_2 = 1$ et $F_1^2 = 1$
- Induction. Soit $n \in \mathbb{N}^*$. Supposons $\mathcal{P}(n)$ et montrons $\mathcal{P}(n+1)$

$$\begin{aligned} F_1^2 + F_2^2 + \cdots + F_n^2 + F_{n+1}^2 &= F_n F_{n+1} + F_{n+1}^2 \text{ d'après } \mathcal{P}(n) \\ &= F_{n+1}(F_n + F_{n+1}) \\ &= F_{n+1}F_{n+2} \text{ par définition de } F_{n+2} \end{aligned}$$

- Conclusion. $\forall n \in \mathbb{N}^*, F_1^2 + F_2^2 + \dots + F_n^2 = F_n F_{n+1}$

$\varphi^{n-2} \leq F_n \leq \varphi^{n-1}$ où $\varphi = \frac{1+\sqrt{5}}{2}$ est la racine du polynôme $x^2 - x - 1$

Soit $n \in \mathbb{N}^*$. On note $\mathcal{P}(n)$ la proposition : $\varphi^{n-2} \leq F_n \leq \varphi^{n-1}$

- Base. Pour $n = 1$, on a bien $\frac{2}{1+\sqrt{5}} = \varphi^{-1} \leq F_1 = 1 \leq \varphi^0 = 1$
 Pour $n = 2$, on a bien $1 = \varphi^0 \leq F_2 = 1 \leq \varphi^1 = \frac{1+\sqrt{5}}{2}$
- Induction. Soit $n \in \mathbb{N}$. Supposons $\mathcal{P}(k)$ vraie pour tout $k \in [1, n]$ et montrons $\mathcal{P}(n+1)$
 On suppose $\varphi^{k-2} \leq F_k \leq \varphi^{k-1}$ pour tout $k \in [1, n]$ et on veut montrer $\varphi^{n-1} \leq F_{n+1} \leq \varphi^n$
 En ajoutant F_{n-1} à chaque membre de l'inéquation $\varphi^{n-2} \leq F_n \leq \varphi^{n-1}$, on a :
 $\varphi^{n-2} + F_{n-1} \leq F_n + F_{n-1} \leq \varphi^{n-1} + F_{n-1}$
 De plus, par $\mathcal{P}(k)$, $\varphi^{n-3} \leq F_{n-2} \leq \varphi^{n-2}$, donc :
 $\varphi^{n-2} + \varphi^{n-3} \leq F_{n+1} \leq \varphi^{n-1} + \varphi^{n-2}$ par définition de F_{n+1}
 Or φ est la racine de $x^2 - x - 1$, donc $\varphi^2 - \varphi - 1 = 0 \Leftrightarrow \varphi^2 = \varphi + 1$
 On a alors : $\varphi^{n-3}(\varphi + 1) \leq F_{n+1} \leq \varphi^{n-2}(\varphi + 1)$
 Puis $\varphi^{n-1} \leq F_{n+1} \leq \varphi^n$ en appliquant $\varphi^2 = \varphi + 1$
- Ainsi, $\forall n \in \mathbb{N}^*, \varphi^{n-2} \leq F_n \leq \varphi^{n-1}$

Exercice (Induction structurelle). On considère le sous-ensemble \mathcal{D} de $\mathbb{N} \times \mathbb{N}$ défini inductivement par :

$$\begin{cases} (n, 0) \in \mathcal{D} \\ \text{Si } (n, n') \in \mathcal{D}, \text{ alors } (n, n + n') \in \mathcal{D} \end{cases}$$

- (i) Donner quelques éléments de \mathcal{D} .
- (ii) Montrer par récurrence sur k que si $n' = kn$, alors $(n, n') \in \mathcal{D}$.
- (iii) Montrer que si $(n, n') \in \mathcal{D}$, alors pour tout $k \in \mathbb{N}$, on a $n' = kn$.

Correction. On a :

Donner quelques éléments de \mathcal{D} .

$(0, 0)$ $(1, 0)$ $(2, 0)$ sont des éléments de \mathcal{D}
 $(1, 1)$ $(2, 2)$
 $(1, 2)$ $(2, 4)$
 $(1, 3)$ $(2, 6)$
 $(1, 4)$ $(2, 8)$

Montrer par récurrence sur k que si $n' = kn$, alors $(n, n') \in \mathcal{D}$.

Soit $n \in \mathbb{N}$. On note $\mathcal{P}(k)$ la propriété : $n' = kn$, alors $(n, n') \in \mathcal{D}$.

- Base. Soit $k = 0$. On a alors $n' = 0$ et $(n, 0) \in \mathcal{D}$ par définition
 Donc $(n, n') \in \mathcal{D}$
- Induction. Soit $n \in \mathbb{N}$. Supposons $\mathcal{P}(k)$ et montrons $\mathcal{P}(k + 1)$
 Soit $n' = (k + 1)n$. Montrons que $(n, n') \in \mathcal{D}$
 $n' = kn + n$. Or $(n, kn) \in \mathcal{D}$, alors $(n, \underbrace{kn + n}_{n'}) \in \mathcal{D}$
 Donc $(n, n') \in \mathcal{D}$
- Conclusion. Pour tout $n \in \mathbb{N}$, si $n' = kn$, alors $(n, n') \in \mathcal{D}$.

Montrer que si $(n, n') \in \mathcal{D}$, alors pour tout $k \in \mathbb{N}$, on a $n' = kn$.

Soit $n \in \mathbb{N}$. On note $\mathcal{P}(n, n')$ la propriété : si $(n, n') \in \mathcal{D}$, alors pour tout $k \in \mathbb{N}$, on a $n' = kn$.

- Base. On montre que $\mathcal{P}(n, n')$ est vraie pour tout élément de la base de \mathcal{D} .
 On a : $0 = 0 \times n$ pour tout $n \in \mathbb{N}$
- Induction. Soit $n \in \mathbb{N}$. On suppose $\mathcal{P}(n, n')$ vraie pour tout élément $(n, n') \in \mathcal{D}$
 On montre alors que \mathcal{P} est vraie pour le/les élément(s) de \mathcal{D} construit(s) à partir de (n, n')
 On a : $n' = kn$, $k \in \mathbb{N}$ par hypothèse de récurrence.
 $n + n' = n + kn = (k + 1)n$
 Or $(n, n + n') \in \mathcal{D}$, donc $(n, n + n')$ vérifie la propriété $\mathcal{P}(n, n + n')$ vraie.
- Conclusion. Pour tout $n \in \mathbb{N}$, si $(n, n') \in \mathcal{D}$, alors pour tout $k \in \mathbb{N}$, on a $n' = kn$.

Exercice (Arbre binaire). Donner une définition inductive de la hauteur $h(t)$, du nombre de noeuds $n(t)$, du nombre d'arêtes $ar(t)$ et du nombre de feuilles $f(t)$ d'un arbre binaire.

On suppose que t est un arbre binaire (0, 1 ou deux fils par noeud). Montrer que :

(i) $n(t) \leq 2^{h(t)} - 1$

(ii) $f(t) \leq 2^{h(t)-1}$

Correction. On a :

Hauteur de l'arbre	$h(t)$	Base	$h(\emptyset) = 0$
		Induction	$h((a, g, d)) = 1 + \max(h(g), h(d))$
Nombre de noeud	$n(t)$	Base	$n(\emptyset) = 0$
		Induction	$n((a, g, d)) = 1 + n(g) + n(d)$
Nombre d'arêtes	$ar(t)$	Base	$ar(\emptyset) = 0$
		Induction	$ar((a, g, d)) = 2 + ar(g) + ar(d)$ si $g \neq \emptyset$ et $d \neq \emptyset$
			$= ar(g) + 1$ si $g \neq \emptyset$ et $d = \emptyset$
			$= ar(d) + 1$ si $g = \emptyset$ et $d \neq \emptyset$
			$= 0$ si $g = \emptyset$ et $d = \emptyset$
Nombre de feuilles	$f(t)$	Base	$f(\emptyset) = 0$
		Induction	$f((a, g, d)) = f(g) + f(d)$ si $g \neq \emptyset$ ou $d \neq \emptyset$ $= 1$ si $g = d = \emptyset$

Montrons maintenant :

$n(t) \leq 2^{h(t)} - 1$

- Base. On a : $n(\emptyset) = 0$
Or $h(\emptyset) = 0$ et $2^{h(\emptyset)} - 1 = 1 - 1 = 0 \geq n(\emptyset)$
- Induction. On suppose $n(g) \leq 2^{h(g)} - 1$ et $n(d) \leq 2^{h(d)} - 1$. Montrons $n(t) \leq 2^{h(t)} - 1$ où $t = (a, g, d)$
 $n(t) = 1 + n(g) + n(d) \leq 1 + 2^{h(g)} - 1 + 2^{h(d)} - 1 = 2^{h(g)} + 2^{h(d)} - 1$ par hypothèse de récurrence
Or $h(t) = 1 + \underbrace{\max(h(g), h(d))}_m = 1 + m$
Donc $n(t) \leq 2^m + 2^m - 1 = 2^{m+1} - 1 = 2^{h(t)} - 1$
- Conclusion. On a donc bien $n(t) \leq 2^{h(t)} - 1$

$f(t) \leq 2^{h(t)-1}$

- Base. On a : $f(\emptyset) = 0$
Or $h(\emptyset) = 0$ et $2^{h(\emptyset)-1} = 2^{-1} = \frac{1}{2} \geq f(\emptyset)$
- Induction. On suppose $f(g) \leq 2^{h(g)-1}$ et $f(d) \leq 2^{h(d)-1}$. Montrons $f(t) \leq 2^{h(t)-1}$ où $t = (a, g, d)$
Supposons $g \neq \emptyset$ ou $d \neq \emptyset$
 $f(t) = f(g) + f(d) \leq 2^{h(g)-1} + 2^{h(d)-1}$ par hypothèse de récurrence
Or $h(t) = 1 + \underbrace{\max(h(g), h(d))}_m = 1 + m$
Donc $f(t) \leq 2^{m-1} + 2^{m-1} = 2^m = 2^{h(t)-1}$ car $m = h(t) - 1$
- Conclusion. On a donc bien $f(t) \leq 2^{h(t)-1}$

Exercice (Arbre binaire strict). Soit t un arbre binaire strict (ie. t est non vide, chaque noeud de t a exactement 0 ou 2 fils et il n'y a aucun noeud avec un seul fils non vide). Soit $n(t)$ le nombre de ses noeuds, $f(t)$ le nombre de ses feuilles et $ar(t)$ le nombre de ses arêtes.

- (i) Donner une définition de l'ensemble ABS des arbres binaires stricts.
- (ii) Montrer que si t est un arbre binaire strict non vide, $n(t) = ar(t) + 1$
- (iii) Montrer que si t est un arbre binaire strict non vide, $n(t) = 2f(t) - 1$

Correction. On a :

Donner une définition de l'ensemble ABS des arbres binaires stricts.

