Discussion:

- Solutions only matter within the bounds to the iteration variables.
- Every integer solution there provides a conflict.
- Fusion of loops is possible if no conflicts occur.
- The given special cases suffice to solve the case of two variables over \( \mathbb{Q} \) and of one variable over \( \mathbb{Z} \).
- The number of variables in the inequations corresponds to the nesting-depth of for-loops in general, is quite small.

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

\[
2x + 3y = 24 \\
x - y + 5z = 3
\]
Discussion:

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

  $\sum_{i=1}^{n} a_i \cdot x_i = b$ (or $\sum_{i=1}^{n} a_i \cdot x_i \geq b$), $a_i \in \mathbb{Z}$

- 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**

   \[
   2x + 3y = 24 \\
   x - y + 5z = 3
   \]

---

**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
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:

\[
2x + 3y = 24 \\
x - y + 5z = 3
\]

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

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:

\[
\begin{align*}
2x + 3y &= 24 \\
x - y + 5z &= 3
\end{align*}
\]

\[\Rightarrow\]

\[
\begin{align*}
5y - 10z &= 18 \\
x - y + 5z &= 3
\end{align*}
\]

Observation 3:

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

\[
\begin{bmatrix}
1 & 0 & 0 & | & 2y - 10z &= 18 \\
0 & 1 & 0 & | & x - y + 5z &= 3 \\
0 & 0 & 1 & | & x - y + 3z &= 3
\end{bmatrix}
\]

\[\Rightarrow\]

\[
\begin{bmatrix}
1 & 0 & 0 & | & 5y &= 18 \\
0 & 1 & 0 & | & x - y + 3z &= 3 \\
0 & 0 & 1 & | &
\end{bmatrix}
\]

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:-))
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:**

\[
\begin{align*}
19 & \geq x \\
y & \geq 13 \\
y & \geq x - 7
\end{align*}
\]

- There are at most 2 variables per inequality;
- No scaling factors.

**Idea:** Represent the system by a graph:

---

**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.

---

**Idea:** Represent the system by a **graph**:

![Graph diagram]

\[15 + 5 = 20 \leq 19\]
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 ;-)}
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 $:-)$. 

**Example:**

\[
\begin{align*}
9 & \leq 4x_1 + x_2 & (1) \\
4 & \leq x_1 + 2x_2 & (2) \\
0 & \leq 2x_1 - x_2 & (3) \\
6 & \leq x_1 + 6x_2 & (4) \\
-11 & \leq -x_1 - 2x_2 & (5) \\
-17 & \leq -6x_1 + 2x_2 & (6) \\
-4 & \leq -x_2 & (7)
\end{align*}
\]
For \( x_1 \) we obtain:

\[
\begin{align*}
0 & \leq 4x_1 + x_2 \quad (1) \\
\frac{9}{4} - \frac{1}{4}x_2 & \leq x_1 \quad (1) \\
4 & \leq x_1 + 2x_2 \quad (2) \\
4 - 2x_2 & \leq x_1 \quad (2) \\
0 & \leq 2x_1 - x_2 \quad (3) \\
\frac{1}{2}x_2 & \leq x_1 \quad (3) \\
6 & \leq x_1 + 6x_2 \quad (4) \\
6 - 6x_2 & \leq x_1 \quad (4) \\
-11 & \leq -x_1 - 2x_2 \quad (5) \\
x_1 & \leq 11 - 2x_2 \quad (5) \\
-17 & \leq -6x_1 + 2x_2 \quad (6) \\
x_1 & \leq \frac{17}{6} + \frac{1}{3}x_2 \quad (6) \\
-4 & \leq -x_2 \quad (7) \\
-4 & \leq -x_2 \quad (7)
\end{align*}
\]

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

\[
\max \{ -1, \frac{1}{3}, -\frac{5}{2}, -\frac{1}{2} \} \leq x_2 \leq \min \{ 5, \frac{37}{6}, 17, \frac{1}{12} \}
\]

From which we conclude: \( x_2 \in \left[ \frac{1}{3}, 4 \right] \quad \therefore \)

**In General:**

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

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

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

\[
\begin{align*}
6 \cdot x_1 & \leq 17 + 2x_2 \\
9 - x_2 & \leq 4 \cdot x_1
\end{align*}
\]

