Décidabilité de l'arithmétique de Presburger

Leçons : 909 ; 914 ; 924

Références : CARTON, *Langages formels, calculabilité et complexité* (p.178) [?]

Introduction :

On va prouver que la théorie de Presburger est décidable, c'est à dire que, pour toute formule close (écrite sur la signature[1] comprenant les constantes 0 et 1, la fonction *successeur* et le prédicat d'égalité), on peut dire si $\mathbb{N}$ est un modèle de cette formule ou non.

Théorème 1. Le problème

$PRES$ $\begin{cases} \text{entrée : une formule } \varphi \text{ close sur la signature } \sigma = \{0, 1, +, =\} \\ \text{sortie : oui si } \mathbb{N} \vDash \varphi, \text{ non sinon} \end{cases}$

est décidable.

Démonstration.

Étape 1 : *On veut construire un automate $\mathcal{A}$ telle que $\mathbb{N} \vDash \varphi$ si et seulement si $L(\mathcal{A}) \neq \emptyset$*

Pour tout $k \in \mathbb{N}^*$, on va se donner un alphabet pour représenter les $k-$uplet $(n_1, \dots, n_k) \in \mathbb{N}^k$. On pose $\Sigma_k = \{0, 1\}^k$. On représente les entiers en binaire avec le bit de poids faible à gauche. On ajoute des 0 à droite pour que l'on ait des représentations qui font toutes la même taille.

$$\begin{array}{ccccc} n_1 & = & \alpha_1^1 & \dots & \alpha_1^r \\ n_2 & = & \alpha_2^1 & \dots & \alpha_2^r \\ \vdots & \vdots & \vdots & & \vdots \\ n_k & = & \alpha_k^1 & \dots & \alpha_k^r \end{array}$$

Chaque colonne appartient à Σ_k, donc on peut représenter un $k-$uplet $(n_1, \dots, n_k)$ par un mot w sur l'alphabet Σ_k.

Soit φ une formule sur la signature σ.

On commence par remplacer tous les $\forall x \psi(x)$ par la formule $\neg(\exists x \neg\psi(x))$ qui est sémantiquement équivalente, cela a pour avantage d'enlever tous les quantificateurs $\forall$ de φ.

On va construire pour toute sous-formule ψ de φ un automate $\mathcal{A}_\psi$ tel que

$$\begin{aligned} L(\mathcal{A}_\psi) = \{&(n_1, \dots, n_k) \in \mathbb{N}^k, \mathbb{N}[x_1 = n_1, \dots, x_k = n_k] \vDash \psi \\ &\text{où } x_1, \dots, x_k \text{ sont les variables libres de } \psi\} \end{aligned}$$

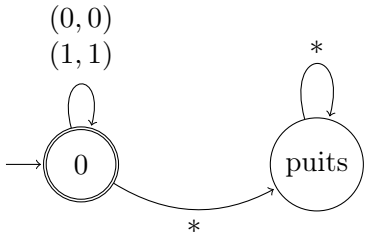
Procédons par induction sur la structure de la formule φ :

— cas où ψ est une formule atomique, deux cas se présentent :[2]

  — $\psi := x_i = x_j$

1. on parle parfois de langage qui comprend des fonctions qui ont chacune une arité positive (on parle de constantes pour les fonctions d'arité nulle) et des relations qui ont chacune une arité strictement positive.
2. on peut traiter les cas $x_i = 0$, $x_i = 1$ de manière similaire.

