

Discussion:

- **Integer Linear Programming** (ILP) can decide satisfiability of a finite set of equations/inequations over \mathbb{Z} of the form:

$$\sum_{i=1}^n a_i \cdot x_i = b \quad \text{bzw.} \quad \sum_{i=1}^n a_i \cdot x_i \geq b, \quad a_i \in \mathbb{Z}$$

- Moreover, a (linear) cost function can be optimized :-)
- **Warning:** The decision problem is in general, already NP-hard !!!
- Notwithstanding that, surprisingly efficient implementations exist.
- Not just loop fusion, but also other re-organizations of loops yield ILP problems ...

Background 5: Presburger Arithmetic

Many problems in computer science can be formulated **without multiplication** :-)

Let us first consider two **simple** special cases ...

1. Linear Equations

$$\begin{array}{rclclcl} 2x & + & 3y & & & = & 24 \\ x & - & y & + & 5z & = & 3 \end{array}$$

Question:

- Is there a solution over \mathbb{Q} ?
- Is there a solution over \mathbb{Z} ?
- Is there a solution over \mathbb{N} ?

Let us reconsider the equations:

$$\begin{array}{rclcl} 2x & + & 3y & & = & 24 \\ x & - & y & + & 5z & = & 3 \end{array}$$

Answers:

- Is there a solution over \mathbb{Q} ? **Yes**
- Is there a solution over \mathbb{Z} ? **No**
- Is there a solution over \mathbb{N} ? **No**

Complexity:

- Is there a solution over \mathbb{Q} ? **Polynomial**
- Is there a solution over \mathbb{Z} ? **Polynomial**
- Is there a solution over \mathbb{N} ? **NP-hard**

Solution Method for Integers:

Observation 1:

$$a_1x_1 + \dots + a_kx_k = b \quad (\forall i : a_i \neq 0)$$

has a solution iff

$$\gcd\{a_1, \dots, a_k\} \mid b$$

Example:

$$5y - 10z = 18$$

has no solution over \mathbb{Z} :-)

Example:

$$5y - 10z = 18$$

has no solution over \mathbb{Z} :-)

Observation 2:

Adding a multiple of one equation to another does not change the set of solutions :-)

Example:

$$\begin{array}{rclclcl} 2x & + & 3y & & & = & 24 \\ x & - & y & + & 5z & = & 3 \end{array}$$

Example:

$$2x + 3y = 24$$

$$x - y + 5z = 3$$



$$5y - 10z = 18$$

$$x - y + 5z = 3$$

Observation 3:

Adding multiples of columns to another column is an invertible transformation which we keep track of in a separate matrix ...

$$\begin{array}{ccc|c} 1 & 0 & 0 & \\ 0 & 1 & 0 & x \\ 0 & 0 & 1 & \end{array} \begin{array}{l} \\ - \\ + \end{array} \begin{array}{l} 5y \\ y \\ 5z \end{array} \begin{array}{l} = \\ \\ = \end{array} \begin{array}{l} 18 \\ 3 \end{array}$$



$$\begin{array}{ccc|c} 1 & 0 & 0 & \\ 0 & 1 & 2 & x \\ 0 & 0 & 1 & \end{array} \begin{array}{l} \\ - \\ + \end{array} \begin{array}{l} 5y \\ y \\ 3z \end{array} \begin{array}{l} = \\ \\ = \end{array} \begin{array}{l} 18 \\ 3 \end{array}$$

Observation 3:

Adding multiples of columns to another column is an invertible transformation which we keep track of in a separate matrix ...

$$\begin{array}{ccc|c} 1 & 0 & 0 & 5y = 18 \\ 0 & 1 & 2 & x - y + 3z = 3 \\ 0 & 0 & 1 & \end{array}$$



$$\begin{array}{ccc|c} 1 & 0 & -3 & 5y = 18 \\ 0 & 1 & 2 & x - y = 3 \\ 0 & 0 & 1 & \end{array}$$

 triangular form !!

