Décidabilité de l'arithmétique de Presburger

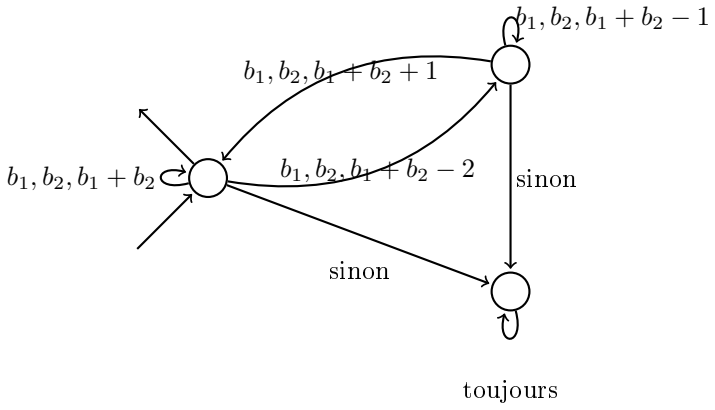
Benjamin Hellouin

Sipser, 6.2

Théorème 1. *L'arithmétique de Presburger est une théorie décidable. Autrement dit, il existe un algorithme qui, étant donné une formule de l'arithmétique de Presburger, décide si elle est démontrable ou non. Plus précisément, l'ensemble des n-uplets qui satisfont cette formule est rationnel.*

On donne un algorithme qui détermine si son entrée, une formule de l'arithmétique de Presburger est vraie dans le modèle $(\mathbb{N}, +)$. Soit $\phi = Q_1 x_1 \dots Q_n x_n \psi$. On note $\phi_i = Q_i \dots Q_l \psi$, qui possède i variables libres. On présente un algorithme qui, pour chaque i en ordre décroissant, construit un automate fini qui reconnaît les i-uplets de nombres qui rendent ϕ_i vraie. Ces i-uplets sont représentés sur l'alphabet $\Sigma_i = \{0,1\}^i$,le k-ième élément de chaque symbole représentant un bit de la k-ème variable libre. On représente ainsi l'entrée $(a_1 \dots a_i)$ avec éventuellement des 0 précédant le premier 1.

Initialisation.
Pour le cas de la première formule $\phi_n = \psi$, il s'agit simplement d'opérations booléennes sur des formules atomiques qui ne peuvent elles-mêmes qu'être de la forme $+(a_i, a_j, a_k)$. On va construire $A_n(i,j,k)$ un automate sur l'alphabet Σ_n qui reconnait l'entrée si et seulement si $a_i + a_j = a_k$, en supposant que le dernier bit est placé en premier. Pour chaque symbole x, on note (x_i, x_j, x_k) les bits correspondants, et le choix de la transition ne dépendent que d'eux.



On construit l'automate A_n qui reconnait l'entrée si et seulement si $\psi(a_1 \dots a_n)$ est satisfaite en utilisant les opération d'union ($\vee$), d'intersection ($\wedge$) et de complément ($\neg$) sur les automates ainsi construits.

Hérédité.
Supposons qu'on a construit l'automate A_{i+1} sur l'alphabet Σ_{i+1}, et que $\phi_i = \exists x_i, \phi_{i+1}$. L'automate non déterministe A_i^{nd} est constitué d'un nouvel état de départ D et de l'ensemble des états de A_{i+1}. Intuitivement, cet automate lit un symbole $x \in \{0,1\}^i$ et branche nondéterministiquement sur les deux états accessibles dans A_{i+1} par $x0$ et $x1$, afin que A_i^{nd} accepte $(a_1 \dots a_i)$ si et seulement si il existe un a_{i+1} tel que A_{i+1} accepte $(a_1 \dots a_{i+1}$. Cependant, cet automate n'essaie que les entiers au plus aussi longs que tous les autres, et il faut travailler un peu plus.

Notons Acc_{i+1} l'ensemble des états de A_{i+1} accessibles depuis l'état de départ en utilisant seulement $(0, \dots 0, 0)$ et $(0, \dots 0, 1)$. Les transitions de A_i^{nd} sont exactement :
- $(D, \varepsilon) \to Acc_i$
- $(q, x) \to \{q_0, q_1\}$ où $q_0 = \delta(q, x0)$ et $q_1 = \delta(q, x1)$ dans A_{i+1}.

Alors on a bien que A_i^{nd} accepte $(a_1 \dots a_i)$ si et seulement si il existe un a_{i+1} tel que A_{i+1} accepte $(a_1 \dots a_{i+1})$, et il suffit de le déterminiser pour trouver A_i.

Si $\phi_i = \forall x_i, \phi_{i+1}$, on se ramène au cas précédent en voyant que $\phi_i = \neg\exists x_i, \neg\phi_{i+1}$ et en passant au complémentaire deux fois.

Conclusion.
Une fois construit l'automate A_0, il suffit de tester si A_0 reconnait ε, ce qui revient exactement à tester si ϕ est vraie.