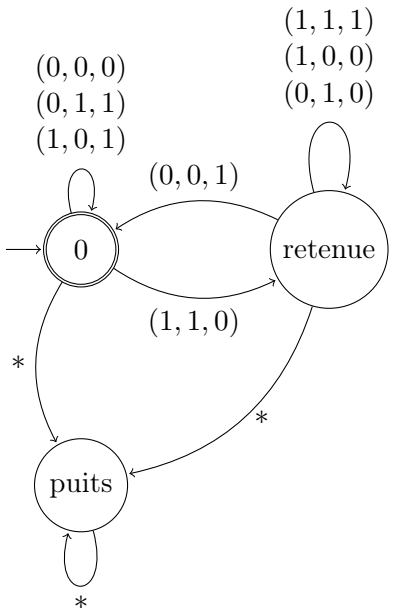
On pose $\mathcal{A}_\psi :=$



En notant $(0,0)$ et $(1,1)$ sur les transitions, on ne donne que les indices i et j des étiquettes, mais ce sont en réalité des étiquettes sur l'alphabet Σ_k. De plus, les étoiles $*$ désignent toutes les autres étiquettes possibles sur Σ_k.

— $\psi := x_i + x_j = x_k$

On pose $\mathcal{A}_\psi :=$



— cas $\psi := \psi_1 \vee \psi_2$, on dispose des automates $\mathcal{A}_{\psi_1}$ et $\mathcal{A}_{\psi_2}$, on construit alors l'automate $\mathcal{A}_\psi$ comme l'automate de l'union des automates $\mathcal{A}_{\psi_1}$ et$\mathcal{A}_{\psi_2}$.

— cas $\psi := \psi_1 \wedge \psi_2$, similaire en prenant l'automate de l'intersection.

— cas $\psi := \neg\psi_1$, similaire en prenant l'automate du complémentaire [3].

— cas $\psi := \exists x \psi_1(x)$ si $x \notin FV(\psi)$, on peut prendre $\mathcal{A}_\psi := \mathcal{A}_{\psi_1}$.

— cas $\psi := \exists x \psi_1(x)$ si $x \in FV(\psi)$, on note $\mathcal{A}_{\psi_1} = (Q, I, F, \delta)$. On pose l'automate

$$\mathcal{A}_\psi := (Q, I, F, \delta')$$

où δ' est telle que

$$p \xrightarrow{(x_1,\dots,x_{k-1})} q \text{ dans } \mathcal{A}_\psi$$

si et seulement si

$$\text{il existe } x \text{ tel que } p \xrightarrow{(x_1,\dots,x_{k-1},x)} q \text{ dans } \mathcal{A}_{\psi_1}$$

3. attention, il faut compléter et déterminiser l'automate, ce qui est coûteux

On obtient à la fin un automate $\mathcal{A}$ sur le langage vide (puisque φ est close) qui est tel que

$\mathbb{N} \vDash \varphi$ si et seulement un état final de $\mathcal{A}$ est accessible en partant de l'état initial.

Comme un parcours d'un graphe[4] est un algorithme d'accessibilité dans un graphe, la théorie de Presburger est donc bien une théorie décidable.

Étape 2 : *Calcul de la complexité*

Lors de la construction de l'automate, l'opération la plus couteuse est la création de l'automate du complémentaire car il faut compléter et déterminiser l'automate précédent. Pour déterminiser et compléter un automate, on passe d'un automate à n états vers un automate à 2^n états. Ainsi si la formule φ contient k symboles $\neg$ et m symboles $\forall$, on va devoir déterminiser $k+2m$ fois et donc cela coutera $2^{2^{\cdot^{\cdot^{\cdot^{2^n}}}}}$ où on itère l'exponentielle $k+2m$ fois et là, on ne compte même pas l'expansion de l'automate dûe aux autres opérateurs (union, intersection). Donc l'algorithme que l'on décrit ici est dans $(k+2m)$ExpTime. On a donc un problème décidable mais avec une complexité non élémentaire, qui n'est pas calculable par une machine efficacement. □

Remarques :

- L'arithmétique de Presburger est cohérente et complète.
- L'arithmétique de Presburger est décidable, mais si on ajoute le symbole $\times$, on obtient l'arithmétique de Peano qui est indécidable.
- Le problème de décidabilité de Presburger est en réalité dans 3ExpTime et est 2ExpTime-dur.