ABS sur $A = \{a\}$

- Base. $a \in ABS$
- Induction. $g \in ABS, d \in ABS, a \in A, (a, g, d) \in ABS$

Montrer que si t est un arbre binaire strict non vide, $n(t) = ar(t) + 1$

- Base. On a $n(a) = 1 = 0 + 1 = ar(a) + 1$
- Induction. Supposons $n(g) = ar(g) + 1$ et $n(d) = ar(d) + 1, g, d \in ABS$
 Soit $t = (a, g, d)$
 $n(t) = 1 + n(g) + n(d)$ par définition de $n(t)$
 Puis, $n(t) = 3 + ar(g) + ar(d)$ par hypothèse de récurrence
 Or $ar(t) = ar(g) + ar(d) + 2$ par définition
 Donc $n(t) = ar(t) + 1$
- Conclusion. $n(t) = ar(t) + 1$ pour tout $t \in ABS$

Montrer que si t est un arbre binaire strict non vide, $n(t) = 2f(t) - 1$

- Base. On a $n(a) = 1$ et $2 \cdot f(a) - 1 = 2 - 1 = 1$
- Induction. Supposons $g, d \in ABS$ vérifiant $n(g) = 2f(g) - 1$ et $n(d) = 2f(d) - 1$
 Pour $t = (a, g, d)$, on a :

$$\begin{aligned}
 n(t) &= 1 + n(g) + n(d) \\
 &= 1 + 2f(g) - 1 + 2f(d) - 1 \text{ par hypothèse de récurrence} \\
 &= 2 \underbrace{[f(g) + f(d)]}_{f(t) \text{ par déf}} - 1 \\
 n(t) &= 2f(t) - 1
 \end{aligned}$$

- Conclusion. Pour tout $t \in ABS, n(t) = 2f(t) - 1$

Exercice (Alphabet). Soit A^* le monoïde libre engendré par l'alphabet A . Le miroir d'un mot $u = a_1 a_2 \dots a_n$ est $\tilde{u} = a_n \dots a_2 a_1$. Donner une définition inductive du miroir d'un mot.

Correction. Définition inductive de la construction de A^* , monoïde de A :

- Base. $\varepsilon \in A^*$
- Induction. Si $a \in A$, $m \in A^*$, alors $a.m \in A^*$

Définition inductive du miroir d'un mot :

- Base. $\tilde{\varepsilon} = \varepsilon$
- Induction. Soit $a \in A$, $m \in A^*$, alors $\widetilde{a.m} = \tilde{m}.a$

Exercice (Alphabet). On définit inductivement les langages rationnels (ou bien réguliers) par :

- (i) Un langage fini est rationnel
- (ii) Si L_1 et L_2 sont rationnels, alors $L_1 \cup L_2$ est rationnel
- (iii) Si L_1 et L_2 sont rationnels, alors $L_1.L_2$ est rationnel
- (iv) Si L est rationnel, alors $L^* = \bigcup_{n \geq 0} L^n$ est rationnel ($L^0 = \{\varepsilon\}$)

On appelle miroir du mot $u = a_1 \dots a_n$, le mot $\tilde{u} = a_n \dots a_1$. Si L est un langage rationnel, on définit $\tilde{L} = \{\tilde{u}, u \in L\}$.

Montrer que si L est un langage rationnel, alors \tilde{L} est également rationnel.

Correction. Soit \mathcal{P}_L la proposition : \tilde{L} est un langage rationnel. Montrons que \mathcal{P}_L est vrai pour tout langage rationnel $L \in \text{Rat}(A)$

- (i) Si L est fini, \mathcal{P}_L est vrai car \tilde{L} est également fini.
- (ii) Soient L_1 et L_2 deux langages rationnels vérifiant \mathcal{P}_{L_1} et \mathcal{P}_{L_2} . Montrons que $L = L_1 \cup L_2$ vérifie \mathcal{P}_L (ie. L est rationnel).
 $\tilde{L} = \tilde{L}_1 \cup \tilde{L}_2$ et \tilde{L}_1 et \tilde{L}_2 sont rationnels, donc $\tilde{L}_1 \cup \tilde{L}_2$ est rationnel par (ii) et L vérifie \mathcal{P}_L
- (iii) Supposons \mathcal{P}_{L_1} et \mathcal{P}_{L_2} vraies pour L_1 et L_2 rationnels et montrons que \mathcal{P}_L est vraie avec $L = L_1.L_2$ (ie. \tilde{L} rationnel)
 $\tilde{L} = \tilde{L}_1.\tilde{L}_2$. Or par hypothèse d'induction, \tilde{L}_1 et \tilde{L}_2 sont rationnels, alors $\tilde{L}_1.\tilde{L}_2$ est rationnel par (iii) et L vérifie \mathcal{P}_L
- (iv) Supposons \mathcal{P}_L vraie et montrons \mathcal{P}_{L^*} (ie. \tilde{L}^* est rationnel).
Montrons que $\tilde{L}^n = (\tilde{L})^n$ par récurrence sur n :

- Base. Pour $n = 0$, on a $\tilde{L}^0 = \varepsilon = (\tilde{L})^0$
- Induction. Soit $n \in \mathbb{N}$. Supposons la propriété vraie au rang n et montrons qu'elle est vraie au rang $n + 1$

$$\widetilde{\tilde{L}^{n+1}} = \widetilde{\tilde{L}^n.\tilde{L}} = \tilde{L}^n.\tilde{\tilde{L}} \underset{HR}{=} (\tilde{L})^n.\tilde{\tilde{L}} = (\tilde{L})^{n+1}$$

$$\text{On a alors : } \tilde{L}^* = \widetilde{\bigcup_{n \in \mathbb{N}} L^n} = \bigcup_{n \in \mathbb{N}} (\tilde{L}^n) = \bigcup_{n \in \mathbb{N}} (\tilde{L})^n = (\tilde{L})^*$$

Ainsi, \tilde{L} est un langage rationnel.

Exercice (Ordre bien fondé). Soient (A, \leq_1) et (B, \leq_2) deux ordres bien fondés. Les ordres suivants sont-ils bien fondés ?

- (i) Sur $A \times B$, l'ordre produit $[(a, b) \leq (a', b') \Leftrightarrow (a, \leq_1 a') \wedge (b \leq_2 b')]$.
- (ii) Sur $A \times B$, l'ordre lexicographique.
- (iii) Sur A^* , l'ordre lexicographique (A est un alphabet totalement ordonné)

Correction. On a :

Sur $A \times B$, l'ordre produit $[(a, b) \leq (a', b') \Leftrightarrow (a, \leq_1 a') \wedge (b \leq_2 b')]$.

On suppose qu'il existe une suite décroissante dans $A \times B$:

$$(a_1, b_1) \geq (a_2, b_2) \geq \dots \geq (a_n, b_n) \geq \dots$$

Montrons qu'elle est stationnaire à partir d'un certain rang.

On a :

- $a_1 \geq a_2 \geq \dots \geq a_n \geq \dots$ une suite décroissante dans A
Or A est un ordre bien fondé, alors il existe $n_1 \in \mathbb{N}$ tel que $\forall n \geq n_1, a_n = a_{n_1}$
- $b_1 \geq b_2 \geq \dots \geq b_n \geq \dots$ une suite décroissante dans B
Or B est un ordre bien fondé, alors il existe $n_2 \in \mathbb{N}$ tel que $\forall n \geq n_2, b_n = b_{n_2}$

On pose alors $n_0 = \max(n_1, n_2)$, alors $\forall n \geq n_0, (a_n, b_n) = (a_{n_0}, b_{n_0}) = (a_{n_1}, b_{n_2})$

Ainsi, à partir d'un certain rang, la suite $(a_n, b_n)_{n \in \mathbb{N}}$ est stationnaire.

Donc l'ordre défini est bien fondé.

Sur $A \times B$, l'ordre lexicographique.

Montrons que l'ordre produit lexicographique sur $A \times B$ est bien fondé.

Rappel : Ordre lexicographique sur $A \times B$: $(a, b) < (a', b')$ si $a < a'$ ou $(a = a' \text{ et } b < b')$

Supposons qu'il existe dans $A \times B$ une suite strictement décroissante :

$$(a_1, b_1) \geq (a_2, b_2) \geq \dots \geq (a_n, b_n) \geq \dots$$

Montrons qu'une telle suite est stationnaire à partir d'un certain rang.

On a : $a_1 \geq a_2 \geq \dots \geq a_n \geq \dots$ une suite décroissante dans A

Donc il existe $n_1 \in \mathbb{N}$ tel que $\forall n \geq n_1, a_n = a_{n_1}$

$\forall n \geq n_1$, on a donc $(a_n, b_n) = (a_{n_1}, b_n)$

Comme $(a_n, b_n) \geq (a_{n+1}, b_{n+1})$ et $a_{n+1} = a_{n_1}$, alors on a $(a_{n_1}, b_n) \geq (a_{n_1}, b_{n+1})$

Ainsi, $b_n \geq b_{n+1}$

La suite $b_{n_1}, b_{n_1+1}, \dots$ est décroissante infinie.

Or B est un ordre bien fondé, alors il existe $n_2 \in \mathbb{N}$ tel que $\forall n \geq n_2, b_n = b_{n_2}$

On pose alors $n_0 = \max(n_1, n_2)$, on a alors $\forall n \geq n_0, (a_{n+1}, b_{n+1}) = (a_n, b_n)$ - la suite est stationnaire.

Donc c'est un ordre bien fondé.

Sur A^* , l'ordre lexicographique (A est un alphabet totalement ordonné)

Soit $A = \{a, b\}$, $a < b$

Rappel. L'ordre lexicographique sur A^* est défini par :

$\underbrace{x_1 \dots x_k \dots x_n}_m \preceq \underbrace{y_1 \dots y_k \dots y_m}_{m'}$ s'il existe $k \in \mathbb{N}$ tel que $x_1 = y_1, \dots, x_k = y_k$ et $x_{k+1} < y_{k+1}$ (où $n = k$ et $m > k$)

Contre-exemple : la suite $(a^n b)_{n \in \mathbb{N}}$ est infinie strictement décroissante.

$$ab \succeq aab \succeq aaab \succeq aaaab \succeq \dots$$

Ainsi, cet ordre n'est pas un ordre bien fondé.

Exercice (Fonction d'Ackerman). Soit $f: \mathbb{N} \rightarrow \mathbb{N}$ une fonction définie par :

- $f(0, n) = n + 1$
- $f(m, 0) = f(m - 1, 1)$
- $f(m, n) = f(m - 1, f(m, n - 1))$

Résoudre alors les questions suivantes :

- (i) Montrer que $f(m, n)$ est définie pour tout couple $(m, n) \in \mathbb{N}^2$
- (ii) Soit $g: \mathbb{N}^2 \rightarrow \mathbb{N}$ définie par :

- $g(0, n) = n + 1$
- $g(m, 0) = g(m - 1, 1)$
- $g(m, n) = g(m - 1, g(m, n + 1))$

Pour quels couples $(m, n) \in \mathbb{N}^2$ la fonction $g(m, n)$ est-elle définie ?

- (iii) Calculer $f(1, n)$, $f(2, n)$ et $f(3, n)$

Correction. On a :

Montrer que $f(m, n)$ est définie pour tout couple $(m, n) \in \mathbb{N}^2$

Considérons l'ordre lexicographique, un ordre bien fondé, total sur \mathbb{N}^2 .

$(a, b) \leq (a', b')$ si $(a < a') \vee ((a = a') \wedge (b \leq b'))$

Montrons alors que $f(m, n)$ est bien défini pour tout couple $(m, n) \in \mathbb{N}^2$ (ie. Il n'existe pas de suite infinie strictement décroissante).

Pour tout $(0, n) \in \mathbb{N}^2$, $f(n, m)$ est bien défini.

Soit $(m, n) \in \mathbb{N}^2$, $m \neq 0$

Supposons que pour tout $(a, b) \in \mathbb{N}^2$ tel que $(a, b) < (m, n)$, $f(a, b)$ est bien définie. Montrons alors que $f(m, n)$ est bien défini.

Cas 1. $m = 0$

$$f(m, 0) = f(m - 1, 1)$$

Or comme $(m - 1, 1) < (m, 0)$, alors par hypothèse d'induction, $f(m - 1, 1)$ est bien défini, alors $f(m, 0)$ est bien défini.

Cas 2. $n \neq 0$

$f(m, n)$ existe si, et seulement si, :

- $A = f(m, n - 1)$ existe
- $B = f(m - 1, A)$ existe.

Or on a $(m, n - 1) < (m, n)$. Donc par hypothèse d'induction, $f(m, n - 1)$ existe.

Comme $f(m, n - 1)$ existe, et $(m - 1, A) < (m, n)$, alors par hypothèse d'induction, $f(m - 1, A)$ existe

Et $f(m, n)$ existe.

