2. One Polynomial Special Case:

\[ x \geq y + 5 \]
\[ 19 \geq x \]
\[ y \geq 13 \]
\[ y \geq x - 7 \]

- 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.
$$13 + 5 \leq 19$$
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.

$\Rightarrow$

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:

\[
\begin{align*}
9 & \leq 4x_1 + x_2 \\
4 & \leq x_1 + 2x_2 \\
0 & \leq 2x_1 - x_2 \\
6 & \leq x_1 + 6x_2 \\
-11 & \leq -x_1 - 2x_2 \\
-17 & \leq -6x_1 + 2x_2 \\
-4 & \leq -x_2
\end{align*}
\]
For $x_1$ we obtain:

\begin{align*}
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)
\end{align*}

If such an $x_1$ exists, all lower bounds must be bounded by all upper bounds, i.e.,
\[
\begin{align*}
\frac{9}{4} - \frac{1}{4}x_2 & \leq 11 - 2x_2 & (1, 5) & \quad -35 \leq -7x_2 & (1, 5) \\
\frac{9}{4} - \frac{1}{4}x_2 & \leq \frac{17}{6} + \frac{1}{3}x_2 & (1, 6) & \quad -\frac{7}{12} \leq \frac{7}{12}x_2 & (1, 6) \\
4 - 2x_2 & \leq 11 - 2x_2 & (2, 5) & \quad -7 \leq 0 & (2, 5) \\
4 - 2x_2 & \leq \frac{17}{6} + \frac{1}{3}x_2 & (2, 6) & \quad \frac{7}{6} \leq \frac{7}{3}x_2 & (2, 6) \\
\frac{1}{2}x_2 & \leq 11 - 2x_2 & (3, 5) & \quad \text{or} & \quad -22 \leq -5x_2 & (3, 5) \\
\frac{1}{2}x_2 & \leq \frac{17}{6} + \frac{1}{3}x_2 & (3, 6) & \quad -\frac{17}{6} \leq -\frac{1}{6}x_2 & (3, 6) \\
6 - 6x_2 & \leq 11 - 2x_2 & (4, 5) & \quad -5 \leq 4x_2 & (4, 5) \\
6 - 6x_2 & \leq \frac{17}{6} + \frac{1}{3}x_2 & (4, 6) & \quad \frac{19}{6} \leq \frac{19}{3}x_2 & (4, 6) \\
-4 & \leq -x_2 & (7) & \quad -4 \leq -x_2 & (7)
\end{align*}
\]
\[
\begin{align*}
\frac{9}{4} - \frac{1}{4} x_2 & \leq 11 - 2x_2 \quad (1, 5) & -5 & \leq -x_2 \quad (1, 5) \\
\frac{9}{4} - \frac{1}{4} x_2 & \leq \frac{17}{6} + \frac{1}{3} x_2 \quad (1, 6) & -1 & \leq x_2 \quad (1, 6) \\
4 - 2x_2 & \leq 11 - 2x_2 \quad (2, 5) & -7 & \leq 0 \quad (2, 5) \\
4 - 2x_2 & \leq \frac{17}{6} + \frac{1}{3} x_2 \quad (2, 6) & \frac{1}{2} & \leq x_2 \quad (2, 6) \\
\frac{1}{2} x_2 & \leq 11 - 2x_2 \quad (3, 5) & \text{or} & -\frac{22}{5} & \leq -x_2 \quad (3, 5) \\
\frac{1}{2} x_2 & \leq \frac{17}{6} + \frac{1}{3} x_2 \quad (3, 6) & -17 & \leq -x_2 \quad (3, 6) \\
6 - 6x_2 & \leq 11 - 2x_2 \quad (4, 5) & -\frac{5}{4} & \leq x_2 \quad (4, 5) \\
6 - 6x_2 & \leq \frac{17}{6} + \frac{1}{3} x_2 \quad (4, 6) & \frac{1}{2} & \leq x_2 \quad (4, 6) \\
-4 & \leq -x_2 \quad (7) & -4 & \leq -x_2 \quad (7)
\end{align*}
\]

This is the one-variable case which we can solve exactly:
\[
\max \left\{ -1, \frac{1}{2}, -\frac{5}{4}, \frac{1}{2} \right\} \leq x_2 \leq \min \left\{ 5, \frac{22}{5}, 17, 4 \right\}
\]

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

**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 $\pm 1$, we apply Fourier-Motzkin elimination :-) 
- Otherwise, we provide a bound for a positive multiple of $x$ ...

