Feuille 1

Exercice 1

Nous supposons un ensemble dénombrable $\mathcal{X}$ de variables $x, y, z, \cdots, x_0, x_1, \cdots$.

1. L'ensemble des expressions, dénoté par Exp, est défini inductivement de la manière suivante :
   - Cas de base une variable dans $\mathcal{X}$ est une expression, une constante dans $\boldsymbol{Z}$ est une expression.
   - Induction : Si e_1 et e_2 sont des expressions alors (e_1), $e_1 + e_2$, $e_1 - e_2$, $e_1 * e_2$ sont des expressions.
2. Pour donner une sémantique à une expression, il faut donc supposer des valeurs associées aux variables. Une telle association c.a.d. une application $\sigma : \mathcal{X} \longrightarrow \boldsymbol{Z}$ est apppelée état. L'ensemble des états est dénoté Σ.
   Pour un état $\sigma \in \Sigma$ et une expression e, on dénote par $[\![e]\!]\sigma$ la valeur de e dans l'état σ. Nous définissons l'application $[\![\ ]\!] : Exp \longrightarrow (\Sigma \longrightarrow \boldsymbol{Z})$:
   - Cas de base :
     – pour une variable x, on a $[\![x]\!]\sigma = \sigma(x)$
     – pour une constante n, on a $[\![n]\!]\sigma = n$
   - Pas d'induction : $[\![e_1\, op\, e_2]\!]\sigma = ([\![e_1]\!]\sigma)\ op\ ([\![e_2]\!]\sigma)$ et $[\![(e)]\!]\sigma = [\![e]\!]\sigma$, ou $op \in \{+, -, *...\}$.
3. Notation:
   - Soit $\sigma : \mathcal{X} \longrightarrow \boldsymbol{Z}$ un état et $n \in \boldsymbol{Z}$. Alors, $\sigma[n/x]$ dénote l'état qui associe à toute variable y autre que x la valeur $\sigma(y)$ et à x la valeur n.
   - Soit e, e' deux expressions. Alors, $e[e'/x]$ dénote l'expression obtenue en remplaçant par (e') toutes les occurences de la variable x dans e.

Exercices

1. Soit σ un état tel que $\sigma(x) = 1$, $\sigma(y) = -1$ et $\sigma(z) = 3$.
   Quelle est la valeur des expressions suivantes dans σ :
   - $x + y$
   - $0 - y$
   - $x + z + y$
   - $x + y * z$
   - $(x + y) * z$
2. Soit $e = (x * y) + (2 * x)$, $e' = y + 1$, $e'' = x + y$. Donnez les expressions obtenues pour:
   - $e[e'/x]$
   - $e[e''/x]$
   - $e[e'/y]$
   - $e'[e/x]$
3. Montrer les egalités suivantes:

- $\sigma[n/x](x) = n$, ou n est une constante.
- $\sigma[n/x](y) = \sigma(y)$, si y est différent de x.
- $\sigma[\sigma(x)/x] = \sigma$.
- $[\![e]\!]\sigma[n/x] = [\![e]\!]\sigma$, si x n'apparait pas dans e.
- $[\![e[e'/x]]\!]\sigma = [\![e]\!]\sigma[[\![e']\!]\sigma/x]$.

Exercice 2

1. **Prédicats et formules logiques** L'ensemble des prédicats (formules logiques) de 1er-ordre est défini inductivement par :
   - Base :
     - Les constantes V et F sont des prédicats.
     - Si e_1, e_2 sont des expressions alors $e_1 = e_2$, $e_1 < e_2$, $e_1 \leq e_2$, $e_1 > e_2$, $e_1 \geq e_2$ sont des prédicats.
   - Induction : Si P_1 et P_2 sont des prédicats alors $\neg P_1$, $P_1 \wedge P_2$, $P_1 \vee P_2$, $\exists x \cdot P_1$ et $\forall x \cdot P_1$ sont des prédicats.
2. **Sémantique des prédicats** Pour chaque prédicat P et état σ, nous voulons définir quand σ satisfait P, dénoté par $\sigma \models P$. Cette définition est par induction sur la structure de P :
   - Cas de base :
     - $\sigma \models \mathsf{V}$ et $\sigma \not\models \mathsf{F}$
     - $\sigma \models e_1 \sim e_2$ ssi $[\![e_1]\!]\sigma \sim [\![e_2]\!]\sigma$, pour $\sim \in \{<, \leq, =, >, \geq, ...\}$.
   - Induction :
     - $\sigma \models \neg P$ ssi $\sigma \not\models P$
     - $\sigma \models P_1 \wedge P_2$ ssi $\sigma \models P_1$ et $\sigma \models P_2$
     - $\sigma \models P_1 \vee P_2$ ssi $\sigma \models P_1$ ou $\sigma \models P_2$
     - $\sigma \models \exists x \cdot P$ ssi il existe $n \in \boldsymbol{Z}$ tel que $\sigma[n/x] \models P$
     - $\sigma \models \forall x \cdot P$ ssi pour tout $n \in \boldsymbol{Z}$, on a $\sigma[n/x] \models P$
3. **Validité et satisfiabilité**
   - Un prédicat P est valide, si pour tout état σ on a $\sigma \models P$.
   - Un prédicat P est satisfiable, s'il existe un état σ tel que $\sigma \models P$.
   - Un prédicat P est insatisfiable, si pour tout état σ on a $\sigma \not\models P$.

