

# Presburger Arithmetic

- Which arithmetical problems can be solved using automata?
- **Presburger arithmetic (PA)**: a logical language to define arithmetical properties of (tuples of) natural numbers

Is there an integer solution?

$$3x - 4y = 5$$

$$-x + y = 3$$

Is there an integer solution?

$$\begin{aligned}2x + 3y &\geq 5 \\ -x + 4y &\leq 3\end{aligned}$$

Are there integers  $x, y$  such that

$$3x - 4y = 5$$

$$-x + y = 3$$

but not

$$2x + 3y \geq 2$$

$$-x + 4y \leq 4 \quad ?$$

For every integer solution  $x, y$  of

$$2x + 3y \geq 5$$

$$-x + 4y \leq 3$$

is there is an integer solution  $z, u$  of

$$3z - 2u \geq 3$$

$$-z + 4u \leq -2$$

such that  $x + z = y + u$  ?

# Syntax of PA

- Symbols:

Variables

$x, y, z \dots$

Constants

$0, 1$

Arithmetical symbols

$+, \leq$

Logical symbols

$\forall, \neg, \exists$        $(\wedge, \vee, \rightarrow, \dots)$

Parenthesis

$(, )$

- Terms:

Variables,  $0$  and  $1$  are terms.

If  $t$  and  $u$  are terms, then  $t + u$  is a term.

# Syntax of PA

- Atomic formulas:

$t \leq u$ , where  $t$  and  $u$  are terms

- Formulas:

Atomic formulas are formulas.

If  $\varphi_1, \varphi_2$  are formulas, then so are  $\varphi_1 \vee \varphi_2, \neg \varphi_1, \exists x \varphi_1$

- Free and bound variables:

A variable is **bound** if it is in the scope of an existential quantifier, otherwise it is **free**.

- Sentences: formulas without free variables.



# Abbreviations

- Logical abbreviations:

$$\varphi_1 \wedge \varphi_2 \equiv \neg(\neg\varphi_1 \vee \neg\varphi_2)$$

$$\varphi_1 \rightarrow \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2$$

$$\varphi_1 \leftrightarrow \varphi_2 \equiv \neg(\varphi_1 \vee \varphi_2) \vee \neg(\neg\varphi_1 \vee \neg\varphi_2)$$

$$\forall x \varphi \equiv \neg \exists x \neg \varphi$$

- Arithmetic abbreviations:

$$\begin{array}{ll} n & := \underbrace{1 + 1 + \dots + 1}_{n \text{ times}} & t \geq t' & := t' \leq t \\ nx & := \underbrace{x + x + \dots + x}_{n \text{ times}} & t = t' & := t \leq t' \wedge t \geq t' \\ & & t < t' & := t \leq t' \wedge \neg(t = t') \\ & & t > t' & := t' < t \end{array}$$

# Semantics (intuition)

- The semantics of a sentence is **true** or **false**.
- The semantics of a formula with free variables  $(x_1, \dots, x_k)$  is the set containing all tuples  $(n_1, \dots, n_k)$  of natural numbers that "satisfy the formula"

# Semantics (more formally)

- An **interpretation of a formula  $\varphi$**  is a function  $\mathcal{J}$  that assigns a natural number to every free variable appearing in  $\varphi$  (and perhaps also to others).
- Given an interpretation  $\mathcal{J}$ , a variable  $x$ , and a number  $n$ , we denote by  $\mathcal{J}[n/x]$  the interpretation that assigns to  $x$  the number  $n$ , and to all other variables the same value as  $\mathcal{J}$ .

# Semantics (more formally)

- We inductively define when an interpretation  $\mathcal{J}$  satisfies a formula  $\varphi$ , denoted by  $\mathcal{J} \models \varphi$  :

$$\mathcal{J} \models t \leq u \quad \text{iff} \quad \mathcal{J}(t) \leq \mathcal{J}(u)$$

$$\mathcal{J} \models \neg\varphi_1 \quad \text{iff} \quad \mathcal{J} \not\models \varphi_1$$

$$\mathcal{J} \models \varphi_1 \vee \varphi_2 \quad \text{iff} \quad \mathcal{J} \models \varphi_1 \text{ or } \mathcal{J} \models \varphi_2$$

$$\mathcal{J} \models \exists x \varphi \quad \text{iff} \quad \text{there exists } n \geq 0 \text{ such that } \mathcal{J}[n/x] \models \varphi$$