Observation 4:

- A special solution of a triangular system can be directly read off :-)
- All solutions of a homogeneous triangular system can be directly read off :-)
- All solutions of the original system can be recovered from the solutions of the triangular system by means of the accumulated transformation matrix:-))

Example

$$\begin{array}{ccc|c} 1 & 0 & -3 & 5y \\ 0 & 1 & 2 & x - y \\ 0 & 0 & 1 & \end{array} = \begin{array}{c} 15 \\ 3 \end{array}$$

One special solution:

$$[6, 3, 0]^T$$

All solutions of the homogeneous system are spanned by:

$$[0, 0, 1]^T$$

Solving over \mathbb{N}

- ... is of major practical importance;
- ... has led to the development of many new techniques;
- ... easily allows to encode NP-hard problems;
- ... remains difficult if just three variables are allowed per equation.

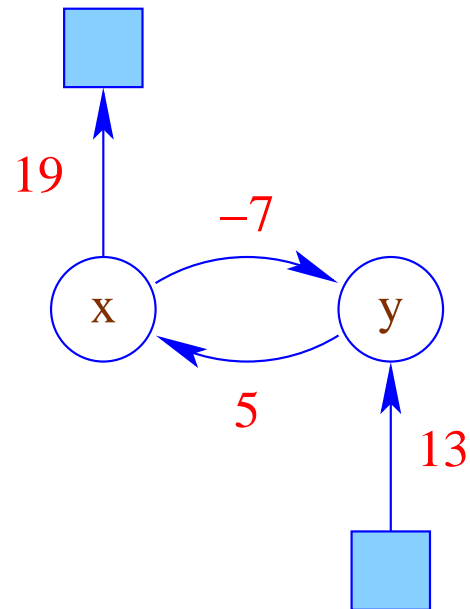
2. One Polynomial Special Case:

$$\begin{aligned}x &\geq y + 5 \\19 &\geq x \\y &\geq 13 \\y &\geq x - 7\end{aligned}$$

- There are at most 2 variables per **in**-equation;
- no scaling factors.

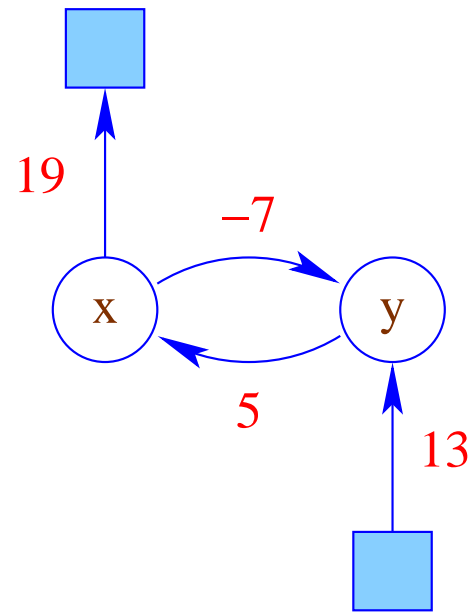
Idea:

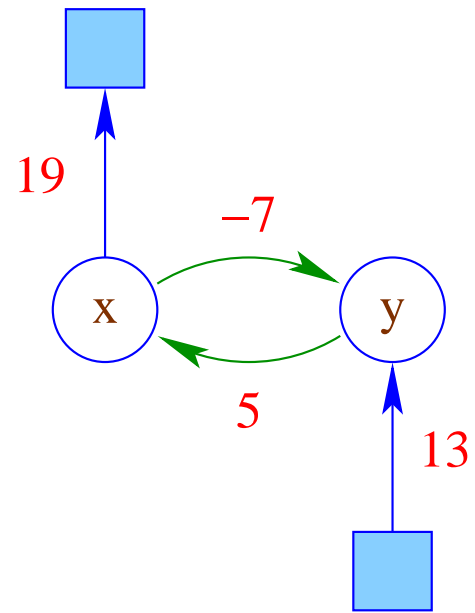
Represent the system by a **graph**:

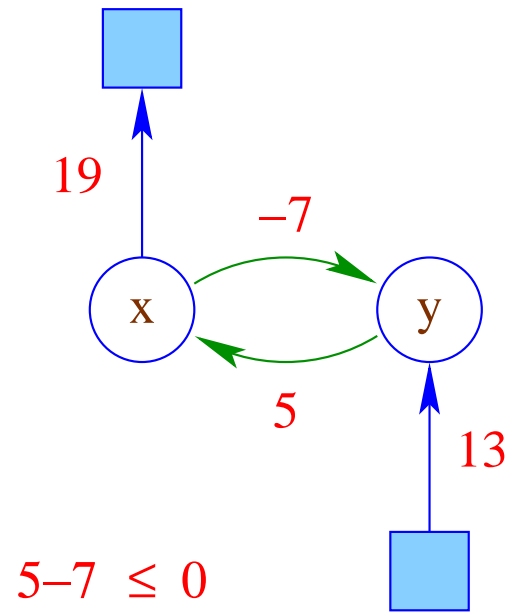


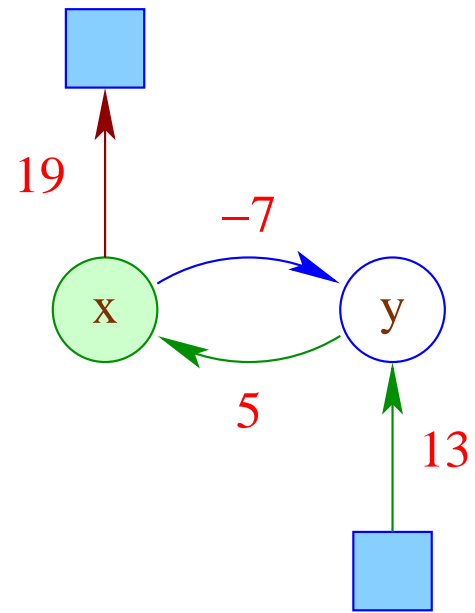
The in-equations are **satisfiable** iff

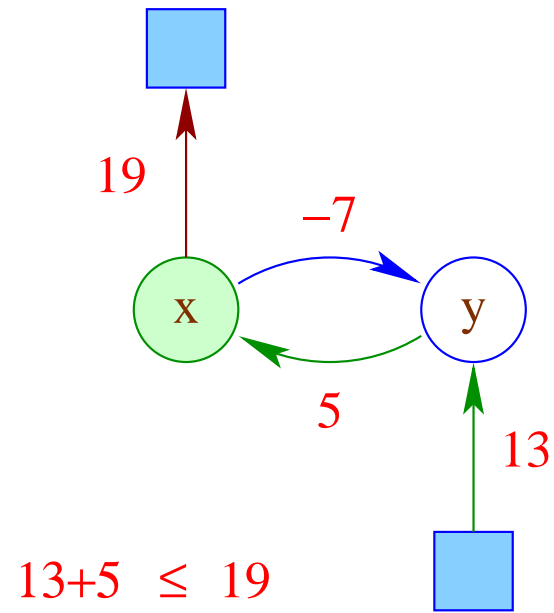
- the weight of every **cycle** are at most **0**;
- the weights of paths **reaching** x are bounded by the weights of edges from x into the **sink**.











The in-equations are **satisfiable** iff

- the weight of every **cycle** are at most **0**;
- the weights of paths **reaching** x are bounded by the weights of edges from x into the **sink**.



Compute the **reflexive** and **transitive** closure of the edge weights!