Pour quels couples $(m, n) \in \mathbb{N}^2$ la fonction $g(m, n)$ est-elle définie ?

On a : $g(1, 0) = g(0, 1)$ est bien défini.

Or $g(1, 1) = g(0, g(1, 2))$ n'est pas défini car $g(1, 2)$ n'est pas défini.

De plus, $g(0, n) = n + 1$ est bien défini sur \mathbb{N}^2

Conclusion. La fonction g est définie pour les couples $(1, 0)$ et $(0, n)$

Calculer $f(1, n)$, $f(2, n)$ et $f(3, n)$

Montrons par induction sur n que $f(1, n) = n + 2$

- Base. Pour $n = 0$, on a : $f(1, 0) = f(0, 1) = 1 + 1 = 2 = 0 + 2$
- Induction. Supposons que pour un certain n , $f(1, n) = n + 2$.
Montrons alors $f(1, n + 1) = n + 3$
 $f(1, n + 1) = f(0, f(1, n)) = f(0, n + 2) = n + 3$ par hypothèse d'induction et par définition de f
- Conclusion. $f(1, n) = n + 2$

Montrons par induction que $f(2, n) = 2n + 3$

- Base. Pour $n = 0$, on a : $f(2, 0) = f(1, 1) = 1 + 2 = 3 = 2 \times 0 + 3$
- Induction. Supposons que pour un certain n , $f(2, n) = 2n + 3$.
Montrons alors $f(2, n + 1) = 2n + 5$
 $f(2, n + 1) = f(1, f(2, n)) = (2n + 3) + 2 = 2n + 5$ par hypothèse d'induction et par définition de $f(1, n)$
- Conclusion. $f(2, n) = 2n + 3$

Montrons par induction que $f(3, n) = 2^{n+3} - 3$

- Base. Pour $n = 0$, on a : $f(3, 0) = f(2, 1) = 2 + 3 = 5 = 2^{0+3} - 3$
- Induction. Supposons que pour un certain n , $f(3, n) = 2^{n+3} - 3$.
Montrons alors $f(3, n + 1) = 2^{n+4} - 3$
 $f(3, n + 1) = f(2, f(3, n)) = 2 \cdot (2^{n+3} - 3) + 3 = 2^{n+4} - 3$ par hypothèse d'induction et par définition de $f(2, n)$
- Conclusion. $f(3, n) = 2^{n+3} - 3$

Exercice . Soit $F_0 = \{a\}$, $F_1 = \{s\}$, $F = F_0 \cup F_1$. L'ensemble T des termes construits sur F est $T = \{a, s(a), s(s(a)), \dots\}$

Posons $V = \mathbb{N}$. Soit $h: F_0 \rightarrow V$, et $h_s: V \rightarrow V$; il existe une et une seule fonction h^* de T dans V telle que :

- Base. Si $t = a \in F_0$, alors $h^*(t) = h(t)$
- Induction. Si $t = s(t_1)$, alors $h^*(t) = h_s(h^*(t_1))$

Calculer h^* dans les cas suivants :

- (i) $h_1(a) = 0$ et $h_{1_s}(n) = n + 1$
- (ii) $h_2(a) = 1$ et $h_{2_s}(n) = 2n$
- (iii) $h_3(a) = 1$ et $h_{3_s}(n) = n + 2$

Correction. On a :

$$h_1(a) = 0 \text{ et } h_{1_s}(n) = n + 1$$

Montrer par récurrence sur n que $h^*(s^n(a)) = n$

- Base. Au rang $n = 1$, on a : $h^*(s(a)) = h_{1_s}(h^*(a)) = h(0) = 0 + 1 = 1$
- Induction. Soit $n \in \mathbb{N}$. Supposons la propriété vraie au rang n et montrons qu'elle est vraie au rang $n + 1$.
On a alors : $h^*(s^{n+1}(a)) = h^*(s(s^n(a))) = h_{1_s}(h^*(s^n(a))) = h_{1_s}(n)$ par hypothèse de récurrence.
Ainsi, on a $h^*(s^{n+1}(a)) = n + 1$
- Conclusion. $h^*(s^n(a)) = n$

$$h_2(a) = 1 \text{ et } h_{2_s}(n) = n + 2$$

Montrons par récurrence sur n que $h^*(s^n(a)) = 2n + 1$

- Base. Au rang $n = 1$, on a $h^*(s(a)) = h_{2_s}(h^*(a)) = h_{2_s}(1) = 1 + 2 = 3$
- Induction. Soit $n \in \mathbb{N}$. Supposons la propriété vraie au rang n et montrons qu'elle est vraie au rang $n + 1$.
On a alors : $h^*(s^{n+1}(a)) = h^*(s(s^n(a))) = h_{2_s}(h^*(s^n(a))) = h_{2_s}(2n + 1)$ par hypothèse de récurrence.
Ainsi, on a $h^*(s^{n+1}(a)) = 2n + 3$
- Conclusion. $h^*(s^n(a)) = 2n + 1$

$$h_3(a) = 1 \text{ et } h_{3_s}(n) = 2n$$

Montrons par récurrence sur n que $h^*(s^n(a)) = 2^n$

- Base. Au rang $n = 1$, on a $h^*(s(a)) = h_{3_s}(h^*(a)) = h_{3_s}(1) = 2 \times 1 = 2$
- Induction. Soit $n \in \mathbb{N}$. Supposons la propriété vraie au rang n et montrons qu'elle est vraie au rang $n + 1$.
On a alors : $h^*(s^{n+1}(a)) = h^*(s(s^n(a))) = h_{3_s}(h^*(s^n(a))) = h_{3_s}(2^n)$ par hypothèse de récurrence.
Ainsi, on a $h^*(s^{n+1}(a)) = 2 \times 2^n = 2^{n+1}$
- Conclusion. $h^*(s^n(a)) = 2^n$

2 Ensembles et ordre

Exercice . Calculer l'ensemble $\mathcal{P}(S)$ des parties de S pour :

(i) $S = \{1, 2, 3\}$

(ii) $S = \{1, \{1, 4\}\}$

Correction. On a :

$$S = \{1, 2, 3\}$$

$$\mathcal{P}(S) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$$

De plus, $\#S = |S| = 3$, donc $\#\mathcal{P}(S) = |\mathcal{P}(S)| = 2^3 = 8$

$$S = \{1, \{1, 4\}\}$$

$$\mathcal{P}(S) = \{\emptyset, \{1\}, \{\{1, 4\}\}, \{1, \{1, 4\}\}\}$$

De plus, $\#S = |S| = 2$, donc $\#\mathcal{P}(S) = |\mathcal{P}(S)| = 2^2 = 4$

Exercice . Soient A, B, C trois parties de E . Montrer que :

(i) $A \cap \bar{B} = A \cap \overline{A \cap B}$

(ii) $A \cap \bar{B} = A \cap \bar{C} \Leftrightarrow A \cap B = A \cap C$

Correction. On a :

$$A \cap \bar{B} = A \cap \overline{A \cap B}$$

$$\begin{aligned} A \cap \overline{A \cap B} &= A \cap (\bar{A} \cup \bar{B}) \text{ d'après les lois de Morgan} \\ &= (A \cap \bar{A}) \cup (A \cap \bar{B}) \text{ par distributivité} \\ A \cap \overline{A \cap B} &= A \cap \bar{B} \text{ car } A \cap \bar{A} = \emptyset \end{aligned}$$

$$A \cap \bar{B} = A \cap \bar{C} \Leftrightarrow A \cap B = A \cap C$$

D'après la question précédente, on a :

– (*) $A \cap \bar{B} = A \cap \overline{A \cap B}$

– (**) $A \cap \bar{C} = A \cap \overline{A \cap C}$

Supposons $A \cap B = A \cap C$

On a alors en injectant dans (*) : $A \cap \bar{B} = A \cap \overline{A \cap C}$

En injectant dans (**) : $A \cap \bar{C} = A \cap \overline{A \cap B}$

Or $A \cap \overline{A \cap B} = A \cap \overline{A \cap C}$ par hypothèse. Ainsi, $A \cap \bar{B} = A \cap \bar{C}$

Supposons $A \cap \bar{B} = A \cap \bar{C}$

On pose $B' = \bar{B}$ et $C' = \bar{C}$

On a montré que $A \cap B' = A \cap C'$

Ainsi, $A \cap B' = A \cap C' \Leftrightarrow A \cap \bar{B} = A \cap \bar{C} \Leftrightarrow A \cap B = A \cap C$ car $\bar{\bar{A}} = A$

Exercice . On a :

- (i) Trouver un exemple d'application qui n'est bijective, ni surjective.
- (ii) Soit $f: A \rightarrow B$ une application. Montrer que

$$f \text{ injective} \Leftrightarrow \forall X, Y \subset A, f(X \cap Y) = f(X) \cap f(Y)$$

Correction. On a :

Trouver un exemple d'application qui n'est bijective, ni surjective.

L'application constante f définie par : $f: \mathbb{N} \rightarrow \mathbb{N}$ n'est ni injective, ni surjective.

$$n \mapsto 0$$

$$f \text{ injective} \Leftrightarrow \forall X, Y \subset A, f(X \cap Y) = f(X) \cap f(Y)$$

Montrons l'équivalence dans les deux sens.

$(\Rightarrow) \forall x \in X \cap Y, x \in X \text{ et } x \in Y, \text{ donc } f(x) \in f(X) \text{ et } f(x) \in f(Y)$

Ainsi, $f(x) \in f(X) \cap f(Y)$

Donc $f(X \cap Y) \subset f(X) \cap f(Y)$

Supposons f injective. On veut montrer $f(X) \cap f(Y) \subset f(X \cap Y)$

Soit $z \in f(X) \cap f(Y)$

Il existe $x \in X$ et $y \in Y$ tels que $z = f(x)$ et $z = f(y)$, ainsi, $f(x) = f(y)$

Or f est injective, donc $f(x) = f(y) \Rightarrow x = y$ par définition.

Donc $x \in X \cap Y$ et $z = f(x) \in f(X \cap Y)$

Ainsi, $f(X) \cap f(Y) \subset f(X \cap Y)$

Par double inclusion, on a donc $f(X \cap Y) = f(X) \cap f(Y)$

(\Leftarrow) Supposons $f(X \cap Y) = f(X) \cap f(Y)$.

Soit $x, y \in A$ tels que $f(x) = f(y)$

Posons $X = \{x\}$ et $Y = \{y\}$. On a $f(X) = f(Y)$

De plus, $f(X) \cap f(Y) = f(X \cap Y) = f(X) = f(Y)$ par hypothèse

On a $f(X \cap Y) = f(X) \cap f(Y) = f(X) = \{f(x)\} \neq \emptyset$

Ainsi, $X \cap Y \neq \emptyset$ et $x = y$

Donc f est injective

Exercice . Montrer que $(\mathbb{N}, +, 0)$ et $(\mathcal{P}(E), \cup, \emptyset)$ sont des monoïdes commutatifs.

Correction. On a :

$(\mathbb{N}, +, 0)$

Montrons que la loi interne $+$ est :

- Associative : $\forall x, y, z \in \mathbb{N}, (x + y) + z = x + (y + z)$
- Admet un élément neutre : soit $e = 0$, alors $\forall x \in \mathbb{N}, x + e = e + x = x$
- Commutative : $\forall x, y \in \mathbb{N}, x + y = y + x$

$(\mathcal{P}(E), \cup, \emptyset)$

Montrons que la loi interne \cup est :

- Associative : $\forall X, Y, Z \in \mathcal{P}(E), (X \cup Y) \cup Z = X \cup (Y \cup Z)$
- Admet un élément neutre : soit $\varepsilon = \emptyset$, alors $\forall X \in \mathcal{P}(E), X \cup \varepsilon = \varepsilon \cup X = X$

- Commutative : $\forall X, Y \in \mathcal{P}(E), X \cup Y = Y \cup X$

Remarque. $(\mathcal{P}(E), \cap, E)$ est un monoïde commutatif.