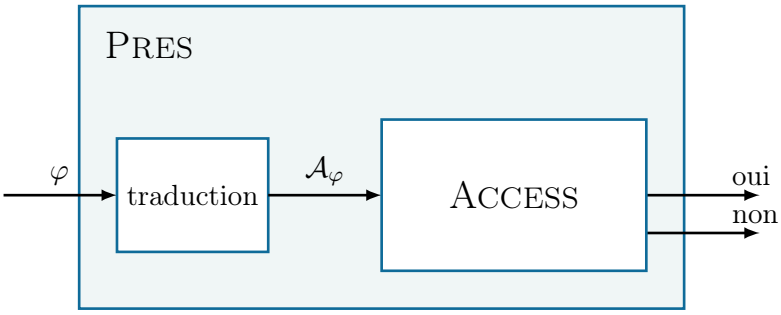
Astuces de l'agrégatif :

Il y a plein de rédaction de ce développement, je mets en page suivante un exemple de rédaction qu'il ne faut pas faire, au début je l'avais rédigé comme ci-après mais après un passage en leçon avec certains encadrants, ils nous avaient mis en garde contre cette rédaction car ce n'était vraiment pas propre.

Remarques :

On peut voir ce résultat comme une réduction de Pres à Access où

Access $\begin{cases} \underline{\text{entrée :}} \text{ un graphe } G=(S,A) \text{ et deux sommets } u,v \in S \\ \underline{\text{sortie :}} \text{ oui si il existe un chemin de } u \text{ à } v \text{ dans le graphe } G, \text{ non sinon} \end{cases}$



Comme Access est décidable, alors Pres est bien décidable.

4. en largeur ou en profondeur par exemple

Décidabilité de l'arithmétique de Presburger (Version à ne pas faire)

Leçons : 909 ; 914 ; 924

Références : CARTON, *Langages formels, calculabilité et complexité* (p.178)

Introduction :

Le développement qui suit est un exemple de ce qu'il ne faut pas faire, il est tiré du *Carton* qui est un mauvaise référence pour ce développement. Je l'avais tapé en pensant qu'il était bien mais en fait il y a plein de choses bizarres dedans. Ainsi j'ai tapé une meilleure version (à mon sens) de ce développement, cela peut donc faire la différence à l'agreg avec des candidats qui auraient "bêtement" recopié le *Carton*. Les problèmes les plus flagrants sont : la mise sous forme prénexe inutile (en effet les formules en elle-même sont définies par induction, pourquoi séparer les cas inductifs ?), le fait de s'arrêter à 1 et non à 0 (cela rajoute une disjonction de cas inutile) et on ne parle pas de complexité (alors que c'est primordial ici puisque décidable est équivalent à avoir un algorithme sauf que celui-ci est de complexité très médiocre il est dans kExpTime où k est le nombre de négations (après avoir fait le changement des $\forall$ en $\exists$)).

Enfin bref tout cela pour dire qu'il faut faire attention à ce développement qui est sournois.

Prérequis :

- la décidabilité de Vide pour un langage rationnel [1]
- la décidabilité de Univ pour un langage rationnel [2]
- Théorème de Kleene

Théorème 1. Le problème Pres $\begin{cases} \text{entrée :} & \text{une formule } \varphi \text{ close sur le langage } L = \{0, 1, +, =\} \\ \text{sortie :} & \text{oui si } \mathbb{N} \vDash \varphi, \text{ non sinon} \end{cases}$
est décidable.

Démonstration.

