EXERCICES TYPE X–ENS

Exercice 3.1 – Grammaires contextuelles

p. 5

On admet le théorème suivant :

Théorème – Lemme de pompage algébrique

Soit L un langage algébrique (ou hors-contexte). Alors il existe $n \in \mathbb{N}$ tel que pour tout mot $u \in L$ vérifiant $|u| \geqslant n$, on peut écrire $u = vwxyz$ avec :

1. $|wxy| \leqslant n$;
2. $|wy| > 0$;
3. $\forall k \in \mathbb{N}, vw^k xy^k z \in L$.

On pose $L_0 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ et $L_1 = \{uu \mid u \in \{a,b\}^*\}$.

1. Montrer que L_0 n'est pas algébrique. L_1 est-il algébrique ?

Définition

Une **grammaire** (non restreinte) est un quadruplet $G = (V, T, P, S)$ où V est un alphabet dit de **variables**, T est un alphabet disjoint de V dit de **terminaux**, $S \in V$ est le **symbole initial** et P est un ensemble de **règles de production** de la forme $\alpha \to \beta$, avec $\alpha \in (V \cup T)^+$ et $\beta \in (V \cup T)^*$.

Pour $u, v \in (V \cup T)^*$, on dit que u se **dérive immédiatement** en v, noté $u \Rightarrow v$, s'il existe $\alpha \to \beta \in P$, $\gamma, \delta \in (V \cup T)^*$ tels que $u = \gamma\alpha\delta$ et $v = \gamma\beta\delta$. On note $\Rightarrow^*$ la clôture réflexive et transitive de $\Rightarrow$. Le **langage engendré par** G, noté $L(G)$, est $L(G) = \{u \in T^* \mid S \Rightarrow^* u\}$.

Une grammaire est dite **monotone** si toute règle est de la forme $S \to \varepsilon$ ou $\alpha \to \beta$, $|\alpha| \leqslant |\beta|$, $\beta \in (V \cup T \setminus \{S\})^+$. Un langage est dit **monotone** s'il est engendré par une grammaire monotone.

On notera par des lettres capitales les éléments de V et par des minuscules les éléments de T.

2. Déterminer le langage engendré par $S \to aSBa \mid aba$, $aB \to Ba$, $bB \to bb$. Ce langage est-il monotone ?
3. Montrer que L_0 est monotone. En étudiant la grammaire donnée par $S \to S'F$, $S' \to aS'A \mid bS'B \mid F$, $F \to a$, $xX \to Xx$ (pour $x \in \{a,b\}$, $X \in \{A,B\}$), montrer que L_1 est monotone.
4. Montrer que tout langage algébrique est monotone.

Définition

Une grammaire $G = (V, T, P, S)$ est dite **contextuelle** si toute règle est de la forme $S \to \varepsilon$ ou $\alpha X \gamma \to \alpha\beta\gamma$, $X \in V$, $\alpha, \gamma \in (V \cup T)^*$, $\beta \in (V \cup T \setminus \{S\})^+$. Un langage est dit **contextuel** s'il est engendré par une grammaire contextuelle.

5. Montrer que L_0 est contextuel.
   On cherchera un moyen de remplacer une règle $XY \to YX$ *par plusieurs règles contextuelles.*

Définition

Une grammaire monotone est en **forme normale de Kuroda** si ses règles sont de l'une des formes suivantes : $S \to \varepsilon$, $X \to a$, $X \to Y$, $X \to YZ$, $XY \to ZU$.

6. Montrer que tout langage monotone est engendré par une grammaire en forme normale de Kuroda.

7. En déduire que les langages monotones sont exactement les langages contextuels.
8. On admet que pour $G = (V, T, P, S)$ une grammaire non restreinte et $u \in T^*$, le problème qui consiste à déterminer si $u \in L(G)$ est indécidable. Qu'en est-il si on impose à G d'être monotone ?

Exercice 3.2 – Décidabilité de l'arithmétique de Presburger

p. 6

On considère dans cet exercice la logique de Presburger, c'est-à-dire la logique du premier ordre sur le langage $\{0, 1, +, =\}$. Formellement, un **terme** est défini par induction comme une constante (0 ou 1), une variable (x, y, x_1, x_2, ...) ou une addition ($+$) entre deux termes. Une **formule** est définie par induction comme une égalité entre deux termes ($=$), une négation ($\neg$), conjonction ($\wedge$), disjonction ($\vee$), implication ($\rightarrow$) de formules, ou une quantification existentielle ($\exists$) ou universelle ($\forall$) d'une formule.

Par exemple, φ_0 définie par $\forall x\, \exists y\, (x = 0 \vee x = y + 1)$ et φ_1 définie par $\forall x\, \forall y\, \forall z\, (x + y) + z = x + (y + z)$ sont des formules en logique de Presburger.

Pour réaliser des démonstrations en arithmétique de Presburger, on considère les règles données en annexes, correspondant aux règles de la logique classique auxquelles on ajoute des axiomes et le raisonnement par récurrence.

1. Montrer que les règles suivantes (où t, u et v désignent des termes) sont dérivables à partir des règles $(=_i)$ et $(=_e)$:

$$\frac{\Gamma \vdash u = v}{\Gamma \vdash t[x := u] = t[x := v]}\ =_c \qquad \frac{\Gamma \vdash t = u}{\Gamma \vdash u = t}\ =_s \qquad \frac{\Gamma \vdash t = u \quad \Gamma \vdash u = v}{\Gamma \vdash t = v}\ =_t$$

2. Montrer que la règle (R_2) est dérivable à partir des règles de la logique classique et des autres règles de l'arithmétique de Presburger.
3. Montrer que la formule φ_1 est démontrable.