For $x_1$ we obtain:

\[
\begin{align*}
0 & \leq 4x_1 + x_2 & (1) & \quad \frac{9}{4} - \frac{1}{2}x_2 & \leq x_1 & (1) \\
4 & \leq x_1 + 2x_2 & (2) & \quad 4 - 2x_2 & \leq x_1 & (2) \\
0 & \leq 2x_1 - x_2 & (3) & \quad \frac{1}{2}x_2 & \leq x_1 & (3) \\
6 & \leq x_1 + 6x_2 & (4) & \quad 6 - 6x_2 & \leq x_1 & (4) \\
-11 & \leq -x_1 - 2x_2 & (5) & \quad x_1 & \leq 11 - 2x_2 & (5) \\
-17 & \leq -6x_1 + 2x_2 & (6) & \quad x_1 & \leq \frac{17}{6} + \frac{1}{3}x_2 & (6) \\
-4 & \leq -x_2 & (7) & \quad -4 & \leq -x_2 & (7)
\end{align*}
\]

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

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

\[
\begin{align*}
6 \cdot x_1 & < 18 + 2x_2 \\
8 - x_2 & < 4 \cdot x_1
\end{align*}
\]

... where we always divide by gcds:

\[
\begin{align*}
3 \cdot x_1 & < 9 + x_2 \\
8 - x_2 & < 4 \cdot x_1
\end{align*}
\]

This implies:

\[
3 \cdot (8 - x_2) < 4 \cdot (9 + x_2)
\]
W.l.o.g., we only consider strict in-equations:

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

... where we always divide by geds:

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

This implies:

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

\[\eta_5\]

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*}

\[\eta_4\]

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:

\[a \cdot b < a \cdot \beta - b \cdot \alpha\]
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 $\alpha < a \cdot x$ and $b \cdot x < \beta$.

Then it should hold that:

\[ b \cdot \alpha < a \cdot \beta \]

and moreover:

\[ \frac{a - b}{b} < a \cdot x - b \cdot \alpha \]

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:

\[ a \cdot \beta \leq b \cdot \alpha + a \cdot b \]

or:

\[ a \cdot \alpha < a \cdot x < \alpha + a \cdot \beta \]

Division with $b$ yields:

\[ \alpha < a \cdot x < \alpha + \frac{a}{b} \]

for some $i \in \{1, \ldots, a - 1\}$.
Discussion (cont.):

→ Fourier-Motzkin Elimination is not the best method for rational systems of inequalities.

→ 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 :-(

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:

\[ b \cdot \alpha < ab \cdot x < b \cdot \alpha + a \cdot b \]

or:

\[ b \cdot \alpha < ab \cdot x < b \cdot \alpha + a \cdot \alpha \]

Division with \( b \) yields:

\[ \alpha < a \cdot x < \alpha + \alpha \]

\[ \Rightarrow \alpha + i = a \cdot x \] for some \( i \in \{\ldots, \alpha\} \) !!!
4. Generalization to a Logic

Disjunction:

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

Quantors:

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

\[\implies \text{Presburger Arithmetic}\]
Mojzesz Presburger, 1904–1943 (?)

**Presburger Arithmetics**

- **Full arithmetic without multiplication**
- **Arithmetic**: highly undecidable, even incomplete
- **Hilbert’s 10th Problem**
- **Gödel’s Theorem**

**Presburger Formulas over \( \mathbb{N} \):**

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

\[
\forall x, y \varphi \equiv \neg \forall (\exists x, y \varphi)
\]

**Goal:**

PSAT

Find values for the free variables in \( \mathbb{N} \) such that \( \phi \) holds ...
Presburger Formulas over $\mathbb{N}$:

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

**Goal:** PSAT

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

---

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

```
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 \( \Rightarrow \)

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

Approximation of Horn Clauses

Step 1:

Simplification of pre-conditions by splitting, simplification and guard simplification (as before \( \Rightarrow \))

Step 2:

Introduction of copies of variables \( X \). Every copy receives all literals of \( X \) as pre-condition.

\[
\begin{align*}
p(f(X, X)) & \leftarrow q(X) \quad \text{yields} \\
p(f(X, X')) & \leftarrow q(X), q(X')
\end{align*}
\]

Projects away the \( x \)-component:

<table>
<thead>
<tr>
<th>213</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>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>

Assume \( A = \{p_1, \ldots, p_m\} \). Then:

\[
\begin{align*}
[A](b) & \leftarrow \text{whenever } p_i(b) \leftarrow \text{ for all } i. \\
[A](f(X_1, \ldots, X_k)) & \leftarrow [B_i](X_1), \ldots, [B_k](X_k) \\
& \text{whenever } B_i = \{p_{ij} \mid X_{ij} = X_i\} \text{ for } \ 
p_{ij}(f(X_1, \ldots, X_k)) & \leftarrow p_{ij1}(X_{ij1}), \ldots, p_{ijk}(X_{ijk})
\end{align*}
\]

Warning:
Automata for Basic Predicates:

\[ x = 5 \]

![Automata Diagram](image-url)