Exercices

1. Pour chacun des prédicats suivants, dites s'il est valide, satisfiable ou insatisfiable.
   - $\forall y \exists x \cdot (y = 2 * x \vee y = 2 * x + 1)$
   - $\forall x \exists y \cdot (y < x \wedge y \geq 0)$
   - $\forall x \forall y \cdot (y < x \vee y \geq 0 \vee x < 0)$
   - $\forall y \cdot (x = 2 * y)$
   - $\exists y \cdot (x = 2 * y)$
   - $\exists x \exists y \cdot (x = 2 * y)$
2. Montrez que pour tout prédicat P, P est valide ssi $\neg P$ est insatisfiable.

Exercice 3

On considère l'automate étendu de la Fig. 7. L'état q_0 est l'unique état initial et q_2

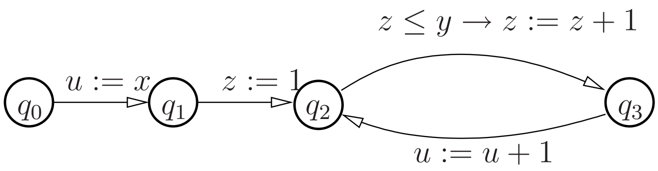

Figure 1: Figure

l'unique état terminal.

1. Calculer les exécutions de cet automate dans les états initiaux suivants :
   (a) $\sigma(x) = 2$, $\sigma(y) = 3$ et $\sigma(v) = 0$, pour toute variable v autre que x et y.
   (b) $\sigma(x) = 2$, $\sigma(y) = 0$ et $\sigma(v) = 0$, pour toute variable v autre que x et y.
   (c) $\sigma(x) = 2$, $\sigma(y) = -3$ et $\sigma(v) = 0$, pour toute variable v autre que x et y.
2. Déterminer la relation entre états induite par cet automate.
3. Donner une spécification (P, Q) satisfaite par cet automate (sans preuve pour l'instant).

Exercice 4

1. Déterminer l'ensemble des automates étendus partiellement correct par rapport à (V, F).
2. Déterminer l'ensemble des automates étendus partiellement correct par rapport à (V, V).
3. Déterminer l'ensemble des automates étendus partiellement correct par rapport à (F, F).
4. Déterminer l'ensemble des automates étendus partiellement correct par rapport à (F, V).

Exercice 5

On considère l'automate A donné par le quintuplet suivant :

1. $y : \boldsymbol{Z}, z : \boldsymbol{Z}$
2. $Q = \{q_0, q_1, q_2\}$
3. $Q_0 = \{q_0\}$
4. $\mathcal{T} = \{(q_0, z := 0, q_1), (q_1, z < y - 1 \to z := z + 1, q_1), (q_1, \neg z < y - 1, q_2)\}$
5. $Q_t = \{q_2\}$

1. Dessiner cet automate.
2. Calculer les exécutions de cet automate dans les états initiaux suivants :
   (a) $\sigma(y) = 3$ et $\sigma(v) = 0$, pour toute variable v autre que y.
   (b) $\sigma(y) = 1$ et $\sigma(v) = 0$, pour toute variable v autre que y.
   (c) $\sigma(x) = 2$, $\sigma(y) = -3$ et $\sigma(v) = 0$, pour toute variable v autre que y.

3. Déterminer la relation entre états induite par cet automate.
4. Donner une spécification (P, Q) satisfaite par cet automate (sans preuve pour l'instant).

Exercice 6

Donner un automate qui satisfait la spécification

$$(x = x_0 \wedge y = y_0 \wedge x_0 \geq 0 \wedge y_0 \geq 0, z = x_0 * y_0)$$

et qui n'utilise que l'addition comme opération.