Pour la suite, on cherche à démontrer que l'arithmétique de Presburger est décidable, c'est-à-dire qu'il existe un algorithme qui, étant donnée une formule en logique de Presburger, détermine si elle est démontrable ou non. On introduit pour cela une sémantique : on dit qu'une formule φ est **vraie dans** $\mathbb{N}$, et on note $\mathbb{N} \vDash \varphi$, si la formule est vraie en considérant les variables quantifiées comme appartenant à $\mathbb{N}$. Pour une formule non close $\varphi(x_1, \ldots, x_n)$ et des entiers $a_1, \ldots, a_n \in \mathbb{N}^n$, on dit que φ est **vraie dans** $\mathbb{N}$ **aux valeurs** $a_1, \ldots, a_n$, et on note $\mathbb{N} \vDash \varphi(a_1, \ldots, a_n)$, si elle est vraie en substituant chaque x_i par l'entier a_i. On admet que la logique de Presburger est **complète** pour cette sémantique, c'est-à-dire que si $\mathbb{N} \vDash \varphi$, alors φ est démontrable.

À un n-uplet d'entiers $(a_1, \ldots, a_n) \in \mathbb{N}^n$, on associe un mot de Σ^*, $\Sigma = \{0, 1\}^n$, noté également $(a_1, \ldots, a_n)$ par abus de notation, défini comme le mot

$$(a_1^{(0)}, \ldots, a_n^{(0)})(a_1^{(1)}, \ldots, a_n^{(1)}) \ldots (a_1^{(m)}, \ldots, a_n^{(m)}),$$

où $(a_i^{(0)} a_i^{(1)} \ldots a_i^{(m)})_2$ correspond à la décomposition binaire de a_i, bits de poids faible en tête. On suppose qu'on rajoute autant de 0 que nécessaire pour que toutes les écritures binaires soient de même taille. Par exemple, au quadruplet $(2, 7, 1, 3)$, on associe le mot $(0, 1, 1, 1)(1, 1, 0, 1)(0, 1, 0, 0)$.

Pour $x_1, \ldots, x_n$ des variables, on appelle **formule élémentaire** une formule de la forme $x_i = 0$, $x_i = 1$, $x_i = x_j$ ou $x_i + x_j = x_k$. On dit qu'une formule $\varphi(x_1, \ldots, x_n)$ est **rationnelle** si le langage sur Σ $L(\varphi) = \{(a_1, \ldots, a_n) \in \Sigma^* \mid \mathbb{N} \vDash \varphi(a_1, \ldots, a_n)\}$ est rationnel.

4. Montrer que les formules élémentaires sont rationnelles.
5. Montrer qu'une formule obtenue par conjonctions, disjonctions, négations et implications de formules élémentaires est rationnelle. En déduire que toute formule sans quantificateur est rationnelle.
6. On suppose $\varphi(x_1, \ldots, x_n)$ rationnelle. Montrer que $\exists x_1\, \varphi(x_1, \ldots, x_n)$ est rationnelle.
7. En déduire que toute formule de la logique de Presburger est rationnelle, puis que la logique de Presburger est décidable.

Annexe : règles d'inférence de la logique de Presburger

Règles de la déduction naturelle

Opérateur	Introduction	Élimination
Implication $\to$	$\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}\ \to_i$	$\dfrac{\Gamma \vdash \varphi \to \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi}\ \to_e$
Conjonction $\wedge$	$\dfrac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi}\ \wedge_i$	$\dfrac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi}\ \wedge_e^g$ et $\dfrac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi}\ \wedge_e^d$
Disjonction $\vee$	$\dfrac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi}\ \vee_i^g$ et $\dfrac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi}\ \vee_i^d$	$\dfrac{\Gamma \vdash \varphi \vee \psi \quad \Gamma, \varphi \vdash \sigma \quad \Gamma, \psi \vdash \sigma}{\Gamma \vdash \sigma}\ \vee_e$
Négation $\neg$	$\dfrac{\Gamma, \varphi \vdash \bot}{\Gamma \vdash \neg\varphi}\ \neg_i$	$\dfrac{\Gamma \vdash \varphi \quad \Gamma \vdash \neg\varphi}{\Gamma \vdash \bot}\ \neg_e$
Atomiques $\top$ et $\bot$	$\dfrac{}{\Gamma \vdash \top}\ \top_i$	$\dfrac{\Gamma \vdash \bot}{\Gamma \vdash \varphi}\ \bot_e$
Égalité $=$	$\dfrac{}{\Gamma \vdash t = t}\ =_i$	$\dfrac{\Gamma \vdash \varphi[x := t] \quad \Gamma \vdash t = u}{\Gamma \vdash \varphi[x := u]}\ =_e$
Existentiel $\exists$	$\dfrac{\Gamma \vdash \varphi[x := t]}{\Gamma \vdash \exists x \varphi}\ \exists_i$	$\dfrac{\Gamma \vdash \exists x\, \varphi \quad \Gamma, \varphi \vdash \psi \quad x \notin V_F(\Gamma) \cup V_F(\psi)}{\Gamma \vdash \psi}\ \exists_e$
Universel $\forall$	$\dfrac{\Gamma \vdash \varphi \quad x \notin V_F(\Gamma)}{\Gamma \vdash \forall x\, \varphi}\ \forall_i$	$\dfrac{\Gamma \vdash \forall x\, \varphi}{\Gamma \vdash \varphi[x := t]}\ \forall_e$

où x désigne une variable, t et u des termes et $V_F(\Gamma)$ et $V_F(\psi)$ désignent les ensembles des variables libres (*Free*) des formules de Γ et de ψ respectivement.

