



LICENCE D'INFORMATIQUE

Année 2009–2010

STRUCTURES DISCRÈTES LI 214

Ce document comprend les exercices de Travaux Dirigés,
les TME et des rappels sur la Théorie des ensembles.

Informations sur [http ://www.liafa.jussieu.fr/~ig/l2.html](http://www.liafa.jussieu.fr/~ig/l2.html)

LICENCE STRUCTURES DISCRÈTES – T.D. INDUCTION

EXERCICE 1.1 Montrer que, pour $n \geq 2$,

$$\overline{A_1 \cup A_2 \cup \dots \cup A_n} = \overline{A_1} \cap \overline{A_2} \cap \dots \cap \overline{A_n}$$

$$\overline{A_1 \cap A_2 \cap \dots \cap A_n} = \overline{A_1} \cup \overline{A_2} \cup \dots \cup \overline{A_n}$$

(Lois de Morgan généralisées)

◇

EXERCICE 1.2 La suite harmonique est définie par $H_n = 1 + 1/2 + \dots + 1/n$ pour $n \geq 1$. Montrer que $H_{2^n} \geq 1 + n/2$ pour tout $n \geq 0$.

◇

EXERCICE 1.3 On prouve la proposition “Toutes les roses ont le même parfum.” par récurrence sur le nombre n de roses. Si $n = 1$, c’est évident. Soit un ensemble $c_1, c_2, \dots, c_{n+1}$ de $n + 1$ roses. Par hypothèse de récurrence $c_1, \dots, c_n$ est un ensemble de n roses qui ont donc le même parfum. De même, $c_2, c_3, \dots, c_{n+1}$ est un ensemble de n roses qui ont donc le même parfum. Puisque ces deux ensembles ont en commun $c_2, \dots, c_n$, les $n + 1$ roses ont le même parfum. Où est l’erreur ?

◇

EXERCICE 1.4 On considère le polynôme à coefficient réels $P(x) = \frac{1}{3}x^3 + ax^2 + bx$.

- 1) Trouver a et b pour que $\forall x \in \mathbb{R}, P(x+1) - P(x) = x^2$. On suppose dans la suite que cette propriété est vérifiée.
- 2) Montrer que $\forall n \in \mathbb{N}, P(n)$ est un entier.
- 3) $\forall n \geq 0$, on pose $S_n = \sum_{k=0}^n k^2$. Montrer que

$$\forall n \geq 0, \quad S_n = P(n+1) = \frac{n(n+1)(2n+1)}{6}.$$

◇

EXERCICE 1.5 Soit $H_n = 1 + 1/2 + \dots + 1/n$ pour $n \geq 2$. Montrer que H_n n’est pas entier pour $n \geq 2$.

◇

EXERCICE 1.6 On rappelle que les nombres de Fibonacci sont définis par $F_n = F_{n-1} + F_{n-2}$ pour $n > 1$ et $F_0 = 0, F_1 = 1$. Montrer que pour tout $n > 0$:

- 1) $F_1 + F_3 + \dots + F_{2n-1} = F_{2n}$.
- 2) $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$.
- 3) $F_1^2 + F_2^2 + \dots + F_n^2 = F_n F_{n+1}$
- 4) F_{3n} est pair.

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

EXERCICE 1.7 Donner une définition inductive de $f(n) = a^{2^n}$.

Indication. On pourra remarquer que $a^{2^{n+1}} = \left(a^{2^n}\right)^2$. $\diamond$

EXERCICE 1.8 1) Montrer que $\forall n \in \mathbb{N}, (n+1)^2 - (n+2)^2 - (n+3)^2 + (n+4)^2 = 4$.

2) En déduire que tout entier m peut s'écrire comme somme et différence des carrés $1^2, 2^2, \dots, n^2$ pour un certain n , c'est-à-dire

$$\forall m \in \mathbb{N}, \exists n \in \mathbb{N}, \exists \varepsilon_1, \dots, \varepsilon_n \in \{-1, 1\}, m = \varepsilon_1 1^2 + \varepsilon_2 2^2 + \dots + \varepsilon_n n^2$$

(Indication : montrer d'abord le résultat pour $m \in \{0, 1, 2, 3\}$). $\diamond$

EXERCICE 1.9 On considère le sous-ensemble D de $\mathbb{N} \times \mathbb{N}$ défini inductivement par :

(i) $(n, 0) \in D$

(ii) si $(n, n') \in D$, alors $(n, n + n') \in D$

1) Donner quelques éléments de D .

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

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

EXERCICE 1.10 1) On dit qu'un ensemble ordonné est un *treillis* si tout sous-ensemble fini non vide d'éléments de E admet une borne supérieure et une borne inférieure.

Montrer qu'un ensemble ordonné $(E, \leq)$ est un treillis si et seulement si tout sous-ensemble à deux éléments de E admet une borne supérieure et une borne inférieure.

Indication : montrer par récurrence sur n que si tout sous-ensemble à deux éléments de E admet une borne supérieure et une borne inférieure, alors tout sous-ensemble fini à n éléments de E admet une borne supérieure et une borne inférieure.

2) Un treillis est un *treillis complet* si tout sous-ensemble d'éléments de E admet une borne supérieure et une borne inférieure.

Montrer que si un ensemble ordonné $(E, \leq)$ est un treillis complet, alors $\sup\{e \mid e \in E\} = \inf\{e \mid e \in \emptyset\} = \top$ est le plus grand élément de E et $\inf\{e \mid e \in E\} = \sup\{e \mid e \in \emptyset\} = \perp$ est le plus petit élément de E . $\diamond$

EXERCICE 1.11 Donner une définition inductive de la hauteur $h(t)$, du nombre de nœuds $n(t)$, du nombre d'arêtes $ar(t)$ et du nombre de feuilles $f(t)$ d'un arbre binaire. $\diamond$

EXERCICE 1.12 Soit t un arbre, $h(t)$ sa hauteur, $n(t)$ le nombre de ses nœuds, $f(t)$ le nombre de ses feuilles. On suppose que t est un arbre binaire (0, 1 ou 2 fils par nœud). Montrer que $n(t) \leq 2^{h(t)} - 1$, et que $f(t) \leq 2^{h(t)-1}$. $\diamond$

EXERCICE 1.13 Soit t un arbre binaire *strict* (c'est-à-dire que t est non vide, chaque nœud de t a exactement 0 ou 2 fils et il n'y a aucun nœud avec un seul fils non vide). Soit $n(t)$ le nombre de ses nœuds, $f(t)$ le nombre de ses feuilles et $a(t)$ le nombre de ses arêtes.

1) Donner une définition de l'ensemble *ABS* des arbres stricts.

2) Montrer que si t est un arbre binaire *strict* non vide $n(t) = a(t) + 1$.

3) Montrer que si t est un arbre binaire *strict* non vide $n(t) = 2f(t) - 1$. $\diamond$

EXERCICE 1.14 Un arbre n -aire *complet* est un arbre où :

- chaque nœud interne a exactement n fils,
- toutes les feuilles sont à la même profondeur.

L'arbre vide est un arbre n -aire de profondeur 0.

- 1) Donner une définition inductive des arbres n -aires complets et étiquetés sur l'alphabet $A = \{a\}$ (toutes les nœuds sont étiquetés a).
- 2) Donner une définition inductive des arbres n -aires complets et étiquetés sur un alphabet A non réduit à un unique élément.
- 3) Donner une définition inductive du nombre de nœuds et du nombre d'arêtes d'un arbre n -aire complet de profondeur k . $\diamond$

EXERCICE 1.15 Définir le parcours préfixe d'un arbre binaire. $\diamond$

EXERCICE 1.16 (Fonction d'Ackerman) Soit $f: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ définie par :

$$\begin{aligned} f(0, n) &= n + 1, \\ f(m, 0) &= f(m - 1, 1), \\ f(m, n) &= f(m - 1, f(m, n - 1)). \end{aligned}$$

- 1) Montrer que $f(m, n)$ est définie pour tout couple $(m, n) \in \mathbb{N} \times \mathbb{N}$.
- 2) Soit $g: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ définie par :

$$\begin{aligned} g(0, n) &= n + 1, \\ g(m, 0) &= g(m - 1, 1), \\ g(m, n) &= g(m - 1, g(m, n + 1)). \end{aligned}$$

Pour quels couples $(m, n) \in \mathbb{N} \times \mathbb{N}$ la fonction $g(m, n)$ est-elle définie ?

- 3) Calculer $f(1, n)$, $f(2, n)$ et $f(3, n)$. $\diamond$

EXERCICE 1.17 Les listes de lettres L de l'alphabet A sont définies inductivement par :

- (B) $\varepsilon \in L$,
- (I) $\forall l \in L, \forall a \in A, (al) \in L$.

Définissons $g(x, y)$ sur $L \times L$ par : $\forall a \in A, \forall l \in L, \forall y \in L$,

$$\begin{aligned} g(\varepsilon, y) &= y, \\ g((al), y) &= g(l, (ay)). \end{aligned}$$

- 1) Soit $Q(x)$ le prédicat ' $\forall y, g(x, y)$ est défini'. Prouver par induction sur x que $Q(x)$ est vrai sur L .
- 2) Calculer $g((a_1), y)$, pour $a_1 \in A, y \in L$.
- 3) Prouver par induction sur n (pour $n \geq 1$) que

$$g((a_n(a_{n-1}(\dots(a_1)\dots))), y) = g(\varepsilon, (a_1(\dots(a_{n-1}(a_n y))\dots))).$$

- 4) Soit $rev(x) = g(x, \varepsilon)$. Dédurre de la question 3 que, pour $a_1, \dots, a_n \in A$,

$$rev((a_n(a_{n-1}(\dots(a_1)\dots)))) = (a_1(\dots(a_{n-1}(a_n))\dots)). \quad \diamond$$

Les notations qui suivent sont communes aux exercices 1.18 à 1.25. Un ensemble E muni d'une opération $*$ associative est un semi-groupe. Si de plus E possède un élément neutre e pour $*$ on dit que $(E, *, e)$ est un monoïde. Etant donné un ensemble A , le monoïde libre sur A , noté A^* , est l'ensemble des suites finies d'éléments de A (dont la suite vide, notée ε). Dans ce cas, l'opération qui fait de A^* un monoïde est la concaténation, notée $.$ ("point"), qui admet la suite vide ε , pour élément neutre.

EXERCICE 1.18 On considère le monoïde libre A^* sur un alphabet fini A . On note u^p le mot $u \dots u$, où p exemplaires de u ont été concaténés. On dit que mot u est un *facteur gauche* d'un mot v , s'il existe un mot w tel que $v = uw$, (de même, u est facteur droit de v si on peut écrire $v = wu$). Soit α et β deux mots.

1) Montrer qu'une condition nécessaire et suffisante pour que l'un d'entre eux soit facteur gauche de l'autre est qu'il existe deux mots u et v tels que $\alpha u = \beta v$.

2) On suppose que $\alpha\beta = \beta\alpha$. Montrer que cela équivaut à : $\exists \gamma \in A^*, \exists (p, q) \in \mathbb{N} \times \mathbb{N}, \alpha = \gamma^p, \beta = \gamma^q$. $\diamond$

On appelle *langage* (sur l'alphabet A) un sous-ensemble de A^* . Si L_1 et L_2 sont deux langages de A^* , on définit leur concaténation par :

$$L_1.L_2 = \{u.v \mid u \in L_1, v \in L_2\}$$

La concaténation de langages est une opération associative admettant $\{\varepsilon\}$ comme élément neutre. On peut alors définir les puissances d'un langage L par : $L^0 = \{\varepsilon\}$ et $L^{n+1} = L^n.L = L.L^n$. Enfin, l'étoile d'un langage L est le sous-monoïde de A^* engendré par L c'est à dire :

$$L^* = \bigcup_{n \in \mathbb{N}} L^n$$

EXERCICE 1.19 Les notations sont les mêmes que dans l'exercice 1.18. On définit inductivement les langages *rationnels* (on dit aussi *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 $L_1.L_2 = \{uv, u \in L_1, v \in L_2\}$ 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, on définit $\tilde{L} = \{\tilde{u}, u \in L\}$. Montrer que si L est un langage rationnel, alors $\tilde{L}$ est également rationnel. $\diamond$

EXERCICE 1.20 [Lemme d'Arden] Soient L et M deux langages de A^* tels que $\varepsilon \notin L$, montrer que l'équation $X = L.X \cup M$ admet pour unique solution le langage $L^*.M$. $\diamond$

EXERCICE 1.21

- 1) Définir la relation "est un préfixe de" sur A^* . S'agit-il d'une relation d'ordre ? si oui, s'agit-il d'un ordre total ou d'un ordre partiel ?
- 2) En supposant que A est muni d'un ordre total $\preceq_A$, définir l'ordre lexicographique sur A^* . S'agit-il d'un ordre total ou d'un ordre partiel ?
- 3) Soit $A = \{a, b\}$ tel que $a \preceq_A b$. Montrer que l'ordre lexicographique défini à la question précédente n'est pas un ordre bien fondé. $\diamond$

EXERCICE 1.22 Soient $(A, \leq_1)$ et $(B, \leq_2)$ deux ordres bien fondés. Les ordres suivants sont-ils bien fondés ?

- 1. sur $A \times B$, l'ordre produit $((a, b) \leq (a', b') \iff (a \leq_1 a') \wedge (b \leq_2 b'))$.
- 2. sur $A \times B$, l'ordre lexicographique. $\diamond$

EXERCICE 1.23 Les ordres suivants sont-ils bien fondés ?

1. sur A^2 , l'ordre lexicographique (A alphabet totalement ordonné).
2. sur $\mathbb{N} \setminus \{0\}$, $m \leq n$ ssi m divise n .
3. sur l'ensemble des diviseurs d'un entier donné, la relation du 2.
4. sur A^* , l'ordre préfixe.
5. sur A^* , l'ordre $u \leq v$ ssi u est un sous-mot de v .
6. sur A^* , l'ordre lexicographique (A alphabet totalement ordonné).
7. sur $A^*/\equiv$, l'ordre des longueurs $u \leq v$ ssi $|u| \leq |v|$ (où $\equiv$ est défini par $u \equiv v$ si et seulement si $|u| = |v|$). $\diamond$

EXERCICE 1.24 L'ordre préfixe sur A^* peut être défini de deux manières différentes :

- (1) $\forall m_1, m_2 \in A^* \quad m_1 \preceq_{pref}^1 m_2$ si et seulement si $\exists m_3 \in A^* \quad m_1 m_3 = m_2$
- (2) Définition inductive :
 - (B) $\forall m \in A^* \quad \varepsilon \preceq_{pref}^2 m$
 - (I) Si $m_1, m_2 \in A^*$ sont tels que $m_1 \preceq_{pref}^2 m_2$, alors pour tout $a \in A$, $am_1 \preceq_{pref}^2 am_2$

Montrer que ces deux définitions sont équivalentes. $\diamond$

EXERCICE 1.25 Soit A^* le monoïde libre engendré par l'alphabet A .

Le miroir d'un mot $u = a_1 a_2 \cdots a_n$ est le mot $\tilde{u} = a_n \cdots a_2 a_1$. Donner une définition inductive du miroir. $\diamond$

EXERCICE 1.26 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 $h: F_0 \longrightarrow V$, et $h_s: V \longrightarrow V$; il existe une et une seule fonction h^* de T dans V telle que :

- (B') Si $t = a \in F_0$, $h^*(t) = h(t)$,
- (I') Si $t = s(t_1)$, $h^*(t) = h_s(h^*(t_1))$.

Calculer h^* dans les cas suivants :

- 1) $h_1(a) = 0$, $h_{1s}(n) = n + 1$.
- 2) $h_2(a) = 1$, $h_{2s}(n) = 2n$.
- 3) $h_3(a) = 1$, $h_{3s}(n) = n + 2$. $\diamond$

EXERCICE 1.27 Soit X un ensemble de symboles de variable et F un ensemble de symboles de fonction, on définit inductivement l'ensemble $T(F, X)$ des termes sur F et X de la façon suivante :

- (B) Si $x \in X$ alors $x \in T(F, X)$; si $f \in F_0$, alors $f \in T(F, X)$;
- (I) Si $t_1, \dots, t_n \in T(F, X)$, $n \geq 1$ et $f \in F_n$, alors $f(t_1, \dots, t_n) \in T(F, X)$. (F_n est l'ensemble des symboles de fonction d'arité n).

- 1) Une application φ de $T(F, X)$ dans $T(F, X)$ est un *morphisme* si elle vérifie $\varphi(f_0) = f_0$ pour $f_0 \in F_0$ et $\varphi(f(t_1, \dots, t_n)) = f(\varphi(t_1), \dots, \varphi(t_n))$ pour $f \in F_n, n > 0$. Soit une application h de X dans $T(F, X)$, montrer qu'il existe un morphisme unique h^* de $T(F, X)$ dans $T(F, X)$ qui prolonge h (c'est à dire, tel que $\forall x \in X, h^*(x) = h(x)$).

- 2) Une substitution est une opération sur les termes permettant de remplacer certaines variables d'un terme par un terme. Plus formellement, l'ensemble Θ des substitutions est l'ensemble des fonctions de X dans $T(F, X)$:

$$\Theta = \{\theta : X \rightarrow T(F, X)\}$$

On notera s_{id} la substitution identité, c'est-à-dire la substitution telle que pour toute variable $x \in X$, $s_{id}(x) = x$. Une substitution θ se prolonge de façon unique en une fonction θ^* de $T(F, X)$ dans $T(F, X)$ définie par :

$$\forall x \in X \quad \theta^*(x) = \theta(x) \quad \text{et} \quad \forall f_0 \in F_0 \quad \theta^*(f_0) = f_0$$

$$\forall f \in F_n \quad \forall t_1, t_2, \dots, t_n \in T(X, F) \quad \theta^*(f(t_1, t_2, \dots, t_n)) = f(\theta^*(t_1), \theta^*(t_2), \dots, \theta^*(t_n))$$

Appliquer une substitution à un terme c'est donc instancier certaines de ses variables par des termes et donc faire augmenter sa taille. Montrer que :

$$\forall \theta \in \Theta \quad \forall t \in T(X, F) \quad \tau(\theta^*(t)) \geq \tau(t)$$

où la taille d'un terme est donnée par l'application $\tau : T(X, F) \rightarrow \mathbb{N}$ définie par :

$$\tau(t) = \begin{cases} 0 & \text{si } t = x, \\ 1 & \text{si } t = f_0 \in F_0, \\ 1 + \sum_{i=1}^n \tau(t_i) & \text{si } t = f(t_1, \dots, t_n). \end{cases}$$

- 3) On peut définir un préordre (i.e. une relation réflexive et transitive) sur $T(X, F)$ comme suit : $t_1 \preceq t_2$ si et seulement si il existe une substitution θ telle que $\theta^*(t_1) = t_2$. Montrer que cette relation définit bien un préordre (ce préordre est appelé préordre de filtrage).
- 4) Pourquoi le préordre de filtrage ne définit-il pas une relation d'ordre (i.e. une relation réflexive, antisymétrique et transitive) ?
- 5) Dans cette question nous allons voir qu'il est possible de définir une relation d'ordre à partir d'un préordre.

