Décidabilité de l'arithmétique de Presburger

Ref Librement inspiré du Carton p (!!)

Leçons 914, 917, 924 NB : pour 909 faire une autre version.

Déf Le langage de l'arithmétique de Presburger est le langage $\mathcal{L}$ constitué de → 0 symbole de constante, → s ——— fonction unaire, → $+$ ——— binaire, → $=$ ——— relation binaire. [DNR] P136.

Déf On appellera ici formule atomique simple sur $\mathcal{L}$ une formule de la forme $x_i = 0$ ou $x_i = x_j$ ou $s(x_i) = x_j$ ou $x_i + x_j = x_k$ avec i, j, k 2à2 $\neq$.

Lemme Pour toute formule Ψ combinaison booléenne de formules atomiques simples sur $\mathcal{L}$, on peut construire A_Ψ un automate qui reconnait exactement les codages des entiers naturels satisfaisant Ψ pour un codage bien choisi.

ADMIS ici (À developper dans la 909) Cf dernier encadré.

Pté PRES = { entrée : Ψ une $\mathcal{L}$ formule close ; sortie : oui si $\mathbb{N} \models \Psi$, non sinon } est décidable

Pour démontrer cette propriété il nous faudra dans un premier temps donner un codage des entiers, ou plutôt des k-uplets d'entiers, afin d'utiliser le lemme.

Une fois ce codage expliqué on montrera comment ramener notre problème au problème de la vacuité d'un langage, un langage de nos codages même.

Enfin il restera à montrer qu'on peut construire un automate reconnaissant ce langage, puisqu'on saura alors décider sa vacuité : il suffira de résoudre un pb d'accessibilité dans le graphe qu'est l'automate.

1) Codage des k-uplets d'entiers

2) Se ramener à un problème de vacuité d'un langage

3) Construire un automate reconnaissant ce langage.

1) Codage des k-uplets d'entiers

Soit $(n_i)_{i\in[\![1..k]\!]} \in \mathbb{N}^k$

Pour $i\in[\![1..k]\!]$, on peut écrire n_i en binaire $n_i : c_i^{r_i} \dots c_i^2 c_i^1$

(r_i : on a écrit n_i en r_i chiffres ici ; c_i^1 : bit de poids faible)

Quitte à poser $r = \max_{i\in[\![1..k]\!]} r_i$ on a

$$n_1 = c_1^r \dots c_1^1$$
$$n_2 = c_2^r \dots c_2^1$$
$$\vdots$$
$$n_k = c_k^r \dots c_k^1$$

On pose alors :

$$\langle (n_i)_{i\in[\![1..k]\!]} \rangle_k^r = \begin{pmatrix} c_1^r \\ \vdots \\ c_k^r \end{pmatrix} \cdots \underbrace{\begin{pmatrix} c_2^2 \\ \vdots \\ c_k^2 \end{pmatrix} \begin{pmatrix} c_1^1 \\ c_k^1 \end{pmatrix}}_{\in \Sigma_k := \{0,1\}^k} \in \Sigma_k^*$$

($\langle\cdot\rangle_k^r$: codage de k-uplets sur r chiffres)

ex $r=3$ $\quad 0\to 000 \quad 1\to 001 \quad \langle 10,1\rangle_2^3 = \begin{smallmatrix} 0&0&0 \\ 0&0&1 \end{smallmatrix}$

Il nous faut aussi un "décodage" : pour $\omega\in\Sigma_k^*$ on pose

$$[\![\omega]\!]_k = \rangle \omega \langle_k^r \quad \text{où } r = |\omega|$$

Ici pas besoin de préciser quel "décodage" on s'applique, celui ci est évident, ou plutôt donné par la longueur du mot ω

ex $[\![\begin{smallmatrix} 0&0&0&0 \\ 0&0&1&1 \end{smallmatrix}]\!]_2 = (0,1)$

$[\![\begin{smallmatrix} 0&0 \\ 0&1 \end{smallmatrix}]\!]_2 = (0,1)$

⚠ Il faut néanmoins préciser le "k".

ex $[\![\varepsilon]\!]_1 = 0$ mais $[\![\varepsilon]\!]_3 = (0,0,0)$.