Exercice . A étant un alphabet, l'ensemble des mots de A^* de longueur paire est-il un monoïde ?

Correction. Soit $\mathcal{B} = \{\text{mots de longueur paire sur } A\}$ et soit $.$ l'opération concaténation.

Montrons que $(\mathcal{B}, .)$ est un monoïde :

- Associativité : $\forall a, b, c \in \mathcal{B}, a.(b.c) = (a.b).c$
- Loi interne à \mathcal{B} : soit $m = u.v$ avec $|u| = 2k$ et $|v| = 2k', k, k' \in \mathbb{N}$
Alors $|m| = 2(k + k')$
- Élément neutre : Soit $\varepsilon \in \mathcal{B}$ tel que $|\varepsilon| = 0$

Exercice . Considérer la relation $\mathcal{R} = \{(1, 1), (2, 3), (3, 2)\}$ sur $X = \{1, 2, 3\}$.

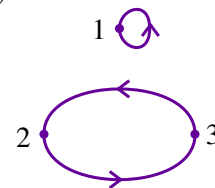
Déterminer si \mathcal{R} est :

- Réflexive
- Symétrique
- Transitive

Correction. La relation \mathcal{R} peut être représentée par un graphe (ci-contre).

On a :

- Réflexive : non car $(2, 2) \notin \mathcal{R}$ et $(3, 3) \notin \mathcal{R}$
- Symétrique : oui car $\forall (x, y) \in \mathcal{R}, (y, x) \in \mathcal{R}$
- Transitive : non car si $(2, 3) \in \mathcal{R}$ et $(3, 2) \in \mathcal{R}$ mais $(2, 2) \notin \mathcal{R}$



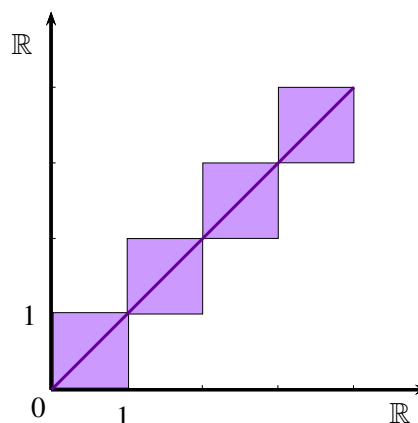
Exercice . Soit T la relation sur l'ensemble des réels \mathbb{R} défini comme suit :

$$xTy \Leftrightarrow \exists n \in \mathbb{N}, x \in [n, n+1] \text{ et } y \in [n, n+1]$$

Tracer le graphe de la relation T .

Correction. Cherchons les propriétés de T .

- T est réflexive : $\forall x \in \mathbb{R}, xTx$
- Si $x \in [0, 1], y \in [0, 1]$ alors $0Ty$ et $0Tx$
- $1Ty$ pour $y \in [0, 1] \cup [1, 2]$



Le graphe de T est présenté ci-contre :

Exercice . La relation \mathcal{R} sur \mathbb{N} définie par $n\mathcal{R}m \Leftrightarrow m = n + 1$ est-elle symétrique ? Réflexive ? Transitive ? Quelles sont les relations \mathcal{R}^+ et \mathcal{R}^* ?

Correction. On a :

- La relation \mathcal{R} n'est pas symétrique : $3\mathcal{R}4$, mais $4\not\mathcal{R}3$
- La relation \mathcal{R} n'est pas réflexive : $3\not\mathcal{R}3$
- La relation n'est pas transitive : Si $3\mathcal{R}4$ et $4\mathcal{R}5$, alors $3\mathcal{R}5$

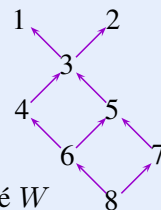
Rappels :

- \mathcal{R}^0 est défini par : $\mathcal{R}^0 = \{(x, x) \in \mathcal{R}, x \in E\}$
- \mathcal{R}^2 est défini par : $\mathcal{R}^2 = \mathcal{R} \cdot \mathcal{R} = \{(x, y) \in E^2, \exists z \in E, x\mathcal{R}z \text{ et } z\mathcal{R}y\}$
- \mathcal{R}^n est défini inductivement par : $\mathcal{R}^n = \{(x, y) \in E^2, \exists z \in E, x\mathcal{R}^{n-1}z \text{ et } z\mathcal{R}^{n-1}y\}$
- \mathcal{R}^* est défini par : $\mathcal{R}^* = \bigcup_{n \in \mathbb{N}} \mathcal{R}^n$
- \mathcal{R}^+ est défini par : $\mathcal{R}^+ = \bigcup_{n \in \mathbb{N}^*} \mathcal{R}^n$

Ainsi, $n\mathcal{R}^+m$ si, et seulement si, $m > n$ et $n\mathcal{R}^*m$ si, et seulement si, $m \geq n$

Exercice . Soit $W = \{1, 2, \dots, 7, 8\}$ ordonné comme dans la figure ci-contre par un ordre large (ie. $\forall v \in W, v \leq v$). On considère le sous-ensemble $V = \{4, 5, 6\}$, puis les sous-ensemble $V = \{1, 3, 4, 5\}$ et $V = \{4, 7\}$.

- Trouver l'ensemble des majorants de V
- Trouver l'ensemble des minorants de V
- Est-ce que $\sup(V)$ existe ?
- Est-ce que $\inf(V)$ existe ?



Représentation de l'ensemble ordonné W

Correction. On a :

$$V = \{4, 5, 6\}$$

- $\text{Maj}(V) = \{1, 2, 3\}$
- $\text{Min}(V) = \{6, 8\}$
- $\sup(V)$ existe et $\sup(V) = 3$
- $\inf(V)$ existe et $\inf(V) = 6$
- 4 et 5 sont maximaux
- 6 est minimal

$$V = \{1, 3, 4, 5\}$$

- $\text{Maj}(V) = \{1\}$
- $\text{Min}(V) = \{6, 8\}$
- $\sup(V)$ existe et $\sup(V) = 1$
- $\inf(V)$ existe et $\inf(V) = 6$
- 1 est maximal
- 4 et 5 sont minimaux

$$V = \{4, 7\}$$

- $\text{Maj}(V) = \{1, 2, 3\}$
- $\text{Min}(V) = \{8\}$
- $\sup(V)$ existe et $\sup(V) = 3$
- $\inf(V)$ existe et $\inf(V) = 8$
- 4 est maximal
- 7 est minimal

Exercice . Soit $A = \{1, 2, 3, 4, \dots\} = \mathbb{N} \setminus \{0\}$. A est ordonnée par la relation « x divise y ».

- (i) Vérifier que cette relation est un ordre.
- (ii) Déterminer les éléments minimaux et maximaux de $A \setminus \{1\}$
- (iii) Existe-t-il une borne sup et une borne inf pour tout sous-ensemble de 2 éléments ?
- (iv) Soient les ensembles $A = \{6, 15, 21\}$, $B = \{1, 6, 14, 21\}$ et $C = \{3, 6, 12, 15\}$. Donner les majorants, minorant, borne sup, borne inf, éléments minimaux, maximaux de A , B et C .
- (v) On se place dans \mathbb{N} ordonné par la relation « x divise y » avec la convention que 0 divise 0. Existe-t-il une borne sup et une borne inf pour tout sous-ensemble de 2 éléments ? Existe-t-il un minimum, un maximum ?

Correction. On a :

Vérifier que cette relation est un ordre.

- Réflexive : $\forall x \in A, x|x$
- Antisymétrique : $\forall x, y \in A, x|y \text{ et } y|x \Rightarrow \begin{cases} x = ky \\ y = k'x \end{cases} \Rightarrow y = kk'y \Rightarrow kk' = 1 \Rightarrow k = k' = 1$
Et $x = y$
- Transitive : $\forall x, y, z \in A, x|y \text{ et } y|z \Rightarrow \begin{cases} y = kx \\ z = k'y \end{cases} \Rightarrow z = kk'x$
Et $x|z$

Déterminer les éléments minimaux et maximaux de $A \setminus \{1\}$

Les éléments minimaux de A sont les nombres premiers

Les éléments maximaux de A sont $\forall n \in A, n|2n$

Existe-t-il une borne sup et une borne inf pour tout sous-ensemble de 2 éléments ?

Soit $\{x, y\} \in A$, alors : $\sup(\{x, y\}) = \text{ppcm}(x, y)$ et $\inf(\{x, y\}) = \text{pgcd}(x, y)$

Soient les ensembles $A = \{6, 15, 21\}$, $B = \{1, 6, 14, 21\}$ et $C = \{3, 6, 12, 15\}$. Donner les majorants, minorant, borne sup, borne inf, éléments minimaux, maximaux de A , B et C .

$A = \{6, 15, 21\}$	– $\text{Min}(B) = \{1\}$	– $\text{Min}(C) = \{1, 3\}$
– $\text{Min}(A) = \{1, 3\}$	– $\text{Maj}(B) = \{42k, k \in A\}$	– $\text{Maj}(C) = \{60k, k \in A\}$
– $\text{Maj}(A) = \{210k, k \in A\}$	– $\sup(B) = 42$ et $\inf(B) =$	– $\sup(C) = 60$ et $\inf(C) =$
– $\sup(A) = 210$ et $\inf(A) =$	1	3
3	– $\min(B) = 1, \max(B)$	– $\min(C) = 3, \max(C)$
– $\min(A)$ et $\max(A)$	n'existe pas	n'existe pas
n'existent pas	– Minimal : 1. Maximaux : 6,	– Minimal : 3. Maximaux :
– Minimaux : non. Maxi-	14, 21	12, 15
maux : 6, 15, 21	$C = \{3, 6, 12, 15\}$	

On se place dans \mathbb{N} ordonné par la relation « x divise y » avec la convention que 0 divise 0. Existe-t-il une borne sup et une borne inf pour tout sous-ensemble de 2 éléments ? Existe-t-il un minimum, un maximum ?

Rappel. $0|y$ si $y = k \cdot 0 : 0$ ne divise que 0

$y|0$ si $0 = k \cdot y$ vrai si $k = 0$: tout nombre divise 0

Il faut distinguer plusieurs cas :

- $x \neq y$, alors $\sup(\{x, y\}) = \text{ppcm}(x, y)$ et $\inf(\{x, y\}) = \text{pgcd}(x, y)$
- $x = y = 0$, alors $\sup(\{0\}) = 0$ et $\inf(\{0\}) = 0$
- $x = 0$ et $y \neq 0$, alors $\text{Min}(\{0, y\}) = \{y\}$ car $0 \nmid y$, ainsi, $\inf(\{0, y\}) = y = \min(\{0, y\})$
 $\text{Maj}(\{0, y\}) = 0$, ainsi, $\sup(\{0, y\}) = 0 = \max(\{0, y\})$

Exercice . Soient A et B deux parties disjointes d'un ensemble E . Réaliser une bijection entre $\mathcal{P}(A) \times \mathcal{P}(B)$ et $\mathcal{P}(A \cup B)$.

Correction. Soit l'application $f: \mathcal{P}(A) \times \mathcal{P}(B) \longrightarrow \mathcal{P}(A \cup B)$.
 $(C, C') \longmapsto C \cup C' = f(C, C')$

Montrons que f est injective.

Soient $(C, C'), (D, D') \in \mathcal{P}(A) \times \mathcal{P}(B)$. On suppose que $f(C, C') = f(D, D')$. Montrons que $(C, C') = (D, D')$.

Par hypothèse, $C \cup C' = D \cup D'$.

Or $C = (C \cup C') \cap A$ car $C' \cap A = \emptyset$ (en effet, $A \cap B = \emptyset$)

Et $D = (D \cup D') \cap A$

Donc $C = D$

De même, $C' = (C \cup C') \cap B$ car $C \cap B = \emptyset$

Et $D' = (D \cup D') \cap B$

Donc $C' = D'$

Ainsi, $(C, C') = (D, D')$. Donc f est injective.