Soit E un ensemble muni d'un préordre $\preceq$.

- a) Montrer que la relation $\approx$ définie par $x \approx y$ ssi soit $x = y$ soit $x \preceq y$ et $y \preceq x$ est une relation d'équivalence (i.e., une relation réflexive, symétrique et transitive).
- b) On note $[x]_{/\approx} = \{y \mid y \approx x\}$ la classe d'équivalence de x et $E_{/\approx}$ l'ensemble des classes d'équivalence de E pour $\approx$ (ensemble quotient). Montrer que la relation $\overline{\preceq}$ définie sur $E_{/\approx}$ par $[x]_{/\approx} \overline{\preceq} [y]_{/\approx}$ ssi $x \preceq y$ est une relation d'ordre (cet ordre est appelé ordre quotient du préordre $\preceq$).
- 6) D'après la question précédente, on peut définir une relation d'équivalence sur les termes comme suit :

$$t_1 \equiv t_2 \text{ ssi } t_1 = t_2 \text{ ou } (t_1 \preceq t_2 \text{ et } t_2 \preceq t_1)$$

En fait, la relation $\equiv$ identifie les termes équivalents à un renommage près des variables. A partir de cette relation d'équivalence entre termes, il est alors possible de construire une relation d'ordre $\overline{\preceq}$ sur l'ensemble quotient $T(X, F)_{/\equiv}$ définie par :

$$[t_1]_{/\equiv} \overline{\preceq} [t_2]_{/\equiv} \text{ ssi } t_1 \preceq t_2$$

On remarquera que tous les termes d'une même classe d'équivalence ont la même taille.

- (i) Soit $\Lambda(t)$ le nombre de variables distinctes apparaissant dans le terme t . Montrer que $t \prec t'$ et $\tau(t) = \tau(t') \implies \Lambda(t') < \Lambda(t)$.
- (ii) Montrer que $\supseteq$ est un ordre bien fondé sur $T(X, F)_{/\equiv}$. ◇

EXERCICE 1.28 1. On suppose qu'une propriété P définie sur $\mathbb{N}$ vérifie :

- (i) $P(1)$ est vraie
- (ii) si $P(n)$ est vraie alors $P(2n)$ est vraie (pour $n \geq 1$)
- (iii) si $P(n)$ est vraie alors $P(n-1)$ est vraie (pour $n \geq 2$).

Montrer par récurrence sur n que $P(n)$ est vraie pour tout $n \geq 1$.

2. On veut montrer que la moyenne arithmétique est supérieure à la moyenne géométrique. Soient $a_1, \dots, a_n$ n nombres réels positifs, avec $n \geq 1$; on pose $A = (a_1 + \dots + a_n)/n$ et $G = (a_1 \cdots a_n)^{1/n}$, montrer que $A \geq G$. ◇

LICENCE STRUCTURES DISCRÈTES – T.D. ENSEMBLES ET ORDRES

ENSEMBLES ET FONCTIONS

EXERCICE 2.1 Calculer l'ensemble $\mathcal{P}(S)$ des parties de S pour

1. $S = \{1, 2, 3\}$

2. $S = \{1, \{1, 4\}\}$. ◇

EXERCICE 2.2 Soient A, B, C trois parties de E . Montrer que :

1) $A \cap \overline{B} = A \cap \overline{A \cap B}$.

2) $A \cap \overline{B} = A \cap \overline{C} \iff A \cap B = A \cap C$. ◇

EXERCICE 2.3 Soient A, B, C trois parties de E . Montrer que

$$(A \cup B \subseteq A \cup C \text{ et } A \cap B \subseteq A \cap C) \implies B \subseteq C.$$

Dans quel cas a-t-on l'égalité $B = C$? ◇

EXERCICE 2.4 1) Trouver un exemple d'application qui n'est ni injective, ni surjective.

2) Soit $f: A \longrightarrow B$ une application. Montrer que

$$f \text{ injective} \iff \forall X, Y \subseteq A, f(X \cap Y) = f(X) \cap f(Y). \quad \diamond$$

EXERCICE 2.5 Soit $f: E \longrightarrow F$ une application. Montrer que :

- i) Si $E \neq \emptyset$ alors f est injective si et seulement si f a un inverse à gauche, c'est-à-dire il existe une application $r: F \longrightarrow E$ telle que $r \circ f = id_E$. L'application r est surjective et s'appelle une rétraction de f .
- ii) f est surjective si et seulement si f a un inverse à droite, c'est-à-dire il existe une application $s: F \longrightarrow E$ telle que $f \circ s = id_F$. L'application s est injective et s'appelle une section de f .
- iii) f est bijective si et seulement si f a un inverse, c'est-à-dire il existe une application $f^{-1}: F \longrightarrow E$ telle que $f \circ f^{-1} = id_F$ et $f^{-1} \circ f = id_E$. L'application f^{-1} est une bijection et s'appelle la bijection réciproque de f . ◇

EXERCICE 2.6 Soit A et B deux parties disjointes d'un ensemble E . Réaliser une bijection entre $\mathcal{P}(A) \times \mathcal{P}(B)$ et $\mathcal{P}(A \cup B)$. ◇

EXERCICE 2.7 Démontrer que $A \times (B \cap C) = (A \times B) \cap (A \times C)$. ◇

EXERCICE 2.8 Si $n < m$, il n'existe pas d'injection de $[m]$ dans $[n]$. Une formulation plus imagée de ce résultat est le "principe des tiroirs" (ou "pigeon-hole principle"), à savoir : si

$n < m$ et que l'on veut placer m chemises dans n tiroirs alors un tiroir au moins contiendra plusieurs chemises. Plus précisément, montrer qu'un tiroir contiendra au moins un nombre entier $p \geq \frac{m}{n}$ chemises. $\diamond$

EXERCICE 2.9 1. On dit qu'un ensemble E est *dénombrable* s'il a le même cardinal que $\mathbb{N}$. Montrer que $\mathbb{N}^2$ est dénombrable (on pourra montrer que $(x, y) \mapsto 2^y(2x+1)-1$, ou bien $f(x, y) = y + (0 + 1 + 2 + \dots + (x+y))$, définit une bijection de $\mathbb{N}^2$ dans $\mathbb{N}$). $\diamond$

EXERCICE 2.10 Montrer que l'ensemble des nombres rationnels est dénombrable. $\diamond$

EXERCICE 2.11 * Soit A un ensemble infini. Soit $B = \{b_1, b_2, \dots\}$ un ensemble dénombrable tel que A et B sont disjoints. Démontrer qu'il existe une bijection entre $A \cup B$ et A . $\diamond$

OPÉRATIONS ET RELATIONS

EXERCICE 2.12 Soit $*$ une opération binaire sur un ensemble E . Démontrer que si $*$ admet un élément neutre, cet élément neutre est unique. $\diamond$

EXERCICE 2.13 Montrer que $(\mathbb{N}, +, 0)$ (resp. $(P(E), \cup, \emptyset)$) sont des monoïdes commutatifs. $\diamond$

EXERCICE 2.14 A étant un alphabet, l'ensemble des mots de A^* de longueur paire est-il un monoïde ? $\diamond$

EXERCICE 2.15 Considérer la relation $R = \{(1, 1), (2, 3), (3, 2)\}$ sur $X = \{1, 2, 3\}$. Déterminer si R est

1. réflexive,
2. symétrique,
3. transitive. $\diamond$

EXERCICE 2.16 Soient $A = \{1, 2, 3, 4\}$ et $B = \{1, 3, 5\}$, et soit $R \subset A \times B$ définie par $(a, b) \in R$ ssi $a < b$.

1. Écrire R comme un ensemble de paires ordonnées.
2. Tracer R sur un diagramme $A \times B$.
3. Trouver le domaine de R et le co-domaine de R^{-1} .
4. Trouver $R^{-1}.R$. $\diamond$

EXERCICE 2.17 Soit T la relation sur l'ensemble des réels $\mathbb{R}$ définie comme suit :

$$xTy \text{ ssi il existe } n \in \mathbb{N} \text{ t.q. } x \in [n, n+1] \text{ et } y \in [n, n+1]$$

Tracer le graphe de la relation T . $\diamond$

EXERCICE 2.18 La relation $\mathcal{R}$ sur $\mathbb{N}$ définie par " $n \mathcal{R} m$ si et seulement si $m = n+1$ " est-elle symétrique ? réflexive ? transitive ? Quelles sont les relations $\mathcal{R}^+$ et $\mathcal{R}^*$? $\diamond$

EXERCICE 2.19 La relation $\mathcal{R}$ sur $\mathbb{N}$ définie par " $n \mathcal{R} m$ si et seulement si n et m ont un diviseur commun différent de 1" est-elle transitive ? $\diamond$

EXERCICE 2.20 Soit T la relation sur l'ensemble des réels $\mathbb{R}$ définie par

$$xTy \text{ ssi } 0 \leq x - y \leq 1$$

- i. Exprimer T et T^{-1} comme sous-ensembles de $\mathbb{R} \times \mathbb{R}$ et tracer le graphe des relations.
- ii. Montrer que $T \circ T^{-1} = \{(x, z) : |x - z| \leq 1\}$. $\diamond$

EXERCICE 2.21 Démontrer que pour toutes paires de relations $R \subset X \times Y$ et $S \subset Y \times Z$, $(S \circ R)^{-1} = R^{-1} \circ S^{-1}$. $\diamond$

EXERCICE 2.22 Soit E un ensemble fini : $E = \{e_1, \dots, e_n\}$, et soit $\mathcal{R}$ une relation binaire sur E . On représente $\mathcal{R}$ par une matrice $M_{\mathcal{R}}$ de dimension $n \times n$, à éléments dans $\{0, 1\}$ de la façon suivante :

$$m_{i,j} = \begin{cases} 1 & \text{si } e_i \mathcal{R} e_j, \\ 0 & \text{sinon.} \end{cases}$$

- 1) Quelle est la propriété de $M_{\mathcal{R}}$ caractérisant le fait que la relation $\mathcal{R}$ est symétrique ? réflexive ? irréflexive ? antisymétrique ?
- 2) Si on connaît $M_{\mathcal{R}}$ et $M_{\mathcal{R}'}$, comment peut-on calculer $M_{\mathcal{R}^{-1}}$, $M_{\overline{\mathcal{R}}}$ et $M_{\mathcal{R} \cdot \mathcal{R}'}$. $\diamond$

EXERCICE 2.23 Soit $R \subset A \times A$ une relation. Démontrer que :

1. R est réflexive ssi $\Delta_A \subset R$, où $\Delta_A = \{(a, a) : a \in A\}$ est appelé *relation diagonale*.
2. R est symétrique ssi $R = R^{-1}$
3. R est transitive ssi $R \cdot R \subset R$
4. R réflexive implique $R \cdot R \supset R$ et $R \cdot R$ réflexive
5. R symétrique implique $R \cdot R^{-1} = R^{-1} \cdot R$
6. R transitive implique $R \cdot R$ transitive. $\diamond$

EXERCICE 2.24 Considérons l'ensemble $\mathbb{N} \times (\mathbb{N} \setminus \{0\})$, c'est-à-dire l'ensemble des paires ordonnées d'entiers positifs dont le second est non nul. Soit R définie sur $\mathbb{N} \times (\mathbb{N} \setminus \{0\})$ par

$$(a, b)R(c, d) \text{ ssi } ad = bc$$

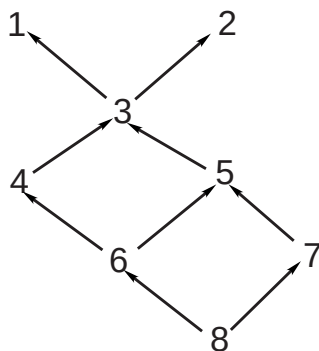
Démontrer que R est une relation d'équivalence. $\diamond$

EXERCICE 2.25 Soit R une relation d'équivalence dans A et soit $[a]$ la classe d'équivalence de $a \in A$. Montrer que :

1. pour chaque $a \in A$, $a \in [a]$
2. $[a] = [b]$ ssi $(a, b) \in R$
3. si $[a] \neq [b]$ alors $[a]$ et $[b]$ sont disjointes.
4. l'ensemble des classes d'équivalence forme une partition de A . $\diamond$

EXERCICE 2.26 Soit E un ensemble d'ensembles ; on définit sur E la relation $\sim$ par : $A \sim B$ ssi il existe une bijection $f : A \rightarrow B$. Démontrer que pour tout ensemble d'ensembles E , la relation $\sim$ est une relation d'équivalence. En particulier, $|A| = |B|$ est une relation d'équivalence. $\diamond$

RELATIONS D'ORDRE

FIGURE 2.1 Représentation de l'ensemble ordonné W

EXERCICE 2.27 Soit $W = \{1, 2, \dots, 7, 8\}$ ordonné comme dans la figure 2.1, par un ordre large (i.e. on suppose que $\forall v \in W \ v \leq v$). Considérer le sous-ensemble $V = \{4, 5, 6\}$ de W .

1. Trouver l'ensemble des majorants de V
2. Trouver l'ensemble des minorants de V .
3. Est-ce que $\sup(V)$ existe ?
4. Est-ce que $\inf(V)$ existe ?

◇

EXERCICE 2.28 Soit $A = \{a, b, c\}$ ordonné comme l'indique le diagramme suivant

$$b \longrightarrow a \longleftarrow c$$

Soit $\mathcal{A}$ l'ensemble de tous les sous-ensembles non-vides et totalement ordonnés de A ; $\mathcal{A}$ est partiellement ordonné par inclusion. Représenter graphiquement l'ordre de $\mathcal{A}$.

◇

EXERCICE 2.29 Soit $A = \{2, 3, 4, \dots\} = \mathbb{N} \setminus \{0, 1\}$; A est ordonné par la relation "x divise y".

1. Vérifier que cette relation est un ordre.
2. Déterminer les éléments minimaux de A
3. Déterminer les éléments maximaux de A .

◇

EXERCICE 2.30 On se place dans $\mathbb{N} \setminus \{0\}$ ordonné par la relation "x divise y".

- 1) Existe-t-il une borne sup et une borne inf pour tout sous-ensemble de 2 éléments ?
- 2) Soient les ensembles $A = \{6, 15, 21\}$ et $B = \{1, 6, 14, 21\}$. Donner les minorants et majorants de A (resp. B). A (resp. B) possède-t-il un minimum ? un maximum ?
- 3) Soit $A = \{3, 6, 12, 15\}$. Donner les majorants, minorants, la borne supérieure, la borne inférieure, les éléments maximaux, minimaux. Discuter.
- 4) On se place maintenant dans $\mathbb{N}$ ordonné par la relation "x divise y" et 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 maximum ? un minimum ?

◇

EXERCICE 2.31 Montrer que les ensembles ordonnés $\mathcal{P}\{a, b, c\}$ muni de la relation d'inclusion et $\{1, 2, 3, 6, 7, 14, 21, 42\}$ muni de la relation de division (dans $\mathbb{N}$) sont isomorphes.

◇

EXERCICE 2.32 Soit $\mathcal{A}$ un ensemble d'ensembles. $\mathcal{A}$ est partiellement ordonné par inclusion et soit $\mathcal{B}$ un sous-ensemble de $\mathcal{A}$.

1. Démontrer que si $A \in \mathcal{A}$ est un majorant de $\mathcal{B}$, alors $\bigcup\{B : B \in \mathcal{B}\} \subset A$
2. Est-ce que $\bigcup\{B : B \in \mathcal{B}\}$ est un majorant de $\mathcal{B}$ dans $\mathcal{A}$?

◇

EXERCICE 2.33 Donner un exemple d'ensemble ordonné qui a exactement un élément maximal mais qui n'a pas de maximum. $\diamond$

EXERCICE 2.34 ** Soit X un ensemble partiellement ordonné. Alors il existe un sous-ensemble Y totalement ordonné de X t.q. Y n'est pas un sous-ensemble propre des autres sous-ensembles totalement ordonnés de X . $\diamond$

EXERCICE 2.35 * Soit R une relation, $R \subset A \times B$. Supposons que le domaine de R soit A . Alors il existe un sous-ensemble f^* de R tel que f^* est une fonction de A vers B . $\diamond$

LICENCE STRUCTURES DISCRÈTES – T.D. LOGIQUE

EXERCICE 3.1 On appelle *fonction caractéristique* d'une partie A de E la fonction $\chi_A: E \longrightarrow \{0, 1\}$ définie par :

$$\chi_A(e) = \begin{cases} 1 & \text{si } e \in A, \\ 0 & \text{si } e \notin A. \end{cases}$$

1. Exprimer en termes de fonction caractéristique les opérations d'union, intersection, différence, différence symétrique, complémentation.
2. On suppose que E est fini de taille n .
 - (i) Définir l'algèbre de Boole F des fonctions de E dans $\{0, 1\}$
 - (ii) Montrer qu'on a un isomorphisme ψ de $\mathcal{P}(E)$ dans F (qui associe à une partie A la fonction $\psi(A) = \chi_A$).
 - (iii) Définir l'algèbre de Boole $\{0, 1\}^n$.
 - (iv) Montrer qu'on a un isomorphisme φ de F dans $\{0, 1\}^n$.
 - (iv) En déduire un isomorphisme entre les algèbres de Boole $\mathcal{P}(E)$ et $\{0, 1\}^n$. ◇

EXERCICE 3.2 1. Montrer que la relation $x \leq y$ ssi $x \sqcup y = y$ est une relation d'ordre sur une algèbre de Boole et que pour cet ordre, $\sup(x, y) = x \sqcup y$.