Étape 1 : *Se ramener à un problème de vacuité ou d'universalité d'un langage*

Pour tout $k \in \mathbb{N}^*$, on va se donner un alphabet pour représenter les $k-$uplet $(n_1, \dots, n_k) \in \mathbb{N}^k$. On pose $\Sigma_k = \{0, 1\}^k$. On représente les entiers en binaire avec le bit de poids faible à gauche. On ajoute des 0 à droite pour que l'on ait des représentations qui font toutes la même taille.

$$\begin{array}{ccccc} n_1 & = & \alpha_1^1 & \dots & \alpha_1^r \\ n_2 & = & \alpha_2^1 & \dots & \alpha_2^r \\ \vdots & \vdots & \vdots & & \vdots \\ n_k & = & \alpha_k^1 & \dots & \alpha_k^r \end{array}$$

Chaque colonne appartient à Σ_k, donc on peut représenter un $k-$uplet $(n_1, \dots, n_k)$ par un mot w sur l'alphabet Σ_k. On notera $[\![w]\!] = (n_1, \dots, n_k)$ le décodage du mot.

1. Vide $\begin{cases} \text{entrée :} & \text{un langage rationnel } L \text{ sur un alphabet } \Sigma \\ \text{sortie :} & \text{oui si } L = \emptyset, \text{ non sinon} \end{cases}$
2. Univ $\begin{cases} \text{entrée :} & \text{un langage rationnel } L \text{ sur un alphabet } \Sigma \\ \text{sortie :} & \text{oui si } L = \Sigma^*, \text{ non sinon} \end{cases}$

Soit φ une formule sur le langage L, on peut la supposer sous forme prénexe[3].

$$\varphi = Q_1 x_1 \dots Q_n x_n \psi$$

où $Q_i \in \{\exists, \forall\}$ et ψ est une formule sans quantificateur.

On pose

$$\varphi_k = Q_{k+1} x_{k+1} \dots Q_n x_n \psi$$

D'où $\varphi_0 = \varphi$ et $\varphi_n = \psi$. Les variables $x_1, \dots, x_k$ sont libres dans φ_k.

On introduit, pour tout $k \in \mathbb{N}^*$,

$$\mathcal{N}_k = \{(n_1, \dots, n_k) \in \mathbb{N}^k, \mathbb{N}[x_1 = n_1, \dots, x_k = n_k] \vDash \varphi_k\}$$

et

$$\mathcal{L}_k = \{w \in \Sigma_k^*, [\![w]\!] \in \mathcal{N}_k\}$$

On remarque que

— Si $\varphi = \exists x_1 \varphi_1$, alors $\big(\mathbb{N} \vDash \varphi \iff \mathcal{N}_1 \neq \emptyset \iff \mathcal{L}_1 \neq \emptyset\big)$.

— Si $\varphi = \forall x_1 \varphi_1$, alors $\big(\mathbb{N} \vDash \varphi \iff \mathcal{N}_1 = \mathbb{N} \iff \mathcal{L}_1 = \Sigma_1^*\big)$.

Étape 2 : *Prouver que le langage $\mathcal{L}_1$ est rationnel*

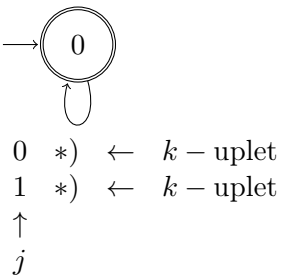
Montrons par récurrence sur k allant de n à 1 que

"il existe un automate $\mathcal{A}_{\varphi_k}$ tel que $\mathrm{L}(A_{\varphi_k}) = \mathcal{L}_k$"

Initialisation : On veut montrer qu'il existe A_ψ tel que $\mathrm{L}(A_\psi) = \mathcal{L}_n$. On va le montrer par induction structurelle sur la forme de ψ