On dispose aussi des règles suivantes (usuelles) :

$$\frac{}{\Gamma, \varphi \vdash \varphi}\ \text{ax} \qquad \frac{\Gamma \vdash \varphi}{\Gamma, \psi \vdash \varphi}\ \text{aff} \qquad \frac{}{\Gamma \vdash \varphi \vee \neg\varphi}\ \text{te}$$

Règles supplémentaires pour la logique de Presburger

On ajoute les axiomes suivants pour l'addition :

$$\frac{}{\Gamma \vdash \forall x\, \neg(0 = x + 1)}\ R_1 \qquad \frac{}{\Gamma \vdash \forall x\, (x = 0 \vee \exists y\, x = y + 1)}\ R_2$$

$$\frac{}{\Gamma \vdash \forall x\, \forall y\, (x + 1 = y + 1) \to (x = y)}\ R_3 \qquad \frac{}{\Gamma \vdash \forall x\, (x + 0 = x)}\ R_4$$

$$\frac{}{\Gamma \vdash \forall x\, \forall y\, (x + y) + 1 = x + (y + 1)}\ R_5$$

Enfin, on ajoute une infinité dénombrable de règles exprimant le principe de récurrence : pour toute formule $\varphi(x, y_1, \ldots, y_n)$ ayant pour variables libres $x, y_1, \ldots, y_n$:

$$\frac{\Gamma \vdash \varphi(0, y_1, \ldots, y_n) \quad \Gamma \vdash \forall x\, (\varphi(x, y_1, \ldots, y_n) \to \varphi(x + 1, y_1, \ldots, y_n))}{\Gamma \vdash \forall x\, \varphi(x, y_1, \ldots, y_n)}\ \text{rec}$$

Exercice 3.3 – Largeur de bande

p. 9

Définition

Soit $G = (V, E)$ un graphe non orienté connexe. Pour une fonction dite **d'étiquetage** $f : V \longrightarrow \mathbb{N}$ injective, on définit la quantité : $\varphi(f, G) = \max\{|f(u) - f(v)|, \{u, v\} \in E\}$.
La **largeur de bande** de G, notée $\varphi(G)$ est le plus petit élément de l'ensemble des $\varphi(f, G)$ lorsque G parcours l'ensemble des fonctions d'étiquetage.

1. Calculer la largeur de bande du graphe suivant :

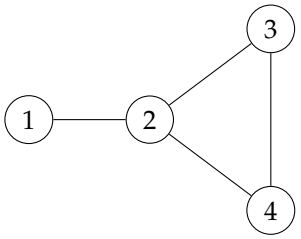
2. Montrer qu'il suffit, dans la définition, de se limiter aux fonctions bijectives $f : V \longrightarrow [\![1; |V|]\!]$. Pour la suite, les fonctions d'étiquetage seront considérées à valeurs dans $[\![1; |V|]\!]$.
3. Calculer la largeur de bande d'un cycle à $n \geqslant 3$ sommets.
4. Montrer que $\deg(G)$, le degré maximal d'un sommet de G, est inférieur ou égal à $2\varphi(G)$.
5. On considère un coloriage des sommets de G tel que chaque arête de G relie deux sommets de couleurs distinctes. Montrer que le nombre minimal de couleurs utilisées pour un tel coloriage est majoré par $\varphi(G) + 1$.
6. Montrer que $\dfrac{|V| - 1}{\delta(G)} \leqslant \varphi(G) \leqslant |V| - \delta(G)$ où $\delta(G)$ désigne le diamètre du graphe, c'est-à-dire la plus grande distance entre deux sommets.

Exercice 3.4 – Dégénérescence de graphe

p. 9

On ne considère dans cet exercice que des graphes simples non orientés.
Pour $k \in \mathbb{N}$, on dit qu'un graphe non-vide G est k**-dégénéré** si tout sous-graphe non vide H de G contient un sommet de degré au plus k dans H. La **dégénérescence** de G est le plus petit $k \in \mathbb{N}$ tel que G est k-dégénéré.

1. Montrer que le graphe G_0 suivant est 2-dégénéré. Quelle est sa dégénérescence ?



2. On note Δ_G le degré maximal d'un sommet d'un graphe G. Montrer que tout graphe G est Δ_G-dégénéré. Est-il vrai que tout graphe G est de dégénérescence Δ_G ?
3. Montrer que dans la définition d'un graphe k-dégénéré, il suffit de considérer les sous-graphes induits.
4. Déterminer un algorithme naïf pour calculer la dégénérescence d'un graphe. Préciser la complexité.
5. Caractériser les graphes de dégénérescence 1.

Un graphe G est dit **régulier** si tous ses sommets ont le même degré.

6. Montrer que tout graphe régulier G est de dégénérescence Δ_G. La réciproque est-elle vraie ? Et si le graphe est connexe ?

Pour $k \in \mathbb{N}$, un k**-arrangement** d'un graphe G est un ordre total sur ses sommets $v_1 < \ldots < v_n$ tel que, pour $i \in [\![1, n]\!]$, on ait $|\{v_j \in V | j < i \text{ et } \{v_i, v_j\} \in E\}| \leqslant k$.

7. Montrer qu'un graphe a un k-arrangement si et seulement s'il est k-dégénéré.
8. Proposer un algorithme efficace pour calculer la dégénérescence.