# Semantics (more formally)

- **Lemma:** If two interpretations of a formula  $\varphi$  assign the same values to all **free** variables of  $\varphi$ , then either both satisfy  $\varphi$  or none satisfy  $\varphi$ .
- **Corollary:** if  $\varphi$  is a sentence, either all interpretations satisfy  $\varphi$ , or none satisfy  $\varphi$ .
- A sentence is **true** if it is satisfied by all interpretations, and **false** if it is not satisfied by any interpretation.
- A **model** or **solution** of  $\varphi$  is the projection of an interpretation that satisfies  $\varphi$  onto the free variables of  $\varphi$ . The set of solutions or **solution space** is denoted by  $Sol(\varphi)$ .

# Formulating questions

Are there integers  $x, y$  such that

$$2x + 3y \geq 5$$

$$-x + 4y \leq 3 \quad ?$$

$$\exists x \exists y (2x + 3y \geq 5 \wedge -x + 4y \leq 3)$$

# Formulating questions

For every solution  $x, y$  of

$$2x + 3y \geq 5$$

$$-x + 4y \leq 3$$

is there is a solution  $z, u$  of

$$3z - 2u \geq 3$$

$$-z + 4u \leq -2$$

such that  $x + z = y + u$ ?

$$\forall x \forall y$$

$$(2x + 3y \geq 5 \wedge -x + 4y \leq 3)$$

$\rightarrow$

$$(\exists z \exists u$$

$$(3z - 2u \geq 3 \quad \wedge$$

$$-z + 4u \leq -2 \quad \wedge$$

$$x + z = y + u \quad ) )$$

# Language of a formula

- We encode natural numbers with the *lsbf encoding*.
- If  $\varphi$  has free variables  $x_1, \dots, x_k$ , we encode a solution of  $\varphi$  as a word over  $\{0,1\}^k$  in the usual way. E.g, the encoding of  $(x_1, x_2, x_3) = (5, 10, 0)$  is

$$\begin{array}{l} x_1 \\ x_2 \\ x_3 \end{array} \quad \begin{array}{cccc} \left[ \begin{array}{c} 1 \\ 0 \\ 0 \end{array} \right] & \left[ \begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right] & \left[ \begin{array}{c} 1 \\ 0 \\ 0 \end{array} \right] & \left[ \begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right] \end{array}$$

- The *language of  $\varphi$* , denoted by  $L(\varphi)$ , is the set of encodings of the solutions of  $\varphi$ .



# An NFA for the solution space

- Given  $\varphi$ , we construct an NFA  $A_\varphi$  such that  $L(A_\varphi) = L(\varphi)$
- We can take:

$$A_{\neg\varphi} \quad := \quad \text{CompNFA}(A_\varphi)$$

$$A_{(\varphi_1 \vee \varphi_2)} \quad := \quad \text{UnionNFA}(A_{\varphi_1}, A_{\varphi_2})$$

$$A_{\exists x\varphi} \quad := \quad \text{Projection}_x(A_\varphi)$$

where  $\text{Projection}_x$  projects onto all variables but  $x$

- It remains to construct  $A_\varphi$  for an atomic formula  $\varphi$ .

# DFA for atomic formulas

- Every atomic formula has the same solutions as a formula of the form

$$a_1x_1 + \dots + a_nx_n \leq b := a \cdot x \leq b$$

where the  $a_i$  and  $b$  are arbitrary integers (possibly negative).

- Given  $a \cdot x \leq b$  we construct a DFA with integers as states and  $b$  as initial state satisfying:

Each state  $q \in \mathbb{Z}$  recognizes the tuples  $c \in \mathbb{N}^n$  such that  $a \cdot c \leq q$

# Transitions

- Given  $q \in \mathbb{Z}$  and a letter  $\zeta \in \{0,1\}^n$  we compute the target state  $q' \in \mathbb{Z}$  of the transition  $(q, \zeta, q')$ .
- For every word  $w \in (\{0,1\}^n)^*$  we have:
  - $w$  is accepted from  $q'$  iff  $\zeta w$  is accepted from  $q$and so for every tuple  $c \in \mathbb{N}^n$ :
  - $c$  is accepted from  $q'$  iff  $2c + \zeta$  is accepted from  $q$
- Hence we choose  $q'$  so that
  - $a \cdot c \leq q'$  iff  $a \cdot (2c + \zeta) \leq q$
- Since  $a \cdot (2c + \zeta) \leq q$  iff  $2(a \cdot c) + a \cdot \zeta \leq q$  we take

$$q' = \left\lfloor \frac{q - a \cdot \zeta}{2} \right\rfloor$$

# Final states

- A state is final iff it accepts the empty word
- So  $q \in \mathbb{Z}$  is final iff it accepts  $(0, \dots, 0) \in \mathbb{N}^n$
- So we take  $q \in \mathbb{Z}$  final iff  $a \cdot (0, \dots, 0) \leq q$  iff  $q \geq 0$