2. Montrer qu'un homomorphisme est une application monotone pour l'ordre sous-jacent à l'algèbre de Boole défini dans la question 1.
3. Montrer que toute application monotone n'est pas forcément un homomorphisme. ◇

EXERCICE 3.3 Trouver un polynôme booléen pour la fonction f définie par :

$f(x, y)$	x		0	1
	y			
	0		1	0
	1		0	0

Quelle est la table de vérité de la fonction duale ? Trouver un polynôme booléen pour la fonction duale. ◇

EXERCICE 3.4 Ecrire les fonctions duales de

1. $x \bar{y} \bar{z}$, 2. $xyz + t$, 3. $(p \vee ff) \wedge (q \vee t)$. ◇

EXERCICE 3.5 Soit une formule F écrite en utilisant les variables propositionnelles $p_1, \dots, p_n$: soit G la formule obtenue en remplaçant chaque occurrence de p_i par $\bar{p}_i$, pour $i = 1, \dots, n$.

T.S.V.P.

- 1) Calculer G pour $F = p_1 \vee (p_2 \wedge p_3)$.
- 2) (i) Montrer que si F est une tautologie, G est aussi une tautologie. (ii) Montrer que si $F \equiv F'$, alors on a aussi $G \equiv G'$.
- 3) Dualité. Soient deux formules F, G formées avec uniquement des $\wedge, \vee, \neg$. Montrer que si $F \equiv G$ alors, on peut permuter les $\wedge$ et $\vee$ et on obtient encore deux formules équivalentes.
- 4) Appliquer la question 3 aux équivalences : $p \vee (p \wedge q) \equiv p$, $\neg(p \wedge q) \equiv (\neg p \vee \neg q)$, $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$. $\diamond$

EXERCICE 3.6 Quel est le nombre maximum de formules non équivalentes qu'on peut former avec n variables propositionnelles $p_1, \dots, p_n$? $\diamond$

EXERCICE 3.7 Ecrire la table de vérité et la fonction duale de $xy + \bar{x}\bar{y}$. $\diamond$

EXERCICE 3.8 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. $\diamond$

EXERCICE 3.9 Une fonction est *auto-duale* ssi $f(x) = \overline{f(\bar{x})}$. Les fonctions suivantes sont-elles auto-duales ?

1. $f(x, y) = x$
2. $f(x, y) = x + y$
3. $f(x, y) = xy + \bar{x}\bar{y}$
4. $f(x, y) = xy + \bar{x}y$ $\diamond$

EXERCICE 3.10 Vérifiez que les fonctions f_1 et f_2 suivantes sont égales : $f_1(x, y, z) = xy + yz + xz$, $f_2(x, y, z) = (x + y)(y + z)(x + z)$. En déduire que f_1 et f_2 sont auto-duales. $\diamond$

EXERCICE 3.11 On définit $x \text{ NAND } y = \bar{x}\bar{y}$. Montrer que toutes les opérations booléennes sont exprimables en fonction de NAND . $\diamond$

EXERCICE 3.12 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 conjonctive les formules précédentes et dessiner les arbres correspondants à ces FND et FNC. $\diamond$

EXERCICE 3.13 Les formules peuvent être écrites en notation infixe (cours) ou préfixe, et on omet parfois les parenthèses. Parmi les mots suivants, écrits en notation préfixe, reconnaître ceux qui sont des formules, les écrire en notation infixe et dessiner l'arbre syntaxique les représentant.

1. $\supset \neg p \vee qr$;
2. $\supset \neg pq \vee r$;
3. $\supset \neg \supset \vee \wedge p q r s \vee st$;
4. $\supset \vee \neg p q r$;
5. $\supset \neg ps \vee qr$. ◇

EXERCICE 3.14 Construire la table de vérité de XOR . ◇

EXERCICE 3.15 On suppose un univers partitionné en 4 types d'individus

- chat heureux
- chat malheureux
- souris heureuse
- souris malheureuse

On choisit une espèce (chat , souris) et une attitude (heureux, malheureux) ; on dit qu'un type est OK s'il a soit l'espèce choisie, soit l'attitude choisie, mais pas les deux. Sachant qu'un chat malheureux est OK, quel(s) autre(s) type(s) sont OK ? ◇

EXERCICE 3.16 (du livre de Lassaing de Rougemont) 1) Soit la formule F suivante : $pq\bar{r} + p\bar{q}r + \bar{p}qr$. F et $\neg F$ sont-elles satisfaisables ? Sont-elles des tautologies ?

2) Mettre F et $\neg F$ sous forme normale conjonctive (ou clausale) puis disjonctive.

3) Trouver une formule G telle $(F \wedge G) \vee (\neg F \wedge \neg G)$ soit une tautologie.

4) Soit F_1 obtenue en substituant $\bar{p}$ à p dans F . F_1 est-elle conséquence de F ? F est-elle conséquence de F_1 ? ◇

EXERCICE 3.17 Soient p, q, r des propositions. Exprimer par des formules

1. Si p alors q . 2. Si p alors q sinon r . 3. p est une condition nécessaire pour que q soit vraie.
4. p est une condition suffisante pour que q soit vraie. 5. q si p . 6. q seulement si p .

Mettre les formules que vous avez trouvées en FND (forme normale disjonctive). ◇

EXERCICE 3.18 Soient $\mathcal{F}, \mathcal{G}$ deux ensembles de formules. On dit que $\mathcal{F} \models \mathcal{G}$ si et seulement si pour toute interprétation I , on a

$$I(F) = \text{vrai pour toute } F \in \mathcal{F}$$

implique que

$$I(G) = \text{vrai pour toute } G \in \mathcal{G} .$$

Montrer que $(\mathcal{F} \models \mathcal{F}' \text{ et } \mathcal{F}' \models \mathcal{F}'')$ implique que $\mathcal{F} \models \mathcal{F}''$. ◇

EXERCICE 3.19 Soient $G = p \supset (q \supset r)$ et $F = (p \supset q) \supset r$. 1. A-t-on $\{F\} \models \{G\}$? 2. A-t-on $\{G\} \models \{F\}$? ◇

EXERCICE 3.20 Ecrire, en utilisant les symboles de prédicats du monde de Tarski, des formules exprimant que :

1. **e** et **d** sont entre **b** et **a**.
2. Ni **e** ni **d** ne sont entre **b** et **a**.

3. Il est faux que **e** et **d** sont entre **b** et **a**. ◇

EXERCICE 3.21

1. Ecrire, en utilisant les symboles de prédicats du monde de Tarski, **Small**, **Large**, **Cube**, et uniquement ceux-ci, des formules exprimant que :

- 1.a **a** est petit, ou **c** et **d** sont grands.
- 1.b **a** est petit et **c** est grand, ou **a** est petit et **d** est grand.
- 1.c Soit **a** et **e** sont des cubes, soit **a** et **f** sont des cubes.

2. Les formules trouvées en 1.a et 1.b sont-elles équivalentes ? Justifiez votre réponse.

3. La formule $\text{Cube}(\mathbf{a}) \wedge (\text{Cube}(\mathbf{e}) \vee \text{Cube}(\mathbf{f}))$ est-elle équivalente à la formule trouvée en 1.c ? Justifiez votre réponse. ◇

EXERCICE 3.22 Les formules du premier ordre seront définies inductivement (en notation infixe) par :

- Si R est un symbole de relation d'arité n , et si $t_1, \dots, t_n \in T$, alors $R(t_1, \dots, t_n)$ est une formule.
- Si F et F' sont des formules, alors $\neg F$, $(F \supset F')$, $(F \wedge F')$, et $(F \vee F')$ sont des formules.
- Si F est une formule et x est une variable, alors $(\forall x F)$ et $(\exists x F)$ sont des formules.

1. Dessiner l'arbre représentant la formule $\neg \left(\left((\forall x P(x)) \wedge Q(x) \right) \supset \left(R(x) \vee \neg P(x) \right) \right)$. Ecrire cette formule en notation préfixe (c'est-à-dire en mettant les opérateurs $\wedge, \supset$, etc. d'abord).

2. Donner une définition inductive de la fonction transformant la notation infixe en notation préfixe. Pouvez-vous modifier cette définition inductive pour obtenir une fonction transformant la notation infixe en un arbre représentant la formule ? ◇

EXERCICE 3.23 On considère un langage avec le prédicat unaire p et les prédicats binaires r et s .

1) Déterminer les variables libres et liées dans les formules 1 à 5 suivantes :

- 1. $(\forall x p(x)) \vee (\exists y p(y))$
- 2. $(\forall x p(x)) \wedge (\exists x p(x))$
- 3. $(\forall x p(x)) \wedge (\exists y r(x, y))$
- 4. $(\forall x \exists y s(x, y)) \supset (\exists y \forall x r(x, y))$
- 5. $(\forall x s(x, y)) \supset (\exists y r(x, y))$

2) Mettre sous forme prénexe les formules 1 à 5 précédentes ; combien de formes prénexes différentes peut-on trouver pour chacune des fomules (on dira que deux formes prénexes sont différentes si on ne peut pas passer de l'une à l'autre par un renommage des variables) ? ◇

EXERCICE 3.24 On considère un langage avec les prédicats binaires $=, \geq$ et $<$. Soient F et G les formules $\forall y(y \geq x) \wedge \forall x \exists y(x < y)$ et $(x + x = x)$.

1. Mettre sous forme prénexe la formule $F \wedge G$.

2. Montrer que $F \wedge G$ est satisfaisable. L'interprétation I telle que $I, v \models F \wedge G$ pour une valuation v est-elle un modèle de $F \wedge G$. ◇

EXERCICE 3.25 On considère un langage avec les prédicats unaires *prime*, *even*, *odd* (premier, pair et impair) et les prédicats binaires $\geq$ et $|$ (divise).

1. Traduire en formules logiques : "certains nombres premiers sont pairs" et "tous les nombres premiers supérieurs à 5 sont impairs". Expliquer pourquoi les formules $\exists x(\text{prime}(x) \supset \text{even}(x))$

, $(\exists x(\text{prime}(x))) \supset \text{even}(x)$ et $\forall x(\text{prime}(x) \wedge (x \geq 5) \wedge \text{odd}(x))$, $(\forall x(\text{prime}(x) \wedge (x \geq 5))) \wedge \text{odd}(x)$ ne sont pas des traductions correctes des énoncés précédents.

2. Mêmes questions pour "il existe un nombre pair divisible par 9" et "tout nombre pair est divisible par 2", et les formules $\exists x(\text{even}(x) \supset (9|x))$, $\forall x(\text{even}(x) \wedge (2|x))$. $\diamond$

EXERCICE 3.26 On se donne un langage comprenant deux symboles de prédicats (ou relations) binaires R et $=$ (= sera toujours interprété comme l'égalité).

1. Ecrivez une formule du calcul des prédicats exprimant que la relation binaire R est une relation d'ordre large.

2. Ecrivez une formule du calcul des prédicats exprimant que l'ordre R a un minimum.

3. Ecrivez une formule du calcul des prédicats exprimant que l'ordre R a un élément minimal. $\diamond$

EXERCICE 3.27 On considère un langage avec les prédicats unaires prime , even , la constante 1, et la fonction unaire s (successeur). Traduire en formules logiques les énoncés :

1. tout nombre pair est premier

2. aucun nombre pair n'est premier

3. certains nombres premiers sont pairs

4. certains nombres premiers ne sont pas pairs

5. tout nombre premier est soit pair, soit égal à 2

Parmi les formules obtenues, lesquelles sont vraies dans $\mathbb{N}$. Justifier les réponses. $\diamond$

EXERCICE 3.28 1. Précisez à l'aide d'un quantificateur le sens de "un" dans les phrases suivantes et formalisez-les en logique des prédicats :

a) Jean suit un cours

b) Un français a été champion du monde de cyclisme

c) Un entier naturel est pair ou impair

d) Un enseignant a toujours un nouveau sujet à étudier

e) Dans un triangle isocèle une médiane est également hauteur

f) Dans un triangle équilatéral une médiane est également hauteur

g) Un homme a besoin d'avoir un idéal

2. Formaliser les énoncés suivants dans le langage de la logique des prédicats du premier ordre :

a) Tous les hommes sont doués de raison

b) Seuls les hommes sont doués de raison

c) Aucun homme n'est doué de raison

d) Tous les poissons, sauf les requins, sont gentils avec les enfants

e) Chaque individu aime quelqu'un et personne n'aime tout le monde

f) Tout homme a des amis et des ennemis

g) N'importe qui peut apprendre la logique, s'il travaille assez

3. Représenter la phrase "Tout nombre entier naturel x a un successeur qui est inférieur ou égal à tout entier strictement supérieur à x " par une formule de la logique des prédicats en utilisant les prédicats suivants :

$\text{entier}(x)$: " x est un entier naturel"

$\text{successeur}(x, y)$: " x est successeur de y "

$\text{inf}(x, y)$: " x est inférieur ou égal à y "

Vous pouvez ajouter un prédicat $\text{egal}(x, y) : "x = y"$ si vous le souhaitez.

Soit la formule : $\forall x(p(x) \supset \exists y(q(y, x) \wedge \forall z(r(z, x) \supset s(z, y))))$. Est-elle satisfaisable ? Est-elle universellement valide (tautologie) ? $\diamond$

EXERCICE 3.29 Écrire, en utilisant les symboles de prédicats du monde de Tarski, des formules traduisant

1. Un dodécaèdre est grand.
2. Tous les dodécaèdres sont grands.
3. Tous les dodécaèdres ne sont pas grands. Remarquer que cette assertion est ambiguë : on peut la traduire par une formule commençant par $\forall$ ou bien par une formule commençant par $\neg$. $\diamond$

EXERCICE 3.30 Écrire, en utilisant les symboles de prédicats du monde de Tarski, des formules traduisant

1. Il y a au moins deux cubes.
2. Il y a au plus deux cubes.
3. Il y a exactement deux cubes. $\diamond$

EXERCICE 3.31 On se place dans le calcul des prédicats. Les formules sont définies inductivement par :

- (B) Si R est un symbole de relation d'arité n , et si $t_1, \dots, t_n \in T$, alors $R(t_1, \dots, t_n)$ est une formule.
- (I) Si F et F' sont des formules, alors $\neg F$, $(F \supset F')$, $(F \wedge F')$, $(F \vee F')$, $\forall x F$ et $\exists x F$ sont des formules.

1. Donnez une définition inductive de l'ensemble des variables libres dans une formule F .

On notera $V(t)$ l'ensemble des variables figurant dans le terme t . On notera $L(F)$ l'ensemble des variables libres dans la formule F .

2. Donnez une définition inductive de l'ensemble $B(F)$ des variables liées dans une formule F . $\diamond$

EXERCICE 3.32 Soient les formules

- (1) $\forall x(\neg R(x, x) \wedge (\forall y \forall z(R(x, y) \wedge R(y, z) \supset R(x, z))))$
- (2) $\exists x \forall y R(x, y)$
- (3) $\forall x \exists y R(x, y)$
- (4) $\exists x \forall y R(y, x)$
- (5) $\forall x \exists y R(y, x)$
- (6) $\forall x \exists y(R(x, y) \wedge \forall z(R(x, z) \supset (z = y \vee R(y, z))))$
- (7) $\forall x \forall y(R(x, y) \supset \exists z(R(x, z) \wedge R(z, y)))$

Dites si oui ou non chacune des formules ci-dessus est valide dans les interprétations ci-dessous

- (i) les entiers naturels (positifs) comme domaine et $<$ pour R .
- (ii) les entiers relatifs comme domaine et $<$ pour R .
- (iii) les rationnels comme domaine et $<$ pour R .
- (iv) l'ensemble des parties de $\mathbb{N}$ comme domaine et $\subset$ (l'inclusion stricte) pour R .
- (v) l'ensemble des parties de $\mathbb{N}$ comme domaine et $\subseteq$ (l'inclusion large) pour R .

Montrer que $\exists x \forall y R(x, y) \supset \forall x \exists y R(y, x)$ et $\exists x \forall y R(y, x) \supset \forall x \exists y R(x, y)$. Les réciproques sont-elles vraies ? $\diamond$

EXERCICE 3.33 Un monde de Tarski infini. On représente la grille du monde de Tarski par l'ensemble $\mathbb{N} \times \mathbb{N}$ des coordonnées ligne et colonne. La case (0,0) est en haut à gauche. Une figure est représentée par son nombre de face (4, 6 ou 12) et sa taille par l'entier 0, 1 ou 2.

1. donnez une définition d'un ensemble T permettant de représenter une figure dans le monde de Tarski.
2. proposez une interprétation des prédicats **Tet Cube Dodec Small Medium Large Smaller Larger LeftOf RightOf BackOf FrontOf Between**
3. écrire, en utilisant les symboles de prédicats du monde de Tarski, et uniquement ceux-ci, les définitions des formules suivantes :
 - (i) $LessFig[x, y]$ pour “ x est une figure strictement plus petite que y ” (fonction du nombre de faces).
 - (ii) $EqFig[x, y]$ pour “ x et y sont la même figure”.
 - (iii) $EqSize[x, y]$ pour “ x est de la même taille que y ”.
 - (iv) $\Phi[x, y]$ qui donne l'ordre lexicographique (strict) sur l'ensemble T (on utilisera les définitions ci-dessus, et le nombre de faces est prioritaire par rapport à la taille, c'est-à-dire qu'un petit dodécaèdre est supérieur à un grand carré par exemple). $\diamond$

EXERCICE 3.34 On considère les formules suivantes :

$$F_1: \forall x \forall y (LeftOf(x, y) \supset RightOf(y, x))$$

$$F_2: \forall x \forall y ((Small(x) \wedge Small(y) \wedge BackOf(x, y)) \supset Dodec(x))$$

$$F_3: \forall x \forall y ((Cube(x) \wedge Cube(y) \wedge x \neq y) \supset (Larger(x, y) \vee Smaller(x, y)))$$

$$F_4: \forall x \forall y \forall z ((Smaller(x, y) \wedge Smaller(y, z)) \supset Smaller(x, z))$$

$$F_5: \forall x \forall y (Larger(x, y) \supset x \neq y)$$

$$F_6: \forall x \forall y \forall z (Between(x, y, z) \vee \neg Between(x, y, z))$$

$$F_7: \forall x \forall y \forall z (Between(x, y, z) \vee \neg Between(x, z, y))$$

1. Quelles sont les formules qui sont valides dans tous les mondes de Tarski's world ? Pour chaque formule qui n'est pas valide dans tous les mondes de Tarski's world dites comment construire un monde qui falsifie la formule.
2. Quelle formule est une tautologie ?
3. Pour chaque formule qui est valide dans tous les mondes de Tarski's world mais qui n'est pas une tautologie, dites comment changer l'interprétation des prédicats pour falsifier la formule. $\diamond$

EXERCICE 3.35 Formaliser en logique des prédicats du premier ordre chacune des phrases du texte suivant emprunté à Lewis Carroll :

Aucune des choses que quelqu'un rencontre en mer et qu'il ne remarque pas n'est une sirène. Les choses qui sont rencontrées en mer et qu'on inscrit sur le livre de bord sont toujours dignes d'intérêt. Personnellement, je n'ai jamais rien rencontré, au cours de mes voyages en mer, qui soit digne d'intérêt. Les choses que quelqu'un rencontre en mer et qu'il remarque sont toujours inscrites sur le livre de bord.