Montrons que f est surjective.

Soit $X \in \mathcal{P}(A \cup B)$. On montre qu'il existe $(C, C') \in \mathcal{P}(A) \times \mathcal{P}(B)$ tel que $X = f(C, C')$

On pose $C = A \cap X$ et $C' = B \cap X$.

On a alors :

$$\begin{aligned} C \cup C' &= (A \cap X) \cup (B \cap X) \\ &= [A \cup (B \cap X)] \cap [X \cup (B \cap X)] \\ &= (A \cup B) \cap (A \cup X) \cap (X \cup B) \cap (X \cup X) \\ C \cup C' &= X \text{ car } X \subset A \cup B, X \subset A \cup X \text{ et } X \subset B \cup X \end{aligned}$$

Ainsi, f est surjective, donc f est bijective.

3 Logique

Exercice . Résoudre les trois questions de cet exercice :

- (i) Montrer que la relation $x \leq y$ si, et seulement si, $x \cup y = y$ est une relation d'ordre sur une algèbre de Boole.
- (ii) Montrer qu'un homomorphisme est une application monotone par l'ordre sous-jacent à l'algèbre de Boole définie dans la question 1.
- (iii) Montrer que toute application monotone n'est pas forcément un homomorphisme.

Correction. On a :

Montrer que la relation $x \leq y$ si, et seulement si, $x \cup y = y$ est une relation d'ordre sur une algèbre de Boole.

Montrons que cette relation est :

- Réflexive : $\forall x \in \mathcal{B}, x \cup x = x \Rightarrow x \leq x$
- Antisymétrique : $\forall x, y \in \mathcal{B}, (x \leq y \text{ et } y \leq x) \Rightarrow x = y$
Or $x \cup y = y$ et $y \cup x = x$, et par commutativité, $x = y$
- Transitivité : $\forall x, y, z \in \mathcal{B}, (x \leq y \text{ et } y \leq z) \Rightarrow x \leq z$
Soient $x \cup y = y$ et $y \cup z = z$
Alors $x \cup z = x \cup (y \cup z) = (x \cup y) \cup z = y \cup z = z$ par associativité.

Montrer qu'un homomorphisme est une application monotone par l'ordre sous-jacent à l'algèbre de Boole définie dans la question 1.

Soit $h: \mathcal{B} \rightarrow \mathcal{B}'$ un homomorphisme défini par :

- $h(x \cap y) = h(x) \cap' h(y)$
- $h(x \cup y) = h(x) \cup' h(y)$
- $h(\perp) = \perp'$
- $h(\top) = \top'$
- $h(\bar{x}) = \bar{x}'$

$h: (\mathcal{B}, \leq) \rightarrow (\mathcal{B}', \leq')$ est monotone si $\forall x, y \in \mathcal{B}, x \leq y \Rightarrow h(x) \leq' h(y)$

On suppose h un homomorphisme. Montrons que h est pour cet ordre.

Soient $x, y \in \mathcal{B}$ tels que $x \leq y$.

$$x \cup y = y$$

$$h(x \cup y) = \underbrace{h(x) \cup' h(y)}_{\text{def. : } h(x) \leq' h(y)} = h(y)$$

Ainsi, h est monotone.

Montrer que toute application monotone n'est pas forcément un homomorphisme.

On cherche h monotone qui ne soit pas un homomorphisme.

h vérifie alors $\forall x, y \in \mathcal{B}, x \leq y \Rightarrow h(x) \leq' h(y)$

$$\forall x \in \mathcal{B}, h(x) = \perp'$$

h est constante, donc monotone, mais $h(\top) = \perp' \neq \top'$

Soit $x \neq \perp$, alors $h(x \cap \bar{x}) = h(\perp) = \perp'$

$$h(x) \cap h(\bar{x}) = \top' \cap \top' = \top' \neq \perp'$$

Ainsi, h ainsi définie n'est pas un homomorphisme.

Exercice . Trouver un polynôme booléen pour la fonction f définie par :

f	0	1
0	1	0
1	0	0

Quelle est la table de vérité de la fonction duale \tilde{f} ? Trouver un polynôme pour la fonction duale \tilde{f} .

Correction. Table de vérité de \tilde{f} :

\tilde{f}	0	1
0	1	1
1	1	0

Polynôme booléen de f : $f(x, y) = \bar{x}.\bar{y}$

Polynôme booléen de \tilde{f} : $\tilde{f}(x, y) = \bar{x} + \bar{y}$

Exercice . Ecrire les fonctions duales de :

- $x.\bar{y}.\bar{y}$
- $x.y.z + t$
- $(p \vee ff) \wedge (q \vee ff)$

Correction. On a :

- $\widetilde{x.\bar{y}.\bar{z}} = \overline{\bar{x}.\bar{y}.\bar{z}} = x + \bar{y} + \bar{z}$
- $\widetilde{x.y.z + t} = \overline{\bar{x}.\bar{y}.\bar{z} + \bar{t}} = (x + y + z) . t$
- $\widetilde{(p \vee ff) \wedge (q \vee ff)} = (t + ff) . (q + ff) = p.ff + q.ff = (p \wedge ff) \vee (q \wedge ff)$

Exercice . Ecrire la table de vérité et la fonction duale de $f(x, y) = x.y + \bar{x}.\bar{y}$

Correction. Calcul de la fonction duale \tilde{f} :

$$\tilde{f}(x, y) = \overline{f(\bar{x}, \bar{y})} = \overline{\bar{x}.\bar{y} + \bar{\bar{x}}.\bar{\bar{y}}} = (\bar{x} + \bar{y}) . (x + y) = \bar{x}.x + \bar{x}.y + \bar{y}.x + \bar{y}.y = \bar{x}.y + \bar{y}.x$$

Table de vérité de f :

f	0	1
0	0	1
1	1	0

Table de vérité de \tilde{f} :

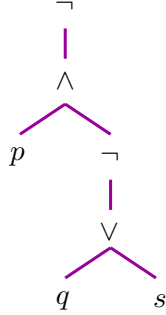
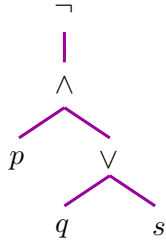
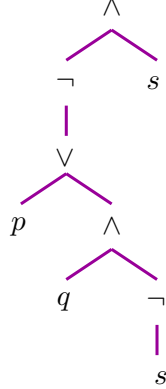
\tilde{f}	0	1
0	1	0
1	0	1

Exercice . Dessiner les arbres correspondant aux formules suivantes :

$$F = \neg(p \wedge \neg(q \vee s)) \quad G = \neg(p \wedge (q \vee s)) \quad H = \neg(p \vee (q \wedge \neg s)) \wedge s$$

Mettre en forme normale disjonctive et conjonctives les formules précédentes.

Correction. On a :

Formule	$F = \neg(p \wedge \neg(q \vee s))$	$G = \neg(p \wedge (q \vee s))$	$H = \neg(p \vee (q \wedge \neg s)) \wedge s$
Arbre			
FNC	$(\neg p \vee q \vee s)$	$(\neg p \vee \neg q) \wedge (\neg p \vee \neg s)$	$\neg p \wedge (\neg q \vee s) \wedge s$
FND	$(\neg p) \vee q \vee s$	$\neg p \vee (\neg q \wedge \neg s)$	$(\neg p \wedge \neg q \wedge \neg s) \vee (\neg p \wedge s)$

Exercice . Soient p, q, r des propositions. Exprimer par des formules les expressions du tableau.

Correction. On a :

Si p , alors q	$p \rightarrow q$	$\neg p \vee q$
Si p , alors q , sinon r	$(p \rightarrow q) \wedge (\neg p \rightarrow r)$	$(p \wedge q) \vee (\neg p \wedge r)$
p est une condition nécessaire pour que q soit vraie	$q \rightarrow p$	$\neg p \vee q$
p est une condition suffisante pour que q soit vraie	$p \rightarrow q$	$\neg p \vee q$
q si p	$p \rightarrow q$	$\neg p \vee q$
q seulement si p	$q \rightarrow p$	$\neg q \vee p$
q si, et seulement si, p	$p \equiv q$	$(\neg p \wedge \neg q) \vee (q \wedge p)$

Exercice . Trouver un polynôme booléen pour la fonction f définie par :

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Quelle est la table de vérité de la fonction duale?
Trouver un polynôme booléen pour la fonction duale.

Correction. On a :

$$\begin{aligned}
 f(x_1, x_2, x_3) &= (x_1 + x_2 + x_3) \cdot (\bar{x}_1 + x_2 + x_3) \cdot (\bar{x}_1 + \bar{x}_2 + x_3) \cdot (\bar{x}_1 + \bar{x}_2 + \bar{x}_3) \\
 &= \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 + \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot x_2 \cdot x_3 + x_1 \cdot \bar{x}_2 \cdot x_3 \\
 &= \bar{x}_1 \cdot x_2 \cdot \underbrace{(\bar{x}_3 + x_3)}_1 + \bar{x}_2 \cdot x_3 \cdot \underbrace{(\bar{x}_1 + x_1)}_1 \\
 f(x_1, x_2, x_3) &= \bar{x}_1 \cdot x_2 + \bar{x}_2 \cdot x_3
 \end{aligned}$$

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	$\tilde{f}(x_1, x_2, x_3)$	x_1	x_2	x_3	$\tilde{f}(x_1, x_2, x_3)$
0	0	0	0	1	1	1	1	1
0	0	1	1	1	1	1	0	0
0	1	0	1	0	1	0	1	0
0	1	1	1	1	1	0	0	0
1	0	0	0	0	0	1	1	1
1	0	1	1	0	0	1	0	0
1	1	0	0	0	0	0	1	1
1	1	1	0	1	0	0	0	1

La fonction duale de f , notée \tilde{f} est alors définie par :

$$\begin{aligned}
\tilde{f}(x_1, x_2, x_3) &= \overline{f(\bar{x}_1, \bar{x}_2, \bar{x}_3)} \\
&= \overline{(\bar{x}_1 + \bar{x}_2 + \bar{x}_3) \cdot (\bar{\bar{x}}_1 + \bar{\bar{x}}_2 + \bar{\bar{x}}_3) \cdot (\bar{\bar{\bar{x}}}_1 + \bar{\bar{\bar{x}}}_2 + \bar{\bar{\bar{x}}}_3)} \\
&= \overline{(\bar{x}_1 + \bar{x}_2 + \bar{x}_3) + (\bar{x}_1 + \bar{x}_2 + \bar{x}_3) + (\bar{x}_1 + \bar{x}_2 + \bar{x}_3) + (\bar{x}_1 + \bar{x}_2 + \bar{x}_3)} \\
&= \bar{x}_1.\bar{x}_2.\bar{x}_3 + \bar{x}_1.\bar{\bar{x}}_2.\bar{\bar{x}}_3 + \bar{x}_1.\bar{x}_2.\bar{\bar{x}}_3 + \bar{x}_1.\bar{\bar{x}}_2.\bar{x}_3 \\
&= x_1.x_2.x_3 + \bar{x}_1.x_2.x_3 + \bar{x}_1.\bar{x}_2.x_3 + \bar{x}_1.\bar{\bar{x}}_2.\bar{x}_3 \\
&= x_2.x_3.(x_1 + \bar{x}_1) + \bar{x}_1.\bar{x}_2.(x_3 + \bar{x}_3) \\
\tilde{f}(x_1, x_2, x_3) &= x_2x_3 + \bar{x}_1.\bar{x}_2
\end{aligned}$$

Exercice . On considère un langage avec le prédicat unaire p et les prédicats binaires r et s . Déterminer les variables libres et liées dans les formules du tableau.