Remarquons qu'on a par construction

$$[\![\left(\begin{smallmatrix} \omega \\ c \end{smallmatrix}\right)]\!]_{k+1} = (\underbrace{n_1, \dots n_k}_{[\![\omega]\!]_k}, \underbrace{n_{k+1}}_{[\![c]\!]_1}) \quad \text{pour } \begin{matrix} c\in\Sigma_1^* \\ \omega\in\Sigma_k^* \end{matrix}$$

(notation pour $\left(\begin{smallmatrix} \omega_1 \\ c_1 \end{smallmatrix}\right)\left(\begin{smallmatrix} \omega_2 \\ c_2 \end{smallmatrix}\right)\cdots\underbrace{\left(\begin{smallmatrix} \omega_n \\ c_n \end{smallmatrix}\right)}_{\in\Sigma_{k+1}} \in \Sigma_{k+1}^*$)

2) Se ramener à un problème de vacuité

Soit φ une $\mathcal{L}$-formule close.
On se ramène à $\tilde\varphi \equiv \varphi$ sous forme prénexe.
Et m'quitte à ajouter des variables on se ramène à $\hat\varphi \equiv \varphi$, sous forme prénexe et dont les formules atomiques sont simples.

ex $\varphi \quad \exists x_1 \big((\forall x_2\ x_1 + x_2 = s(x_2))\big) \wedge (\exists x_3,\ s(x_3) = x_1)$

$\tilde\varphi = \exists x_1, \forall x_2, \exists x_3, (x_1 + x_2 = s(x_2)) \wedge (s(x_3) = x_1)$

$\hat\varphi = \exists x_1, \forall x_2, \exists x_3, \exists x_4, (x_1 + x_2 = x_4) \wedge (s(x_2) = x_4) \wedge (s(x_3) = x_1)$

On écrit $\hat\varphi : Q_1 x_1 \dots Q_k x_k \underbrace{Q_{k+1} x_{k+1} \dots Q_m x_m\ \psi}_{:=\psi_k}$, où ψ sans quantif.

Ainsi $\psi_0 = \hat\varphi$, $\psi_m = \psi$ et $\forall k$, $\mathrm{Var.lib}(\psi_k) = \{x_i \mid i\in[\![1..k]\!]\}$

On note aussi $\mathcal{N}_k = \{(n_i)_{i\in[\![1..k]\!]} \mid \mathbb{N}[^{x_i := n_i}_{i\in[\![1..k]\!]}] \models \hat\varphi\}$

$\mathcal{L}_k = \{\omega\in\Sigma_k^* \mid [\![\omega]\!]_k = (n_1 - n_k) \in \mathcal{N}_k\}$

$\psi : \hat\varphi : \exists x_1 \psi_1$, alors $\mathbb{N}\models\hat\varphi$ ssi $\mathcal{N}_1 \neq \emptyset$ ssi $\mathcal{L}_1 \neq \emptyset$

$\psi : \hat\varphi : \forall x_1 \psi_1$ —————— $\mathcal{N}_1 = \mathbb{N}$ ssi $\mathcal{L}_1 = \Sigma_1^*$ ssi $\mathcal{L}_1^c = \emptyset$.

↳ On est ramené au problème de vacuité de $\mathcal{L}_1$.

3) Montrer par récurrence que $\mathcal{L}_1$ est reconnaissable

On montre par réc. finie, pour k allant de m à 1 la propriété H_k [∃ il existe un automate $\mathcal{A}_k$ tq $\mathcal{L}(\mathcal{A}_k) = \mathcal{L}_k$]

- Puisque les formules atomiques de $\hat\varphi$ et donc de ψ sont simples, le lemme assure H_m ($\psi_m = \psi$)
- Soit $k>1$ tel que H_k. On note $\mathcal{A}_k = (\Sigma_k, Q, q_0, F, T)$.
  On suppose que $\psi_{k-1} : \exists x_k \psi_k$.

  alors $\mathcal{N}_{k-1} = \{(n_1 - n_{k-1}) \mid \exists n_k\in\mathbb{N}, (n_1 - n_k)\in\mathcal{N}_k\}$

  donc $\mathcal{L}_{k-1} = \{\omega\in\Sigma_{k-1}^* \mid \exists n_k\in\mathbb{N},\ ([\![\omega]\!]_{k-1}, n_k)\in\mathcal{N}_k\}$

  $= \{ \text{——} \mid \exists c\in\Sigma_1^+ ([\![\omega]\!]_{k-1}, [\![c]\!]_1)\in\mathcal{N}_k\}$

  $= \{ \text{——} \mid \exists c\in\Sigma_1^* \quad [\![\begin{smallmatrix} 0 & \cdots & 0 & \omega \\ c & & & \end{smallmatrix}]\!]_k \in \mathcal{N}_k\}$ ⚠

  $= \{ \text{——} \mid \exists w\in\Sigma_1^+ \quad (\begin{smallmatrix} 0 & \cdots & 0 & \omega \\ w & & & \end{smallmatrix}) \in \mathcal{L}_k\}$

