10. Presburger Arithmetic

Presburger Arithmetic is the first-order theory over the natural numbers ($\mathbb{N}_0$) with addition ($+$) as relation. It is convenient to also allow the constants 0 and 1 and the relations $\leq$ and $<$, with the canonical interpretation.

PA is named in honor of Mojżesz Presburger (1904–1943?):

- born in Warsaw
- died in Holocaust (1943?)
- student of Alfred Tarski
- MA-thesis: About the completeness of a certain system of integer arithmetic in which addition is the only operation (1930)

Again we are interested in which arithmetical problems can be solved using automata!

Syntax of PA

- Symbols:

variables	x, y, z …
constants	0, 1
arithmetic symbols	+, =<
logical symbols	or, not, Exists
parenthesis	

- Terms:
  - a variable is a term
  - 0 and 1 are terms
  - if t and u are terms, then t + u is a term

- Atomic formulas: t =< u , where t and u are terms

Syntax of PA

- Formulas:
  - every atomic formula is a formula;
  - if φ_1, φ_2 are formulas, then so are $\neg\varphi_1$, $\varphi_1 \vee \varphi_2$, and $\exists x \varphi_1$.
- Free and bound variables:
  - a variable is bound if it is in the scope of an existential quantifier, otherwise it is free.
- A formula without free variables is called a sentence

Abbreviations

Conjunction, implication, bi-implication, universal quantification

$$
\begin{aligned}
n &= \underbrace{1 + 1 + \ldots + 1}_{n \text{ times}} \\
nx &= \underbrace{x + x + \ldots + x}_{n \text{ times}}
\end{aligned}
\qquad
\begin{aligned}
t \geq t' &= t' \leq t \\
t = t' &= t \leq t' \wedge t \geq t' \\
t < t' &= t \leq t' \wedge \neg(t = t') \\
t > t' &= t' < t
\end{aligned}
$$

Semantics (intuition)

- The semantics of a sentence is "true" or "false"

- The semantics of a formula with free variables (x_1, ..., x_k) is the set containing all tuples (n_1, ..., n_k) of natural numbers that "satisfy the formula"

Semantics (more formally)

- An interpretation of a formula F is any function that assigns a natural number to every variable appearing in f (and perhaps also to others).

  Given an interpretation I, a variable x, and a number n, we denote by I[n/x] the interpretation that assigns to x the number n , and to all other variables the same value as I.

Semantics (more formally)

- We define when an interpretation satisfies a formula F.

$$
\begin{array}{lll}
\mathfrak{I} \models t \leq u & \text{iff} & \mathfrak{I}(t) \leq \mathfrak{I}(u) \\
\mathfrak{I} \models \neg\varphi_1 & \text{iff} & \mathfrak{I} \not\models \varphi_1 \\
\mathfrak{I} \models \varphi_1 \vee \varphi_2 & \text{iff} & \mathfrak{I} \models \varphi_1 \text{ or } \mathfrak{I} \models \varphi_2 \\
\mathfrak{I} \models \exists x\, \varphi & \text{iff} & \text{there exists } n \geq 0 \text{ such that } I[n/x] \models \varphi
\end{array}
$$

- Lemma: Let F be a formula, and let I1, I2 be two interpretations of F. If I1 and I2 assign the same values to all FREE variables of F, then either they both satisfy F or none of them satisfies F.

- Consequence: if F is a sentence, either all interpretations satisfy F, or none of them satisfies F.

Semantics (more formally)

- We say a sentence is true if it is satisfied by all interpretations.

- We say a sentence is false if it is not satisfied by any interpretation.

- A model or solution of a formula F is the projection of any interpretation that satisfies F onto the free variables of F.

- The set of models or solutions of F is also called the solution space of F, and denoted by Sol(F).

Language of a formula

we encode natural numbers as strings over $\{0, 1\}$ using the least-significant-bit-first encoding *lsbf*. If we have free variables $x_1, \ldots, x_k$, the elements of the solution space are encoded as a word over $\{0, 1\}^k$. For instance, the word

$$\begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \quad \begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix}$$

is an encoding of the solution $(3, 10, 0)$. The language of a formula is then defined to be

$$\mathcal{L}(\varphi) = \{lsbf(s) \mid s \in Sol(\varphi)\}$$

Constructing an NFA for the solution space

Given a formula F, we construct an NFA Aut(F) such that L(Aut(F)) = L(F).

We can take:

- Aut(not F) = CompNFA(Aut(F))
- Aut(F or G) = UnionNFA(Aut(F), Aut(G))
- Aut(Exists x F) = Projection_x(Aut (F))

So it remains to define Aut(F) for an atomic formula F.

All atomic formulas equivalent (same solutions) to atomic formulas of the form

$$\varphi = \quad a_1 x_1 + \ldots + a_n x_n \leq b = \quad a \cdot x \leq b$$

where the a_i and b can be arbitrary integers (possibly negative).