Solutions

Correction de l'exercice 3.1 page 1

1. Supposons que L_0 soit algébrique et soit n la longueur de pompage donnée par le lemme de pompage algébrique. On pose $u = a^n b^n c^n \in L$. D'après le lemme, u admet une décomposition $u = vwxyz$ vérifiant les trois conditions. Mais si u vérifie les deux premières conditions, alors wy est un mot non vide qui est soit dans a^*b^*, soit dans b^*c^*. Dans les deux cas, vxz ne contient pas autant de a que de b que de c, ce qui contredit la troisième condition. On conclut par l'absurde que L n'est pas algébrique.
   Pour L_1, on pose $u = a^n b^n a^n b^n$. On distingue :
   - si $wy \in a^* \mid b^*$, alors comme $|wxy| \leqslant n$, w et y sont contenus dans la même des quatre portions du mot. Dans vw^2xy^2z, il y a un déséquilibre entre deux sections correspondant à la même lettre, et le mot ne peut donc pas s'écrire comme $u'u'$.
   - si wxy est à cheval sur deux portions, alors on a le même type de déséquilibre dans vxz.
2. Le langage engendré est $\{a^n b^n a^n \mid n \in \mathbb{N}^*\}$. L'idée est que si $S \Rightarrow^* \alpha$ et α contient un S, alors $\alpha = a^n S \beta$, avec $\beta \in \{a, B\}^*$, $|\beta|_a = |\beta|_B = n$. Lors de la dérivation immédiate qui fait disparaître S, on obtient un mot de la forme $a^{n+1} aba\beta$. Dès lors, les variables B ne peuvent disparaître qu'au « contact » d'un b, donc il y a inversion de tous les couples aB en Ba pour faire revenir les B « au centre » et les transformer en b. La grammaire donnée n'est pas monotone, car S apparaît du côté droit d'une règle, mais on peut facilement la transformer sans changer le langage, en supprimant cette règle et en rajoutant $S \to aS'Ba$ et $S' \to aS'Ba \mid aba$. La nouvelle grammaire est monotone.
3. On propose pour L_0 :
   - $S \to aS'Bc \mid \varepsilon$;
   - $S' \to aS'Bc \mid abc$;
   - $cB \to Bc$;
   - $bB \to bb$.

   C'est la même idée que pour le langage précédent.
   Pour L_1, la grammaire donnée par l'énoncé engendre le langage $L_a = \{uaua \mid u \in \{a, b\}^*\}$. Il suffit alors de remarquer que $L_1 = L_a \cup L_b \cup \{\varepsilon\}$ et que les langages monotones sont stables par union (il suffit de faire l'union des alphabets, des règles, d'ajouter un nouveau symbole de départ, et une règle $S \to S_1 \mid S_2$).
4. Soit L un langage engendré par une grammaire algébrique $G = (V, T, P, S)$. On commence par créer un nouveau symbole de départ S' et à rajouter une règle $S' \to S$, pour faire disparaitre le symbole de départ du côté droit de chaque règle. Dès lors, toutes les règles sont monotones, sauf celles de la forme $X \to \varepsilon$. On appelle variable annulable une variable telle que $X \Rightarrow^* \varepsilon$. On peut calculer l'ensemble des variables annulables récursivement (X est annulable si $X \to \varepsilon$ ou $X \to X_1 \ldots X_n$ et tous les X_i sont annulables). Dès lors, pour X annulable et $Y \to \alpha X \beta$ une règle telle que $\alpha\beta \neq \varepsilon$, on ajoute la règle $Y \to \alpha\beta$, puis on supprime $X \to \varepsilon$ (si elle existe). La grammaire obtenue est bien monotone.
5. On repart de la grammaire précédente : on remplace c dans les règles par C, on ajoute une règle $C \to c$, et il suffit alors simplement de remplacer la règle $CB \to BC$. On peut la supprimer et rajouter $CB \to CX$, $CX \to YX$, $YX \to YC$, $YC \to BC$. Ces règles sont bien contextuelles.
6. On commence par remplacer chaque terminal a du côté droit d'une règle par X_a, et on rajoute des règles $X_a \to a$ pour $a \in T$. Après cette transformation, les règles restantes sont de la forme $X_1 \ldots X_n \to Y_1 \ldots Y_m$, avec $n \leqslant m$. On distingue :
   - si $m \leqslant 2$, la règle est autorisée par la forme normale de Kuroda ;
   - si $n = 1$ et $m > 2$, on remplace $X_1 \to Y_1 \ldots Y_m$ par $X_1 \to Y_1 Z_1$, $Z_1 \to Y_2 Z_2$, …, $Z_{m-2} \to Y_{n-1} Y_n$;
   - si $n \geqslant 2$ et $m > 2$, on remplace $X_1 \ldots X_n \to Y_1 \ldots Y_m$ par $X_1 X_2 \to Y_1 Z_1$, $Z_1 X_3 \to Y_2 Z_2$, …, $Z_{n-2} X_n \to Y_{n-1} Z_{n-1}$, $Z_{n-1} \to Y_n \ldots Y_m$, et on remplace cette dernière règle comme à l'étape précédente.

   La grammaire obtenue est bien en forme normale de Kuroda.

7. On remarque qu'une grammaire contextuelle est monotone. Réciproquement, si un langage est monotone, par la question précédente, il est engendré par une grammaire en forme normale de Kuroda. Les seules règles non contextuelles sont de la forme $XY \to ZU$. On peut les remplacer par $XY \to XU'$, $XU' \to Z'U'$, $Z'U' \to Z'U$, $Z'U \to ZU$. Ces règles sont bien contextuelles.
8. Il suffit de remarquer que dans une dérivation de S en u, tous les mots intermédiaires sont de taille $\leqslant$ à celle de u. Par ailleurs, pour $\alpha, \beta \in (V \cup T)^*$, on peut vérifier en temps polynomial si $\alpha \Rightarrow \beta$. On peut alors tester si $u \in L(G)$ en créant un graphe orienté dont les sommets sont les mots de taille $\leqslant |u|$, tel qu'il existe une arête entre α et β si $\alpha \Rightarrow \beta$. Il suffit de tester l'existence d'un chemin de S à u dans ce graphe. Le problème est donc décidable.