3. A General Solution Method:

Idea: Fourier-Motzkin Elimination

- Successively remove individual variables x !
- All in-equations with **positive** occurrences of x yield **lower bounds**.
- All in-equations with **negative** occurrences of x yield **upper bounds**.
- All lower bounds must be at most as big as all upper bounds ;-))



Jean Baptiste Joseph Fourier, 1768–1830

Example:

$$9 \leq 4x_1 + x_2 \quad (1)$$

$$4 \leq x_1 + 2x_2 \quad (2)$$

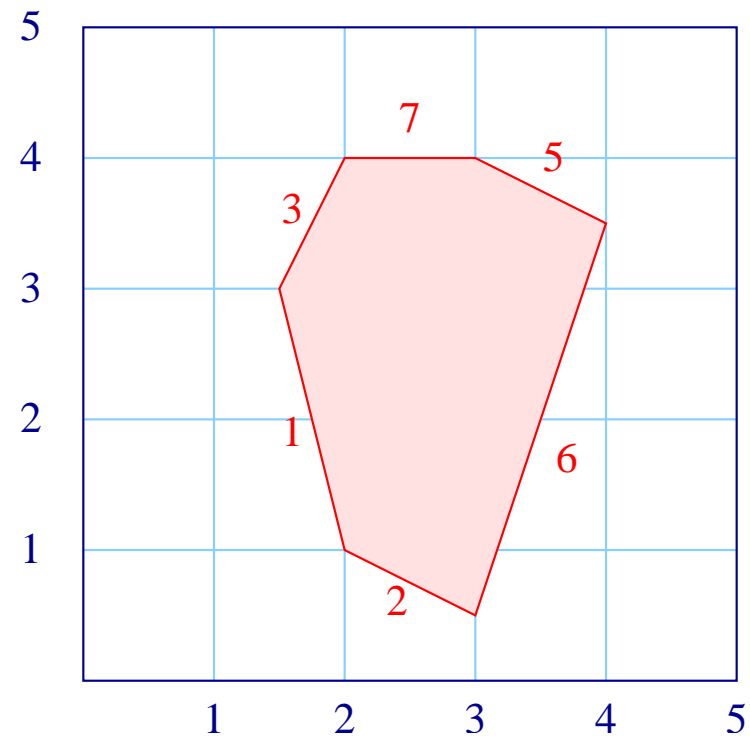
$$0 \leq 2x_1 - x_2 \quad (3)$$

$$6 \leq x_1 + 6x_2 \quad (4)$$

$$-11 \leq -x_1 - 2x_2 \quad (5)$$

$$-17 \leq -6x_1 + 2x_2 \quad (6)$$

$$-4 \leq -x_2 \quad (7)$$



For x_1 we obtain:

$$9 \leq 4x_1 + x_2 \quad (1)$$

$$4 \leq x_1 + 2x_2 \quad (2)$$

$$0 \leq 2x_1 - x_2 \quad (3)$$

$$6 \leq x_1 + 6x_2 \quad (4)$$

$$-11 \leq -x_1 - 2x_2 \quad (5)$$

$$-17 \leq -6x_1 + 2x_2 \quad (6)$$

$$-4 \leq -x_2 \quad (7)$$

$$\frac{9}{4} - \frac{1}{4}x_2 \leq x_1 \quad (1)$$

$$4 - 2x_2 \leq x_1 \quad (2)$$

$$\frac{1}{2}x_2 \leq x_1 \quad (3)$$

$$6 - 6x_2 \leq x_1 \quad (4)$$

$$x_1 \leq 11 - 2x_2 \quad (5)$$

$$x_1 \leq \frac{17}{6} + \frac{1}{3}x_2 \quad (6)$$

$$-4 \leq -x_2 \quad (7)$$

If such an x_1 exists, all lower bounds must be bounded by all upper bounds, i.e.,