Que peut-on dire des choses rencontrées en mer ? Formaliser votre réponse. $\diamond$

STRUCTURES DISCRÈTES – T.D. LANGAGES ET AUTOMATES

LANGAGES ET MORPHISMES

EXERCICE 4.1 Soit A l'alphabet $\{a, b\}$ et B l'alphabet $\{a, b, c\}$. On note P l'ensemble des mots sur A formé par ε et les mots commençant par a et sans facteur bbb . Vérifier que P est un sous-monoïde de A^* .

Soit ϕ l'homomorphisme de B^* dans A^* défini par $\phi(a) = abb, \phi(b) = ab, \phi(c) = a$. Montrer que c'est un isomorphisme de B^* sur P .

Rappel : par définition, un isomorphisme de monoïdes est un homomorphisme bijectif dont l'inverse est un homomorphisme. Cependant tout homomorphisme de monoïdes qui est bijectif est un isomorphisme. $\diamond$

EXERCICE 4.2 Soit A un alphabet contenant au moins les lettres a et b .

- 1) Donner un homomorphisme de A^* dans $(A \cdot A)^*$.
- 2) Donner un homomorphisme de A^* dans $(A \setminus \{b\})^*$.
- 3) Y a-t-il un isomorphisme de monoïdes entre $(A \setminus \{a\})^*$ et $(A \setminus \{b\})^*$? $\diamond$

EXERCICE 4.3 1) Montrer que $(\mathcal{P}(A^*), \cdot, \{\varepsilon\})$ est un monoïde.

2) Montrer que si $(L_i)_{i \in I}$ est une famille quelconque de langages, alors

$$\left(\bigcup_{i \in I} L_i\right) \cdot L = \bigcup_{i \in I} L_i \cdot L,$$

- 3) Montrer que $L^* = (L + \{\varepsilon\})^*$ et que $L^* = \{\varepsilon\} + L \cdot L^*$.
- 4) Montrer que $\emptyset^* = \{\varepsilon\}$. $\diamond$

EXERCICE 4.4 Soit $X = \{b\}$ et $Y = (A \setminus \{b\}) \cdot \{b\}^*$.

- 1) Décrire informellement les éléments de X^* , Y et Y^* .
- 2) Montrer que tout mot de A^* commençant par une lettre distincte de b appartient à Y^* .
- 3) Montrer que tout mot u de A^* s'écrit de façon unique sous la forme $u = vw$, où $v \in X^*$ et $w \in Y^*$. $\diamond$

EXERCICE 4.5 **Lemme de Lévi :** Soient $u, v, x, y \in A^*$. Montrer que $uv = xy$ si et seulement si l'un des trois cas suivants est vérifié :

- (i) $|u| = |x|$, $u = x$, et $v = y$.
- (ii) $|u| < |x|$ et il existe $t \in A^*$ tel que $ut = x$ et $v = ty$.
- (iii) $|u| > |x|$ et il existe $t \in A^*$ tel que $u = xt$ et $tv = y$. $\diamond$

AUTOMATES COMPLETS ET DÉTERMINISTES

EXERCICE 4.6 Expliquez pourquoi l'automate suivant sur $\{a, b\}$ n'est pas complet. Quel langage reconnaît-il ? Donnez un automate complet équivalent. $\diamond$

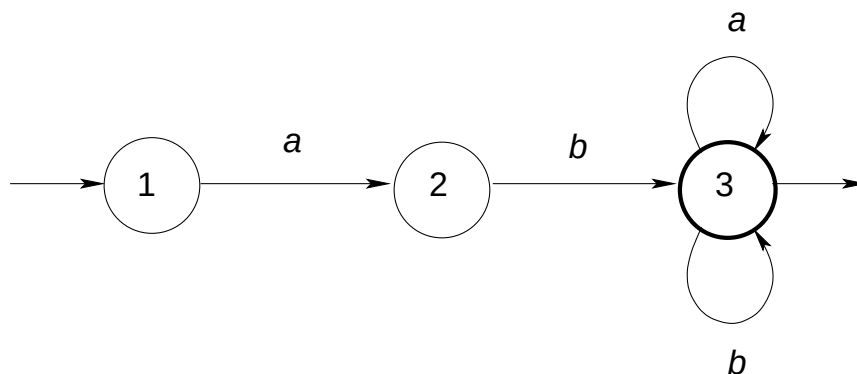


FIGURE 4.1 Automate non complet

EXERCICE 4.7 Représenter l'automate $\mathcal{A}$ sur l'alphabet $\{a, b, c\}$ d'états 0, 1, 2, 3, d'état initial 0, d'état terminal 3 et de 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 ? déterministe ? justifier.
- Les mots *baba* et *cabcb* sont-ils reconnus par $\mathcal{A}$?
- Décrire $L(\mathcal{A})$ en langage ordinaire, puis en donner une expression rationnelle. $\diamond$

CONSTRUCTION D'AUTOMATES

EXERCICE 4.8 Construire un automate déterministe reconnaissant le langage fini $\{a, ba, aba, bab, bbba\}$. $\diamond$

EXERCICE 4.9 Soit $A = \{a, b, c\}$. Donner des automates déterministes complets reconnaissant les langages suivants :

- 1) L'ensemble des mots de longueur paire.
- 2) L'ensemble des mots où le nombre d'occurrences 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 par un "c". $\diamond$

OPÉRATIONS

EXERCICE 4.10 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 $\mathcal{A}_1$, $\mathcal{A}_2$ et les automates déterministes construits sont-ils complets ? Que remarquez-vous ? $\diamond$

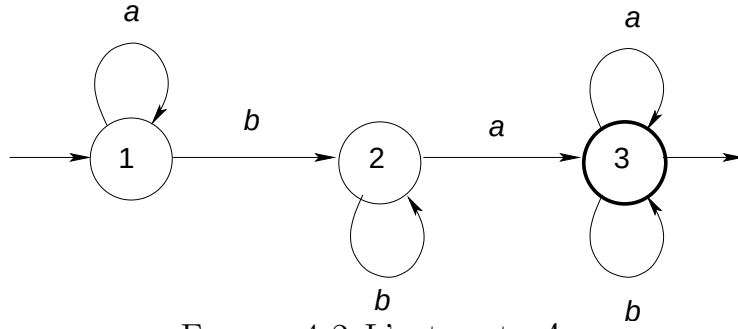


FIGURE 4.2 L'automate $\mathcal{A}_1$

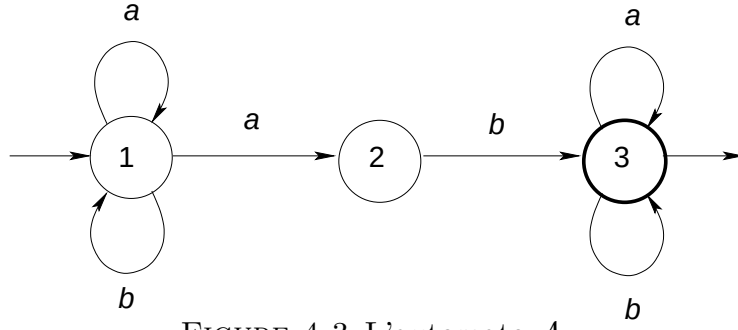


FIGURE 4.3 L'automate $\mathcal{A}_2$

EXERCICE 4.11 Soit $A = \{a, b\}$; Soient L_1 le langage comprenant tous les mots contenant un nombre pair de b et L_2 le langage comprenant tous les mots contenant un nombre impair de a .

1. Donner pour chaque L_i un automate déterministe complet A_i reconnaissant L_i et exprimer L_i sous forme d'expression rationnelle.
2. Construire à partir des A_i un automate déterministe reconnaissant $L_1 \cap L_2$.
3. Construire à partir des A_i un automate déterministe reconnaissant $L_1 \cup L_2$. $\diamond$

EXERCICE 4.12 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. $\diamond$

LANGAGES RECONNAISSABLES

EXERCICE 4.13 Soit L un langage et $\text{Pref}(L) = \{u \in A^* / \exists v \in A^* : uv \in L\}$ l'ensemble des préfixes des mots de ce langage L . Montrer que si un langage L est reconnaissable l'ensemble $\text{Pref}(L)$ est aussi reconnaissable. $\diamond$

EXERCICE 4.14 **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 de m), on a $m = \alpha u \beta$ avec

- (i) $u \neq \varepsilon$,
- (ii) $|u| < N_0$,
- (iii) $\alpha u^* \beta \subset L$. $\diamond$

EXERCICE 4.15 Montrer que les langages suivants sur l'alphabet $\{a, b\}$ ne peuvent pas être reconnus par un automate fini.

$$L_1 = \{a^n b^n, n \geq 0\},$$

$$L_2 = \{a^p, p \text{ premier}\}.$$

$\diamond$

EXERCICE 4.16 Les langages ci-dessous sont-ils reconnaissables ?

$$L = \{a^n b^p \mid n, p \geq 0, n = p \bmod 3\}$$

$$L' = \{a^m b^n \mid m \geq 1, n \geq 1, m \neq n\}$$

◇

AUTOMATES MINIMAUX ET EXPRESSIONS RATIONNELLES

EXERCICE 4.17 Soit l'alphabet $A = \{a, b\}$ et soient L_1 le langage comprenant tous les mots contenant "aa" et L_2 le langage comprenant tous les mots ne contenant pas "bb".

1. Donner pour chaque L_i un automate déterministe complet A_i reconnaissant L_i .
2. Construire à partir des A_i un automate déterministe et complet $\mathcal{A}$ reconnaissant $L_1 \cap L_2$.
3. $\mathcal{A}$ est-il minimal ? Justifiez votre réponse et construisez un automate minimal $\mathcal{B}$ reconnaissant $L_1 \cap L_2$.

◇

EXERCICE 4.18 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 $\mathcal{A}$. Minimiser $\mathcal{A}$.

◇

EXERCICE 4.19 Soit $A = \{a, b, c\}$. Construire un automate non déterministe qui reconnaît l'ensemble L des mots de longueur paire qui se terminent par ab . En déduire un automate déterministe complet reconnaissant L .

◇

EXERCICE 4.20 Construire un automate déterministe complet minimal reconnaissant l'ensemble des mots sur $\{a, b, c\}$ comportant au moins trois lettres et dont la troisième lettre à partir de la fin est un "a" ou un "c".

◇

EXERCICE 4.21 Donner les automates déterministes complets minimaux reconnaissant les langages sur l'alphabet $A = \{a, b\}$ donnés par les expressions rationnelles suivantes

$$1) (a + b)^* b (a + b)^*.$$

$$2) ((a + b)^2)^+ + ((a + b)^3)^+.$$

$$3) ba^* + ab + (a + bb)ab^*.$$

◇

SYSTÈMES

EXERCICE 4.22 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 $L_{I,F}$, pour donner une expression rationnelle dénotant L .

◇

EXERCICE 4.23 Soit $A = \{a, b\}$ et L le langage comprenant tous les mots ayant 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 pour L .

◇

EXERCICE 4.24 Soit L le langage sur l'alphabet $A = \{a, b\}$ 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 pour L .

◇

EXERCICE 4.25 Soit $A = \{a, b, c, d\}$ et L le langage comprenant tous les mots où tout "a" est suivi d'un "b" et tout "c" est suivi d'un "b". Donner l'automate déterministe complet minimal reconnaissant L et résoudre le système d'équations correspondant ; en déduire une expression rationnelle pour L .

◇

EXERCICE 4.26 1) Soit l'automate $\mathcal{A}$ d'états 0, 1, d'état initial 0, d'état terminal 1 et de transitions : $(0, a, 0)$, $(0, b, 1)$, $(1, a, 0)$ et $(1, b, 1)$. (Dessiner l'automate $\mathcal{A}$.) Soit L le langage reconnu par $\mathcal{A}$. Donner le système d'équations associé à $\mathcal{A}$ et en déduire une expression rationnelle pour L .

2) Mêmes questions avec $\mathcal{B}$ d'états 0, 1, 2 d'état initial 0, d'état terminal 0 et de transitions : $(0, a, 1)$, $(0, b, 0)$, $(0, c, 0)$, $(1, b, 1)$, $(1, c, 2)$, $(2, a, 2)$, $(2, b, 0)$, $(2, c, 1)$. $\diamond$

T.M.E. LOGIQUE

Machine : sous WINDOWS, ENSEIGNEMENT

Tarski se trouve dans *tout le réseau / Enseignement / C/ Program Files*

Faire une copie du fichier d'exercices dans *votre directory personnel h* : vous ne pourrez travailler que sur cette copie.

Tarski's world est un petit logiciel illustrant de façon simple et récréative les notions de *modèle* et d'*interprétation* des formules du calcul des prédicats.

5.1 Présentation

Domaine d'interprétation

Le “monde” servant de support au domaine d'interprétation de formules est une grille carrée de 8 sur 8 sur laquelle on peut disposer trois sortes de figures géométriques de tailles différentes. Les figures sont : la pyramide, le cube et le dodécaèdre. Les tailles sont : petit, moyen et grand.

Langage

Pour décrire ce monde géométrique, le logiciel propose un *langage* réduit (ensemble de symboles de variables, constantes et prédicats).

- 6 symboles de constantes : `a b c d e f`
- 6 symboles de variables : `u v w x y z`
- 13 symboles de prédicats dont
 - 6 symboles unaires : `Tet Cube Dodec Small Medium Large`
 - 6 symboles binaires : `Smaller Larger LeftOf RightOf BackOf FrontOf`

T.S.V.P.

- 1 symbole ternaire : **Between**

La sémantique attendue des symboles est indiquée par la signification du mot anglais utilisé pour les désigner. Pour en être sûr, vous pouvez déterminer la sémantique des symboles de prédicat en saisissant pour chacun d’eux une formule atomique et en construisant un monde où elle est valide et un monde où elle est fausse. A-t-on par exemple **Smaller**(a,b) si a,b sont tous deux *small*, ou bien **FrontOf**(a,b) si a,b sont sur une même horizontale ?

Ergonomie

Le logiciel offre une interface graphique aux fonctionnalités usuelles (usage de la souris, menus, etc.). Il présente à l’utilisateur quatre fenêtres :

1. une représentation graphique de la grille offrant, sur la gauche, les trois figures possibles. Cette fenêtre sert à construire un domaine d’interprétation. Au lancement du programme, cette fenêtre est intitulée **NEW0.WLD**.
2. un “clavier” dont chaque touche permet l’édition des symboles logiques (quantificateurs, connecteurs) et des symboles du langage (variables, constantes, prédicats).
3. une zone de saisie des formules séparées l’une de l’autre par un trait horizontal. L’utilisateur peut éditer une formule à l’aide du clavier du logiciel ou directement (clavier de sa console). Au lancement du programme, cette fenêtre est intitulée **NEW0.SEN**.
4. la dernière fenêtre, intitulée **Inspector** permet deux catégories d’opérations : l’édition des caractéristiques d’une figure ; la vérification de la validité d’une formule (fenêtre 3) dans le monde représenté dans la fenêtre 1. Le choix de l’une ou l’autre de ces opérations est accessible par un petit menu en haut de la fenêtre. L’option **Block** correspond à l’édition des figures ; l’option **Sentence** correspond à la vérification des formules.

COMPLÉMENT : la figure suivante illustre l’interprétation du prédicat **Between**, les cases grisées sont celles qui satisfont **Between**(z, x, y) si x et y sont représentés par les points noirs.

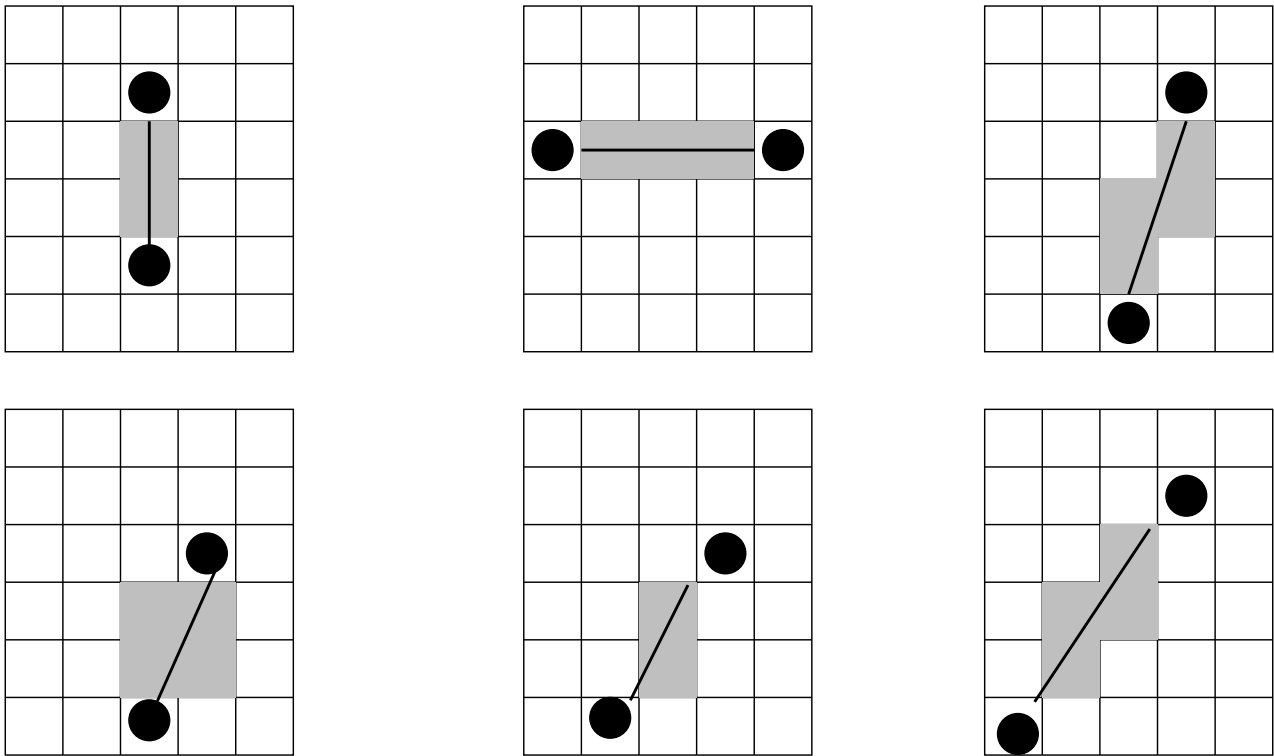


FIGURE 5.1 Interprétation de **Between** : les cases grisées sont les z tels que $\text{Between}(z, x, y)$

Construction d'un monde

- En cliquant sur l'une des figures à gauche de la fenêtre, vous créez une nouvelle instance de cette figure sur la grille.
- Par “glisser-déposer” vous pouvez déplacer les figures d'une case à l'autre. Deux figures ne peuvent cohabiter sur une même case.
- En mode **Block** de la fenêtre **Inspector**, vous pouvez modifier les attributs d'une figure préalablement sélectionnée par un clic. On peut modifier la nature des figures, leur taille ou nommer une figure en utilisant un des symboles de constante (**a**, **b**, **c**, **d**, **e** ou **f**)

REMARQUE 5.1 Une même figure peut recevoir plusieurs noms (symboles de constantes).