Exercice 7

On considère l'automate étendu de la Fig. 8. L'état q_0 est l'unique état initial et q_t

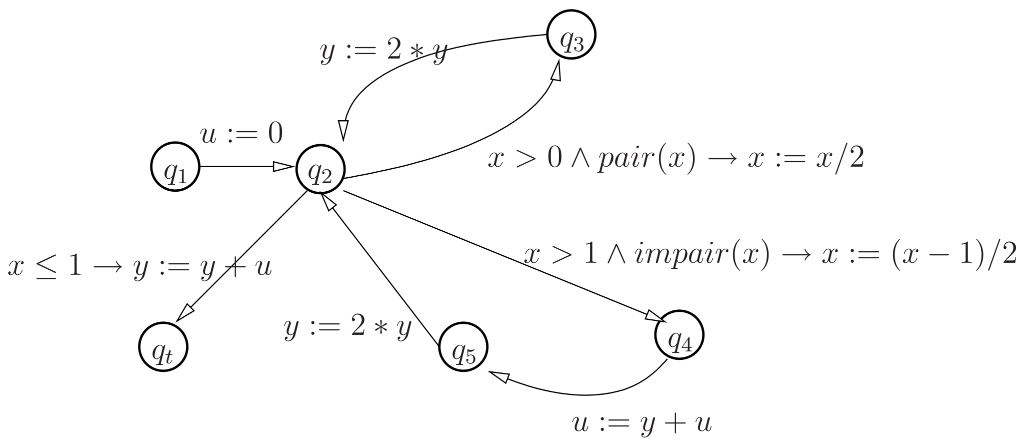

Figure 2: Figure

est l'unique état terminal.

1. Calculer les exécutions de cet automate dans les états initiaux suivants :
   (a) $\sigma(x) = 4$, $\sigma(y) = 3$ et $\sigma(v) = 0$, pour toute variable v autre que x et y.
   (b) $\sigma(x) = 5$, $\sigma(y) = 0$ et $\sigma(v) = 0$, pour toute variable v autre que x et y.
   (c) $\sigma(x) = 2$, $\sigma(y) = -3$ et $\sigma(v) = 0$, pour toute variable v autre que x et y.
   (d) $\sigma(x) = -2$, $\sigma(y) = 3$ et $\sigma(v) = 0$, pour toute variable v autre que x et y.
2. Déterminer la relation entre états induite par cet automate.
3. Donner une spécification (P, Q) satisfaite par cet automate (sans preuve pour l'instant).

Feuille 2

Exercice 8

On considère l'automate A étendu de la Fig. 7. L'état q_0 est l'unique état initial et

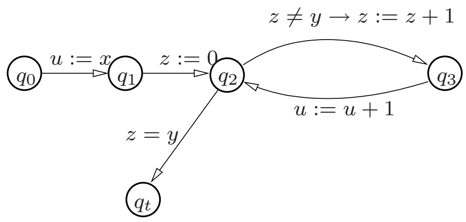

Figure 3: Figure

q_t l'unique état terminal.

- Démontrer que A satisfait la spécification $(y = y_0 \wedge y_0 \geq 1, u = x + y)$.

Exercice 9

On considère l'automate A donné par le quintuplet suivant :

1. $y : \boldsymbol{Z}, z : \boldsymbol{Z}$
2. $Q = \{q_0, q_1, q_2\}$
3. $Q_0 = \{q_0\}$
4. $\mathcal{T} = \{(q_0, z := 0, q_1), (q_1, z < y - 1 \rightarrow z := z + 1, q_1), (q_1, \neg z < y - 1, q_2)\}$
5. $Q_t = \{q_2\}$

Démontrer que A satisfait la spécification $(y = y_0 \wedge y \geq 1, z = y_0 - 1)$.

Exercice 10

On considère l'automate A étendu de la Fig. 8. L'état q_0 est l'unique état initial et q_t est l'unique état terminal.

- $P_{q_1} : x = x_0 \wedge y = y_0 \wedge x_0 > 0.$
- $P_{q_2} : x > 0 \wedge y * x + u = y_0 * x_0$
- $P_{q_3} : x > 0 \wedge \cdots * y * x + u = y_0 * x_0$
- $P_{q_4} : x > 0 \wedge y * (2x + \cdots) + u = y_0 * x_0$
- $P_{q_5} : x > 0 \wedge 2 * x * y + u = y_0 * x_0$
- $P_{q_t} : y = x_0 * y_0$