$\frac{9}{4} - \frac{1}{4}x_2 \leq 11 - 2x_2$	(1, 5)		$-35 \leq -7x_2$	(1, 5)
$\frac{9}{4} - \frac{1}{4}x_2 \leq \frac{17}{6} + \frac{1}{3}x_2$	(1, 6)		$-\frac{7}{12} \leq \frac{7}{12}x_2$	(1, 6)
$4 - 2x_2 \leq 11 - 2x_2$	(2, 5)		$-7 \leq 0$	(2, 5)
$4 - 2x_2 \leq \frac{17}{6} + \frac{1}{3}x_2$	(2, 6)		$\frac{7}{6} \leq \frac{7}{3}x_2$	(2, 6)
$\frac{1}{2}x_2 \leq 11 - 2x_2$	(3, 5)	or	$-22 \leq -5x_2$	(3, 5)
$\frac{1}{2}x_2 \leq \frac{17}{6} + \frac{1}{3}x_2$	(3, 6)		$-\frac{17}{6} \leq -\frac{1}{6}x_2$	(3, 6)
$6 - 6x_2 \leq 11 - 2x_2$	(4, 5)		$-5 \leq 4x_2$	(4, 5)
$6 - 6x_2 \leq \frac{17}{6} + \frac{1}{3}x_2$	(4, 6)		$\frac{19}{6} \leq \frac{19}{3}x_2$	(4, 6)
$-4 \leq -x_2$	(7)		$-4 \leq -x_2$	(7)

$\frac{9}{4} - \frac{1}{4}x_2 \leq 11 - 2x_2$	(1, 5)		$-5 \leq -x_2$	(1, 5)
$\frac{9}{4} - \frac{1}{4}x_2 \leq \frac{17}{6} + \frac{1}{3}x_2$	(1, 6)		$-1 \leq x_2$	(1, 6)
$4 - 2x_2 \leq 11 - 2x_2$	(2, 5)		$-7 \leq 0$	(2, 5)
$4 - 2x_2 \leq \frac{17}{6} + \frac{1}{3}x_2$	(2, 6)		$\frac{1}{2} \leq x_2$	(2, 6)
$\frac{1}{2}x_2 \leq 11 - 2x_2$	(3, 5)	or	$-\frac{22}{5} \leq -x_2$	(3, 5)
$\frac{1}{2}x_2 \leq \frac{17}{6} + \frac{1}{3}x_2$	(3, 6)		$-17 \leq -x_2$	(3, 6)
$6 - 6x_2 \leq 11 - 2x_2$	(4, 5)		$-\frac{5}{4} \leq x_2$	(4, 5)
$6 - 6x_2 \leq \frac{17}{6} + \frac{1}{3}x_2$	(4, 6)		$\frac{1}{2} \leq x_2$	(4, 6)
$-4 \leq -x_2$	(7)		$-4 \leq -x_2$	(7)

This is the **one-variable case** which we can solve exactly:

$$\max \left\{ -1, \boxed{\frac{1}{2}}, -\frac{5}{4}, \frac{1}{2} \right\} \leq x_2 \leq \min \left\{ 5, \frac{22}{5}, 17, \boxed{4} \right\}$$

From which we conclude: $x_2 \in [\frac{1}{2}, 4]$:-)

In General:

- The original system has a solution over \mathbb{Q} iff the system after elimination of one variable has a solution over \mathbb{Q} :-)
- Every elimination step may **square** the number of in-equations
 \implies **exponential** run-time :-((
- It can be modified such that it also decides satisfiability over \mathbb{Z}
 \implies **Omega Test**



William Worthington Pugh, Jr.
University of Maryland, College Park

Idea:

- We successively remove variables. Thereby we omit division ...
- If x only occurs with coefficient ± 1 , we apply Fourier-Motzkin elimination :-)
- Otherwise, we provide a bound for a positive multiple of x ...