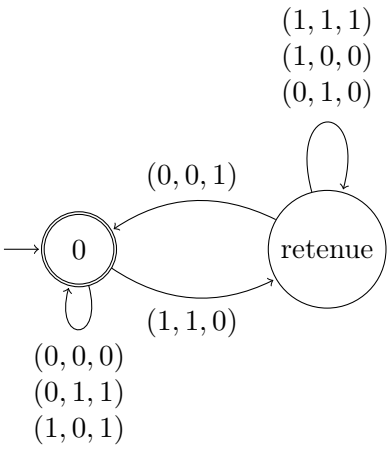
— cas $\psi := x_i = x_j$

On pose $\mathcal{A}_\psi :=$



— cas $\psi := x_i + x_j = x_l$ (on va mettre juste les indices qui nous intéressent)

On pose $\mathcal{A}_\psi :=$



3. c'est-à-dire tous les quantificateurs précèdent la formule

— cas $\psi := \psi_1 \vee \psi_2$ en prenant l'automate de l'union

— cas $\psi := \psi_1 \wedge \psi_2$ en prenant l'automate de l'intersection

— cas $\psi := \neg\psi_1$ en prenant l'automate du complémentaire [4]

Ainsi on peut construire A_ψ tel que $\mathrm{L}(A_\psi) = \mathcal{L}_n$.

Hérédité : On suppose que l'on a $A_{\varphi_{k+1}}$, on veut construire A_{φ_k}.

— Si $Q_{k+1} = \forall$, on se ramène à un $\exists$ par l'automate du complémentaire. [5]

— Si $Q_{k+1} = \exists$, on sait que l'on dispose de

$$\mathcal{A}_{\varphi_{k+1}} = (Q_{k+1}, I_{k+1}, F_{k+1}, T_{k+1}, \Sigma_{k+1})$$

On note $\pi_k : \begin{cases} \Sigma_{k+1} & \to & \Sigma_k \\ (x_1, \dots, x_{k+1}) & \mapsto & (x_1, \dots, x_k) \end{cases}$.

On pose alors

$$\mathcal{A}_{\varphi_k} = (Q_{k+1}, I_k, F_{k+1}, T_k, \Sigma_k)$$

où T_k est définie par

$$\text{si } p \xrightarrow{x} q \text{ est dans } \mathcal{A}_{\varphi_{k+1}} \text{ alors on met } p \xrightarrow{\pi_k(x)} q \text{ dans } \mathcal{A}_{\varphi_k}$$

et $I_k = I_{k+1} \cup \{T_k(i, \underbrace{(0, \dots, 0)}_{k \text{ fois}}), \forall i \in I_{k+1}\}$

On trouve en fait x_{k+1} de manière non déterministe mais il se pourrait que l'entier à "deviner" soit plus grand et nécessite plus de lettres de Σ_{k+1} et donc il faut que l'état qui viendrait d'un état initial de $\mathcal{A}_{\varphi_{k+1}}$ où, sur la transition, on aurait lu que des 0 soit un état initial de $\mathcal{A}_{\varphi_k}$, d'où la modification des états initiaux.

Conclusion : On a donc construit $\mathcal{A}_{\varphi_1}$ tel que $\mathrm{L}(\mathcal{A}_{\varphi_1}) = \mathcal{L}_1$.

Donc $\mathcal{L}_1$ est rationnel en vertu du théorème de Kleene.

Ainsi savoir $\mathcal{L}_1 \neq \emptyset$ et $\mathcal{L}_1 = \Sigma_1^*$ sont des problèmes décidables. Donc Pres est décidable, d'après la remarque de la fin de l'étape 1. □

Remarques :

L'arithmétique de Presburger est décidable, mais si on ajoute le symbole $\times$, on obtient l'arithmétique de Peano qui est indécidable.

Astuces de l'agrégatif :

On pourrait juste parler de langages reconnaissables et ne pas utiliser le théorème de Kleene mais souvent on énonce les problèmes Univ et Vide plutôt pour les langages rationnels donc j'ai préféré garder les utilisations usuelles de ces problèmes.

4. attention, il faut compléter et déterminiser l'automate
5. puisque $\forall x \psi(x)$ est sémantique équivalent à $\neg\exists x(\neg\psi(x))$.