Démontrer que A satisfait la spécification

$$(x = x_0 \wedge y = y_0 \wedge x_0 > 0, y = x_0 * y_0).$$

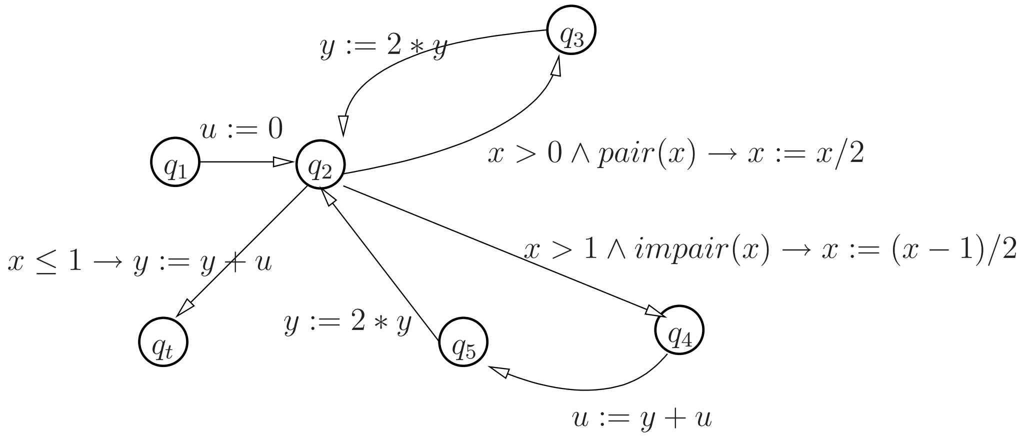

Figure 4: Figure

Exercice 11

Démontrer que l'automate de la figure 9 satisfait la spécification :

$(a > 0 \wedge b \geq 0, x = a^b)$

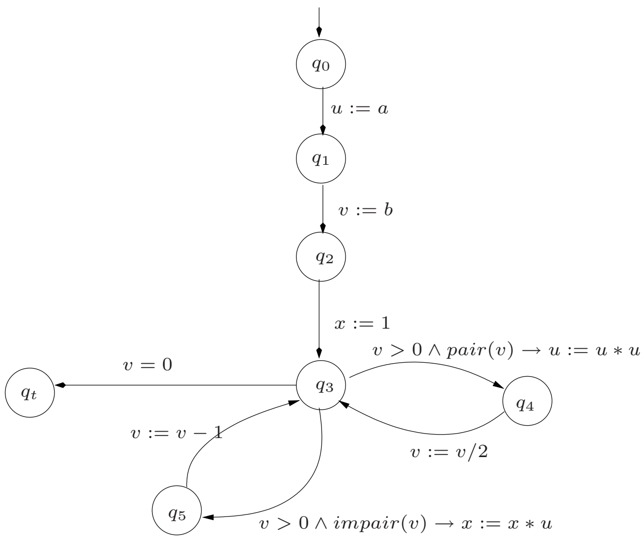

Figure 5: Puissance

Exercice 12

Nous défnissons récursivement la fonction Fibonacci $fib : \mathbb{N} \longrightarrow \mathbb{N}$:

- $fib(0) = 0$ et $fib(1) = 1$ et
- $fib(n+2) = fib(n) + fib(n+1)$.

Démontrer que l'automate de la figure 6 satisfait la spécification :

$$(x = x_0 \wedge x \geq 0, z = fib(x))$$

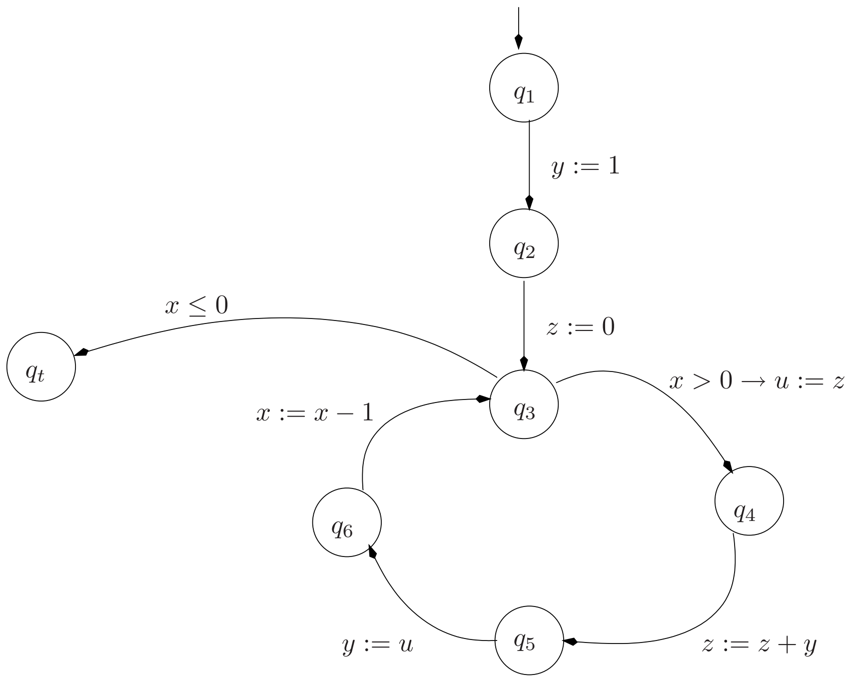

Figure 6: Fibonacci

Feuille 3

Exercice 13

On considère l'automate A étendu de la Fig. 7. L'état q_0 est l'unique état initial et

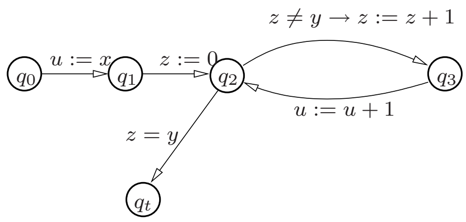

Figure 7: Figure

q_t l'unique état terminal.