Consider, e.g., (1) and (6) :

$$\begin{aligned} 6 \cdot x_1 &\leq 17 + 2x_2 \\ 9 - x_2 &\leq 4 \cdot x_1 \end{aligned}$$

W.l.o.g., we only consider **strict** in-equations:

$$\begin{aligned}6 \cdot x_1 &< 18 + 2x_2 \\8 - x_2 &< 4 \cdot x_1\end{aligned}$$

... where we always divide by gcds:

$$\begin{aligned}3 \cdot x_1 &< 9 + x_2 \\8 - x_2 &< 4 \cdot x_1\end{aligned}$$

This implies:

$$3 \cdot (8 - x_2) < 4 \cdot (9 + x_2)$$

We thereby obtain:

- If one derived in-equation is **unsatisfiable**, then also the overall system **:-)**
- If all derived in-equations are satisfiable, then there is a solution which, however, need not be **integer** **:-(**
- An integer solution is guaranteed to exist if there is **sufficient separation** between lower and upper bound ...
- Assume $\alpha < a \cdot x$ $b \cdot x < \beta$.

Then it should hold that:

$$b \cdot \alpha < a \cdot \beta$$

and moreover:

$$\boxed{a \cdot b} < a \cdot \beta - b \cdot \alpha$$

... in the Example:

$$12 < 4 \cdot (9 + x_2) - 3 \cdot (8 - x_2)$$

or:

$$12 < 12 + 7x_2$$

or:

$$0 < x_2$$

In the example, also these strengthened in-equations are satisfiable

\implies the system has a solution over \mathbb{Z} :-)

Discussion:

- If the strengthened in-equations are satisfiable, then also the original system. The reverse implication may be wrong :-(
• In the case where upper and lower bound are not sufficiently separated, we have:

$$a \cdot \beta \leq b \cdot \alpha + \boxed{a \cdot b}$$

or:

$$b \cdot \alpha < ab \cdot x < b \cdot \alpha + \boxed{a \cdot b}$$

Division with b yields:

$$\alpha < a \cdot x < \alpha + \boxed{a}$$

$$\implies \boxed{\alpha + i = a \cdot x} \text{ for some } i \in \{1, \dots, a - 1\} \quad !!!$$

Discussion (cont.):

- Fourier-Motzkin Elimination is **not** the best method for rational systems of in-equations.
- The **Omega test** is necessarily exponential :-)
If the system is **solvable**, the test generally terminates rapidly.
It may have problems with **unsolvable** systems :-(
- Also for ILP, there are other/smarter algorithms ...
- For programming language problems, however, it seems to behave quite well :-)

4. Generalization to a Logic

Disjunction:

$$\begin{aligned} & (x - 2y = 15 \quad \wedge \quad x + y = 7) \quad \vee \\ & (x + y = 6 \quad \wedge \quad 3x + z = -8) \end{aligned}$$

Quantors:

$$\exists x : \quad z - 2x = 42 \quad \wedge \quad z + x = 19$$

4. Generalization to a Logic

Disjunction:

$$\begin{aligned} & (x - 2y = 15 \quad \wedge \quad x + y = 7) \quad \vee \\ & (x + y = 6 \quad \wedge \quad 3x + z = -8) \end{aligned}$$

Quantors:

$$\exists x : \quad z - 2x = 42 \quad \wedge \quad z + x = 19$$



Presburger Arithmetic



Mojzesz Presburger, 1904–1943 (?)

Presburger Arithmetic \equiv full arithmetic
without multiplication

Presburger Arithmetic \equiv full arithmetic
without multiplication

Arithmetic : highly undecidable :-(
even incomplete :-(

Presburger Arithmetic \equiv full arithmetic
without multiplication

Arithmetic : highly undecidable :-(
even incomplete :-((

\Rightarrow Hilbert's 10th Problem

\Rightarrow Gödel's Theorem

Presburger Formulas over \mathbb{N} :

$$\begin{aligned} \phi \quad ::= & \quad x + y = z \quad | \quad x = n \quad | \\ & \quad \phi_1 \wedge \phi_2 \quad | \quad \neg \phi \quad | \\ & \quad \exists x : \phi \end{aligned}$$

Presburger Formulas over \mathbb{N} :

$$\begin{aligned} \phi \quad ::= & \quad x + y = z \quad | \quad x = n \quad | \\ & \quad \phi_1 \wedge \phi_2 \quad | \quad \neg \phi \quad | \\ & \quad \exists x : \phi \end{aligned}$$

Goal: **PSAT**

Find values for the **free** variables in \mathbb{N} such that ϕ holds ...

