Décidabilité de l'arithmétique de Presburger

Julien Devevey

2018-2019

Ref : Carton - Langages Formels, Calculabilité et Complexité p. 178 et DNR - Introduction à la Logique, p. 136

Définition 1. On dit qu'une théorie logique T est décidable si le problème suivant est décidable : "Etant donné une formule close F du langage de T, est-ce que $T \vdash F$?".

Théorème 2. La théorie au premier ordre des entiers munis de l'addition, mais pas de la multiplication, c'est à dire l'ensemble des formules closes F écrites sur $\{0, S, +, =\}$ telles que $\mathbb{N} \models F$, appelée arithmétique de Presburger, est décidable.

Démonstration.

Pour montrer le résultat, on va passer par des automates. Soit une formule ϕ de l'arithmétique de Presburger. On effectue la preuve par récurrence sur le nombre de quantificateurs de $\phi = Q_1 x_1 \dots Q_n x_n \psi$. On pose alors $\phi_k = Q_{k+1} x_{k+1} \dots Q_n x_n \psi$. Il y a alors k variables libres dans ϕ_k. On décide d'écrire tous les entiers en binaire sur l'alphabet $\{0, 1\}$. Les k-uplets de variables sont donc écrits sur l'alphabet $\Sigma_k = \{0, 1\}^k$, quitte à rajouter des 0 pour avoir des écritures de même longueur. On pose enfin $X_k = \{(n_1, \dots, n_k), \phi_k(n_1, \dots, n_k)$ est vraie.$\}$. On cherche alors à construire des automates $\mathcal{A}_k$ qui ont pour langage l'écriture sur Σ_k des éléments de X_k. On remarque d'ailleurs l'équivalence suivante : $(n_1, \dots n_{k-1}) \in X_{k-1} \Leftrightarrow Q_k n_k, (n_1, \dots, n_k) \in X_k$. On pose donc l'hypothèse de récurrence (H_k) "Il existe un automate $\mathcal{A}_k$ tel que son langage soit X_k".

Commençons par la construction de $\mathcal{A}_n$, qui accepte les n-uplets qui satisfont la formule ψ. Or cette formule, une fois mise sous forme prénexe se décompose en combinaison booléennes d'expressions de la forme $c_i = c_j$ ou $c_i + c_j = c_k$ où les c_i sont soit des constantes, soit une variable x_i. Or les langages rationnels sont stables pour les opérations booléennes, il suffit alors de construire un automate pour chacune de ces formules, cf les deux automates qui lisent les entrées du bit de poids faible au bit de poids fort.

On a donc (H_n).

Soit $0 \leq k < n$ et supposons (H_{k+1}) et on se donne un automate $\mathcal{A}_{k+1}$ vérifiant l'hypothèse de récurrence. Pour passer de $\mathcal{A}_{k+1}$ à $\mathcal{A}_k$, on utilise le fait que $\phi_k = Q_{k+1} x_{k+1} \phi_{k+1}$. Si Q_{k+1} est un $\forall$, on le remplace par $\neg\exists$ et on utilise la

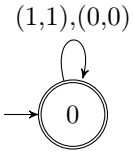

FIGURE 1 – Automate de l'égalité

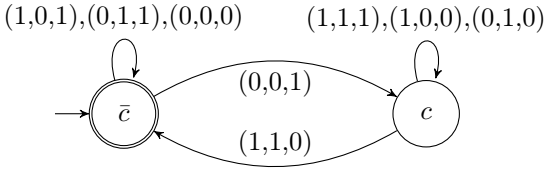

FIGURE 2 – Automate de l'addition

cloture des langages rationnels par passage au complémentaire pour se ramener au cas où on a un $\exists$. Alors, on construit l'automate $\mathcal{A}_k$ en projetant $\mathcal{A}_{k+1}$: on remplace tous les $(k+1)$-uplets par des k-uplets en supprimant la dernière composante, on garde les mêmes états finaux et les états initiaux sont tous ceux qui peuvent être atteints à partir d'un état initial et en suivant uniquement des transitions de la forme $(0,\dots,0)$, ce qui permet de prendre en compte les cas où n_{k+1} ait l'écriture en base 2 la plus longue. Soit alors $(n_1,\dots,n_k) \in X_k$. Donc il existe n_{k+1} tel que $(n_1,\dots,n_{k+1}) \in X_{k+1} = L(\mathcal{A}_{k+1})$ par (H_{k+1}). Par projection et donc construction de $\mathcal{A}_k$, on a alors que $(n_1,\dots,n_k) \in L(\mathcal{A}_k)$. Réciproquement, soit $(n_1,\dots,n_k) \in L(\mathcal{A}_k)$. On regarde l'exécution de $\mathcal{A}_k$ sur $(n_1,\dots,n_k)$: à chaque transition effectuée correspond au moins une transition dans $\mathcal{A}_{k+1}$. On construit alors des n^i_{k+1}, en regardant à chaque étape quel était la $k+1$ème composante de la transition dans $\mathcal{A}_{k+1}$ (comme l'automate n'est pas déterministe, on peut en construire plusieurs), et on sait de plus qu'au moins l'un d'entre eux, noté n_{k+1} correspond à une suite de transitions menant à un état acceptant dans $\mathcal{A}_k$. Alors si on exécute $\mathcal{A}_{k+1}$ sur $(n_1,\dots,n_{k+1})$ (en rajoutant des 0 sur les k premières composantes si nécessaire dans l'écriture en base 2), on a alors un $(k+1)$-uplet accepté par l'automate, ce qui montre alors que $\exists n_{k+1}, (n_1,\dots,n_{k+1}) \in L(\mathcal{A}_{k+1}) = X_{k+1}$. Au final, on a alors prouvé que $L(\mathcal{A}_k) = X_k$, ce qui conclut la récurrence.

Au final, on a bien construit un automate capable de déterminer si une formule ϕ est vraie ou pas, ce qui montre la décidabilité de l'arithmétique de Presburger. □