  $= \{\omega\in\Sigma_{k-1}^* \mid \exists w\in\Sigma_1^*, (\begin{smallmatrix} 0 & \cdots & 0 & \omega \\ w & & & \end{smallmatrix})\in\mathcal{L}(\mathcal{A}_k)\}$ par H_k

  On introduit alors $\rho = \left(\begin{smallmatrix} \Sigma_k & \longrightarrow & \Sigma_{k-1} \\ \left(\begin{smallmatrix} c_1 \\ \vdots \\ c_k \end{smallmatrix}\right) & \longmapsto & \left(\begin{smallmatrix} c_1 \\ \vdots \\ c_{k-1} \end{smallmatrix}\right) \end{smallmatrix}\right)$

  et on pose $\pi(\mathcal{A}_k) = (\Sigma_{k-1}, Q, q_0, F^\pi, T^\pi)$

  où $F^\pi = \{q \mid q \xrightarrow[\mathcal{A}_k]{(\rho^{-1}(\$))^+} q_f\in F\}$

  (i.e. l'ens. des états à partir desquels on pouvait rejoindre l'état final en ne lisant que des 0 sur les $k-1$ premières composantes.)

  $$T^\pi = \{q \xrightarrow{n(a)} q' \mid (q \xrightarrow{a} q')\in T\}$$

Si $w \in \mathcal{L}(\pi(\mathcal{A}_k))$
alors il existe $q_f' \in F^\pi$ tq $q_0 \xrightarrow[\pi(\mathcal{A}_k)]{w} q_f'$.

Par déf de T^π il existe $u \in \Sigma_1^*$ tq $q_0 \xrightarrow[\mathcal{A}_k]{\left[\begin{smallmatrix} w \\ u \end{smallmatrix}\right]} q_f'$

Par déf de F^π ——— $u \in \Sigma_1^*$ tq $q_f' \xrightarrow[\mathcal{A}_k]{\left[\begin{smallmatrix} 0 \\ u \end{smallmatrix}\right]} q_f$
pour un certain $q_f \in F$.

Ainsi $q_0 \xrightarrow[\mathcal{A}_k]{\frac{0/w}{u/v}} q_f \in F$.

Donc $\frac{0/w}{u/v} \in \mathcal{L}(\mathcal{A}_k)$, d'où $\underline{w \in \mathcal{L}_{k-1}}$.

Réc si $w \in \mathcal{L}_{k-1}$.
Il existe $u, v \in \Sigma_1^*$ tel que $\frac{0/w}{u/v} \in \mathcal{L}(\mathcal{A}_k)$
Donc il existe $q_f' \in Q$, et $q_f \in F$ tq
$q_0 \xrightarrow[\mathcal{A}_k]{\left[\begin{smallmatrix} w \\ v \end{smallmatrix}\right]} q_f' \xrightarrow[\mathcal{A}_k]{\left[\begin{smallmatrix} 0 \\ u \end{smallmatrix}\right]} q_f$

Par déf de T^π, $q_0 \xrightarrow[\pi(\mathcal{A}_k)]{w} q_f'$
Par déf de F^π, $q_f' \in F^\pi$
Donc $\underline{w \in \mathcal{L}(\pi(\mathcal{A}_k))}$.

On pose alors $\mathcal{A}_{k-1} = \pi(\mathcal{A}_k)$
ainsi $\mathcal{L}(\mathcal{A}_{k-1}) = \mathcal{L}_{k-1}$.

D'où $\underline{H_{k-1}}$.

Si $\underline{\psi_{k-1} : \forall x_k \, \psi_k}$, $\psi_{k-1} \equiv \neg(\exists x_k \neg\psi_k)$.
Il suffit, puisqu'on sait complémenté, de refaire ce qu'on a fait avant.

On pose ici $\mathcal{A}_{k-1} = \overline{\pi(\overline{\mathcal{A}_k})}$.

$\mathcal{N}_{k-1}^c = \{ m_1 \dots m_{k-1} \mid \mathbb{N}\left[\begin{smallmatrix} x_i := m_i \\ i \in [\![1, k-1]\!] \end{smallmatrix}\right] \models \exists x_k \neg\psi_k \}$
$= \{ \text{———} \mid \exists m_k \in \mathbb{N}, \ \mathbb{N}\left[\begin{smallmatrix} x_i := m_i \\ i \in [\![1, k]\!] \end{smallmatrix}\right] \models \neg\psi_k \}$
$= \{ m_1 \dots m_{k-1} \mid \exists m_k \in \mathbb{N}, \ m_1 \dots m_k \in \mathcal{N}_k^c \}$