Correction de l'exercice 3.2 page 2

1. On pose φ la formule $t[x := u] = t$. On obtient alors l'arbre de preuve suivant :

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash t[x := u] = t[x := u]}\,=_i \quad \Gamma \vdash u = v}{\Gamma \vdash \varphi[x := u]}\,=_e}{\Gamma \vdash t[x := u] = t[x := v]}$$

   Pour la deuxième règle, on pose φ la formule $x = t$. On a :

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash t = t}\,=_i \quad \Gamma \vdash t = u}{\Gamma \vdash \varphi[x := u]}\,=_e}{\Gamma \vdash u = t}$$

   Enfin, pour la troisième règle, c'est une application directe de $=_e$: on pose φ la formule $t = x$. On a alors :

$$\dfrac{\dfrac{\Gamma \vdash t = u \quad \Gamma \vdash u = v}{\Gamma \vdash \varphi[x := u]}\,=_e}{\Gamma \vdash t = v}$$

2. On a l'arbre de preuve :

$$\dfrac{\dfrac{}{\vdash 0 = 0}\,=_i}{\vdash 0 = 0 \vee \exists y\,(0 = y + 1)}\,\vee_i \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash x + 1 = x + 1}\,=_i}{\vdash \exists y\,(x + 1 = y + 1)}\,\exists_i}{\vdash x + 1 = 0 \vee \exists y\,(x + 1 = y + 1)}\,\vee_i}{x = 0 \vee \exists y\,(x = y + 1) \vdash x + 1 = 0 \vee \exists y\,(x + 1 = y + 1)}\,\text{aff}}{\vdash (x = 0 \vee \exists y\,(x = y + 1)) \to (x + 1 = 0 \vee \exists y\,(x + 1 = y + 1))}\,\to_i$$
$$\vdash \forall x\,(x = 0 \vee \exists y\,(x = y + 1))\ \text{rec}$$

   Ce qui montre bien que la règle (R_2) est dérivable à partir des autres.
3. On va faire une preuve par récurrence. On commence par l'initialisation :

$$\dfrac{\dfrac{\dfrac{}{\vdash \forall z\,(z + 0 = z)}\,R_4}{\vdash (x + y) + 0 = x + y}\,\forall_e \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \forall z\,(z + 0 = z)}\,R_4}{\vdash y + 0 = y}\,\forall_e}{\vdash (x + z)[z := y + 0] = (x + z)[z := y]}\,=_c}{\vdash x + (y + 0) = x + y}\,=_s}{\vdash (x + y) + 0 = x + (y + 0)}\,=_t$$

Puis l'hérédité, en notant φ la formule $(x+y)+z = x+(y+z)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \forall y\,\forall z\,(y+z)+1 = y+(z+1)}\,R_5}{\vdash (y+z)+1 = y+(z+1)}\,\forall_e \times 2}{\vdash x+((y+z)+1) = x+(y+(z+1))}\,=_c \quad \dfrac{\dfrac{\dfrac{}{\vdash \forall x\,\forall y\,(x+y)+1 = x+(y+1)}\,R_5}{\vdash \forall y\,(x+y)+1 = x+(y+1)}\,\forall_e}{\vdash (x+(y+z))+1 = x+((y+z)+1)}\,\forall_e}{\dfrac{\vdash (x+(y+z))+1 = x+(y+(z+1))}{\varphi \vdash (x+(y+z))+1 = x+(y+(z+1))}\,\text{aff}}\,=_t$$

puis :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \forall x\,\forall z\,(x+z)+1 = x+(z+1)}\,R_5}{\vdash \forall z\,((x+y)+z)+1 = (x+y)+(z+1)}\,\forall_e}{\vdash ((x+y)+z)+1 = (x+y)+(z+1)}\,\forall_e}{\varphi \vdash ((x+y)+z)+1 = (x+y)+(z+1)}\,\text{aff}}{\varphi \vdash (x+y)+(z+1) = ((x+y)+z)+1}\,=_s \quad \dfrac{\dfrac{}{\varphi \vdash \varphi}\,\text{ax}}{\varphi \vdash ((x+y)+z)+1 = (x+(y+z))+1}\,=_c}{\varphi \vdash (x+y)+(z+1) = (x+(y+z))+1}\,=_t$$

et en regroupant les deux preuves précédentes :

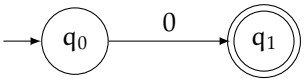
$$\dfrac{\dfrac{\varphi \vdash (x+(y+z))+1 = x+(y+(z+1)) \qquad \varphi \vdash (x+y)+(z+1) = (x+(y+z))+1}{\varphi \vdash (x+y)+(z+1) = x+(y+(z+1))}\,=_t}{\dfrac{\vdash \varphi \to (x+y)+(z+1) = x+(y+(z+1))}{\vdash \forall z\,((x+y)+z = x+(y+z) \to (x+y)+(z+1) = x+(y+(z+1)))}\,\forall_i}\,\to_i$$

On conclut alors :

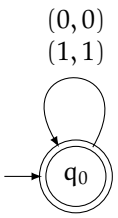
$$\dfrac{\vdash (x+y)+0 = x+(y+0) \qquad \vdash \forall z\,(x+y)+z = x+(y+z) \to (x+y)+(z+1) = x+(y+(z+1))}{\dfrac{\vdash \forall z\,(x+y)+z = x+(y+z)}{\vdash \varphi_1}\,\forall_i \times 2}\,\text{rec}$$