- Démontrer que si les valeurs initiales des variables σ_0 satisfont la pré-condition $y = y_0 \wedge y_0 \geq 1$ alors l'automate términe (c.a.d. ne diverge pas)
- Démontrer que A satisfait **totallement** la spécification $(y = y_0 \wedge y_0 \geq 1, u = x + y)$.

Exercice 14

On considère l'automate A donné par le quintuplet suivant :

1. $y : \boldsymbol{Z}, z : \boldsymbol{Z}$
2. $Q = \{q_0, q_1, q_2\}$
3. $Q_0 = \{q_0\}$
4. $\mathcal{T} = \{(q_0, z := 0, q_1), (q_1, z < y - 1 \to z := z + 1, q_1), (q_1, \neg z < y - 1, q_2)\}$
5. $Q_t = \{q_2\}$

- Démontrer que si les valeurs initiales des variables σ_0 satisfont la pré-condition $y = y_0 \wedge y_0 \geq 1$ alors l'automate términe (c.a.d. ne diverge pas)
- Démontrer que A satisfait **totallement** la spécification $(y = y_0 \wedge y_0 \geq 1, z = y_0 - 1)$.

Exercice 15

On considère l'automate A étendu de la Fig. 8. L'état q_0 est l'unique état initial et q_t est l'unique état terminal.

- Démontrer que si les valeurs initiales des variables σ_0 satisfont la pré-condition $x = x_0 \wedge y = y_0 \wedge x_0 > 0$ alors l'automate términe (c.a.d. ne diverge pas)

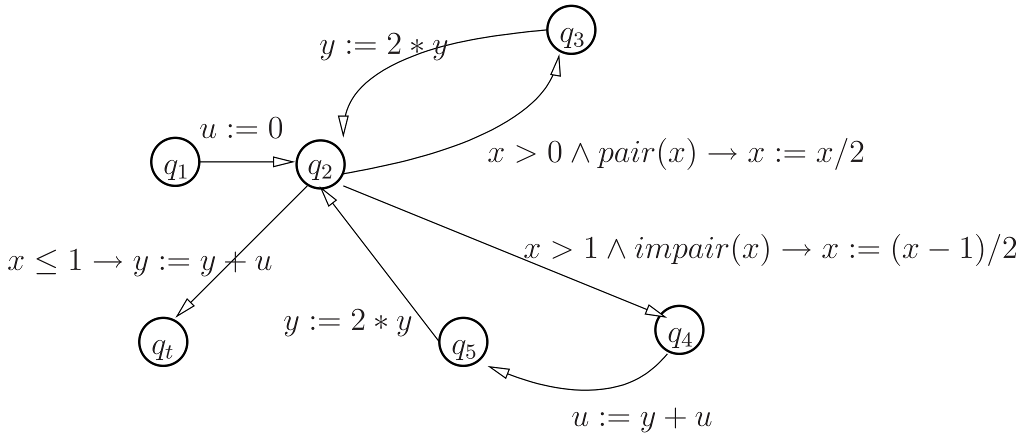

Figure 8: Figure

- Démontrer que A satisfait **totallement** la spécification $(x = x_0 \wedge y = y_0 \wedge x_0 > 0, y = x_0 * y_0)$.

Exercice 16

Démontrer que l'automate de la figure 9 satisfait **totallement** la spécification :

$(a > 0 \wedge b \geq 0, x = a^b)$

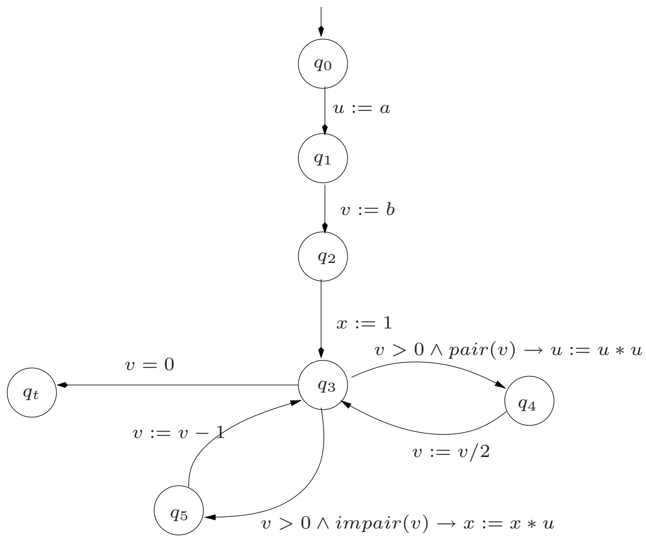

Figure 9: Puissance

Exercice 17

Démontrer que l'automate de la figure 10 satisfait **totallement** la spécification :$(x = x_0 \wedge x_0 \geq 1, x = 1)$.