Validation d'une formule

Il y a trois niveaux de vérifications :

- correction syntaxique : **wff** i.e. *Well-Formed-Formula* ou encore formule bien formée selon les règles d'écriture des formules ;
- énoncé clos (ou non) : **sentence** (ou non) ;
- vérité : la formule est vraie dans le modèle considéré.

Pour chacun de ces niveaux, deux réponses sont possibles (**yes** ou **no**) que l'on sélectionne en cochant la réponse attendue. Un bouton **Verify** permet de demander au logiciel de

contrôler la validité de vos choix. Une croix est affichée sur la ligne correspondante lorsque la réponse proposée est erronée et un signe ressemblant à un V est affiché sur la ligne correspondante lorsque la réponse proposée est correcte.

SYNTAXE les expressions sont *complètement* parenthésées - sauf lorsque le connecteur est associatif (conjonction ou disjonction)

VALIDITÉ seuls les *énoncés clos* peuvent être vrais ou faux.

Le jeu de la vérité

Le logiciel propose un mode “pas-à-pas” pour vérifier la validité d’une formule : le bouton **Game** de la fenêtre **Inspector**. À chaque étape de validation (ou d’invalidation) on peut vous demander de sélectionner une sous-formule, de sélectionner ou de nommer une figure du monde. La succession des étapes constitue une *preuve* de la vérité ou la fausseté de l’énoncé traité. En fait, cette preuve est partielle : par exemple, tous les cas d’une conjonction ne seront pas examinés.

REMARQUE 5.2 avant de pouvoir utiliser cette fonctionnalité, il faut avoir fait vérifier (bouton **Verify**) la vérité ou la fausseté de la formule visée.

Mondes prédéfinis

Le logiciel propose un certain nombre de mondes ou d’ensemble de formules prédéfinis accessibles par le menu **File**, option **Open**, puis **World** ou **Sentence**. Les mondes prédéfinis sont dans le sous répertoire **exercice**.

5.2 Exercices - Calcul propositionnel

Les 3 exercices suivants sont tirés du livre Tarski’s World de Barwise et Etchemendy

EXERCICE 5.1 Chargez le fichier **DeMorg.sen**. Réaliser un monde qui rende toutes les formules vraies. ◇

EXERCICE 5.2 1. Chargez le fichier **Boole.wld**. Créez un fichier **formules.sen** où vous mettrez des formules exprimant que :

1. **f** (le grand dodécaèdre au fond) n’est pas devant **a**
2. **f** est à droite de **a** et à gauche de **b**
3. **f** est soit derrière **a** soit plus petit que **a**.
4. **e** et **d** sont entre **c** et **a**.
5. Ni **e** ni **d** ne sont plus grands que **c**.
6. **e** n’est ni plus grand ni plus petit que **d**.
7. **c** est plus grand que **e** et plus petit que **a**.
8. **c** est plus petit que **f** et de plus il est devant **f**.

2. Changez le monde de manière à ce que toutes vos formules 1 – 8 deviennent fausses dans ce nouveau monde. Vous pourrez commencer par déplacer **f** dans le coin avant de la grille. Ensuite déplacez **e** dans le coin arrière gauche de la grille et faites-en un grand objet. Vérifiez toutes

vos formules 1 – 8 dans ce monde. Si vous les avez formulées correctement elles devraient être toutes fausses. Si l’une d’elles est encore vraie, pouvez vous trouver votre erreur. $\diamond$

EXERCICE 5.3 1. Créez un nouveau fichier de formules et inscrivez-y la traduction en logique du premier ordre des phrases suivantes :

1. **a** est petit ou **c** et **d** sont grands.
2. **d** et **e** sont tous deux derrière **b**.
3. **d** et **e** sont tous deux derrière **b** et sont tous deux plus grands que **b**.
4. **c** et **d** sont tous deux des cubes et de plus aucun des deux n’est petit.
5. Ni **e** ni **a** n’est à droite de **c** et à gauche de **b**.
6. Soit **e** n’est pas grand, soit **e** est derrière **a**.
7. **c** n’est ni entre **a** et **b**, ni en avant de **a**, ni en avant de **b**.
8. Soit **a** et **e** sont des tétraèdres, soit **a** et **f** sont des tétraèdres.
9. Ni **c** ni **d** n’est devant soit **c**, soit **b**.
10. **c** est entre **d** et **f** ou plus petit que **d** et plus petit que **f**.

Enregistrez cet ensemble de propositions sous le nom `exo3.sentences`.

2. Chargez le fichier `Wittgenstein.wld`. Vérifiez que toutes les propositions écrites en français dans la question 1. sont vraies dans ce monde. Vérifiez que vos traductions sont toutes valides ; si ce n’est pas le cas corrigez-les.

3. Modifiez le monde de `Wittgenstein.wld` comme suit : réduisez à la taille *small* tous les objets moyens ou grands, et agrandissez à la taille maximale tous les petits objets. Après ces changements, les formules 1, 3, 4 et 10 deviennent fausses, et toutes les autres restent vraies. Vérifiez qu’il en est bien ainsi dans votre fichier `exo3.sentences` (si ce n’était pas le cas, corrigez vos formules). Ensuite faites pivoter de 90 degrés vers la droite le monde ainsi modifié : les formules 5, 6, 8 et 9 devraient être les seules à être vraies. Corrigez vos formules si ce n’était pas le cas.

Chargez le fichier `Boole.wld` : seule la formule 6 doit être vraie dans ce monde. Corrigez vos formules si ce n’était pas le cas. Modifiez `Boole.wld` en échangeant les positions de **b** et **c**. Les formules 2, 5, 6 et 7 deviennent vraies et les autres restent fausses. Corrigez vos formules si ce n’était pas le cas. $\diamond$

5.3 Exercices - Calcul des prédicats

Les 12 exercices suivants sont tirés du livre Tarski’s World de Barwise et Etchemendy

EXERCICE 5.4

1. Ouvrir `Edgar.sen`. Evaluer les formules dans `Edgar.wld`.
2. Quelle formule traduit “il y a un grand tétraèdre” ?
3. Quelle formule traduit “il y a un cube entre **a** et **b**” ?
4. Exprimer en français les assertions faites par les formules 5 et 6. $\diamond$

EXERCICE 5.5

1. Ouvrir `Allan.sen`.
2. Quelle formule traduit “un dodécaèdre est grand” ? Construisez des mondes pour vérifier votre choix.

3. Quelle formule traduit “tous les tétraèdres sont petits” ? Construisez des mondes pour vérifier votre choix.
4. Construisez un monde où la formule 3 soit fausse et la formule 4 soit vraie.
5. Construisez un monde où les formules 2 et 4 soient vraies et les formules 1 et 3 soient fausses.
6. Peut-on construire un monde où la formule 1 soit vraie et la formule 2 soit fausse. Si oui construisez-le, si non expliquez pourquoi. $\diamond$

EXERCICE 5.6 Ouvrir un nouveau fichier de suffixe `.sen` et y mettre les formules traduisant les assertions suivantes (chaque formule contiendra un seul quantificateur qui sera universel) :

1. Tous les cubes sont petits.
2. Chaque petit cube est à droite de **a**.
3. Tous les dodécaèdres sont grands.
4. **a** est à gauche de tout dodécaèdre.
5. Tout tétraèdre moyen est devant **b**.
6. Chaque cube est devant **b** ou derrière **a**.
7. Tout cube est à gauche de **b** et à droite de **a**.
8. Tout ce qui se trouve entre **b** et **a** est un cube.
9. Tout ce qui est plut petit que **a** est un cube.
10. Tous les dodécaèdres ne sont pas petits. Remarquer que cette assertion est ambiguë : on peut la traduire par une formule commençant par $\forall$ ou bien par une formule commençant par $\neg$. Vous choisirez la première formule, celle qui affirme que tous les dodécaèdres sont soit moyens soit grands.
11. Aucun dodécaèdre n’est petit.
12. **a** n’est pas à droite de tout objet. Remarquer que cette assertion est ambiguë : la traduire comme la négation de “**a** est à droite de tout”.
13. **a** n’est à droite d’aucun objet. Remarquer la différence avec l’assertion précédente.
14. **a** n’est pas à droite d’un cube.
15. Si un objet est un cube, alors cet objet est à gauche de **b** et à droite de **a**.
16. Un objet est un cube, si et seulement si cet objet est à gauche de **b** et à droite de **a**.

Sauvegardez votre fichier dans `exo5.6.sen`.

1. Ouvrez `Claire.wld`. Vérifiez que toutes vos formules sont vraies (sinon il y a une erreur dans vos formules que vous corrigerez).
2. Changez le fichier `Claire.wld` en déplaçant **a** dans l’angle droit au premier rang. Vérifiez que les formules 2, 4, 7, 13, 14, 15, 16 sont fausses et les autres vraies (sinon il y a une erreur dans vos formules que vous corrigerez).
3. Chargez le fichier `Wittgenstein.wld`. Remarquez que les assertions 2, 4, 8, 9, 12, 14 sont vraies et les autres fausses. S’il n’en est pas de même de vos formules, vous les corrigerez.
4. Chargez le fichier `Leibniz.wld`. Remarquez que les assertions 3, 4, 8, 9, 10, 11, 12, 13, 14 sont vraies et les autres fausses. S’il n’en est pas de même de vos formules, vous les corrigerez. $\diamond$

EXERCICE 5.7 1. Chargez le fichier `Leibniz.wld`. Ouvrez un nouveau fichier `exo5.7.sen` où vous mettrez des formules traduisant les assertions 1 à 5 suivantes. Chaque assertion que vous allez traduire est vraie dans ce monde. Vérifiez qu’il en est de même des formules que vous écrirez.

1. **b** est un tétraèdre plus petit que **e**.
 2. Il n'y a pas de cube moyen.
 3. Rien n'est devant **b**.
 4. Tout cube est soit devant soit derrière **e**.
 5. Aucun cube n'est entre deux objets (Il faut 3 quantificateurs pour traduire cette assertion).
1. Changez le monde de sorte que toutes les assertions deviennent fausses. Pour ce faire, transformez **b** en cube moyen, supprimez le tétraèdre le plus à gauche et mettez **b** à sa place, puis mettez un petit cube à l'ancienne place de **b**. Toutes vos formules doivent être fausses dans ce modèle ; si ce n'est pas le cas, vous les corrigerez.
 2. Faites divers changements au monde pour que certaines assertions soient vraies et d'autres fausses, et vérifiez que vos formules prennent la même valeur de vérité que les assertions. ◇

EXERCICE 5.8 1. Chargez le fichier **Ramsey.sen**. Les formules de 1 à 10 sont soit existentielles soit sans quantificateur : on peut donc les rendre vraies en ajoutant des objets nouveaux pour chaque formule, mais un des buts de l'exercice est d'ajouter le moins possible d'objets. Réaliser un monde qui rende les formules 1–10 vraies. On essaiera de faire un tout petit monde : 6 objets doivent suffire (il faudra que l'un des objets ait 2 noms).

2. Si le monde réalisé à la question 1) ne satisfait pas les formules 11–20 (formules universelles), modifiez-le *sans ajouter d'objet* pour que toutes les formules 1–20 soient vraies.

3. Changez le monde de la question 2), *uniquement en ajoutant* des objets pour falsifier le plus grand nombre possible de formules de **Ramsey.sen**. Quelles sont les formules que vous arrivez à falsifier, et quelles sont celles que vous n'arriverez pas à falsifier ? ◇

EXERCICE 5.9 1. Chargez le fichier **Peano.wld**. Créez un fichier **formules.sen** où vous mettrez des formules exprimant que :

1. *Tout dodécaèdre est petit.*
2. *Il y a un cube moyen.*
3. *Il y a au moins deux cubes.*
4. *Il y a un tétraèdre entre deux cubes.*
5. *Tout cube n'est pas devant un dodécaèdre.* Remarquer que cette assertion est ambiguë : la traduire comme la négation de “Tout cube est devant un dodécaèdre”.

2. Changez le monde en remplaçant le cube moyen par un dodécaèdre moyen. Toutes vos formules doivent devenir fausses dans ce nouveau monde. ◇

EXERCICE 5.10 1. Chargez le fichier **Buridan.sen**. Construire un monde où toutes les formules sont vraies et sauvegardez-le dans **exo5.10.wld**.

2. Les assertions suivantes sont des conséquences des formules de **Buridan.sen**, elles doivent donc être vraies dans **exo5.10.wld**.

1. *Il n'y a pas de cube.*
2. *Un tétraèdre n'est pas grand.* remarquer que cette assertion est ambiguë pour certains, vous choisirez de l'interpréter comme “il y a tétraèdre qui n'est pas grand”
3. *Rien n'est derrière a.*
4. *Il n'y que des grands objets derrière b.*

Traduisez les assertions 1 à 4 en formules, ajoutez-les à **Buridan.sen** et sauvegardez le tout dans **exo5.10.sen** et vérifiez que toutes les formules de **exo5.10.sen** sont vraies dans **exo5.10.wld**.

3. Pour chaque formule traduisant une des assertions 1 à 4, modifiez le modèle `exo5.10.wld` pour qu'elle devienne fausse, et cherchez celle(s) des formules initiales de **Buridan.sen** qui sont devenues fausses.
4. Montrez que la formule 9 n'est pas conséquence des 13 autres formules : il suffit de construire un monde (que vous appellerez `exo5.10b.wld`) où 9 est fausse et où toutes les autres sont vraies.
5. Montrez que la formule F suivante (affirmant qu'il y a au moins 2 tétrèdres moyens) est indépendante des formules de **Buridan.sen** : i.e. F n'est pas conséquence de formules de **Buridan.sen** et $\neg F$ non plus.

$$\exists x \exists y ((x \neq y) \wedge Tet(x) \wedge Tet(y) \wedge Medium(x) \wedge Medium(y))$$

Il suffit de construire deux mondes, l'un où F est fausse et où toutes les formules de **Buridan.sen** sont vraies, et l'autre où $\neg F$ est fausse et où toutes les formules de **Buridan.sen** sont vraies. $\diamond$

EXERCICE 5.11 1. Chargez le fichier **Whitehead.sen**.

1. Testez les formules 1 à 7 dans des mondes à 1, 2, 3, ou 4 objets. Quelles formules affirment qu'il y a au moins (au plus, exactement) 2 objets ? Quelles formules affirment qu'il y a au moins (au plus, exactement) 3 objets ?
2. Quelle est la différence entre ce qu'affirment les formules 8 et 9 ?
3. Construire un monde où la formule 10 est vraie.
4. Construire un monde où les formules 11 et 12 sont vraies.
5. A quelle formule est équivalente la formule 13 ? Traduisez en français. Vérifiez en construisant des modèles.
5. Qu'affirme la formule 14 ? Vérifiez en construisant des modèles. $\diamond$

EXERCICE 5.12 1. Chargez le fichier **Montague.sen**. Complétez les formules. Les assertions faites par les formules sont les suivantes :

1. *Tout cube est à gauche de tout tétraèdre* : il faut donc compléter par

$$\forall y (Tet(y) \longrightarrow LeftOf(x, y))$$

2. *Tout petit cube est derrière un grand cube.*
3. *Un cube est devant tous les tétrèdres* (assertion ambiguë).
4. *Un grand cube est devant un petit cube* (assertion ambiguë).
5. *Rien n'est plus grand que tout.*
6. *Tout cube qui est devant tous les tétraèdres est grand.*
7. *Tout ce qui est à droite d'un grand cube est petit.*
8. *Rien de ce qui est derrière un cube et devant un cube n'est grand.*
9. *Tout objet qui n'a rien derrière lui est un cube.*
10. *Tout dodécaèdre est plus petit qu'un tétraèdre.*
2. Chargez le fichier **Pierce.wld**. Vérifiez que toutes vos formules sont vraies.
3. Chargez le fichier **Claire.wld**. Vérifiez que seules les formules 3, 5, 7, 9 sont vraies (sinon il y a une erreur dans vos formules que vous corrigerez).
4. Chargez le fichier **Leibniz.wld**. Vérifiez que seules les formules 5, 6, 8, 10 sont vraies (sinon il y a une erreur dans vos formules que vous corrigerez).

5. Chargez le fichier **Ron.wld**. Vérifiez que seules les formules 2, 3, 4, 5, 8 sont vraies. $\diamond$

EXERCICE 5.13 1. Créez un fichier **exo5.13.sen** où vous mettrez des formules traduisant les assertions suivantes :

1. *Tout tétraèdre est devant tout dodécaèdre.*
2. *Aucun dodécaèdre n'a quelque chose derrière lui.*
3. *Aucun tétraèdre n'a la même taille qu'un cube.*
4. *Tout dodécaèdre a la même taille qu'un cube.*
5. *Tout objet entre deux tétraèdres est un petit cube.*
6. *Tout cube est entre deux objets.*
7. *Tout cube qui a quelque chose derrière lui est petit.*
8. *Tout dodécaèdre qui n'a rien à sa droite est petit.*
9. *Tout dodécaèdre qui n'a rien à sa droite a quelque chose à sa gauche.*
10. *Tout dodécaèdre à gauche d'un cube est grand.*

2. Chargez le fichier **Bolzano.wld**. Vérifiez que toutes vos formules sont vraies.

3. Chargez le fichier **Ron.wld**. Vérifiez que seules les formules 4, 5, 8, 9, 10 sont vraies (sinon il y a une erreur dans vos formules que vous corrigerez).

