## 3.3 Arithmétique de Presburger

[Référence : Olivier Carton, Langages formels, calculabilité et complexité.]

raisonner automatiquement sur des expressions mathématiques sur les entiers naturels avec addition mais sans multiplication (les équations sont linéaires)

À toute formule logique, on peut associer l'ensemble des tuples que les variables libres peuvent prendre pour rendre la formule vraie. On associe un *automate* qui reconnaît cet ensemble.
Autre domaine où on associe un automate à une formule : logique modale temporelle

**Example 61** $\forall y \exists x, 2x + 3y = 3$

**Example 62** *On exprime* $x \leq y$ *avec* $\exists z, y = x + z$.

On s'intéresse aux formules logiques générés par la grammaire suivante :

$$\varphi ::= x = 0 \mid x = 1 \mid z = x + y \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \exists x \varphi$$

où $x, y, z$ sont des symboles de variable.

### 3.3.1 Expressivité

- On réécrit un quantificateur universel comme ceci : $\forall x \varphi$ se réécrit en $\neg\exists\neg\varphi$ ;
- On peut écrire 0 en le remplaçant par $x_0$ et par une variable fraîche $x_0$ et en rajoutant l'équation :

  $x_0 = 0$

  On peut des fois tout simplement le supprimer. Par exemple, on remplace $x + 0$ par $x$.
- On peut écrire tout nombre entier $n \geq 1$ en remplaçant $n$ par une variable fraîche $x_n$ et en rajoutant les équations suivantes :

  $x_n = x_{n-1} + x_1$

  $x_{n-1} = x_{n-2} + x_1$

  $\vdots$

  $x_2 = x_1 + x_2$

  $x_1 = 1$
- On remplace $4x$ par $x + x + x + x$.

- Ainsi une équation ressemble à $y_1 + y_2 + \dots y_k = z_1 + \dots z_\ell$. On regroupe les termes, par exemple, on remplace $y_1 + y_2$ par $z$ et on ajoute

  $z = y_1 + y_2$.

- Une inéquations $y_1 + y_2 + \dots y_k \leq z_1 + \dots z_\ell$ se réécrit comme une équation en regroupant les termes. On arrive à la fin à

  $x \leq y$.

  Là, c'est équivalent à $\exists zy = x + z$.

Au final, nous n'avons que des formules atomiques de la forme $x = 0$, $x = 1$ ou $z = x + y$.

**Example 63** $x + x + y = 1 + 1 + 1$
*se réécrit comme la conjonction de* $x+t = z$, $t = x$, $z+y = u$, $u = v$, $a = 1$, $b = 1$, $c = 1$, $v = n + c$, $n = a + b$.
*On construit donc l'automate de* $x + x + y = 1 + 1 + 1$ *comme intersection des automates de* $x + t = z$, $t = x$, $z + y = u$, $u = v$, $a = 1$, $b = 1$, $c = 1$, $v = n + c$, $n = a + b$.

### 3.3.2 Représentation des mots

On représente les entiers comme des mots sur $\{0, 1\}$ avec le poids faible à gauche. C'est à dire que les automates que l'on va construire vont lire les nombres en commençant par lire les chiffres des unités puis finissent par lire les chiffres de poids forts.

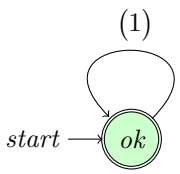
**Example 64** *Par exemple, les mots 001001, 0010010, 0010010000 représentent le nombre* $2^2 + 2^5 = 36$.

On représente un $n$-uplet d'entiers comme un mot sur l'alphabet $\Sigma^{(n)} = \{0, 1\}^n$.

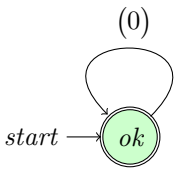
**Example 65** *On représente* $(5, 2, 3)$ *par le mot de trois lettres suivants :*
$\begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}0\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}$.

### 3.3.3 Automates des formules atomiques

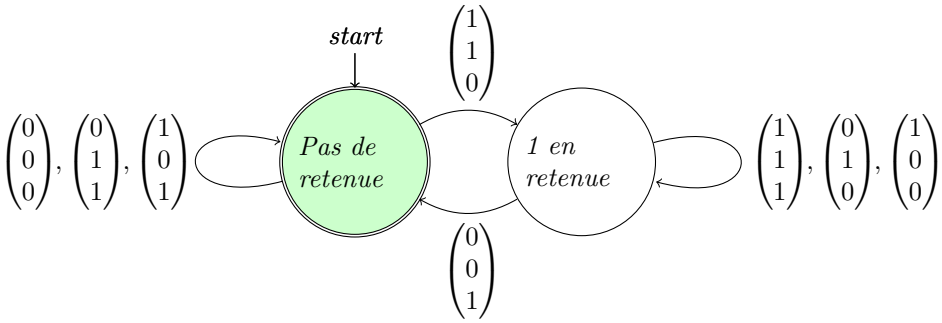
**Example 66** *Voici un automate qui vérifie que* $x_1 = 1$ *:*