Par HR $\mathcal{L}(\mathcal{A}_k) = \mathcal{N}_k$, donc $\mathcal{L}(\overline{\mathcal{A}_k}) = \mathcal{N}_k^c$
Comme prec, et d'après cette reformulation de $\mathcal{N}_{k-1}^c$
$\mathcal{L}(\pi(\overline{\mathcal{A}_k})) = \mathcal{N}_{k-1}^c$ (On reconnaît les $k-1$ uplets projetés des k-uplets reconnus par $\overline{\mathcal{A}_k}$. C'est ce que fait π)

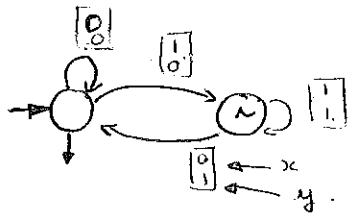
Donc $\mathcal{L}(\overline{\pi(\overline{\mathcal{A}_k})}) = \mathcal{N}_{k-1}$
soit $\mathcal{L}(\mathcal{A}_{k-1}) = \mathcal{N}_{k-1}$, d'où $\underline{H_{k-1}}$.

FIN DVP.

Pour comprendre pourquoi cette histoire d'états finaux est importante

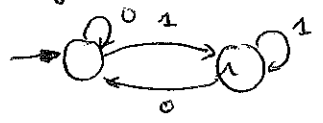
$\varphi \sim \forall x \, \exists y, \ y = x + x$

$\psi_2 \sim y = x + x$. $\mathcal{A}_2$:



$\psi_1 \sim \exists y, \ y = x + x$.

$\underline{\mathcal{A}_1}$

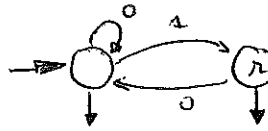


si ici on n'a pas bien déf. les états finals

Pb Cet automate ne reconnaît pas 1

Pourtant il devrait reconnaître 1, puisque pour $x=1$ il existe y tq $y = x+x$, avec $y=2$.
L'idée est que $\mathcal{A}_2$ reconnaissait "(1,2)" codé en $\left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, donc $\mathcal{A}_1$ ——— "1" codé en $\underline{01}$.

$\underline{\mathcal{A}_1}$, le bien défini

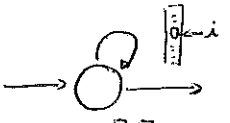


$\left(r \in F_1 \text{ car } \lambda \xrightarrow{\left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]} \bigcirc \in F_2 \right)$

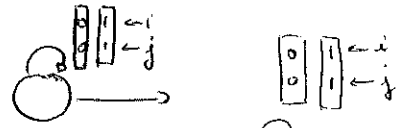
ici $\mathcal{L}(\mathcal{A}_1) = \{0,1\}^* = \Sigma_1^* = \mathcal{L}_1$ puisque $\mathcal{N}_1 = \mathbb{N}$
Donc $\mathbb{N} \models \varphi$, en effet tout entier a un double écrit

Rappels des automates de base pour le lemme

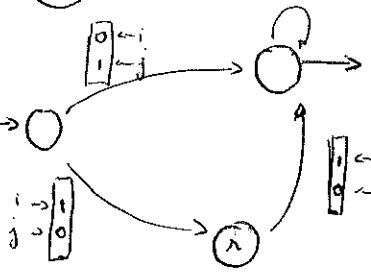
Si $\psi : x_i = 0$ $\mathcal{A}_\psi$:



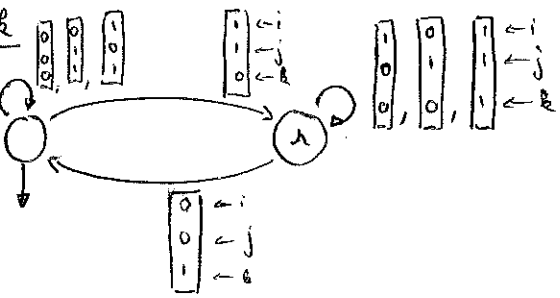
Si $\psi : x_i = x_j$ $\mathcal{A}_\psi$:



Si $\psi : s(x_i) = x_j$ $\mathcal{A}_\psi$



Si $\psi : x_i + x_j = x_k$ $\mathcal{A}_\psi$



Si $\psi = \neg\psi_a$ $\mathcal{A}_\psi = \overline{\det(\mathcal{A}_{\psi_a})}$
Si $\psi = \psi_a \vee \psi_b$ $\mathcal{A}_\psi = \mathcal{A}_{\psi_a} \cup \mathcal{A}_{\psi_b}$
Si $\psi = \psi_a \wedge \psi_b$ $\mathcal{A}_\psi = \mathcal{A}_{\psi_a} \times \mathcal{A}_{\psi_b}$