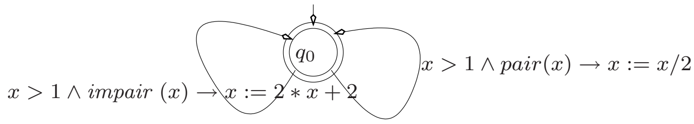

Figure 10: Choice

Exercice 18

Démontrer que l'automate de la figure 11 satisfait **totallement** la spécification :($x = x_0 \wedge x_0 \geq 0, y = x_0^2$).

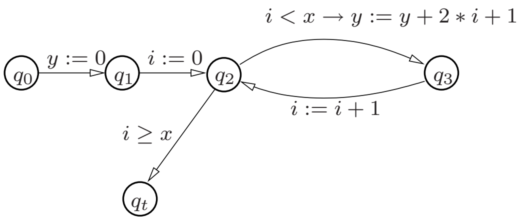

Figure 11: Carré

Exercice 19

On considère l'automate étendu de la Figure 12

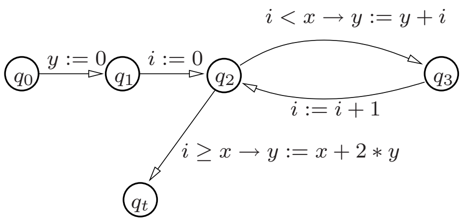

Figure 12: Automate étendu

On considère aussi la pré-condition $P \equiv x = x_0 \wedge x_0 \geq 0$ et la post-condition $Q \equiv y = x_0^2$.

- Calculer les exécutions de cet automate dans les états initiaux suivants :
  1. $\sigma(x) = 3$, $\sigma(y) = 3$ et $\sigma(v) = 0$, pour toute variable v autre que x et y.
  2. $\sigma(x) = 4$, $\sigma(y) = -3$, $\sigma(i) = 4$ et $\sigma(v) = 0$, pour toute variable v autre que x, y et i.
- Compléter les prédicats suivants pour obtenir un automate étendu annoté correct (c'est-à-dire un automate étendu annoté inductif) :
  1. $P_{q_0} : P$
  2. $P_{q_1} : P_{q_0} \wedge y = \cdots$
  3. $P_{q_2} : x = x_0 \wedge i \leq \cdots \wedge y = \frac{(i-1)*i}{2}$
  4. $P_{q_3} : x = x_0 \wedge \cdots < x \wedge y = \frac{i*\cdots}{2}$

5. $P_{q_t} : y = i^2 \wedge i = x_0$

- Démontrer que si les valeurs initiales des variables σ_0 satisfont la pré-condition P, alors l'automate términe (c.a.d. ne diverge pas)
- Démontrer que A satisfait **totallement** la spécification (P, Q).

Feuille 4

Machines de Turing

Exercice 20

Soit M la machine de Turing donnée par $(Q, \Sigma, \Gamma, \Delta, q_0, B, F)$ où :

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$.
- $\Sigma = \{a, b\}$.
- $\Gamma = \{a, b, X, Y, B\}$.
- $$\begin{array}{rl} \Delta = & \{(q_0, a, D, X, q_1), (q_0, Y, D, Y, q_3), (q_1, a, D, a, q_1), \\ & (q_1, b, G, Y, q_2), (q_1, Y, D, Y, q_1), (q_2, a, G, a, q_2), \\ & (q_2, X, D, X, q_0), (q_2, Y, G, Y, q_2), (q_3, Y, D, Y, q_3), \\ & (q_3, B, D, B, q_4)\} \end{array}$$
- q_0 est l'état initial.
- $F = \{q_4\}$

1. Le mot a^2b^2 est-il accepté par M?
2. Le mot a^2b^3 est-il accepté par M?
3. Le mot a^3b^2 est-il accepté par M?
4. Quel langage est reconnu par cette machine?

Exercice 21

Soit $\mathcal{M}$ la machine de Turing avec l'alphabet d'entrée $\Sigma = \{a, b\}$, l'alphabet du ruban $\Gamma = \Sigma \cup \{B\}$, d'états $\{q_0, \ldots q_8\}$, d'état final q_8, et de fonction de transition donnée par la table suivante :

État	Symbole	(état suivant,symbole,mouvement)
q_0	B	(q_1, B, D)
q_1	a	(q_1, a, D)
q_1	b	(q_1, b, D)
q_1	B	(q_2, B, G)
q_2	a	(q_3, B, D)
q_2	b	(q_5, B, D)
q_2	B	(q_8, B, I)
q_3	B	(q_4, a, D)
q_4	a	(q_4, a, D)
q_4	b	(q_4, b, D)
q_4	B	(q_7, a, G)
q_5	B	(q_6, b, D)
q_6	a	(q_6, a, D)
q_6	b	(q_6, b, D)
q_6	B	(q_7, b, G)
q_7	a	(q_7, a, G)
q_7	b	(q_7, b, G)
q_7	B	(q_2, B, G)

On suppose que la machine commence avec sa tête de lecture positionnée sur la première lettre de l'entrée (donc le premier blanc B).