4. Chargez le fichier **Claire.wld**. Vérifiez que seules les formules 1, 3, 5, 7, 9, 10 sont vraies (sinon il y a une erreur dans vos formules que vous corrigerez).

5. Chargez le fichier **Peano.wld**. Vérifiez que seules les formules 8 et 9 sont vraies (sinon il y a une erreur dans vos formules que vous corrigerez). $\diamond$

EXERCICE 5.14 1. Créer un fichier **exo5.14.sen** qui contienne des formules traduisant les propriétés suivantes :

1. *Seuls les grands objets n'ont rien devant eux.*
2. *Si un cube a quelque chose devant lui, alors il est petit*
3. *Tout cube qui est derrière un dodécaèdre est aussi plus petit que ce dodécaèdre.*
4. *Si **e** est entre deux objets, alors ils sont tous les deux petits.*
5. *Si un tétraèdre est entre deux objets, alors ils sont tous les deux petits.*
6. *Tout dodécaèdre est au moins aussi grand que tout cube.*
7. *Si un cube est à droite d'un dodécaèdre mais pas derrière lui, alors il est aussi grand que le dodécaèdre.*
8. *Aucun cube n'ayant rien à sa gauche n'est entre deux cubes.*
9. *Les seuls grands cubes sont **b** et **c**.*
10. *Au plus **b** et **c** sont des grands cubes.*

2. Chargez le fichier **Ron.wld**. Vérifiez que toutes vos formules sont vraies (attention, il y a des objets cachés).

3. Chargez le fichier **Bolzano.wld**. Vérifiez que seules les formules 3, 8, 10 sont vraies (sinon il y a une erreur dans vos formules que vous corrigerez).

4. Chargez le fichier **Wittgenstein.wld**. Vérifiez que seules les formules 5, 7, 8 sont vraies. $\diamond$

EXERCICE 5.15 Ouvrez **Bozo2.sen** et **Leibniz.wld**. La plupart des expressions de ce fichier ne sont pas des formules bien formées ; quelques unes sont des formules bien formées mais ne sont pas des formules closes (sentence). Lisez et examinez chaque expression, si ce n'est pas une formule bien formée, corrigez-la. Si ce n'est pas une sentence faites-en une sentence vraie,

et ce en apportant le moins possible de changements. Si c'est une sentence fausse, essayez de la rendre vraie, ici aussi en apportant le moins possible de changements. Voyez si vous pouvez reconstituer ce qu'on cherchait initialement à dire. Enregistrez votre ensemble de sentences sous **exo5.15.sen**. ◇

EXERCICE 5.16 1. Créer un fichier **quant1.sen** qui contienne des formules traduisant les propriétés suivantes : (on interprète la phrase *le cube est grand* par : il y a exactement un cube, et il est grand ; cette phrase sera supposée fausse si par exemple il n'y a pas de cube, ou s'il y en a deux)

1. *Il y a au moins deux dodécaèdres.*
 2. *Il y a au plus deux tétraèdres.*
 3. *Il y a exactement deux cubes.*
 4. *Il y a seulement trois objets qui ne sont pas petits.*
 5. *Le petit tétraèdre n'a rien devant lui.*
 6. *Le tétraèdre qui a quelque chose devant lui est grand.*
 7. *Aucun dodécaèdre n'est derrière le grand cube.*
 8. *Le cube moyen est à droite du grand cube.*
 9. *Le seul objet n'ayant rien à sa droite est le cube moyen.*
 10. *Le plus petit cube est moyen.*
2. Chargez le fichier **Peano.wld**. Vérifiez que toutes vos formules sont vraies.
3. Chargez le fichier **Bolzano.wld**. Vérifiez que seules les formules 1, 3, 7 sont vraies (sinon il y a une erreur dans vos formules que vous corrigerez).
4. Chargez le fichier **Skolem.wld**. Vérifiez que seules les formules 5, 7 sont vraies.
5. Chargez le fichier **Montague.wld**. Vérifiez que seules les formules 2, 3, 5, 7, 10 sont vraies. ◇

*Dans les deux exercices suivants, nous dirons qu'une formule est T-valide si toute interprétation où les prédicats **Cube**, ..., **Between** sont fixés comme dans Tarski's world est un modèle de cette formule. Noter qu'on ne fixe pas l'interprétation des six constantes.*

EXERCICE 5.17 1. Chargez les fichiers **Carnap.sen** et **Bolzano.wld**. Traduisez les formules en français et vérifiez qu'elles sont vraies dans **Bolzano.wld**.

2. Quelles sont les formules qui sont T-valides ? (Il y en a cinq). Pour chaque formule qui n'est pas T-valide construire un monde qui falsifie la formule.
3. Quelle formule est une tautologie ?
4. Pour chaque formule qui est T-valide mais qui n'est pas une tautologie, dites comment changer l'interprétation des prédicats pour falsifier la formule. ◇

EXERCICE 5.18 1. Chargez le fichier **Post.sen**.

2. Quelles sont les formules qui sont T-valides ?
3. Pour chaque formule qui n'est pas T-valide essayez de construire un monde qui falsifie la formule.
4. Y a-t-il des formules non T-valides mais pour lesquelles vous ne pouvez pas construire de contre-exemple à l'aide de Tarski's world à cause des contraintes sur les mondes de Tarski's world ? ◇

EXERCICE 5.19 Traduire l'ensemble d'énoncés suivants en formules et vérifiez que l'ensemble de formules obtenu est satisfaisable en construisant un monde `monde.wld` qui en est un modèle.

1. *Tout cube est à gauche de tout tétraèdre.*
2. *Il n'y a pas de dodécaèdre.*
3. *Il y a exactement quatre cubes.*
4. *Il y a exactement quatre tétraèdres.*
5. *Aucun tétraèdre n'est grand.*
6. *Aucun objet n'est plus grand que tous les objets à sa droite.*
7. *Un objet est à gauche d'un autre objet uniquement si le second est derrière le premier.* ◇

EXERCICE 5.20 1. Chargez le fichier `Padoa.sen`. Il contient quatre formules, trois quelconques d'entre elles sont satisfaisables, mais pas les quatre. Construire quatre mondes `123.wld`, `124.wld`, `134.wld`, `234.wld` qui sont modèles des sous-ensembles de trois formules.

2. Réinterprétez les prédicats **Tet** et **Dodec** pour que la formule 3 devienne vraie dans `124.wld`.
3. Réinterprétez le prédicat **Between** pour que la formule 4 devienne vraie dans `123.wld`. ◇

EXERCICE 5.21 Cet exercice explore les limites posées par le nombre de variables qui est de six dans Tarski's world.

1. Traduire *Il y a au moins deux objets*. Combien de variables avez-vous utilisées ?
2. Traduire *Il y a au moins trois objets*. Combien de variables avez-vous utilisées ? On peut montrer qu'on ne peut pas exprimer *Il y a au moins sept objets* en n'utilisant que les six variables et le symbole $=$.
3. Chargez les fichiers `Robinson.sen` et `Robinson.wld`. Il y a six cubes, vérifiez la validité de la formule. Ajoutez un petit cube à droite des autres et revérifiez. Jouez le jeu en prétendant (faussetment) que la formule est fausse. Observez la stratégie de Tarski pour choisir un objet pour la variable x . Pouvez-vous en déduire comment on a réutilisé les variables ?
4. Supprimez un des cubes, et rejouez en prétendant que la formule est vraie. Voyez-vous pourquoi on ne peut pas gagner ?
5. Dans la question 2), on a dit qu'il était impossible d'exprimer l'existence de 7 objets avec 6 variables ; ici on exprime l'existence de 7 objets avec 2 variables. Pouvez-vous expliquer ce paradoxe apparent ? (Indication : faites tourner la figure de 90° et réévaluez la formule de Robinson. Reste-t-elle vraie ?)
6. Ecrivez une formule qui dit qu'il y a au moins 4 objets, les uns devant les autres, en utilisant uniquement les variables x et y . ◇

EXERCICE 5.22 On définit un ordre de complexité sur les figures en combinant *lexicographiquement* les deux ordres suivants :

- une figure a est plus complexe qu'une figure b si a contient plus de faces que b ;
- une figure a est plus complexe qu'une figure b si a est plus grande que b .

Ainsi une pyramide est moins complexe qu'un cube, un petit cube est moins complexe qu'un grand cube et une grande pyramide est moins complexe qu'un cube moyen, qui est lui-même moins complexe qu'un petit dodécaèdre.

1. Construire un monde contenant 6 figures rangées de gauche à droite par ordre croissant de complexité.
2. Saisir un ensemble de formules décrivant la propriété de rangement de ce monde.
3. Vérifier. ◇

EXERCICE 5.23

1. Charger l'ensemble de formules du fichier **austin.sen**
2. Réaliser un monde qui satisfasse toutes les formules

◇

EXERCICE 5.24

1. Charger le fichier **montague.sen**
2. Compléter les formules
3. Réaliser un monde pour les formules obtenues

◇

EXERCICE 5.25

1. Créer un monde contenant au moins une figure de chaque espèce.
2. Saisir les deux formules

$$\begin{aligned} &\forall x(Tet(x) \vee Cube(x) \vee Dodec(x)) \\ &\forall x(Tet(x) \rightarrow (Cube(x) \rightarrow Dodec(x))) \end{aligned}$$

3. Vérifier automatiquement puis interactivement (bouton **Game**) la validité de ces formules.

Variante : construire un monde avec *une seule figure* qui valide la formule

$$\forall x(Tet(x) \rightarrow (Cube(x) \rightarrow Dodec(x)))$$

Est-ce le seul monde validant cette formule ?

◇

EXERCICE 5.26

1. Saisir la formule

$$\forall x((Tet(x) \vee Cube(x)) \rightarrow Dodec(x))$$

2. Construire un monde à *une seule figure* validant cette formule.
3. Vérifier interactivement la validité.

◇

EXERCICE 5.27

1. Charger le fichier **morecnf.sen**
2. Donner les formes normales conjonctives de chacune des formules
3. Réaliser un monde pour les formules obtenues

◇

EXERCICE 5.28 Existence et unicité.

1. Charger le fichier **russe1.sen**
2. Réaliser un monde pour les formules obtenues

◇

EXERCICE 5.29

1. Charger le fichier **skolem.sen**
2. Réaliser un monde pour les formules obtenues

◇

EXERCICE 5.30

1. Charger le fichier **post.sen**
2. Réaliser un monde pour les formules obtenues

◇

EXERCICE 5.31 Paradoxe apparent.

1. Charger le fichier `dodgson.sen`
2. Réaliser un monde pour les formules obtenues



EXERCICE 5.32

1. Charger le fichier `bozo.sen`
2. Dire (et vérifier) quelles formules sont correctes et lesquelles ne le sont pas.
3. Corriger les formules incorrectes
4. Réaliser (si possible) un monde pour les formules obtenues



STRUCTURES DISCRÈTES - T.M.E. LANGAGES ET AUTOMATES

PARTIE I : CRÉATION D'AUTOMATES

Le logiciel JFLAP se trouve sous Unix ; si vous le souhaitez, vous pouvez le récupérer à cette adresse : <http://www.cs.duke.edu/csed/jflap/>, où il y a des versions pour Unix, Windows et MacOS X. Lancer le logiciel **JFLAP** avec la commande

```
java -jar /usr/local/jflap/JFLAP.jar
```

et choisir **Finite Automaton**. Il apparaît alors une fenêtre d'édition, dans laquelle on peut dessiner un automate. La barre d'outils située sous l'onglet **Editor** contient quatre boutons ; de gauche à droite :

- **Attribute Tool** que l'on nommera ici bouton **A**
- **State Tool** (bouton **S**)
- **Transition Tool** (bouton **T**)
- **Deleter** (bouton **D**).

EXERCICE 6.1 Soit l'automate $\mathcal{A}$ défini sur l'alphabet $\{a, b\}$, d'états $0, 1, 2$, d'état initial 0 , d'état final 2 et de transitions $0.a = 0$, $0.b = 1$, $1.a = 2$, $1.b = 2$, $2.a = 0$ et $2.b = 1$.

CRÉATION DE L'AUTOMATE $\mathcal{A}$

Pour dessiner l'automate $\mathcal{A}$:

- créer les états : cliquer sur le bouton **S** ; cliquer en trois endroits différents de la fenêtre d'édition, les états apparaissent avec les noms q_0, q_1, q_2
- choisir la nature des états (initial, final) : cliquer sur le bouton **A** ; faire un clic droit sur l'état q_0 et choisir *Initial* ; faire un clic droit sur l'état q_2 et choisir *Final*
- créer les transitions : cliquer sur le bouton **T**
 - transition $0.a = 0$: cliquer sur l'état q_0 et taper a dans le cadre qui apparaît
 - transition $0.b = 1$: promener la souris, bouton gauche enfoncé, de l'état q_0 à l'état q_1 , lâcher le bouton et taper b dans le cadre qui apparaît
 - autres transitions : sur le modèle de $0.b = 1$
 - attention : il faut créer deux transitions pour $1.a = 2$ et $1.b = 2$ (ne pas taper les deux étiquettes a et b dans le même cadre).
- sauver le fichier en le nommant "exo1-tme".

Vous découvrirez tout seul les autres utilisations du bouton **A** et l'utilisation du bouton **D**.

T.S.V.P.

RECONNAISSANCE DE MOTS PAR $\mathcal{A}$

Pour vérifier si des mots sont acceptés, on utilise l'un des menus *Input/Step by State* ou *Input/Fast Run* ou *Input/Multiple Run*.

- Menu *Input/Step by State* : sous la fenêtre contenant l'automate apparaissent une fenêtre montrant les différentes configurations et une barre contenant six boutons. En cliquant sur le bouton **Step**, le mot est lu lettre à lettre, la progression dans le mot est visible dans la fenêtre des configurations et la progression dans l'automate est visible dans la fenêtre de l'automate. Après le dernier **Step**, la configuration est soit verte (le mot est accepté) soit rouge (le mot est rejeté). En cliquant sur une configuration puis sur le bouton **Trace**, on voit apparaître, dans une autre fenêtre, la suite des transitions qui ont amené à cette configuration.
- Menus *Input/Fast Run* et *Input/Multiple Run* : utilisation évidente.

Remarque : il y a aussi un menu *Input/Step by Closure* mais il n'a d'intérêt que pour les automates avec λ -transitions (transitions étiquetées par le mot vide).

En utilisant chacun des trois menus précédents, vérifier si les mots suivants sont acceptés ou non par l'automate $\mathcal{A}$: *ba*, *babb*, *aaabab*, *aaababa*, *abaabb*, *abaabaaab*. $\diamond$

EXERCICE 6.2 On considère l'automate $\mathcal{B}$ obtenu en ajoutant la transition $2.b = 2$ à l'automate $\mathcal{A}$ défini dans l'exercice 1.

1. Dessiner l'automate $\mathcal{B}$ et sauvegarder le fichier en le nommant "exo2-tme".
2. Choisir le menu *Test/Compare Equivalence* et tester si les automates $\mathcal{A}$ et $\mathcal{B}$ sont équivalents.
3. Utiliser le menu *Test/Highlight Nondeterminism* pour vérifier que l'automate $\mathcal{B}$ n'est pas déterministe.
4. Utiliser le menu *Input/Step by State* pour vérifier que le mot *ababbba* est accepté. Dans la fenêtre des configurations, il apparaît cinq configurations correspondant aux cinq lectures possibles du mot *ababbba* ; on remarque que certaines configurations sont acceptées et d'autres pas. Faire afficher la trace de chacune des cinq configurations.
5. Tester d'autres mots, en utilisant *Input/Step by State* ou *Input/Fast Run*. $\diamond$

EXERCICE 6.3 Soit $A = \{a, b, c\}$. Dessiner des automates, non nécessairement déterministes, reconnaissant les langages suivants :

- 1) $\{a, ab, ca, cab, acc\}$
- 2) $[a(b+c)^*abc]^*$
- 3) l'ensemble des mots contenant un nombre impair de a
- 4) l'ensemble des mots contenant le facteur ab
- 5) l'ensemble des mots contenant le facteur ab et se terminant par b $\diamond$

PARTIE II : DÉTERMINISATION D'AUTOMATES

EXERCICE 6.4 Dessiner l'automate $\mathcal{A}$ sur l'alphabet $\{a, b, c\}$ d'états $0, 1, 2, 3$, d'état initial 0 , d'état terminal 3 et de transitions $0.a = 1, 0.a = 0, 0.b = 0, 0.c = 0, 1.a = 2, 1.b = 2, 1.c = 2, 2.a = 3, 2.b = 3, 2.c = 3$ (cf. ex. 7 du TD), en faisant en sorte que l'état i soit nommé q_i dans le dessin. Vérifier que $\mathcal{A}$ n'est pas déterministe.

Pour déterminer l'automate $\mathcal{A}$, on choisit le menu *Convert/Convert to DFA*. La fenêtre se partage en deux :

- la partie gauche contient l'automate $\mathcal{A}$
- la partie droite est une sous-fenêtre de travail, dans laquelle on construit le déterminisé $\mathcal{A}'$ de $\mathcal{A}$, et dont la barre d'outils contient cinq boutons ; de gauche à droite :
 - **Attribute Editor** (bouton **A**)
 - **Expand Group on Terminal** (bouton **T**)
 - **State Expander** (bouton **S**)
 - **Complete**
 - **Done ?**.

Initialement, la sous-fenêtre de travail contient un seul état, l'état initial, nommé q_0 , auquel est attaché un petit cadre contenant tous les indices des états initiaux de $\mathcal{A}$ (ici, le cadre contient 0). Il faut construire toutes les transitions de $\mathcal{A}'$, c'est-à-dire les transitions : $\{0\}.a = \{0, 1\}$, $\{0\}.b = \{0\}$, $\{0\}.c = \{0\}$, $\{0, 1\}.a = \{0, 1, 2\}$, $\{0, 1\}.b = \{0, 2\}$, etc...