Consider a candidate solution

$$\begin{array}{ccccc} & \zeta_0 & \zeta_1 & \cdots & \zeta_m \\ \begin{array}{c} c_1 \\ c_2 \\ \cdots \\ c_n \end{array} & \begin{bmatrix} \zeta_{10} \\ \zeta_{20} \\ \cdots \\ \zeta_{n0} \end{bmatrix} & \begin{bmatrix} \zeta_{11} \\ \zeta_{21} \\ \cdots \\ \zeta_{n1} \end{bmatrix} & \begin{array}{c} \cdots \\ \cdots \\ \cdots \\ \cdots \end{array} & \begin{bmatrix} \zeta_{1m} \\ \zeta_{2m} \\ \cdots \\ \zeta_{nm} \end{bmatrix} \end{array}$$

For every $j \leq m$, let $c^j \in \mathbb{N}^n$ denote the tuple of numbers encoded by the prefix $\zeta_0 \ldots \zeta_{j-1}$. For instance, for the encoding $\zeta_0\zeta_1\zeta_2$ of the tuple $(0, 4, 7, 3)$ given by

$$\begin{matrix} & \zeta_0 & \zeta_1 & \zeta_2 \\ \begin{matrix}0\\4\\7\\3\end{matrix} & \begin{bmatrix}0\\0\\1\\1\end{bmatrix} & \begin{bmatrix}0\\0\\1\\1\end{bmatrix} & \begin{bmatrix}0\\1\\1\\0\end{bmatrix}\end{matrix} \qquad \text{we get} \qquad \begin{matrix} & \zeta_0 & \zeta_1 \\ \begin{matrix}0\\0\\3\\3\end{matrix} & \begin{bmatrix}0\\0\\1\\1\end{bmatrix} & \begin{bmatrix}0\\0\\1\\1\end{bmatrix}\end{matrix}$$

and so $c^2 = (0, 0, 3, 3)$. Define further $c^0 = (0, 0, 0, 0)$; i.e., before reading anything all components of the tuple are 0.

We construct a DFA for the solution space of φ. The idea is that after reading a prefix $\zeta_0 \ldots \zeta_{j-1}$ the automaton should be in the state

$$\left\lfloor \frac{1}{2^j}\left(b - a \cdot c^j\right) \right\rfloor \tag{10.1}$$

Initially we have $c^0 = (0, \ldots, 0)$, and so the initial state is the number $\frac{1}{2^0}(b - a \cdot c^0) = b$. For the transitions, assume that before and after reading the letter ζ_j the automaton is in the states q and q', respectively. Then we have

$$q = \left\lfloor \frac{1}{2^j} \left(b - a \cdot c^j\right) \right\rfloor \qquad \text{and} \qquad q' = \left\lfloor \frac{1}{2^{j+1}} \left(b - a \cdot c^{j+1}\right) \right\rfloor$$

From the definition of c^j we get:

$$c^{j+1} = c^j + 2^j \zeta_j$$

Inserting this in the expression for q', and comparing with q, we obtain the following relation between q and q':

$$q' = \left\lfloor \frac{1}{2}(q - a \cdot \zeta_j) \right\rfloor$$

So for every state q and every letter $\zeta \in \{0, 1\}^n$ we take $\delta(q, \zeta) := \frac{1}{2}(q - a \cdot \zeta)$.

PAtoDFA(φ)

Input: PA formula $\varphi = a \cdot x \leq b$

Output: DFA $A = (Q, \Sigma, \delta, q_0, F)$ such that $\mathcal{L}(A) = \mathcal{L}(\varphi)$

$q_0 \leftarrow s_b$
$W \leftarrow \{s_b\}$
while $W \neq \emptyset$ **do**
    pick s_k **from** W
    add s_k **to** Q
    if $k \geq 0$ **then add** s_k **to** F
    for all $\zeta \in \{0,1\}^n$ **do**
        $j \leftarrow \left\lfloor \frac{1}{2}(k - a \cdot \zeta) \right\rfloor$
        if $s_j \notin Q$ **then add** s_j **to** W
        add (s_k, ζ, s_j) **to** δ

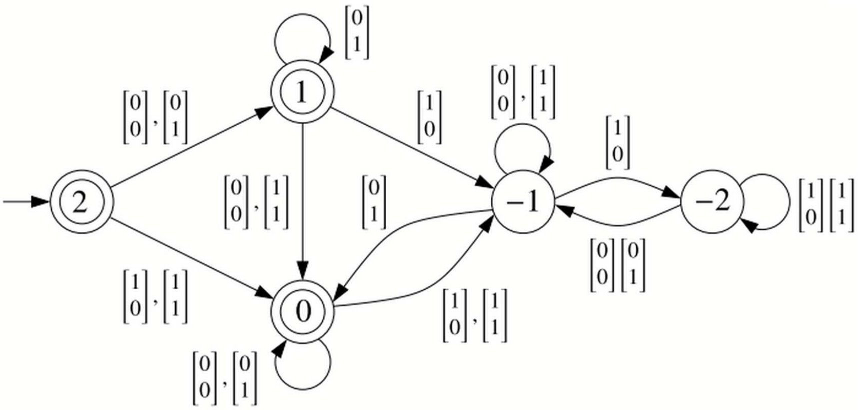

Figure 10.1: DFAs for the formula $2x - y \leq 2$.

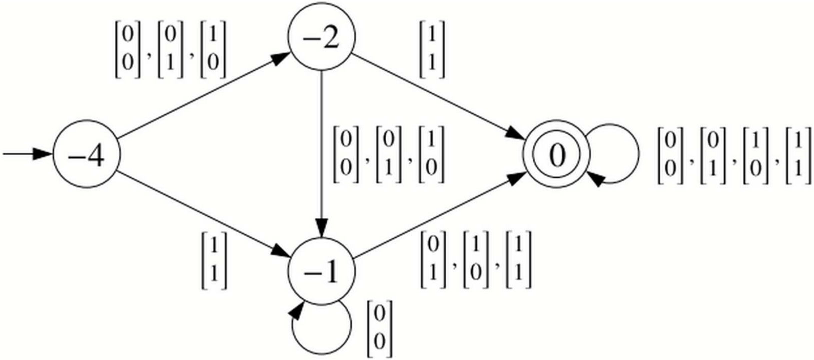

Figure 10.2: DFAs for the formula $x + y \geq 4$.