*AFtoDFA*( $\varphi$ )

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

**Output:** DFA  $A_\varphi = (Q, \Sigma, \delta, q_0, F)$  such that  $L(A_\varphi) = L(\varphi)$

```
1   $Q, \delta, F \leftarrow \emptyset; q_0 \leftarrow s_b$ 
2   $W \leftarrow \{s_b\}$ 
3  while  $W \neq \emptyset$  do
4      pick  $s_k$  from  $W$ 
5      add  $s_k$  to  $Q$ 
6      if  $k \geq 0$  then add  $s_k$  to  $F$ 
7      for all  $\zeta \in \{0, 1\}^n$  do
8           $j \leftarrow \left\lfloor \frac{k - a \cdot \zeta}{2} \right\rfloor$ 
9          if  $s_j \notin Q$  then add  $s_j$  to  $W$ 
10         add  $(s_k, \zeta, s_j)$  to  $\delta$ 
```

# Example: $3x - 2y \geq 6$

Conversion:  $-3x + 2y \leq -6$   
 $a = \begin{pmatrix} -3 \\ 2 \end{pmatrix}, b = -6$

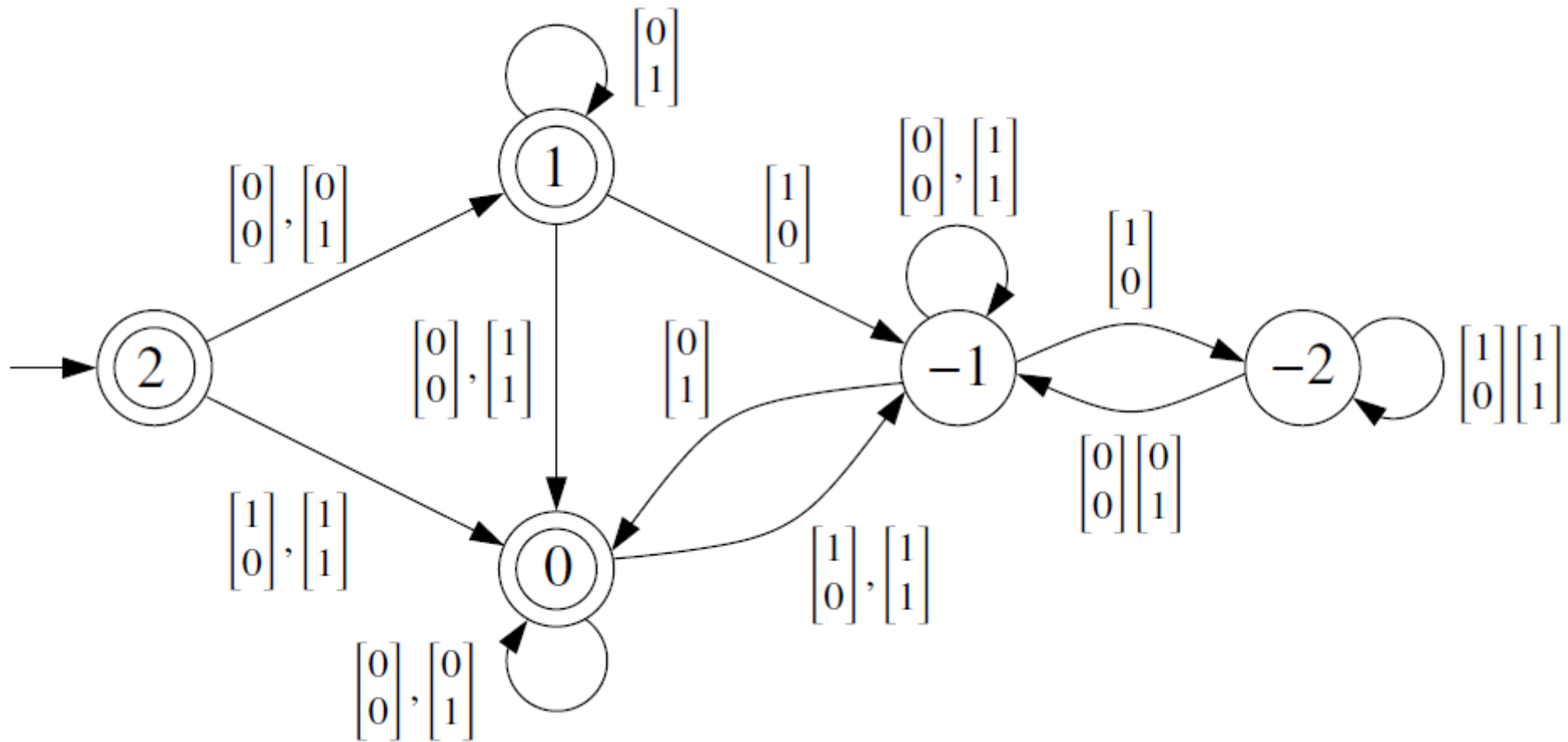
Initial state:  $-6$

Transition from state  $-6$  with letter  $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$  :

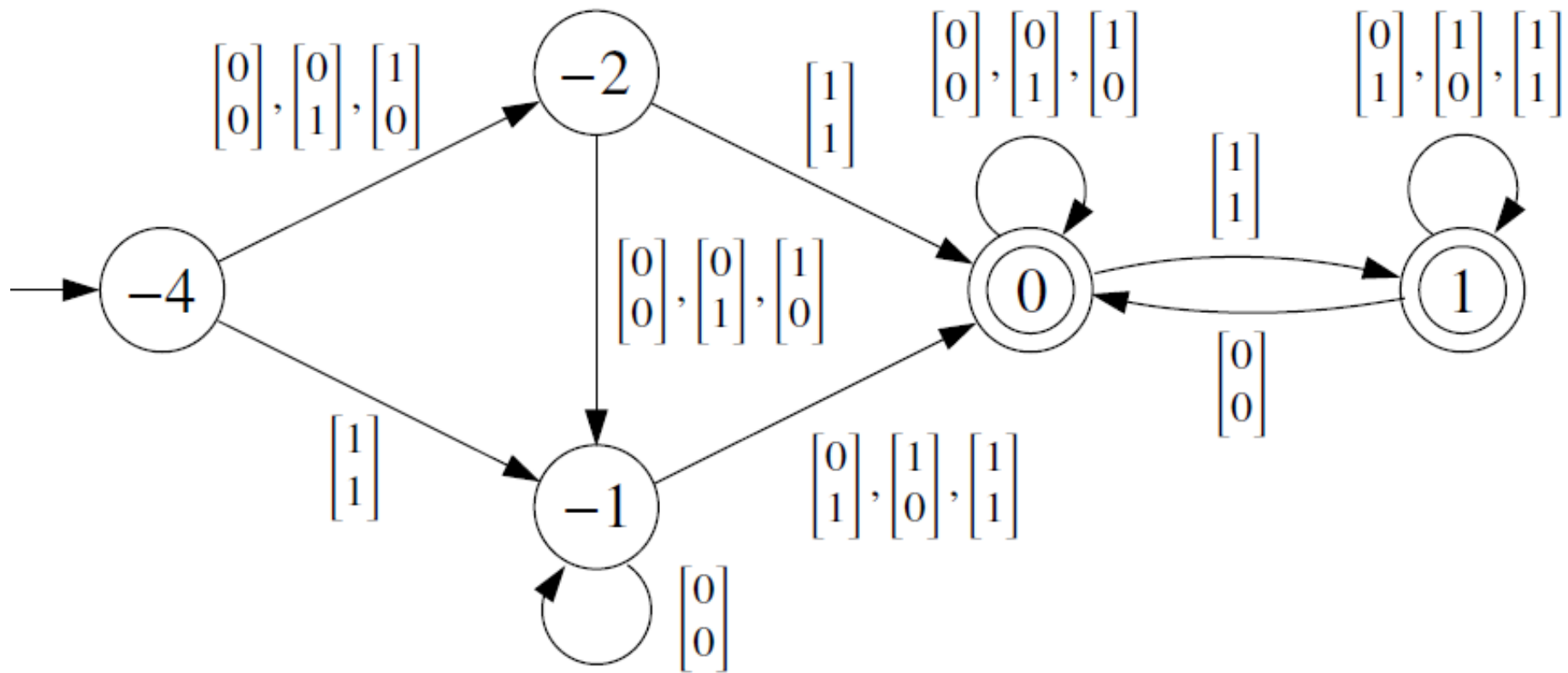
$$q' = \left\lfloor \frac{q - a \cdot \zeta}{2} \right\rfloor$$