**Example 67** *Voici un automate qui vérifie que* $x_1 = 0$ *:*




**Example 68** *Voici un automate qui vérifie que* $x_1 + x_2 = x_3$ *:*



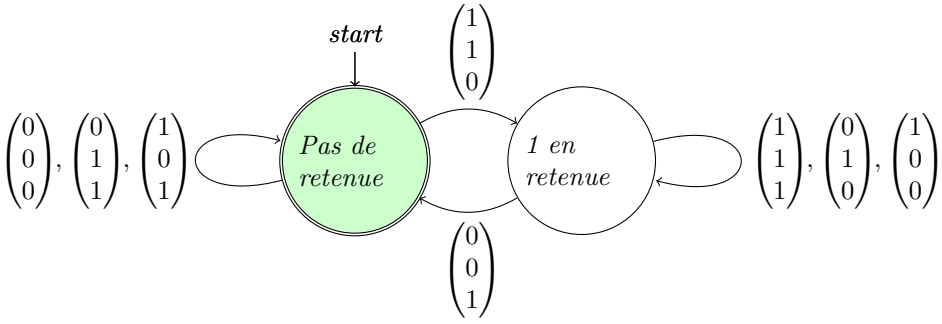

### 3.3.4 Automates des constructions booléennes

| | |
|---|---|
| $\psi_1 \vee \psi_2$ | union des automates pour $\psi_1$ et $\psi_2$ |
| $\neg\psi$ | complémentaire de l'automate pour $\psi$ |

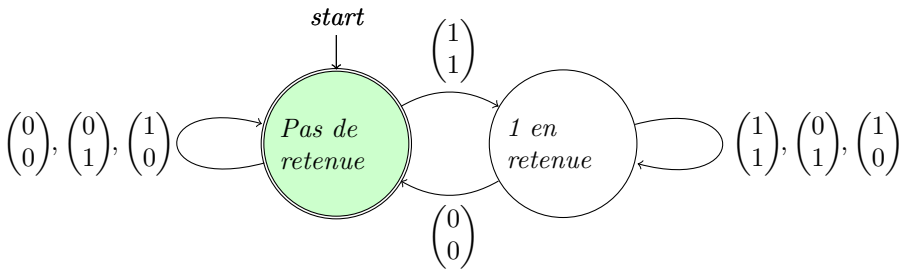
### 3.3.5 Quantificateur existentiel

L'automate correspondant à $\exists x_n \psi$ est l'automate de $\psi$ dans lequel on a effacé la composante qui concerne $x_n$ dans les transitions. Si l'automate de $\psi$, l'automate ainsi calculé $\exists x_n \psi$ est peut-être non-déterministe.

**Example 69** *Par exemple, si on prend l'automate de* $x_1 + x_2 = x_3$ *:*

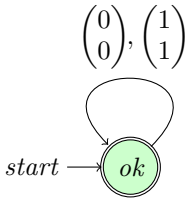


*L'automate pour* $\exists x_3, x_1 + x_2 = x_3$ *est :*

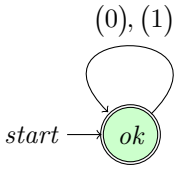



**Example 70** *Construisons l'automate pour* $\forall x \exists y, x = y$*, c'est à dire* $\neg\exists x \neg\exists y, x = y$*.*

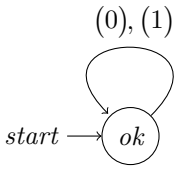
*Automate de* $x = y$*:*



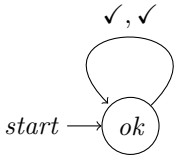

*Automate de* $\exists y, x = y$*:*



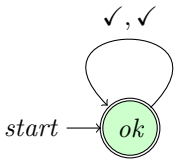

*Automate de* $\neg\exists y, x = y$*:*




*Automate de* $\exists x \neg\exists y, x = y$*:*




*Automate de* $\neg\exists x \neg\exists y, x = y$*:*




### 3.3.6 Complexité

Malheureusement, il nous faut des automates déterministes pour pouvoir passer au complémentaire (négation). Mais, lorsqu'on construire l'automate de $\exists x\psi$ à partir d'un automate $\psi$, on a potentiellement un automate non-déterministe (projection). Toutes les constructions donnent des automates déterministes sauf la projection. Donc à chaque fois que l'on a un quantificateur, on peut potentiellement exploser.

On a donc un algorithme non élémentaire pour tester si une formule close est vraie dans l'arithmétique de Presburger.