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

\begin{align*}
6 \cdot x_1 & \leq 17 + 2x_2 \\
9 - x_2 & \leq 4 \cdot x_1
\end{align*}
W.l.o.g., we only consider strict in-equations:

\[
6 \cdot x_1 \ < \ 18 + 2x_2 \\
8 - x_2 \ < \ 4 \cdot x_1
\]

... where we always divide by gcds:

\[
3 \cdot x_1 \ < \ 9 + x_2 \\
8 - x_2 \ < \ 4 \cdot x_1
\]

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:

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

\[ \implies \text{the system has a solution over } \mathbb{Z} \quad :-) \]
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:

\[ \alpha < a \cdot x < \alpha + \boxed{a} \]

\[ \Rightarrow \boxed{\alpha + i = a \cdot x} \quad \text{for some} \quad i \in \{1, \ldots, a - 1\} \quad !!! \]
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:

\[(x - 2y = 15 \land x + y = 7) \lor (x + y = 6 \land 3x + z = -8)\]

Quantors:

\[\exists x : z - 2x = 42 \land z + x = 19\]
4. Generalization to a Logic

Disjunction:

$$(x - 2y = 15 \land x + y = 7) \lor (x + y = 6 \land 3x + z = -8)$$

Quantors:

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

$\longrightarrow$ Presburger Arithmetic
Mojzesz Presburger, 1904–1943 (?)
Presburger Arithmetic  \[=\] full arithmetic without multiplication
Presburger Arithmetic = full arithmetic
without multiplication

Arithmetic : highly undecidable :-(
甚至 incomplete  :-((
Presburger Arithmetic $\equiv$ full arithmetic without multiplication

Arithmetic $\equiv$ highly undecidable $\equiv$ even incomplete $\equiv$

$\rightarrow$ Hilbert’s 10th Problem
$\rightarrow$ Gödel’s Theorem
Presburger Formulas over $\mathbb{N}$:

$\phi ::= x + y = z \mid x = n \mid \phi_1 \land \phi_2 \mid \neg \phi \mid \exists x : \phi$
Presburger Formulas over $\mathbb{N}$:

$$\phi ::= x + y = z \mid x = n \mid \phi_1 \land \phi_2 \mid \neg \phi \mid \exists x : \phi$$

Goal: PSAT

Find values for the free variables in $\mathbb{N}$ such that $\phi$ holds ...
**Idea:** Code the values of the variables as *Words*  :-)  

<p>| | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td>t</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>z</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>x</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Idea: Code the values of the variables as "Words" :-)
Idea: Code the values of the variables as Words :-)

<table>
<thead>
<tr>
<th>Value</th>
<th>Variable</th>
<th>t</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td>t</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>z</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>x</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
**Idea:** Code the values of the variables as *Words*  :-)  

<table>
<thead>
<tr>
<th>Value</th>
<th>Variable</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td>t</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td>z</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>89</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>x</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Idea: Code the values of the variables as **Words** :-)

<table>
<thead>
<tr>
<th></th>
<th>t</th>
<th>z</th>
<th>y</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>89</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
**Idea:** Code the values of the variables as *Words* :-(

<table>
<thead>
<tr>
<th>Value</th>
<th>Variable</th>
<th>t</th>
<th>z</th>
<th>y</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>89</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
**Idea:** Code the values of the variables as *Words* :-)

<table>
<thead>
<tr>
<th></th>
<th>t</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>89</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Idea: Code the values of the variables as **Words** :-) 

<table>
<thead>
<tr>
<th></th>
<th>t</th>
<th>z</th>
<th>y</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td>1 0 1 0 1 0 1 1</td>
<td>0 1 0 1 0 1 0 0</td>
<td>1 0 0 1 1 0 1 0</td>
<td>1 0 0 0 1 0 0 0</td>
</tr>
</tbody>
</table>
**Idea:** Code the values of the variables as *Words*  :-)

<table>
<thead>
<tr>
<th>Value</th>
<th>Variable</th>
<th>t</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td>t</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>z</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>89</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>x</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Idea: Code the values of the variables as Words 

<table>
<thead>
<tr>
<th></th>
<th>t</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Observation:

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

The set of satisfying variable assignments is regular  \( :-) \)

\[
\begin{align*}
\phi_1 \land \phi_2 & \implies \mathcal{L}(\phi_1) \cap \mathcal{L}(\phi_2) \quad \text{(Intersection)} \\
\neg \phi & \implies \mathcal{L}(\phi) \quad \text{(Complement)} \\
\exists x : \phi & \implies \pi_x(\mathcal{L}(\phi)) \quad \text{(Projection)}
\end{align*}
\]
Projecting away the $x$-component:

<table>
<thead>
<tr>
<th></th>
<th>t</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>z</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>89</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>x</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Projecting away the $x$-component:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>213</td>
<td>t</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>z</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>89</td>
<td>y</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
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 !!
Automata for Basic Predicates:

\[ x = 5 \]
Automata for Basic Predicates:

\[ x + x = y \]
Automata for Basic Predicates:

\[ x + y = z \]