$$q' = \left\lfloor \frac{-6 - (-3, 2) \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}}{2} \right\rfloor = \left\lfloor \frac{-6 + 1}{2} \right\rfloor = -3$$

Example:  $2x - y \leq 2$



Example:  $x + y \geq 4$





# Termination of *AFtoDFA*

- **Lemma:** Let  $\varphi = a \cdot c \leq q$  and  $s = \sum_{i=1}^n |a_i|$ . All states  $s_j$  added by *AFtoDFA*( $\varphi$ ) satisfy

$$-|b| - s \leq j \leq |b| + s$$

**Proof:** Holds for the first state added:  $s_b$

Assume  $s_j$  is added to the workset when processing  $s_k$ .

By ind. hyp.:  $-|b| - s \leq k \leq |b| + s$ .

Together with  $j = \left\lfloor \frac{k - a \cdot \zeta}{2} \right\rfloor$  we get

$$\left\lfloor \frac{-|b| - s - a \cdot \zeta}{2} \right\rfloor \leq j \leq \left\lfloor \frac{|b| + s - a \cdot \zeta}{2} \right\rfloor$$

$$\left\lfloor \frac{-|b| - s - a \cdot \zeta}{2} \right\rfloor \leq j \leq \left\lfloor \frac{|b| + s - a \cdot \zeta}{2} \right\rfloor$$

Some arithmetic yields

$$\begin{aligned} -|b| - s &\leq \frac{-|b| - 2s}{2} \leq \left\lfloor \frac{-|b| - s - a \cdot \zeta}{2} \right\rfloor \\ \left\lfloor \frac{|b| + s - a \cdot \zeta}{2} \right\rfloor &\leq \frac{|b| + 2s}{2} \leq |b| + s \end{aligned}$$

and together we get

$$-|b| - s \leq j \leq |b| + s$$

# Solving a system of inequations

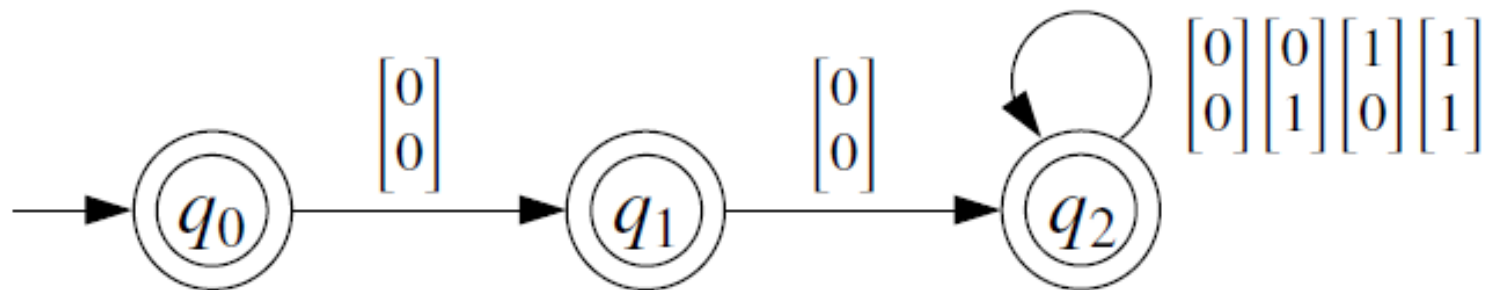
- We compute all solutions of

$$\begin{aligned}2x - y &\leq 2 \\ x + y &\geq 2\end{aligned}$$

s.t.  $x, y$  are multiples of 4. They are the solutions of

$$(\exists z \ x = 4z) \wedge (\exists w \ y = 4w) \wedge (2x - y \leq 2) \wedge (x + y \geq 4)$$

- DFA for  $(\exists z x = 4z) \wedge (\exists w y = 4w)$



- Final result