Il y a plusieurs façons de construire les transitions :

- on peut les construire une à une : cliquer sur le bouton **T**
 - transition $\{0\}.a = \{0, 1\}$: promener la souris, bouton gauche enfoncé, de l'état q_0 de $\mathcal{A}'$ vers un endroit quelconque de la sous-fenêtre de travail ; lâcher le bouton ; une boîte de dialogue apparaît, demandant l'étiquette de la transition que l'on veut construire ("Expand on what terminal ?") ; taper a ; une nouvelle boîte de dialogue apparaît, demandant l'état but de la transition que l'on veut construire ; taper 0 et 1 en les séparant par un espace ; apparaissent alors l'état q_1 avec un petit cadre contenant 0,1 et la transition d'étiquette a qui va de q_0 à q_1 .
 - si l'état but de la transition existe déjà dans $\mathcal{A}'$, procéder comme dans la création d'automates ; par exemple, pour construire la transition $\{0\}.b = \{0\}$: cliquer sur l'état q_0 de $\mathcal{A}'$ puis taper b dans la boîte de dialogue
- on peut construire en une seule fois toutes les transitions issues d'un état q de $\mathcal{A}'$: cliquer sur le bouton **S** puis sur l'état q
- on peut construire toutes les transitions en une seule fois : cliquer sur le bouton **Complete**.

Construire l'automate $\mathcal{A}'$, déterminisé de $\mathcal{A}$, en utilisant uniquement le bouton **T**. ◇

EXERCICE 6.5 Déterminiser l'automate $\mathcal{B}$ défini dans l'exercice 2 (sans utiliser le bouton **Complete**). ◇

EXERCICE 6.6 Tester si les automates construits dans l'exercice 3 sont déterministes et déterminer ceux qui ne le sont pas (sans utiliser le bouton **Complete**). ◇

PARTIE III : MINIMISATION D'AUTOMATES

Pour minimiser un automate $\mathcal{A}$, on utilise l'algorithme suivant :

- initialement l'ensemble des états est partagé en deux sous-ensembles : l'ensemble des états terminaux et l'ensemble des états non terminaux
- on réitère le processus suivant :
 - parmi les ensembles d'états déjà construits $Q_1, Q_2 \dots Q_m$, on choisit un ensemble Q_i
 - on choisit une lettre x

- on partage l'ensemble Q_i en plusieurs sous-ensembles : deux états p et q de Q_i appartiennent au même sous-ensemble ssi les états $p.x$ et $q.x$ de $\mathcal{A}$ appartiennent à un même ensemble Q_j
- et ce jusqu'à ce qu'aucune lettre ne puisse plus partager l'un des ensembles d'états.

EXERCICE 6.7 Soit l'automate $\mathcal{A}$ défini sur l'alphabet $\{a, b\}$, d'états 0, 1, 2, 3, 4, 5, d'état initial 0, d'état final 4 et de transitions $0.a = 1$, $0.b = 2$, $1.a = 3$, $1.b = 3$, $2.a = 3$, $2.b = 3$, $3.a = 4$, $3.b = 5$, $4.a = 4$, $4.b = 4$, $5.a = 5$ et $5.b = 4$.

Pour minimiser l'automate $\mathcal{A}$, on peut dérouler l'algorithme de plusieurs manières.

Une première manière :

- initialement : $\{0, 1, 2, 3, 5\}$ et $\{4\}$
- on partage $\{0, 1, 2, 3, 5\}$ en utilisant a : $\{0, 1, 2, 5\}$, $\{3\}$ et $\{4\}$
- on partage $\{0, 1, 2, 5\}$ en utilisant b : $\{0\}$, $\{1, 2\}$, $\{5\}$ et $\{3\}$, $\{4\}$
- on ne peut pas partager $\{1, 2\}$ (ni en utilisant a , ni en utilisant b).

Une deuxième manière :

- initialement : $\{0, 1, 2, 3, 5\}$ et $\{4\}$
- on partage $\{0, 1, 2, 3, 5\}$ en utilisant a : $\{0, 1, 2, 5\}$, $\{3\}$ et $\{4\}$
- on partage $\{0, 1, 2, 5\}$ en utilisant a : $\{0, 5\}$, $\{1, 2\}$ et $\{3\}$, $\{4\}$
- on partage $\{0, 5\}$ en utilisant a : $\{0\}$, $\{5\}$ et $\{1, 2\}$, $\{3\}$, $\{4\}$
- on ne peut pas partager $\{1, 2\}$ (ni en utilisant a , ni en utilisant b).

Et il y a encore d'autres manières...

MINIMISATION DE L'AUTOMATE $\mathcal{A}$

Dessiner l'automate $\mathcal{A}$, en faisant en sorte que l'état i soit nommé q_i dans le dessin. Pour construire l'automate $\mathcal{A}'$, minimisé de l'automate $\mathcal{A}$, on choisit le menu *Convert/Minimize DFA*. La fenêtre se partage en deux :

- la partie gauche contient l'automate $\mathcal{A}$ (ou le complété de $\mathcal{A}$ si $\mathcal{A}$ n'est pas complet)
- la partie droite est une sous-fenêtre de travail qui contient un arbre dont les feuilles représentent des ensembles d'états.

Initialement, l'arbre a seulement deux feuilles : l'une représente les états non terminaux (ici 0, 1, 2, 3, 5) et l'autre les états terminaux (ici 4). Pour compléter cet arbre, on partage les ensembles d'états comme il est dit dans l'algorithme de minimisation.

En suivant la première manière :

- pour partager $\{0, 1, 2, 3, 5\}$ en utilisant a : cliquer sur le cadre contenant 0, 1, 2, 3, 5 ; cliquer sur le bouton **Set Terminal** ; taper a dans la boîte de dialogue ; sous le cadre contenant 0, 1, 2, 3, 5, apparaissent deux sous-arbres, dont les feuilles sont vides ; on remplit l'une des deux feuilles avec 0, 1, 2, 5 et l'autre feuille avec 3 ; pour cela, cliquer sur l'une des deux feuilles puis, successivement, sur les états q_0 , q_1 , q_2 , q_5 de $\mathcal{A}$; ensuite, cliquer sur l'autre feuille puis sur l'état q_3 de $\mathcal{A}$; remarque : la même démarche (cliquer sur la feuille de l'arbre puis sur l'état de l'automate $\mathcal{A}$) permet d'enlever un état déjà présent dans une feuille ; cliquer sur le bouton **Check Node** pour soumettre la partition ;
- pour partager $\{0, 1, 2, 5\}$ en utilisant b : cliquer sur le cadre contenant 0, 1, 2, 5 ; cliquer sur le bouton **Set Terminal** ; taper b dans la boîte de dialogue ; sous le cadre contenant 0, 1, 2, 5, apparaissent deux sous-arbres ; comme la lettre b partage $\{0, 1, 2, 5\}$ en trois sous-ensembles ($\{0\}$, $\{1, 2\}$ et $\{5\}$), il faut ajouter un sous-arbre (bouton **Add**

Child) ; remplir les trois feuilles ; cliquer sur le bouton **Check Node** pour soumettre la partition ; si elle n'est pas correcte, un message d'erreur s'affiche et il faut alors modifier la composition des feuilles ; si elle est correcte, le message "The expansion is correct !" s'affiche

- comme il n'y a plus d'ensemble d'états à partager, le seul bouton accessible est le bouton **Finish** ; cliquer dessus ; dans la sous-fenêtre de travail, l'arbre est remplacé par l'ensemble des états de l'automate minimal ; construire les transitions de l'automate minimal
- cliquer sur le bouton **Done ?** ; s'il manque des transitions, un message le signale.

Vous pouvez refaire la minimisation de l'automate $\mathcal{A}$ en suivant la deuxième manière de dérouler l'algorithme. $\diamond$

EXERCICE 6.8 Soit l'automate $\mathcal{A}$ d'états 0, 1, 2, 3, 4, 5, d'état initial 0, d'état terminal 5 et de transitions : $0.a = 1$, $1.a = 2$, $2.a = 3$, $3.a = 4$, $4.a = 5$, $5.a = 5$, $0.b = 3$, $1.b = 2$, $2.b = 3$, $3.b = 4$, $4.b = 5$, $5.b = 5$. Dessiner l'automate $\mathcal{A}$. Minimiser $\mathcal{A}$. $\diamond$

EXERCICE 6.9 1) Soit l'automate $\mathcal{A}$ d'états 0, 1, 2, 3, 4, 5, d'état initial 0, d'état terminal 3 et de transitions : $0.a = 1$, $1.a = 1$, $2.a = 4$, $3.a = 5$, $4.a = 4$, $5.a = 5$, $0.b = 2$, $1.b = 3$, $2.b = 2$, $3.b = 3$, $4.b = 5$, $5.b = 5$. Dessiner l'automate $\mathcal{A}$. Minimiser $\mathcal{A}$.

- 2) Soit l'automate $\mathcal{B}$ d'états 0, 1, 2, 3, 4, 5 d'état initial 0, d'états terminaux 3, 4, 5 et de même transitions que $\mathcal{A}$. Dessiner l'automate $\mathcal{B}$. Minimiser $\mathcal{B}$. $\diamond$

EXERCICE 6.10 Soit l'automate $\mathcal{A}$ d'états 0, 1, 2, 3 d'état initial 0, d'état terminal 2 et de transitions : $0.a = 0$, $0.a = 1$, $1.a = 1$, $3.a = 0$, $1.b = 1$, $1.b = 2$, $3.b = 0$, $3.b = 2$, $1.c = 1$, $1.c = 3$, $3.c = 0$.

- 1) Dessiner l'automate $\mathcal{A}$.
- 2) Déterminer $\mathcal{A}$.
- 3) Minimiser $\mathcal{A}$. $\diamond$

PARTIE IV : À DÉCOUVRIR SEUL

- 1) Le menu *Convert/Convert FA to RE* de la fenêtre **Editor**. Il permet de calculer une expression rationnelle pour le langage reconnu par un automate. Pour chacun des exercices 22, 24, 25, 26 de la feuille de TD, on pourra comparer l'expression rationnelle calculée par le logiciel JFLAP à l'expression rationnelle calculée en TD. On pourra au préalable traiter l'exercice suivant :

EXERCICE 6.11 Soient X et Y des langages sur un alphabet A . Montrer que

$$(X + Y)^* = (X^*Y)^*X^* = X^*(YX^*)^*$$

$\diamond$

- 2) Le bouton **Regular Expression** de la fenêtre **New Document**. Il permet de dessiner l'automate reconnaissant un langage donné par une expression rationnelle (ne contenant que des +, . et *). Pour chacune des expressions rationnelles de l'exercice 21 de la feuille de TD, on pourra comparer l'automate construit par le logiciel JFLAP à l'automate construit en TD.

RAPPELS SUR LES ENSEMBLES ET FONCTIONS

Nous rappelons ici quelques notions de base de la théorie des ensembles et les notations utilisées. Nous définirons les ensembles et les notions de fonctions, relations, opérations qui sont fondamentales dans toute la suite.

7.1 Ensembles

Soit E un *ensemble* et e un *élément*, $e \in E$ signifie que e est un élément qui est dans l'ensemble E et se lit e appartient à E . La négation de cette relation, c'est-à-dire e n'appartient pas à E , s'écrit $e \notin E$. L'ensemble vide est un ensemble qui ne contient aucun élément, il est noté $\emptyset$.

A, B étant deux ensembles, on dit que A est un sous-ensemble de B ou une partie de B ou encore que A est inclus dans B , et on note $A \subseteq B$, si et seulement si tout élément de A appartient à B . La négation de $A \subseteq B$ s'écrit $A \not\subseteq B$ et ne veut pas du tout dire $B \subseteq A$. On peut remarquer que $A = B$ si et seulement si $A \subseteq B$ et $B \subseteq A$, c'est-à-dire si et seulement si A et B ont les mêmes éléments. Si $A \subseteq B$ mais que $A \neq B$, on écrit $A \subsetneq B$. Enfin, on note $\mathcal{P}(E)$ l'ensemble des parties de E ; donc $A \subseteq E$ si et seulement si $A \in \mathcal{P}(E)$. Remarquons que les ensembles $\emptyset$ et E sont toujours des éléments de $\mathcal{P}(E)$.

EXEMPLE 7.1 Soit $E = \{0, 1\}$. $\mathcal{P}(E) = \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$.

On appelle *produit cartésien* de deux ensembles E et F l'ensemble des couples formés d'un élément de E et d'un élément de F : $E \times F = \{(x, y) / x \in E \text{ et } y \in F\}$. Le produit cartésien se généralise à une famille finie d'ensembles :

$$E_1 \times \cdots \times E_n = \{(x_1, \dots, x_n) / x_1 \in E_1, \dots, x_n \in E_n\}$$

Finalement, on note $E^n = E \times \cdots \times E$ le produit cartésien de E par lui-même n fois, pour $n \geq 1$. E^n peut être défini récursivement par : $E_1 = E$ et $E^n = E \times E^{n-1}$.

7.1.1 Réunion, intersection, différence, complémentaire, partition

On suppose dans toute la suite que l'on travaille dans un ensemble référentiel E . Soient A, B deux parties de E . On définit :

- l'intersection de A et B : $A \cap B = \{e \in E / e \in A \text{ et } e \in B\}$,
- l'union de A et B : $A \cup B = \{e \in E / e \in A \text{ ou } e \in B\}$,
- la différence de A et B : $A \setminus B = \{e \in E / e \in A \text{ et } e \notin B\}$,
- le complémentaire de A dans E , noté $\overline{A}$ ou A^c : $\overline{A} = E \setminus A = \{e \in E / e \notin A\}$,
- la différence symétrique de A et B : $A \triangle B = (A \setminus B) \cup (B \setminus A)$.

On dit que deux ensembles A et B de E sont *disjoints* si et seulement si $A \cap B = \emptyset$.

EXEMPLE 7.2 1. $\overline{\overline{E}} = \emptyset$, et $\overline{\emptyset} = E$. 2. $A \cap \emptyset = \emptyset$, $A \cup E = E$, $A \triangle E = \overline{A}$. 3. $A \cap A = A \cup A = A \cup \emptyset = A \triangle \emptyset = A \cap E = A$. 4. $A \setminus B = A \cap \overline{B}$. 5. $A \setminus A = A \triangle A = \emptyset$.

Les lois de Morgan permettent de calculer le complémentaire d'une union et d'une intersection : $\overline{A \cup B} = \overline{A} \cap \overline{B}$ et $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

L'union et l'intersection sont des opérations associatives et commutatives . Elles sont de plus distributives l'une par rapport à l'autre, c'est-à-dire

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \quad \text{et} \quad A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

On dit qu'une famille finie ou infinie $(A_i)_{i \in I}$ de parties de E est une partition de E si elle vérifie

- i) $A_i \neq \emptyset$ pour tout $i \in I$,
- ii) $A_i \cap A_j = \emptyset$ pour tous i, j distincts dans I ,
- iii) $E = \bigcup_{i \in I} A_i$.

7.2 Fonctions

7.2.1 Définitions

Intuitivement, une application f d'un ensemble E dans un ensemble F est un procédé pour associer à chaque élément de E un unique élément $f(x)$ de F .

Soit E et F deux ensembles, on appelle *fonction* de E vers F tout triplet $f = (E, F, \Gamma)$ où Γ est une partie de $E \times F$, telle que pour tout élément de E il y a au plus un élément y de F tel que $(x, y) \in \Gamma$; pour tout élément x de E cet élément y , s'il existe, est appelé image de x et est notée $f(x)$. On dit que E est l'ensemble de départ, F l'ensemble d'arrivée et Γ le graphe de f . L'ensemble $\text{Dom}(f) = \{x \in E / \exists y \in F, (x, y) \in \Gamma\}$ est le domaine ou l'ensemble de définition de f . L'ensemble

$$\text{Im}(f) = \{y \in F / \exists x \in E, (x, y) \in \Gamma\}$$

est l'image de f . Soit $X \subseteq E$, on note

$$f(X) = \{y \in F / \exists x \in X, (x, y) \in \Gamma\}$$

l'ensemble des images par f des éléments de X . De même, on appelle *antécédent* par f d'un élément $y \in F$ tout élément $x \in E$ tel que $(x, y) \in \Gamma$. Pour $Y \subseteq F$, on note

$$f^{-1}(Y) = \{x \in E / \exists y \in Y, (x, y) \in \Gamma\}$$

l'ensemble des antécédents par f des éléments de Y .

Avec ces notations, on a $\text{Dom}(f) = f^{-1}(F)$ et $\text{Im}(f) = f(E)$.

La fonction f est notée $f: E \longrightarrow F$. On dit qu'une fonction $f: E \longrightarrow F$ est une *application* si son domaine est l'ensemble E tout entier : $\text{Dom}(f) = E$. L'ensemble des applications de E dans F est noté F^E .

EXEMPLE 7.3 1. On note $\text{id}_E: E \longrightarrow E$ l'application identité de E définie par $f(x) = x$ pour tout x de E .

2. On appelle *fonction caractéristique* d'une partie A de E la fonction $\chi_A: E \longrightarrow \{0, 1\}$ définie par : $\chi_A(e) = \begin{cases} 1 & \text{si } e \in A, \\ 0 & \text{si } e \notin A. \end{cases}$

Une application $f: E \longrightarrow F$ est *injective* (resp. *surjective*, *bijective*) si tout élément $y \in F$ a au plus (resp. au moins, exactement) un antécédent par f . On a donc

- f est injective si et seulement si $\forall x, y \in E, f(x) = f(y) \implies x = y$
- f est surjective si et seulement si $\forall y \in F, \exists x \in E, y = f(x)$
- f est bijective si et seulement si $\forall y \in F, \exists! x \in E, y = f(x)$, c'est-à-dire f est injective et surjective (le symbole $\exists!$ signifie “il existe un et un seul”).

Soient $f: E \longrightarrow F$ et $g: F \longrightarrow G$ deux applications, la composition de f et g est l'application $g \circ f: E \longrightarrow G$ définie par $g \circ f(x) = g(f(x))$. Par exemple, si $f: E \longrightarrow F$ est une application, on a $f \circ \text{id}_E = \text{id}_F \circ f = f$. La composition des applications est associative ce qui permet d'écrire sans parenthèses $h \circ g \circ f$.

Soit $f: E \longrightarrow F$ une application, soient A et B deux parties de E et soient C et D deux parties de F . On a $f(A \cup B) = f(A) \cup f(B)$ et $f(A \cap B) \subseteq f(A) \cap f(B)$, et $f^{-1}(C \cup D) = f^{-1}(C) \cup f^{-1}(D)$ et $f^{-1}(C \cap D) = f^{-1}(C) \cap f^{-1}(D)$. Si de plus f est injective alors $f(A \cap B) = f(A) \cap f(B)$. Ces propriétés se montrent facilement et se généralisent à des unions ou intersections quelconques.