Correction. On a le tableau suivant :

Formule F	$L(F)$	$B(F)$
$(\forall x p(x)) \vee (\exists y p(y))$	\emptyset	$\{x, y\}$
$(\forall x p(x)) \vee (\exists x p(x))$	\emptyset	$\{x\}$
$(\forall x p(x)) \wedge (\exists y r(x, y))$	$\{x\}$	$\{y\}$
$(\forall x \exists y s(x, y)) \rightarrow (\exists y \forall x r(x, y))$	\emptyset	$\{x, y\}$
$(\forall x s(x, y)) \rightarrow (\exists y r(x, y))$	$\{x, y\}$	\emptyset

Exercice . Ecrire, en utilisant les symboles de prédicats du monde de Tarski, des formules traduisant les énoncés du tableau.

Sans les quantificateurs.

Correction. On a alors le tableau suivant :

1. e et d sont entre b et a Between(e, b, a) \wedge Between(d, b, a)
2. Ni e , ni d ne sont entre b et a \neg Between(e, b, a) \wedge \neg Between(d, b, a)
3. Il est faux que e et d sont entre b et a \neg (Between(e, b, a) \wedge Between(d, b, a)) = \neg Between(e, b, a) \vee \neg Between(d, b, a)
4. a est petit ou c et d sont grand Small(a) \vee (Large(c) \wedge Large(d)) = (Small(a) \wedge Large(c)) \vee (Small(a) \wedge Large(c))
5. Soit a et e sont des cubes, soit a et f sont des cubes (Cube(a) \wedge Cube(e)) \vee (Cube(a) \wedge Cube(f))

Exercice . Ecrire, en utilisant les symboles de prédicats du monde de Tarski, des formules traduisant les énoncés du tableau.

Avec les quantificateurs.

Correction. On a alors le tableau suivant :

1. <i>Un dodécaèdre est grand</i>
<i>Tout dodécaèdre est grand</i> $\forall x (\text{Dodec}(x) \rightarrow \text{Large}(x))$
<i>Il existe (au moins) un grand dodécaèdre</i> $\exists x (\text{Dodec}(x) \wedge \text{Large}(x))$
2. <i>Tous les dodécaèdres sont grands</i> $\forall x (\text{Dodec}(x) \rightarrow \text{Large}(x))$
3. <i>Tous les dodécaèdres ne sont pas grands</i>
<i>Chacun des dodécaèdres n'est pas grand</i> $\forall x (\text{Dodec}(x) \rightarrow \neg \text{Large}(x))$
<i>L'ensemble des dodécaèdres n'est pas inclus dans l'ensemble des grands éléments</i> $\neg (\forall x (\text{Dodec}(x) \rightarrow \text{Large}(x))) = \exists x \neg (\text{Dodec}(x) \rightarrow \text{Large}(x))$ $= \exists x (\text{Dodec}(x) \vee \neg \text{Large}(x))$
4. <i>Il y a au moins deux cubes</i> $F_1 = \exists x \exists y (\text{Cube}(x) \wedge \text{Cube}(y) \wedge \neg (x = y))$
5. <i>Il y a au plus deux cubes</i> $F_2 = \forall x \forall y \forall z (\text{Cube}(x) \wedge \text{Cube}(y) \wedge \neg (x = y) \wedge \underbrace{(\text{Cube}(z) \rightarrow ((x = z) \vee (y = z)))}_{\neg \text{Cube}(z) \vee (x = z) \vee (y = z)})$ $F_2 = \neg (\exists x \exists y \exists z (\neg (x = y) \wedge \neg (x = z) \wedge \neg (y = z) \wedge (\text{Cube}(x) \wedge \text{Cube}(y) \wedge \text{Cube}(z))))$
6. <i>Il y a exactement deux cubes</i> $F_1 \wedge F_2$

Exercice . On se donne un langage comprenant deux symboles de prédicats (ou relations) binaires R et $=$ (= sera toujours interprété comme l'égalité).

- Ecrivez une formule du calcul des prédicats exprimant que la relation binaire R est une relation d'ordre large.
- Ecrivez une formule du calcul des prédicats exprimant que l'ordre R a un minimum.
- Ecrivez une formule du calcul des prédicats exprimant que l'ordre R a un élément minimal.

Correction. On a :

- R est un ordre large. Ainsi, R est :

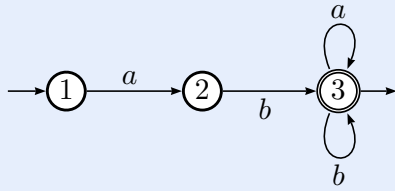
- Réflexive : $F_1 : \forall x R(x, x)$
- Transitive : $F_2 : \forall x \forall y \forall z (R(x, y) \wedge R(y, z) \rightarrow R(x, z))$
- Antisymétrique : $F_3 : \forall x \forall y (R(x, y) \wedge R(y, x) \rightarrow (x = y))$

Ainsi, R est un ordre large si $F_1 \wedge F_2 \wedge F_3$

- R un minimum si $\exists y \forall x R(y, x)$
- R a un élément minimal si $\exists y \forall x (R(x, y) \rightarrow (x = y))$

4 Automates

Exercice . Expliquez pourquoi l'automate suivant sur $\{a, b\}$ n'est pas complet. Quel langage reconnaît-il ? Donner un automate complet équivalent.



Correction. On rappelle la définition d'un automate complet.

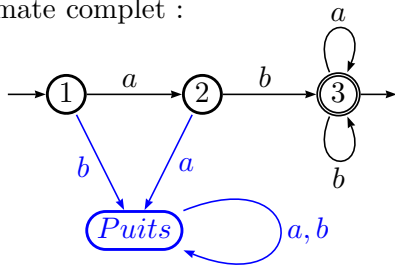
Un automate est complet si de chaque sommet, pour chaque lettre a , il existe une arrête étiquetée a issue de ce sommet.

Or pour $s = 1$, et la lettre b , il n'existe pas d'arrête étiquetée b issue de s . Autrement dit, il n'existe pas de transition (s, b, t) avec t sommet $\neq s$

Soit $L_{1,3}$ le langage reconnu par l'automate. On a alors :

$$L_{1,3} = ab(a+b)^* = ab\{a, b\}^* = \{\text{mots commençant par } ab\}$$

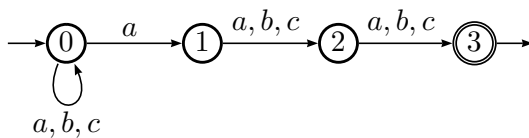
Automate complet :



Exercice . Représenter l'automate \mathcal{A} sur l'alphabet $\{a, b, c\}$ d'états 0, 1, 2, 3 où 0 est l'état initial et 3 l'état final. \mathcal{A} a pour transitions : $(0, a, 1)$, $(0, a, 0)$, $(0, b, 0)$, $(0, c, 0)$, $(1, a, 2)$, $(1, b, 2)$, $(1, c, 2)$, $(2, a, 3)$, $(2, b, 3)$, $(2, c, 3)$

- Cet automate est-il complet ?
- Les mots $baba$ et $cabcb$ sont-ils reconnus par \mathcal{A} ?
- Décrire $L(\mathcal{A})$ en langage ordinaire et en donner une expression rationnelle.

Correction. On a alors l'automate suivant :



Cet automate est non complet et non déterministe :

- Il n'y a en effet aucune transition partant de l'état 3.
- $(0, a, 1)$ et $(0, a, 0)$ sont deux transitions issues de l'état 0

Le mot $baba$ est reconnu par \mathcal{A} , ainsi, $baba \in L(\mathcal{A})$.

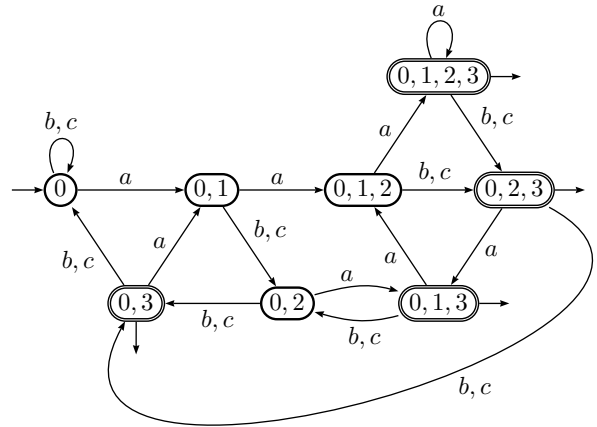
Le mot $cabcb$ n'est pas reconnu par \mathcal{A} , ainsi, $cabcb \notin L(\mathcal{A})$.

Le langage $L_{0,3}(\mathcal{A})$ reconnu par \mathcal{A} est exprimé par : $(a+b+c)^* a (a+b+c)^2$.

Ce sont les mots se terminant par a , suivi de 2 lettres quelconques.

Remarque. L'automate déterministe de l'automate \mathcal{A} est :

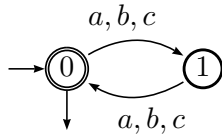
$\mathcal{P}(S)$	a	b	c
$\{0\}$	$\{0, 1\}$	$\{0\}$	$\{0\}$
$\{0, 1\}$	$\{0, 1, 2\}$	$\{0, 2\}$	$\{0, 2\}$
$\{0, 1, 2\}$	$\{0, 1, 2, 3\}$	$\{0, 2, 3\}$	$\{0, 2, 3\}$
$\{0, 2\}$	$\{0, 1, 3\}$	$\{0, 3\}$	$\{0, 3\}$
$\{0, 1, 2, 3\}$	$\{0, 1, 2, 3\}$	$\{0, 2, 3\}$	$\{0, 2, 3\}$
$\{0, 2, 3\}$	$\{0, 1, 3\}$	$\{0, 3\}$	$\{0, 3\}$
$\{0, 1, 3\}$	$\{0, 1, 2\}$	$\{0, 2\}$	$\{0, 2\}$
$\{0, 3\}$	$\{0, 1\}$	$\{0\}$	$\{0\}$



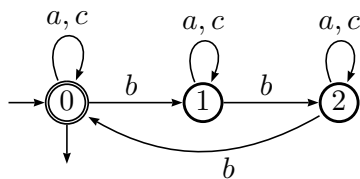
Exercice . Soit $A = \{a, b, c\}$. Donner des automates déterministes complets reconnaissant les langages suivants.

Correction. On a :

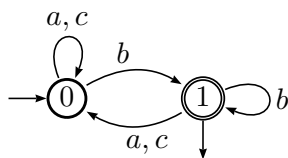
- 1) L'ensemble des mots de longueur paire.



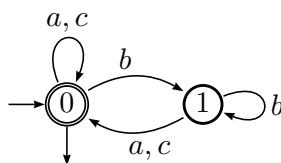
- 2) L'ensemble des mot où le nombre d'occurrence de b est divisible par 3.



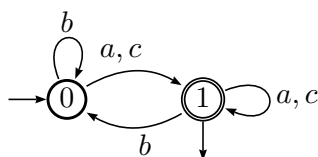
- 3) L'ensemble des mots se terminant par b .



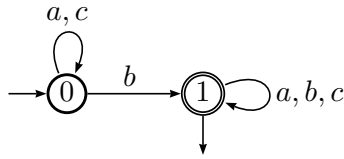
- 4) L'ensemble des mots ne se terminant pas par b .



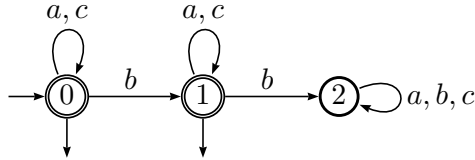
- 5) L'ensemble des mots non vides ne se terminant pas par b .



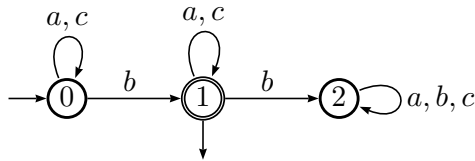
- 6) L'ensemble des mots contenant au moins un b .