1. Que fait cette machine de Turing sur l'entrée $BabBBBBBBBBBBBBB$? (donner les configurations successives)
2. Que fait cette machine sur l'entrée $BbaaBBBBBBBBBBBB$? (ne pas donner les configurations successives)
3. Décrire ce que fait cette machine sur un entrée arbitraire de la forme $BuBBBBBBBBBBBBBB$ avec $u \in \Sigma^*$.
4. Modifier cette machine pour ajouter le caractère a au milieu du mot obtenu.

Exercice 22

Construisez une machine de Turing qui prend en entrée un mot $a_0a_1 \ldots a_nBBBBB\ldots$, où $a_i \in \{0,1\}^+$ et qui renvoie en sortie $a_1 \ldots a_nBBBBB\ldots$, où B est le blanc.

Exercice 23

Construisez une machine de Turing qui prend en entrée un mot $a_1 \ldots a_nBBBBB\ldots$, où $a_i \in \{0,1\}^+$ et qui renvoie en sortie $Ba_1 \ldots a_nBBBBB\ldots$, où B est le blanc.

Exercice 24

Soit Σ un alphabet. L'image $R(u)$ par miroir d'un mot u est le mot qu'on obtient en lisant u de droite à gauche (comme en arabe ou en hébreu). Plus précisemment:

- $R(\epsilon) = \epsilon$ et
- $R(u \cdot a) = a \cdot R(u)$.

Trouvez une machine de Turing qui prend en entrée un mot $w \in \Sigma^*$ et qui renvoie $R(w)$.

Exercice 25

Soit $\Sigma = \{0, 1, \#\}$. Trouvez une machine de Turing qui reconnait le langage

$$\{u \in \Sigma^+ \mid \exists v \in \{0,1\}^+ \; u = v\#v\}.$$

Exercice 26

Soit $\Sigma = \{0, 1\}$. Trouvez une machine de Turing qui reconnait le langage

$$\{u \in \Sigma^+ \mid \exists v \in \Sigma^+ \; u = vv\}.$$

Exercice 27

Soit Σ un alphabet fini et soit P un automate à pile défini par $P = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0, F)$. Trouvez une machine de Turing qui reconnait le langage $L(P)$.

Exercice 28

Soit $\Sigma = \{a, b, c\}$ et $L = \{a^n b^m c^{n+m} \mid n, m \in \mathbb{N}\}$.
Montrer que L est récursif.

Exercice 29

Soit $\Sigma = \{a, b\}$. Par $n|m$, on dénote que n divise m, pour $n, m \in \mathbb{N}$.
Soit $L = \{a^n b^m \mid n|m\}$. Montrer que L est récursif.

Exercice 30

Soit $\Sigma = \{a, b\}$.
Montrer que $L = \{a^{n^2} \mid n \in \mathbb{N}\}$ est récursif.

Feuille 5

Langages récursifs et langages récursivement énumérables - 1

Exercice 31

Soit $\Sigma = \{a, b\}$.
Montrer que les langages suivants sont récursifs:

1. $L = \{a^n b^{2^n} \mid n \in \mathbb{N}\}$.
2. $L' = \{a^n b^{\lceil \log_2(n) \rceil} \mid n \in \mathbb{N}\}$ où $\lceil m \rceil$ dénote le plus petit entier k tel que $k \geq m$.

Exercice 32

Montrer les assertions suivantes:

1. L'union de deux langages récursifs est un langage récursif.
2. L'union de deux langages récursivement énumérables est un langage récursivement énumérable.

Exercice 33

Un ensemble A est dit *dénombrable*, s'il existe une bijection de A vers $\mathbb{N}$. Intuitivement, ceci veut dire qu'on peut associer un entier comme numéro chaque élément de A.
Soit $\Sigma = \{1\}$. Le but de cet exercice est de démontrer que l'ensemble des langages sur Σ n'est pas dénombrable. Soit $\mathcal{L}(\Sigma)$ l'ensemble des langages sur Σ.
On veut démontrer que $\mathcal{L}(\Sigma)$ n'est pas dénombrable par l'absurde. On suppose donc qu'il existe une fonction bijective $f : \mathcal{L}(\Sigma) \to \mathbb{N}$.
On considre la fonction $g : \mathbb{N} \to \Sigma^*$ telle que $g(n) = 1^n$.