4. On commence par des cas particuliers :

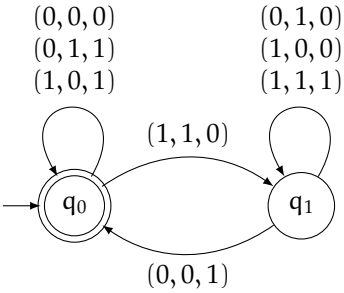
- $x_1 = 0$, pour $n = 1$, est rationnelle, car $L(x_1 = 0)$ est reconnu par :



  De même, $x_1 = 1$ est rationnelle.
- $x_1 = x_2$, pour $n = 2$, est rationnelle, car $L(x_1 = x_2)$ est reconnu par :



- $x_1 + x_2 = x_3$, pour $n = 3$, est rationnelle, car $L(x_1 + x_2 = x_3)$ est reconnu par :



L'état q_0 correspond à l'état sans retenue, et l'état q_1 est l'état qui permet de garder en mémoire la propagation de retenue.

On étend naturellement ces deux solutions à toutes les formules élémentaires en gardant ces transitions pour les variables apparaissant dans la formule et en rajoutant dans chaque uplet toutes les valeurs possibles pour les autres composantes.

5. On peut montrer que $L(\varphi \wedge \psi) = L(\varphi) \cap L(\psi)$, $L(\varphi \vee \psi) = L(\varphi) \cup L(\psi)$, $L(\neg\varphi) = \overline{L(\varphi)}$ et $L(\varphi \to \psi) = L(\neg\varphi \vee \psi)$. Les langages rationnels étant clos par union, intersection et complémentaire, on obtient le résultat voulu.

Par ailleurs, une formule de la forme $t_1 + t_2 + \ldots + t_p = t_{p+1} + \ldots + t_q$, où les t_i sont des variables ou des constantes, peut s'écrire comme une conjonction de formules élémentaires :

- pour chaque t_i qui est une constante, on la remplace par une variable y_i et on rajoute une clause $y_i = t_i$;
- en rajoutant des variables intermédiaires, on peut réduire le nombre d'additions de chaque côté de l'égalité. Par exemple, $y_1 + y_2 + y_3 = y_4 + y_5$ est sémantiquement équivalente à $y_1 + z_2 = z_4 \wedge y_2 + y_3 = z_2 \wedge y_4 + y_5 = z_4$.

En combinant ce résultat avec le précédente, on en déduit que toute formule sans quantificateur est rationnelle.

6. Supposons $\varphi(x_1, \ldots, x_n)$ rationnelle et soit $A = (Q, \Sigma = \{0,1\}^n, \delta, q_0, F)$ un AFD reconnaissant $L(\varphi)$. On pose B l'automate non déterministe défini par $B = (Q, \Sigma' = \{0,1\}^{n-1}, \Delta, \{q_0\}, F')$ tel que :

- pour un état $q \in Q$ et une lettre $(b_1, \ldots, b_{n-1}) \in \Sigma'$, on pose $\Delta(q, (b_1, \ldots, b_{n-1})) = \{\delta(q, (b_1, \ldots, b_{n-1}, 0)), \delta(q, (b_1, \ldots, b_{n-1}, 1)\}$, autrement dit on supprime la dernière composante des lettres dans les transitions de A ;
- $F' = \{q \in Q \mid \exists p \in F \mid \exists u \in (\{0\}^{n-1} \times \{0,1\})^* \mid \delta^*(q, u) = p\}$, autrement dit les états finaux dans B sont ceux qui permettent d'atteindre un état final de A en lisant un mot dont les lettres ne contiennent que des 0 sur les $n-1$ premières composantes (pour gérer le cas où x_n correspond à l'entier a_n avec l'écriture binaire la plus grande).

Montrons que B reconnait $L(\exists x_n\ \varphi)$:

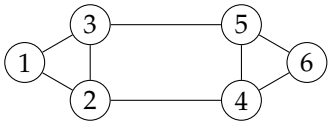
- supposons $(a_1, \ldots, a_{n-1}) \in L(\exists x_n\ \varphi)$. Alors il existe $a_n \in \mathbb{N}$ tel que $\mathbb{N} \vDash \varphi(a_1, \ldots, a_n)$. On en déduit que $(a_1, \ldots, a_n) \in L(\varphi)$. Distinguons deux cas :
  - si $a_n \neq \max\{a_1, \ldots, a_n\}$, alors $\delta^*(q_0, (a_1, \ldots, a_n)) \in F \cap \Delta^*(q_0, (a_1, \ldots, a_{n-1})) \subset F' \cap \Delta^*(q_0, (a_1, \ldots, a_{n-1}))$ donc $(a_1, \ldots, a_{n-1}) \in L(B)$;
  - sinon, $\delta^*(q_0, (a_1, \ldots, a'_n)) \in F' \cap \Delta^*(q_0, (a_1, \ldots, a_{n-1}))$, où a'_n correspond à l'entier dont l'écriture binaire est celle de a_n, tronquée au nombre de bits significatifs le plus grand parmi $a_1, \ldots, a_{n-1}$. À nouveau, on a $(a_1, \ldots, a_{n-1}) \in L(B)$.
- la réciproque se fait sur la même idée : si $(a_1, \ldots, a_{n-1}) \in L(B)$, alors il existe un chemin acceptant dans B. On peut lui associer un chemin acceptant dans A dont les $n-1$ premières composantes des lettres lues correspondent à $(a_1, \ldots, a_{n-1})$. On en déduit qu'il existe $a_n \in \mathbb{N}$ tel que $\mathbb{N} \vDash \varphi(a_1, \ldots, a_n)$, et donc que $\mathbb{N} \vDash \exists x_n\ \varphi(a_1, \ldots, a_{n-1}, x_n)$.