Proposition 7.4 Soient $f: E \longrightarrow F$ et $g: F \longrightarrow G$ deux applications. 1. f et g injectives $\implies g \circ f$ injective. 2. f et g surjectives $\implies g \circ f$ surjective. 3. f et g bijectives $\implies g \circ f$ bijective.

7.3 Cardinaux

7.3.1 Ensembles finis

Pour tout entier n , on note $[n]$ l'ensemble $\{1, \dots, n\}$ des entiers compris entre 1 et n . On démontre la proposition suivante (appelée principe des tiroirs)

Proposition 7.5 *Si $n < m$, il n'existe pas d'injection de $[m]$ dans $[n]$.*

On déduit aisément de cette proposition que deux entiers n et m sont égaux si et seulement si il existe une bijection de $[n]$ vers $[m]$. Un ensemble E est *fini* s'il existe un entier n et une bijection de E vers $[n]$. Cet entier n est alors unique, s'appelle *cardinal* de E et se note $|E|$.

Soient E et F des ensembles finis, et $f: E \longrightarrow F$ une application, avec ces notations, on a

$$\begin{aligned} f \text{ injective} &\iff \forall y \in F, |f^{-1}(\{y\})| \leq 1 \\ f \text{ surjective} &\iff \forall y \in F, |f^{-1}(\{y\})| \geq 1 \\ f \text{ bijective} &\iff \forall y \in F, |f^{-1}(\{y\})| = 1 \end{aligned}$$

De plus, si E est fini et $|E| = |F|$ alors $f \text{ injective} \iff f \text{ surjective} \iff f \text{ bijective}$. Si E n'est pas fini, ces équivalences sont fausses ; par exemple $f: \mathbb{N} \implies \mathbb{N}$ définie par $f(n) = 2n$ est injective, mais non surjective.

Enfin, les propriétés suivantes sont très utiles

Proposition 7.6 *Soient E et F deux ensembles finis.*

- i) *Si E et F sont disjoints alors $|E \cup F| = |E| + |F|$.*
- ii) *Si $(A_i)_{i \in [n]}$ est une partition de E alors $|E| = |A_1| + \dots + |A_n|$.*
- iii) $|E \times F| = |E| \times |F|$.
- iv) $|F^E| = |F|^{|E|}$, où F^E désigne l'ensemble des applications de E dans F .
- v) $|\mathcal{P}(E)| = 2^{|E|}$.

7.3.2 Ensembles dénombrables

Le cardinal peut se généraliser à des ensembles non finis, de telle façon que les propriétés suivantes soient vérifiées pour E et F deux ensembles quelconques :

- i) $|E| \leq |F| \iff$ il existe une injection de E vers F .
- ii) $|E| \geq |F| \iff$ il existe une surjection de E vers F .
- iii) $|E| = |F| \iff$ il existe une bijection de E vers F .

Des relations qui précèdent, on peut déduire que s'il existe à la fois une injection et une surjection de E vers F alors il existe une bijection de E vers F .

Un ensemble E est *dénombrable* s'il est en bijection avec l'ensemble $\mathbb{N}$ des entiers naturels. On note ω le cardinal de $\mathbb{N}$. Une union $\bigcup_{i \in I} A_i$ est dite dénombrable si l'ensemble I d'indices est dénombrable.

Les ensembles dénombrables vérifient les propriétés suivantes :

- Toute partie d'un ensemble dénombrable est finie ou dénombrable.
- Tout produit cartésien fini d'ensembles dénombrables est dénombrable.
- Toute union dénombrable d'ensembles dénombrables est dénombrable.

Enfin signalons qu'il existe des ensembles non dénombrables. Ceci provient du résultat suivant :

Proposition 7.7 Soit E un ensemble et soit $\mathcal{P}(E)$ l'ensemble des parties de E . On a

$$|E| < |\mathcal{P}(E)|.$$

On en déduit que $\mathcal{P}(\mathbb{N})$ n'est pas dénombrable.

7.4 Opérations et relations

Une *opération* Φ sur un ensemble E est une application $\Phi: E^n \longrightarrow E$. On dit que n est l'*arité*, ou le *rang*, de Φ , ou encore que Φ est une opération n -aire, et on note $a(\Phi) = n$; on étudie principalement les opérations d'arité deux, ou opérations binaires.

7.4.1 Opérations binaires

Une *opération binaire* ou *loi de composition interne* $*$ sur un ensemble E est une application $*$: $E \times E \longrightarrow E$. L'image d'un couple (x, y) par cette opération est notée $x * y$ (notation infixe). On définit les propriétés suivantes

- i) $*$ est *associative* si $\forall a, b, c \in E, a * (b * c) = (a * b) * c$,
- ii) $*$ est *commutative* si $\forall a, b \in E, a * b = b * a$,
- iii) $*$ admet l'élément 1 pour *élément neutre* si $\forall e \in E, e * 1 = 1 * e = e$.

Si une opération $*$ est associative, on notera $a * b * c$ (sans parenthèses) le produit des trois éléments a, b et c , qu'on peut indifféremment calculer en faisant $(a * b) * c$ ou $a * (b * c)$. Plus généralement, on notera $e_1 * e_2 * \dots * e_n$ le produit de n éléments.

Définition 7.8 Un ensemble E muni d'une opération $*$ associative est un *semi-groupe*. Si de plus E possède un élément neutre e pour $*$ on dit que $(E, *, e)$ est un *monoïde*. Si la loi $*$ est commutative, le semi-groupe (resp. le monoïde) est dit *commutatif*.

EXEMPLE 7.9 L'ensemble $\mathbb{N}$ muni de l'addition $+$ et de l'élément neutre 0 est un monoïde commutatif. L'ensemble $\mathbb{N}$ muni de la multiplication $\times$ et de l'élément neutre 1 est aussi un monoïde commutatif.

EXEMPLE 7.10 L'ensemble $\mathcal{P}(E)$ muni de l'intersection (ou de l'union) est un monoïde commutatif. L'ensemble des applications de E dans lui-même muni de la composition des applications est un monoïde non commutatif dès que E a au moins deux éléments. L'ensemble des matrices carrées d'ordre n , à coefficients dans $\mathbb{R}$ est un monoïde, non commutatif si $n > 1$, pour l'opération de multiplication des matrices.

Un cas particulier de monoïdes, sont les *monoïdes libres*.

Définition 7.11 Soit A un ensemble fini appelé *alphabet* et dont les éléments sont appelés *lettres*. Le monoïde libre sur A , noté A^* , est l'ensemble des mots écrits sur l'alphabet A . Un mot u est simplement une suite finie de lettres.

Le nombre d'éléments de la suite est appelé *longueur* du mot et noté $|u|$. Si le mot u de longueur n est la suite $(u_1, u_2, \dots, u_n)$, on le note simplement $u = u_1 u_2 \cdots u_n$. Par exemple, $aaba, acebdacebd, a, b, aa, aaa$ sont des mots sur l'alphabet $A = \{a, b, c, d, e\}$. Le mot vide, noté ε , est un mot particulier de A^* qui ne contient aucune lettre (la suite vide). La loi qui fait de A^* un monoïde est appelée *concaténation*. La concaténation de deux mots $u = u_1 u_2 \cdots u_n$ et $v = v_1 v_2 \cdots v_m$ est le mot $u \cdot v = u_1 u_2 \cdots u_n v_1 v_2 \cdots v_m$. Par exemple, $aaba \cdot cde = aabacde$. L'élément neutre pour la concaténation est bien sûr le mot vide ε .

Définition 7.12 Un ensemble E muni d'une opération $*$ est un *groupe* si c'est un monoïde et que tout élément admet un inverse, c'est-à-dire $\forall e \in E, \exists e' \in E$ tel que $e * e' = e' * e = \mathbb{1}$ (où $\mathbb{1}$ désigne l'élément neutre pour $*$). Si de plus $*$ est commutative le groupe est dit *commutatif*.

EXEMPLE 7.13 L'ensemble $\mathbb{Z}$ des entiers relatifs muni de l'addition est un groupe commutatif. Si E muni d'une opération $*$ est un monoïde alors l'ensemble des éléments inversibles de E forme un groupe. En particulier, l'ensemble des matrices carrées inversibles d'ordre n , à coefficients dans $\mathbb{R}$ est un groupe, non commutatif si $n > 1$, pour l'opération de multiplication des matrices.

Soient $\top$ et $\perp$ deux lois sur un ensemble E . $\top$ est *distributive* par rapport à $\perp$ si $\forall a, b, c \in E$,

$$a \top (b \perp c) = (a \top b) \perp (a \top c) \quad \text{et} \quad (a \perp b) \top c = (a \top c) \perp (b \top c).$$

Si la loi $\top$ est commutative, une seule des conditions est bien sûr suffisante.

EXEMPLE 7.14 Dans $\mathbb{R}$, la multiplication est distributive par rapport à l'addition. Dans $\mathcal{P}(E)$, l'intersection et l'union sont distributives l'une par rapport à l'autre.

7.4.2 Relations

Définition 7.15 Une relation sur un ensemble E est la donnée d'une partie $\mathcal{R}$ de $E \times E$. Pour indiquer qu'une paire (e, e') de $E \times E$ est dans cette partie $\mathcal{R}$, on utilisera, selon les cas, l'une des notations suivantes : $(e, e') \in \mathcal{R}$, $e \mathcal{R} e'$, $\mathcal{R}(e, e')$.

EXEMPLE 7.16 Sur $E = \mathbb{N}$, les ensembles suivants sont des relations

- l'ensemble $\{(n, m) / n \leq m\}$,
- l'ensemble $\{(n, m) / n \leq m \leq 2n\}$,
- l'ensemble $\{(n, m) / n \leq m \text{ et } \exists k : n^2 + m^2 = k^2\}$.

Sur $E = \mathcal{P}(A)$, l'inclusion est une relation.

7.4.3 Opérations ensemblistes sur les relations

Puisqu'une relation est un ensemble on peut définir aisément

- le *complémentaire* $\overline{\mathcal{R}}$ d'une relation $\mathcal{R}$; $\overline{\mathcal{R}}$ est le complémentaire de $\mathcal{R}$ dans $E \times E$:

$$(e, e') \in \overline{\mathcal{R}} \iff (e, e') \notin \mathcal{R},$$

- la *réunion* $\mathcal{R}_1 \cup \mathcal{R}_2$ de deux relations $\mathcal{R}_1$ et $\mathcal{R}_2$:

$$(e, e') \in \mathcal{R}_1 \cup \mathcal{R}_2 \iff (e, e') \in \mathcal{R}_1 \text{ ou } (e, e') \in \mathcal{R}_2,$$

- l'*intersection* $\mathcal{R}_1 \cap \mathcal{R}_2$ de deux relations $\mathcal{R}_1$ et $\mathcal{R}_2$:

$$(e, e') \in \mathcal{R}_1 \cap \mathcal{R}_2 \iff (e, e') \in \mathcal{R}_1 \text{ et } (e, e') \in \mathcal{R}_2.$$

On peut définir aussi trois relations particulières

- la *relation vide*, notée $\emptyset_E : \forall e, e' \in E, (e, e') \notin \emptyset_E$,
- la *relation pleine*, notée $\mathbf{\Pi}_E : \forall e, e' \in E, (e, e') \in \mathbf{\Pi}_E$,
- la *relation identité*, notée $Id_E : \forall e, e' \in E, (e, e') \in Id_E \iff e = e'$.

Puisque $\mathcal{R}_1$ et $\mathcal{R}_2$ sont des ensembles, on peut écrire $\mathcal{R}_1 \subseteq \mathcal{R}_2$ pour exprimer que

$$\forall e, e' \in E, (e, e') \in \mathcal{R}_1 \implies (e, e') \in \mathcal{R}_2.$$

7.4.4 Autres opérations sur les relations

Soit $\mathcal{R}$ une relation binaire sur E . La *relation inverse*, notée $\mathcal{R}^{-1}$, est définie par

$$e \mathcal{R}^{-1} e' \iff e' \mathcal{R} e.$$

Soient $\mathcal{R}_1$ et $\mathcal{R}_2$ deux relations binaires sur E . Leur *produit*, noté $\mathcal{R}_1 \cdot \mathcal{R}_2$, est la relation définie par $e (\mathcal{R}_1 \cdot \mathcal{R}_2) e' \iff \exists e'' : e \mathcal{R}_1 e'' \text{ et } e'' \mathcal{R}_2 e'$.

Ce produit est associatif ; il a Id_E pour élément neutre.

Soit $\mathcal{R}$ une relation binaire. La relation $\mathcal{R}^*$ est égale à

$$Id_E \cup \mathcal{R} \cup (\mathcal{R} \cdot \mathcal{R}) \cup (\mathcal{R} \cdot \mathcal{R} \cdot \mathcal{R}) \cup \dots$$

ou encore $\bigcup_{i \geq 0} \mathcal{R}^i$ avec $\mathcal{R}^0 = Id_E$, $\mathcal{R}^{i+1} = \mathcal{R} \cdot \mathcal{R}^i$, pour $i \geq 0$. La relation $\mathcal{R}^+$ est définie par $\mathcal{R}^+ = \bigcup_{i > 0} \mathcal{R}^i$ et donc $\mathcal{R}^* = Id_E \cup \mathcal{R}^+$. On admettra que $\forall i, j \geq 0, \mathcal{R}^{i+j} = \mathcal{R}^i \cdot \mathcal{R}^j$.

7.4.5 Relations d'équivalence

Définition 7.17 Une relation d'équivalence est une relation réflexive, symétrique et transitive.

EXEMPLE 7.18

- i) L'égalité sur un ensemble E est une relation d'équivalence, notée $=$ ou aussi Id_E (cf. section 7.4.3).
- ii) Soit n un entier supérieur ou égal à 2, la relation sur $\mathbb{Z}$: “ x et y ont même reste dans la division par n ” est une relation d'équivalence. On la note $x \equiv y[n]$, ou aussi $x \equiv y \pmod{n}$, et on dit que “ x et y sont congrus modulo n ”.

L'intersection $\mathcal{R} \cap \mathcal{R}'$ de deux relations d'équivalence est encore une relation d'équivalence, mais la réunion $\mathcal{R} \cup \mathcal{R}'$ et le produit $\mathcal{R} \cdot \mathcal{R}'$ peuvent ne pas être des relations d'équivalence.

Proposition 7.19 Si $\mathcal{R}$ est une relation quelconque, $(\mathcal{R} \cup \mathcal{R}^{-1})^*$ est une relation d'équivalence, et c'est la plus petite relation d'équivalence qui contient $\mathcal{R}$.

Définition 7.20 Soit $\mathcal{R}$ une relation d'équivalence sur E , et e un élément de E , l'ensemble $\{e' \in E / e \mathcal{R} e'\}$, noté $[e]_{\mathcal{R}}$, est appelé la classe d'équivalence de e .

Proposition 7.21 1) $\forall e \in E, e \in [e]_{\mathcal{R}}$, 2) $\forall e, e' \in E, e \mathcal{R} e' \implies [e]_{\mathcal{R}} = [e']_{\mathcal{R}}$, 3) si $[e]_{\mathcal{R}} \cap [e']_{\mathcal{R}} \neq \emptyset$, alors $[e]_{\mathcal{R}} = [e']_{\mathcal{R}}$.

Démonstration. Le premier point est évident puisque $e \mathcal{R} e$. Pour montrer le second point, considérons $e'' \in [e']_{\mathcal{R}}$; alors $e' \mathcal{R} e''$, et comme $e \mathcal{R} e'$, on a aussi $e \mathcal{R} e''$ et $e'' \in [e]_{\mathcal{R}}$. D'où $[e']_{\mathcal{R}} \subseteq [e]_{\mathcal{R}}$. Réciproquement, $[e]_{\mathcal{R}} \subseteq [e']_{\mathcal{R}}$ pour les mêmes raisons. Enfin, si $e'' \in [e]_{\mathcal{R}} \cap [e']_{\mathcal{R}}$, alors $e \mathcal{R} e''$ et $e'' \mathcal{R} e'$, donc $e \mathcal{R} e'$ et $[e]_{\mathcal{R}} = [e']_{\mathcal{R}}$. $\square$

L'ensemble $\{[e]_{\mathcal{R}} / e \in E\}$ de parties de E , appelé l'ensemble quotient de E par $\mathcal{R}$ et noté $E/\mathcal{R}$, est donc une partition de E . Réciproquement, si $A \subseteq \mathcal{P}(E)$ est une partition de E (c'est-à-dire, $\forall E_1, E_2 \in A, E_1 \neq E_2 \implies E_1 \cap E_2 = \emptyset$ et $\forall e \in E, \exists E_e \in A : e \in E_e$), on peut définir une relation d'équivalence $\mathcal{R}_A$ par $e \mathcal{R}_A e'$ si et seulement si e et e' sont dans le même élément de la partition; il est facile de vérifier que c'est bien une équivalence.

7.4.6 Congruences

Définition 7.22 Une relation d'équivalence $\mathcal{R}$ définie sur un ensemble muni d'une loi de composition interne $*$ est une congruence si elle est compatible avec la loi $*$ c'est-à-dire si $\forall e, e', d, d' \in E, (e \mathcal{R} e' \text{ et } d \mathcal{R} d') \implies (e * d) \mathcal{R} (e' * d')$.

EXEMPLE 7.23 Soit n un entier supérieur ou égal à 2, la relation sur $\mathbb{Z}$: “ $x \equiv y[n]$ ” est une congruence pour l'addition et pour la multiplication.

Si $\mathcal{R}$ est une congruence sur l'ensemble E muni de la loi $*$, la loi $*$ “passe au quotient”, c'est-à-dire que l'ensemble quotient $E/\mathcal{R}$ est muni d'une loi $[*]$ en posant $[e][*][e'] = [e * e']$, et l'opération $[*]$ est bien définie (elle ne dépend pas des représentants des classes d'équivalence que l'on a choisis). Pour simplifier, la loi $[*]$ sera notée simplement $*$.

Proposition 7.24 Soit $\mathcal{R}$ une congruence sur un monoïde (resp. groupe) $(E, *)$ l'ensemble quotient $E/\mathcal{R}$ muni de $*$ est un monoïde (resp. groupe).

EXEMPLE 7.25 Soit n un entier supérieur ou égal à 2, le quotient de $\mathbb{Z}$ par la relation $x \equiv y[n]$ est noté $\mathbb{Z}/n\mathbb{Z}$, c'est un groupe pour l'addition et un monoïde pour la multiplication.