1. **Démontrer** que $g \circ f : \mathcal{L}(\Sigma) \to \Sigma^*$ est une bijection.
2. Considérer l'ensemble
   $$L = \{u \in \Sigma^* \mid u \notin f^{-1}(g^{-1}(u))\}.$$
   Alors, il existe $u_0 \in \Sigma^*$ tel que $L = f^{-1}(g^{-1}(u_0))$. **Pourquoi?**.
3. Considérer la véracité de $u_0 \in L$ pour **déduire une contradiction**. Et en **conclure**, que $\mathcal{L}(\Sigma)$ n'est pas dénombrable.

Exercice 34

- Soit A un ensemble dénombrable. Démontrer que tout $B \subseteq A$ est dénombrable ou fini.

- Soit $f : A \to B$ une fonction bijective. Démontrer que A est dénombrable, si B est dénombrable.

Exercice 35

On définit la fonction $\mathcal{C} : \mathbb{N}^* \to \mathbb{N}$ inductivement:

- Base: $\mathcal{C}(\epsilon) = 1$
- Pas d'induction: $\mathcal{C}(u \cdot n) = \mathcal{C}(u) * p_m^{n+1}$ o m est la longueur de $u \cdot n$ et p_m est le m-me nombre premier.
- On admet l'assertion suivante:

  Soit p un nombre premier et $n, m \in \mathbb{N}$. Si p divise $n * m$, noté $p|n * m$, alors $p|n$ ou $p|m$.

  Montrer que $\mathcal{C}$ est injective.
- Soit $\mathcal{C}' : \mathbb{N}^* \to \mathcal{C}(\mathbb{N}^*)$, o $\mathcal{C}(\mathbb{N}^*) = \{n \in \mathbb{N} \mid \exists u \in \mathbb{N}^* \cdot \mathcal{C}(u) = n\}$. Démontrer que $\mathcal{C}'$ est bijective.
- En déduire que $\mathbb{N}^*$ est dénombrable.
- Soit Σ un alphabet. Démontrer que Σ^* est dénombrable.

En déduire :

- Les langages régulier sur Σ est dénombrable.
- L'ensemble des langages hors-contexte sur Σ est dénombrable.
- L'ensemble des langages récursifs sur Σ est dénombrable.
- L'ensemble des langages sur Σ récursivement énumérables est dénombrable.
- Est-il vrai que tout langage sur $\Sigma = \{1\}$ est récursivement énumérable? Pourquoi?

Feuille 6

Le codage unaire d'un entier naturel $n \in \mathbb{N}$ est le môt $|^n$. À une fonction $f : \mathbb{N}^n \to \mathbb{N}$ on associe une function $\hat{f} : \{|, \#\}^* \to \{|\}$ telle que $\hat{f}(|^{x_1}\# \cdots \#|^{x_n}) = |^{f(x_1,\cdots,x_n)}$. On dit que f est calculable, si $\hat{f}$ est calculable.

Exercice 36

Montrer que l'addition est calculable.

Exercice 37

Montrer que la division euclidèenne est calculable.

Exercice 38

Montrer que la fonction $x \mapsto 2^x$ est calculable.

Exercice 39

Montrer que si $f : \mathbb{N}^n \to \mathbb{N}$ et $g : \mathbb{N}^2 \to \mathbb{N}$ sont calculables alors $h : \mathbb{N}^{n+1} \to \mathbb{N}$ avec $h(x_1, \cdots, x_{n+1}) = g(f(x_1, \cdots, x_n), x_{n+1})$ est calculable.

Feuille 7

Exercice 40

Par la suite et comme en cours $\langle M \rangle$ représente le mot qui code la machine de Turing M.

Soient:

1. $L_\epsilon = \{\langle M \rangle \mid \epsilon \in L(M)\}$.
2. $L_u = \{\langle M \rangle \mid u \in L(M)\}$.
3. $L_{ne} = \{\langle M \rangle \mid L(M) \neq \emptyset\}$.
4. $L_e = \{\langle M \rangle \mid L(M) = \emptyset\}$.
5. $L_k = \{\langle M \rangle \mid |L(M)| \geq k\}$.
6. $L_{\Sigma^*} = \{\langle M \rangle \mid L(M) = \Sigma^*\}$.
7. $L_f = \{\langle M \rangle \mid L(M)$ **est fini**$\}$.
8. $L_{reg} = \{\langle M \rangle \mid L(M)$ **est régulier**$\}$.
9. $L_{hc} = \{\langle M \rangle \mid L(M)$ **est hors contexte**$\}$.
10. $L_r = \{\langle M \rangle \mid L(M) \in R\}$.
11. $L_{nr} = \{\langle M \rangle \mid L(M) \notin R\}$.

Montrer les assertions suivantes:

1. Aucun des langages ci-dessus n'est récursif.
2. $L_\epsilon \in RE$.
3. $L_u \in RE$.
4. $L_{ne} \in RE$.
5. $L_e \notin RE$.
6. $L_k \in RE$.
7. $L_r \notin RE$.
8. $L_{nr} \notin RE$.