7. On traite le cas $\forall x_n\ \varphi(x_1, \ldots, x_n)$ en remarquant qu'une telle formule est sémantiquement équivalente à $\neg(\exists x_n\ \neg\varphi(x_1, \ldots, x_n))$. De plus, la conjonction, disjonction, négation et implication de formules non élémentaires se traite de la même manière que les formules élémentaires. On conclut par induction sur les formules que toute formule de la logique de Presburger est rationnelle.

On obtient alors un automate sur l'alphabet $\Sigma = \{0,1\}^0$ (c'est-à-dire dont les lettres sont des 0-uplets). Il suffit alors de remarquer que $\mathbb{N} \vDash \varphi \Leftrightarrow L(\varphi) \neq \emptyset$. Or, étant donné un automate fini (déterministe ou non), on peut tester si le langage reconnu est vide par un parcours de graphe : il faut vérifier s'il existe un chemin depuis un état initial vers un état final. Tout cela donne bien un algorithme permettant de tester si une formule est vraie dans $\mathbb{N}$. Comme on a admis la complétude de la logique de Presburger pour cette sémantique, on en déduit que la logique de Presburger est décidable.

Correction de l'exercice 3.3 page 4

1. Il existe un étiquetage montrant que $\varphi(G) \leqslant 2$:s



   De plus, soit f une fonction d'étiquetage. Il existe un sommet u de degré 3, donc au moins l'un de ses voisins v vérifie $|f(u) - f(v)| \geqslant 2$.
   Finalement, on en déduit $\varphi(G) = 2$.

2. Soit f une fonction d'étiquetage d'image $f(V)$. Soit g l'unique fonction strictement croissante de $f(S)$ dans $[\![1; |S|]\!]$. Alors pour tout sommet u, v, $|f(u) - f(v)| \geqslant |\tilde{f}(u) - \tilde{f}(v)|$.

3. Dans un premier temps, remarquons que pour un étiquetage f, on ne peut pas avoir $\varphi(G, f) = 1$. En effet, cela imposerait aux d'étiquettes 1 et n d'être voisins. Dès lors :
   - Si n est pair, on utilise l'étiquetage : $1, 3, 5, \ldots, n-1, n, n-2, \ldots, 4, 2$ qui vérifie $\varphi(G, f) = 2$.
   - Si n est impair, on utilise l'étiquetage : $1, 3, 5, \ldots, n-2, n, n-1, \ldots, 4, 2$ qui vérifie $\varphi(G, f) = 2$.

   On en déduit que $\varphi(G) = 2$.

4. Soit f une fonction d'étiquetage telle que $\varphi(G) = \varphi(G, f)$ et soit v un sommet quelconque. On sait qu'il existe au plus $2\varphi(G)$ entiers $k \neq f(v)$ tels que $|f(v) - k| \leqslant \varphi(G)$. On en déduit donc, par injectivité qu'il existe au plus $2\varphi(G)$ voisins de v.

5. Montrons qu'il existe un coloriage à $\varphi(G) + 1$ couleurs. Soit f une fonction d'étiquetage telle que $\varphi(G) = \varphi(G, f)$. Associons à chaque sommet v la couleur $f(v) \mod (\varphi(G) + 1)$. Deux sommets voisins u et v sont de la même couleur si et seulement si $\varphi(G) + 1$ divise $|f(u) - f(v)|$. Or par définition de $\varphi(f, G)$ cette quantité est comprise entre 1 et $\varphi(G)$. On en déduit le résultat.

6. Soit f une fonction d'étiquetage de G, v_1 et v_p les sommets tels que $f(v_0) = 1$ et $f(v_p) = |S|$. Soit $v_0, v_1, \ldots, v_p$ les sommets d'un chemin simple reliant v_0 à v_p. Alors on a :

$$|S| - 1 = |f(v_0) - f(v_p)| \leqslant \sum_{k=1}^{p} |f(v_{k-1} - f(v_k)| \leqslant p\varphi(G) \leqslant \delta(G)\varphi(G)$$

   On a alors la première inégalité.
   Soit $D = v_0 \ldots v_\delta$ un diamètre (simple) de G. Soit f la fonction d'étiquetage des sommets dans l'ordre d'un parcours en largeur depuis v_0. Posons $V_k = \{v \in V | d(v_0, v) = k\}$ pour k allant de 0 à $\delta(G)$. Alors deux sommets voisins u et v sont :
   - Soit dans le même S_k, et dans ce cas, $|f(u) - f(v)| \leqslant |S_k| - 1 = |S_k| - |S_k \cap D| \leqslant |S| - \delta(G)$.
   - Soit dans deux ensembles consécutifs S_k et S_{k+1}, et alors :

$$\begin{aligned} |f(u) - f(v)| &\leqslant |S_k| + |S_{k+1}| - 1 \\ &\leqslant |S_k| + |S_{k+1}| - 1 + \sum_{i=0}^{k-1} (|S_i| - 1) + \sum_{i=k+1}^{\delta(G)} (|S_i| - 1) \\ &= |S| - 1 - k - (\delta(G) - (k+2) + 1) \\ &= |S| - \delta(G) \end{aligned}$$

   On a bien la majoration attendue.

Correction de l'exercice 3.4 page 4

1. Dans ce graphe, le seul sommet de degré 3 est le sommet 2. Tout sous-graphe contenant un des trois autres sommets contient bien un sommet de degré au plus 2. Un sous-graphe qui ne contient aucun des trois est soit vide, soit réduit au sommet 2. Dans un tel sous-graphe, le sommet 2 est de degré 0. La dégénérescence de G_0 est 2, car le sous-graphe induit à $V' = \{2, 3, 4\}$ ne contient aucun sommet de degré 1.