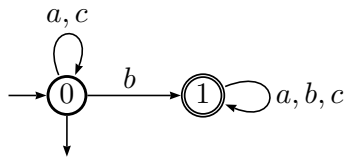
- 7) L'ensemble des mots contenant au plus un b .



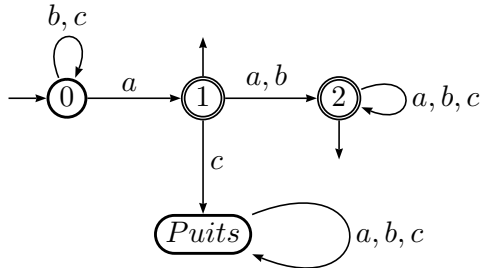
- 8) L'ensemble des mots contenant exactement un b .



- 9) L'ensemble des mots ne contenant aucun b .



- 10) L'ensemble des mots contenant au moins un a et dont la première occurrence de a n'est pas suivie de c .



Exercice . Soient L_1 et L_2 des langages sur un alphabet A . Montrer que si L_1 et L_2 sont respectivement reconnaissables par des automates \mathcal{A}_1 et \mathcal{A}_2 , alors le langage $L_1 \setminus L_2$ est reconnaissable par un automate.

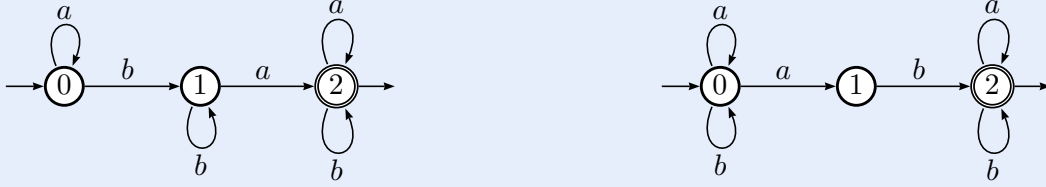
Correction. Par définition, on a $L_1 \setminus L_2 = \{m \in L_1 / m \notin L_2\} = L_1 \cap \mathcal{C}_{A^*}^{L_2} = L_1 \cap (A^* \setminus L_2)$

On sait alors construire un automate \mathcal{A}'_2 qui reconnaît $(A^* \setminus L_2)$ (cf. exercice précédent)

Or les automates \mathcal{A}_1 et \mathcal{A}'_2 sont déterministes complets. On sait également construire un automate déterministe complet reconnaissant $L(\mathcal{A}_1) \cap L(\mathcal{A}'_2) = L_1 \setminus L_2$

Exercice . Les automates \mathcal{A}_1 et \mathcal{A}_2 suivants sur $\{a, b\}$ sont-ils déterministes ? Expliquez pourquoi et si ce n'est pas le cas, déterminez-les.

Les automates \mathcal{A}_1 et \mathcal{A}_2 et les automates déterministes construits sont-ils complets ? Que remarquez-vous ?



Correction. L'automate \mathcal{A}_1 est déterministe et complet.

L'automate \mathcal{A}_2 n'est pas déterministe car pour tout état s , il n'existe pas au plus une arrête étiquetée a issue de s . Or dans l'état 1, on a : $(1, a, 1)$ et $(1, a, 2)$ deux transitions issues de l'état 1, d'étiquette a .

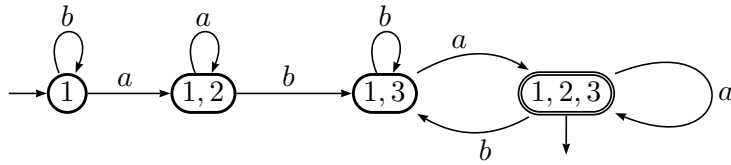
L'automate \mathcal{A}_2 n'est pas complet non plus car il n'y a pas de transition d'étiquette a à l'état 2.

Détermination de l'automate \mathcal{A}_2

On a : $L(\mathcal{A}_2) = \{\text{mots contenant le facteur } ab\}$

On a le tableau suivant :

$\mathcal{P}(S)$	a	b
$\{1\}$	$\{1, 2\}$	$\{1\}$
$\{1, 2\}$	$\{1, 2\}$	$\{1, 3\}$
$\{1, 3\}$	$\{1, 2, 3\}$	$\{1, 3\}$
$\{1, 2, 3\}$	$\{1, 2, 3\}$	$\{1, 3\}$



Exercice (Lemme de l'étoile). Soit L un langage reconnaissable par un automate fini.

Montrer qu'il existe un entier N_0 tel que pour tout mot $m \in L$ vérifiant $|m| \geq N_0$ (où $|m|$ est la longueur du mot m), on a $m = \alpha u \beta$ tel que :

- (i) $u \neq \varepsilon$
- (ii) $|u| < N_0$
- (iii) $\alpha u^* \beta \subset L$

Correction. Soit $m = x_1 \dots x_i x_{i+1} \dots x_j \dots x_p$ avec $p \geq N_0$ où N_0 est égal au nombre d'états de l'automate \mathcal{A} .

Dans T (ie. les transitions), on a : $(q_0, x_1, q_1), (q_1, x_2, q_2), \dots, (q_{p-1}, x_p, q_p)$ et l'état q_p est terminal (ie. $q_p \in F$).

Ainsi, $q_0 \dots q_p$ comporte $p + 1 > N_0$ éléments.

Donc il existe $i < j$ ($i \neq j$) tels que $q_i = q_j$

On choisit i et j de telle sorte que $0 < j - i < N_0$ (toujours possible)

Donc il y a dans l'automate \mathcal{A} une boucle $q_i q_{i+1} \dots q_{j-1} q_j$.

De ce fait, on a :

- $u = x_{i+1} \dots x_j$
- $\alpha = x_1 \dots x_i$
- $\beta = x_{j+1} \dots x_p$

De plus, les trois propriétés (i), (ii), (iii) sont vérifiées.

Exercice . Soit $A = \{a, b\}$. Soient les langages L_1 et L_2 tels que :

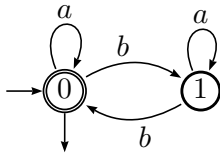
- L_1 est le langage comprenant tous les mots contenant un nombre pair de b .
- L_2 est le langage comprenant tous les mots contenant un nombre impair de a .

Résoudre alors :

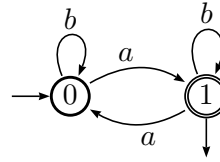
- (i) Donner pour chaque L_i un automate déterministe complet \mathcal{A}_i reconnaissant L_i et exprimer L_i sous forme d'expression rationnelle.
- (ii) Construire à partir des \mathcal{A}_i un automate déterministe reconnaissant $L_1 \cap L_2$.
- (iii) Construire à partir des \mathcal{A}_i un automate déterministe reconnaissant $L_1 \cup L_2$.

Correction. On a :

$$- L_1 = a^* (ba^*ba^*)^* = a^* ((ba^*)^2)^*$$

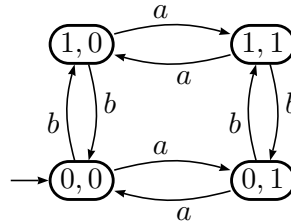


$$- L_2 = b^*a(b^*ab^*a)^*b^*$$



On a le tableau suivant :

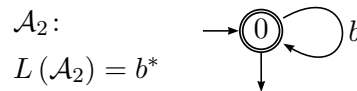
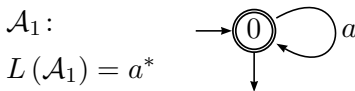
	a	b
$(0, 0)$	$(0, 1)$	$(1, 0)$
$(0, 1)$	$(0, 0)$	$(1, 1)$
$(1, 0)$	$(1, 1)$	$(0, 0)$
$(1, 1)$	$(1, 0)$	$(0, 1)$



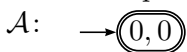
On détermine les états finaux :

- $L_1 \cap L_2$: $(0, 1)$
- $L_1 \cup L_2$: $(0, 0), (0, 1), (1, 1)$

Remarque. Pour l'union de deux langages, les automates considérés doivent être complets.
Contre-exemple : sur l'alphabet $\{a, b\}$.



Automate produit :



$$L(\mathcal{A}) = \emptyset$$

Il y a alors un problème de construction de l'automate produit \mathcal{A} .

En effet : $(0, 0) \xrightarrow{b} (\text{sink}, 0)$ et $(0, 0) \xrightarrow{a} (0, \text{sink})$

Exercice . Soit l'alphabet $A = \{a, b\}$ et soient :

- L_1 le langage comprenant tous les mots contenant "aa".
- L_2 le langage comprenant tous les mots ne contenant pas "bb".

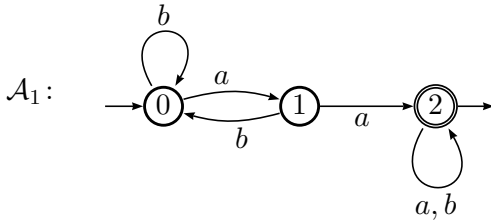
Résoudre les questions suivantes :

- Donner pour chaque L_i un automate déterministe complet \mathcal{A}_i reconnaissant L_i .
- Construire à partir des \mathcal{A}_i un automate déterministe et complet \mathcal{A} reconnaissant $L_1 \cap L_2$.
- Construire un automate minimal \mathcal{B} reconnaissant $L_1 \cap L_2$.

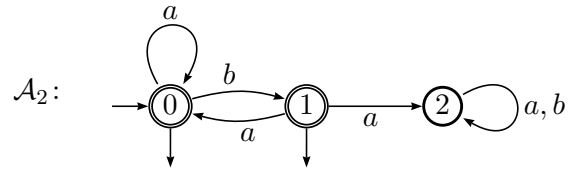
Correction. On a :

Donner pour chaque L_i un automate déterministe complet \mathcal{A}_i reconnaissant L_i .

$L_1 = \{\text{mots contenant } aa\}$

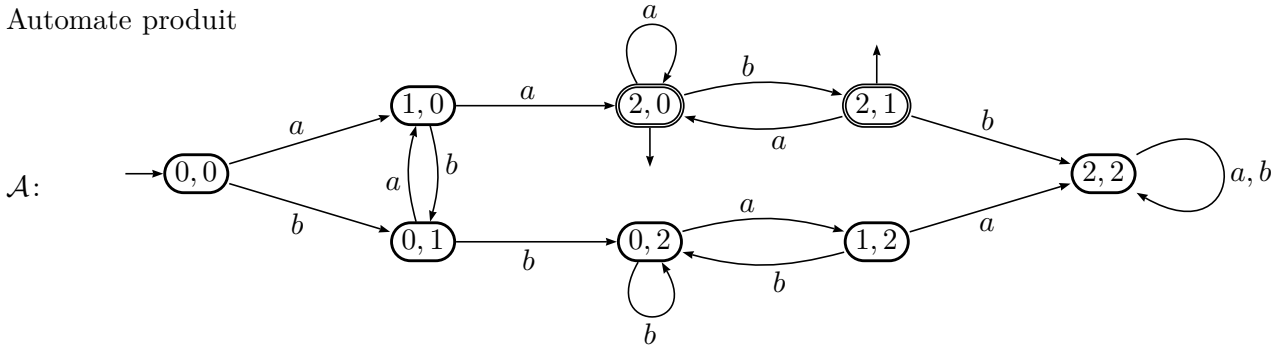


$L_2 = \{\text{mots ne contenant pas } bb\}$



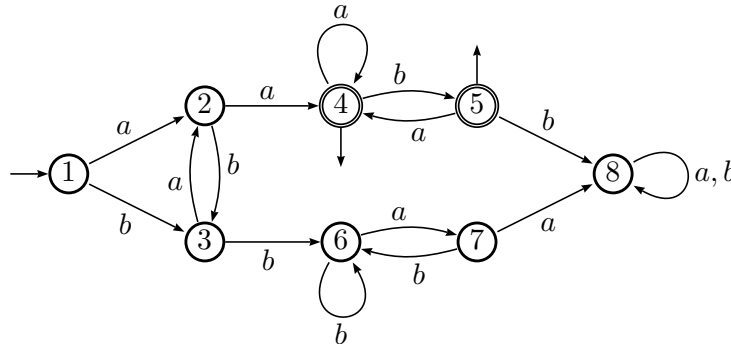
Construire à partir des \mathcal{A}_i un automate déterministe et complet \mathcal{A} reconnaissant $L_1 \cap L_2$.