Idea: Code the values of the variables as Words :-)

213	t	1	0	1	0	1	0	1	1
42	z	0	1	0	1	0	1	0	0
89	y	1	0	0	1	1	0	1	0
17	x	1	0	0	0	1	0	0	0

Idea: Code the values of the variables as Words :-)

213	t	1	0	1	0	1	0	1	1
42	z	0	1	0	1	0	1	0	0
89	y	1	0	0	1	1	0	1	0
17	x	1	0	0	0	1	0	0	0

Idea: Code the values of the variables as Words :-)

213	t	1	0	1	0	1	0	1	1
42	z	0	1	0	1	0	1	0	0
89	y	1	0	0	1	1	0	1	0
17	x	1	0	0	0	1	0	0	0

Idea: Code the values of the variables as Words :-)

213	t	1	0	1	0	1	0	1	1
42	z	0	1	0	1	0	1	0	0
89	y	1	0	0	1	1	0	1	0
17	x	1	0	0	0	1	0	0	0

Idea: Code the values of the variables as Words :-)

213	t	1	0	1	0	1	0	1	1
42	z	0	1	0	1	0	1	0	0
89	y	1	0	0	1	1	0	1	0
17	x	1	0	0	0	1	0	0	0

Idea: Code the values of the variables as Words :-)

213	t	1	0	1	0	1	0	1	1
42	z	0	1	0	1	0	1	0	0
89	y	1	0	0	1	1	0	1	0
17	x	1	0	0	0	1	0	0	0

Idea: Code the values of the variables as Words :-)

213	t	1	0	1	0	1	0	1	1
42	z	0	1	0	1	0	1	0	0
89	y	1	0	0	1	1	0	1	0
17	x	1	0	0	0	1	0	0	0

Idea: Code the values of the variables as Words :-)

213	t	1	0	1	0	1	0	1	1
42	z	0	1	0	1	0	1	0	0
89	y	1	0	0	1	1	0	1	0
17	x	1	0	0	0	1	0	0	0

Idea: Code the values of the variables as Words :-)

213	t	1	0	1	0	1	0	1	1
42	z	0	1	0	1	0	1	0	0
89	y	1	0	0	1	1	0	1	0
17	x	1	0	0	0	1	0	0	0

Idea: Code the values of the variables as Words :-)

213	t	1	0	1	0	1	0	1	1
42	z	0	1	0	1	0	1	0	0
89	y	1	0	0	1	1	0	1	0
17	x	1	0	0	0	1	0	0	0

Observation:

The set of satisfying variable assignments is **regular** :-))

Observation:

The set of satisfying variable assignments is **regular** :-))

$\phi_1 \wedge \phi_2$	\Longrightarrow	$\mathcal{L}(\phi_1) \cap \mathcal{L}(\phi_2)$	(Intersection)
$\neg\phi$	\Longrightarrow	$\overline{\mathcal{L}(\phi)}$	(Complement)
$\exists x : \phi$	\Longrightarrow	$\pi_x(\mathcal{L}(\phi))$	(Projection)

Projecting away the x -component:

213	t	1	0	1	0	1	0	1	1
42	z	0	1	0	1	0	1	0	0
89	y	1	0	0	1	1	0	1	0
17	x	1	0	0	0	1	0	0	0

Projecting away the x -component:

213	t	1	0	1	0	1	0	1	1
42	z	0	1	0	1	0	1	0	0
89	y	1	0	0	1	1	0	1	0

Warning:

- Our representation of numbers is not unique: 011101 should be accepted iff every word from $011101 \cdot 0^*$ is accepted!
- This property is preserved by union, intersection and complement :-)
- It is lost by projection !!!

⇒ The automaton for projection must be enriched such that the property is re-established !!