2. La première affirmation est triviale (le degré d'un sommet dans un sous-graphe est nécessairement inférieur ou égal au degré de ce sommet dans G). La réciproque est fausse, comme le montre

le graphe G_0.

3. Il est clair que si un graphe est k-dégénéré pour la définition initiale, il l'est en ne considérant que les sous-graphes induits. Réciproquement, il suffit de remarquer que le degré d'un sommet dans un sous-graphe est toujours $\leqslant$ au degré du même sommet dans le sous-graphe induit associé aux mêmes sommets.
4. La dégénérescence est exactement :
$$\max_{(V',E') \text{ sous-graphe}} \left(\min_{v \in V'} (\deg(v)) \right)$$
Il suffit alors de faire ce calcul (on peut ne considérer que les sous-graphes induits).
Pour l'écriture du code, on peut envisager d'énumérer tous les sous-ensembles de V par backtracking en utilisant un tableau de booléens (je mets un sommet à `true`, je lance un appel récursif sur la suite, je mets le sommet à `false` et je relance un appel sur la suite). Une fois un sous-ensemble énuméré, on peut calculer le degré minimal en comptant, pour chaque sommet, le nombre de sommets de sa liste d'adjacence qui sont à `true`. La calcul du degré minimal se fait en $\mathcal{O}(|V|+|E|)$ et il y a $2^{|V|}$ sous-graphes possibles.
5. Ces graphes sont exactement les graphes sans cycle contenant au moins une arête (cela se prouve raisonnablement facilement).
6. La dégénérescence d'un graphe est au plus Δ_G (question 2). De plus, si le graphe est régulier, le sous-graphe G lui-même ne contient aucun sommet de degré $< \Delta_G$. La dégénérescence est donc bien Δ_G.
La réciproque est fausse, par exemple en considérant le graphe G_0 dans lequel on supprime l'arête $\{1,2\}$.
La réciproque est en revanche vraie si on suppose le graphe connexe. En effet, supposons G de dégénérescence Δ_G et soit $H = (V', E')$ un sous-graphe induit non vide, minimal pour l'inclusion, dont le degré minimal est Δ_G. Alors ce sous-graphe est régulier (car Δ_G est aussi le degré maximal), et il est égal à G (car sinon, il existerait une arête absente d'un sommet u de V' à un sommet de $V \setminus V'$, et u serait de degré strictement inférieur à Δ_G dans H).
7. Soit un k-arrangement $v_1 < \ldots < v_n$ de G. Montrons que G est k-dégénéré. Soit H un sous-graphe non vide de G et v le maximum de H pour l'ordre $<$. Alors ses voisins dans H sont $< v$, donc il y en a au plus k, donc H contient bien un sommet de degré $\leqslant k$.

   Réciproquement, montrons par induction sur $|V|$ que si G est k-dégénéré, alors il a un k-arrangement.
   - le résultat est vrai si $|V| = 1$, car G est k-dégénéré et admet un k-arrangement pour tout $k \in \mathbb{N}$;
   - supposons le résultat vrai pour $|V| \leqslant n \in \mathbb{N}^*$ et sois G un graphe à $n+1$ sommets qui est k-dégénéré. G admet donc un sommet v_{n+1} de degré $\leqslant k$. Soit G' le sous-graphe induit à $V \setminus \{v_{n+1}\}$. Alors G' est toujours k-dégénéré et a n sommet. Soit alors un k-arrangement $v_1 < v_2 \ldots < v_n$ de G' et considérons $v_n < v_{n+1}$. C'est bien un k-arrangement de G.
8. L'algorithme consiste à construire un k-arrangement. On effectue cela en choisissant le sommet de degré le plus faible, en le mettant à la fin de la liste, et en recommençant en enlevant ce sommet.
   Pour ce faire, on commence par construire une liste des degrés des sommets (linéaire). Pour trouver le sommet de degré minimal se fait alors en temps linéaire, de même que la mise à jour des degrés des autres sommets après sa suppression. L'ordre $v_1 < \ldots < v_n$ ainsi produit peut se faire en temps quadratique en n. Il suffit alors de trouver la valeur maximale du degré de v_i dans le sous-graphe induit par $\{v_1, \ldots, v_i\}$, ce qui se fait à nouveau en temps quadratique.

   Montrons alors que cet ordre total est optimal. Pour cela, soit un ordre total quelconque $u_1 < \ldots < u_n$ qui réalise la meilleure valeur possible pour k et montrons qu'on peut le modifier pour qu'il soit de la forme donnée par l'algorithme. Sachant qu'on peut éventuellement éliminer des suffixes qui coïncident, soit $u_1 < \ldots < u_p$ le plus grand préfixe, c'est-à-dire tel que $u_p \neq v_p$ mais $u_i = v_i$ pour $i > p$. Posons $u_q = v_p$. Dans l'ordre $u_1 < \ldots < u_{q-1} < u_{q+1} < \ldots < u_p < u_q$, la valeur de k réalisée n'est pas pire : en effet, le degré de u_q dans le sous-graphe induit par $\{u_1, \ldots, u_p\}$ est $\leqslant$ à celui de u_p (par définition de l'algorithme) ; de plus, pour chaque autre sommet, il y a un sommet de moins dans la liste des sommets strictement inférieurs. À nouveau, la valeur de k ne peut pas augmenter. On peut réitérer le processus pour retomber sur $v_1 < \ldots < v_n$.