Automate produit



Construire un automate minimal \mathcal{B} reconnaissant $L_1 \cap L_2$.

On reprend l'automate précédent en changeant le nom des états :

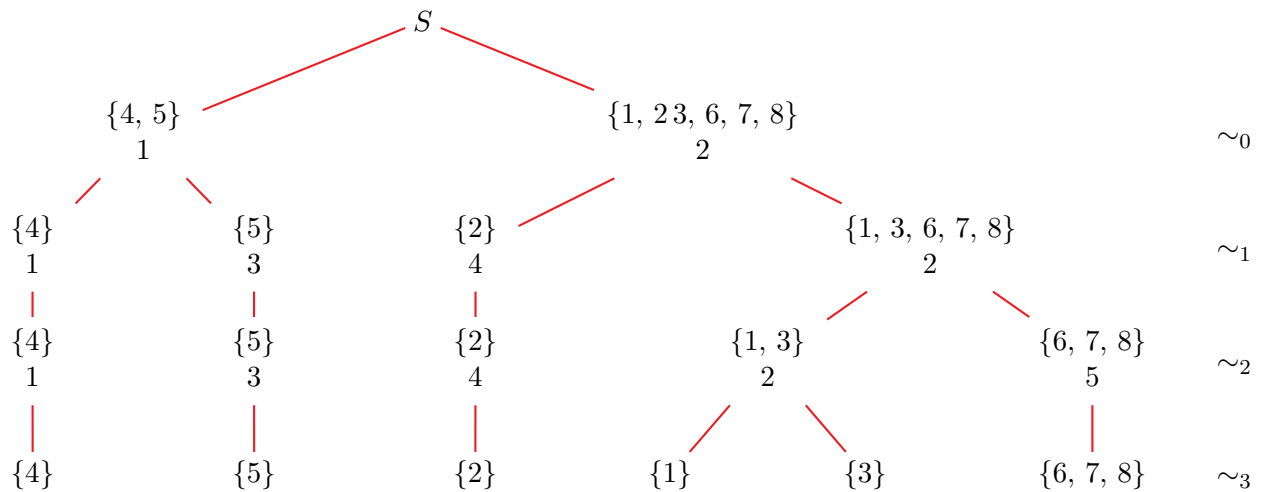


s	1	2	3	4	5	6	7	8
$\text{cl}(s)$	2	2	2	1	1	2	2	2
$\text{cl}(s.a)$	2	1	2	1	1	2	2	2
$\text{cl}(s.b)$	2	2	2	1	2	2	2	2

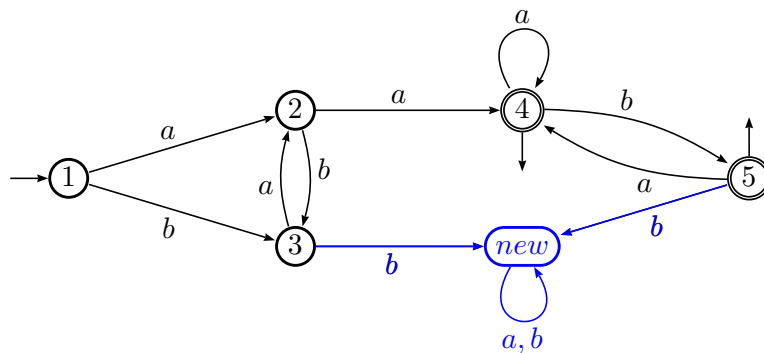
s	1	2	3	4	5	6	7	8
$\text{cl}(s)$	2	4	2	1	3	2	2	2
$\text{cl}(s.a)$	4	1	4	1	1	2	2	2
$\text{cl}(s.b)$	2	2	2	3	1	2	2	2

s	1	2	3	4	5	6	7	8
$cl(s)$	2	4	2	1	3	5	5	5
$cl(s.a)$	4	1	4	1	1	5	5	5
$cl(s.b)$	2	2	5	3	5	5	5	5

On a l'arbre suivant :



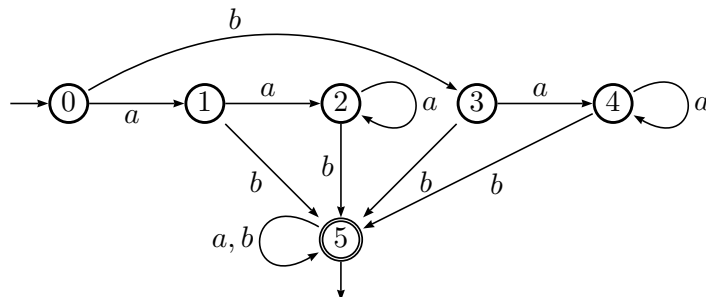
Ainsi, l'automate minimal est :



Exercice . Soit l'automate \mathcal{A} d'états 0, 1, 2, 3, 4, 5 ; d'état initial 0, d'état terminal 5 et de transitions : $(0, a, 1)$, $(1, a, 2)$, $(2, a, 2)$, $(3, a, 4)$, $(4, a, 4)$, $(5, a, 5)$, $(0, b, 3)$, $(1, b, 5)$, $(2, b, 5)$, $(3, b, 5)$, $(4, b, 5)$, $(5, b, 5)$.

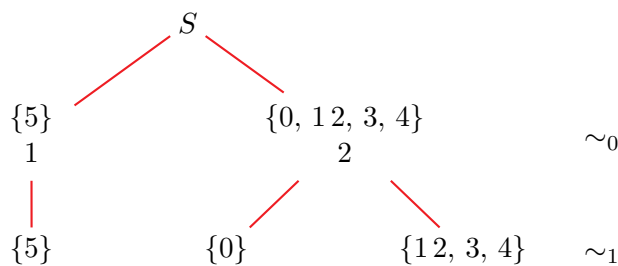
Dessiner l'automate \mathcal{A} et le minimiser.

Correction. On a l'automate \mathcal{A} suivant :

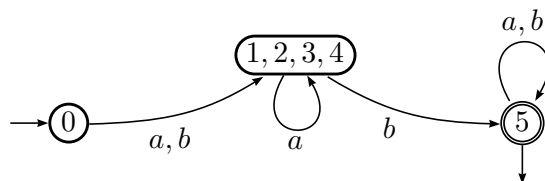


s	0	1	2	3	4	5
$\text{cl}(s)$	2	2	2	2	2	1
$\text{cl}(s.a)$	2	2	2	2	2	1
$\text{cl}(s.b)$	2	1	1	1	1	1

On a l'arbre suivant :



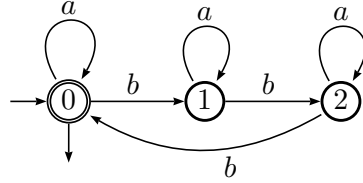
De ce fait, l'automate minimal est :



Exercice . Soit L l'ensemble des mots sur l'alphabet $\{a, b\}$ où le nombre d'occurrences de b est divisible par 3.

Il y a un automate \mathcal{A} à trois états tel que $L = L(\mathcal{A})$. Donner le système d'équations associé aux transitions et résoudre ce système pour $L_{1,F}$, pour donner une expression rationnelle dénotant L .

Correction. On a l'automate \mathcal{A} suivant tel que $L = L(\mathcal{A})$.



On a alors le système suivant où chaque L_i est le langage reconnu par l'automate où i est l'état initial.

$$\begin{cases} L_0 = aL_0 + bL_1 + \varepsilon & (1) \\ L_1 = aL_1 + bL_2 & (2) \\ L_2 = aL_2 + bL_0 & (3) \end{cases}$$

On rappelle alors le lemme d'Arden.

Soit $X = KX + M$. Le plus petit langage de A^* vérifiant cette équation est $X = K^*M$.

En effet : $KX + M = KK^*M + M = K^*M + M = (K^+ + \varepsilon)M = K^*M$

D'après (3), on a :

$L_2 = aL_2 + bL_0 = a^*bL_0$ par le lemme d'Arden

D'après (2), on a :

$L_1 = aL_1 + bL_2 = aL_1 + ba^*bL_0$ par le lemme d'Arden

D'après (1), on a :

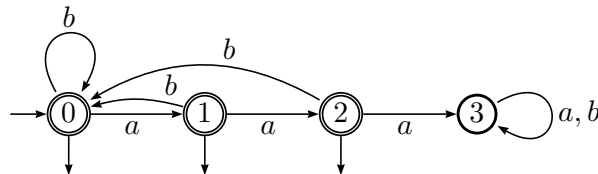
$$\begin{aligned} L_0 &= aL_0 + bL_1 + \varepsilon \\ &= aL_0 + ba^*ba^*L_0 + \varepsilon \\ &= \underbrace{(a + ba^*ba^*b)}_K L_0 + \underbrace{\varepsilon}_M \\ &= (a + ba^*ba^*b)^* \cdot \{\varepsilon\} \\ L_0 &= (a + ba^*ba^*b)^* = \{a, ba^*ba^*b\} \end{aligned}$$

Ainsi, $L(\mathcal{A}) = (a + ba^*ba^*b)^*$.

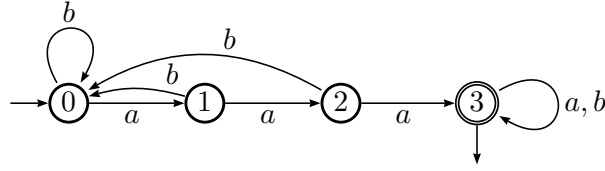
Remarque. Si $M = \emptyset$, alors $X = \emptyset$.

Exercice . Soit L le langage sur l'alphabet $\{a, b\}$ comprenant tous les mots qui n'ont pas trois occurrences successives de a . Donner l'automate déterministe complet minimal reconnaissant L et résoudre le système d'équations correspondant ; en déduire une expression rationnelle de L .

Correction. L'automate \mathcal{A} suivant et l'automate reconnaissant le langage L décrit.



Remarque. L'automate complémentaire reconnaissant le langage comprenant tous les mots ayant trois occurrences successives de a est le suivant :



On a alors le système suivant :

$$\begin{cases} L_0 = aL_0 + bL_0 + \varepsilon & (1) \\ L_1 = aL_2 + bL_0 & (2) \\ L_2 = aL_3 + bL_0 & (3) \\ L_3 = aL_3 + bL_3 & (4) \end{cases}$$

Remarque. L'équation (4) s'écrit également $L_3 = (a + b) L_3$ en mettant L_3 en facteur. Ainsi, d'après le lemme d'Arden, on a $L_3 = \emptyset$.

On résout par substitutions le système précédent.

D'après (3), on a :

$$L_2 = bL_0 + \varepsilon$$

En remplaçant L_2 par sa valeur dans (2), on a :

$$L_1 = a(bL_0 + \varepsilon) + bL_0 + \varepsilon = (ab + b) L_0 + a + \varepsilon$$

En remplaçant L_1 par sa valeur dans (1), on a :

$$L_0 = a((ab + b) L_0 + a + \varepsilon) + bL_0 + \varepsilon = \underbrace{(b + a(ab + b))}_{b(a^2 + a + \varepsilon)} L_0 + a^2 + a + \varepsilon$$

D'après le lemme d'Arden, on a alors :

$$L_0 = (a(ab + b) + b)^* (a^2 + a + \varepsilon) = ((a^2 + a + \varepsilon) b)^* (a^2 + a + \varepsilon)$$

$$\text{Ainsi, } L(\mathcal{A}) = ((a^2 + a + \varepsilon) b)^* (a^2 + a + \varepsilon)$$