Satisfiability Procedure for Arithmetic

- Symbols: \( n, +, -, \geq, \leq, = \)
- Like quantifier-free Presburger arithmetic.
- The sat. problem is to determine the set of a system of linear inequalities.
- Equivalent to the linear programming problem \( \rightarrow \) known to be in \( P \)
- If we add the constraint that \( v \) variables range over integers \( \rightarrow \) integer programming \( \in \text{NP} \) (NP complete)
  \( \rightarrow \) in practice LP algorithms are sound approximations of IP algorithms.
- Pratt observed that most inequalities arising in program verification are of the form \( x - y \leq c \) or \( x \leq c \) or \( y \geq c \)
- There is a simple algorithm for this case.
  Then we will look at Shostak’s generalization of Pratt’s algorithm.
Pratt's algorithm

Let $C$ be a satisfiable set of inequalities $x_i - x_j \leq c$

Note: To handle $x \leq c$ and $x \geq c$ we introduce a variable $z$ (to stand for 0) and we rewrite $x \leq c$ to $x - z \leq c$ and $x \geq c$ to $z - x \leq c$

Claim: the satisfiability of a set of constraints is preserved by this move.

Proof: if $\Psi$ is a sat. interpretation for $C$ then

$\Psi'(x) = \Psi(x) \lor 0$ is another sat. interp. (for any constant $a$)

Think of $C$ representing a directed graph whose nodes are labelled with variables

For $x_i - x_j \leq c$

add an edge $\overrightarrow{x_i \rightarrow x_j}^c$

Let $S_{ij} = \begin{cases} \text{length of the shortest path from } x_i \text{ to } x_j \\ \infty \text{ if no path exists} \end{cases}$

$S_{ij}$ is well defined if only if there are no negative-weight cycles in the graph.
Assume there is a negative-weight cycle.

\[ x_1 - x_2 \leq c_1 \]
\[ x_2 - x_3 \leq c_2 \]
\[ \vdots \]
\[ x_n - x_1 \leq c_n \]

0 \leq c_1 + c_2 + \ldots + c_n < 0

This means that C is not satisfiable.

Thus, satisfiability can be decided by finding negative-weight cycles.

See Bellman-Ford’s algorithm that runs in \( O(|v| \cdot |e| + |v| \cdot |c|) \).

We need to modify slightly the algorithm to make it incremental.

- undoable
- detect all equalities between variables.
- and produce proofs.

First an important lemma.

If C is satisfiable then

\[ \delta_{ij} = \max_{\psi} \psi(x_i - x_j) \]

(defined as \( \infty \) if no such maximum)
Proof

for any \( \psi \vdash C \)  \( (\psi \text{ is a sat. interp. for } C) \)

\[ \psi(x_i-x_j) \leq \delta_{ij} \]

(take the path from \( x_i \) to \( x_j \) and add all the constraints. Get \( C \Rightarrow x_i-x_j \leq \delta_{ij} \))

Thus

\[ \max_{\psi \vdash C} \psi(x_i-x_j) \leq \delta_{ij} \]

- Now we must show that there exists a sat. interp. \( \psi \) such that \( \psi(x_i-x_j) = \delta_{ij} \)

Define \( \psi(x_k) = \psi(x_j) + \delta_{k,j} \) for all \( k \) such that \( \delta_{k,j} < \infty \)

\( \psi \) satisfies \( C \). Take \( x_e-x_m \leq a \in C \)

\[ \psi(x_e-x_m) = \delta_{e,j} - \delta_{m,j} \leq \delta_{e,m} \leq a \]

\[ \text{triangle inequality} \]

\[ \text{a is the length of one path from } e \text{ to } m \]

\[ \text{we still need to consider the case when} \]

- \( \delta_{ij} = \infty \). We must show that \( \psi(x_i-x_j) \) can be arbitrarily large

- \( x_e \) or \( x_m \) are not predecessors of \( j \) in the graph
Lemma 2

- If $C$ is satisfiable and $d_{ij}$ are the shortest path length then

$$C \land x_i - x_j \leq a \text{ is sat } \iff d_{ji} + a \geq 0$$

(This means that we can incrementally check satisfiability)

Proof. $d_{ji} + a < 0$ But $C \Rightarrow x_j - x_i \leq d_{ji}$

Thus $C \land x_i - x_j \leq a \Rightarrow 0 \leq d_{ji} + a$

Now assume $C \land x_i - x_j \leq a$ is not sat

$C \Rightarrow x_i - x_j > a$

$C \Rightarrow x_j - x_i < -a$

But $\max_{x \in \mathcal{E}} \psi(x_j - x_i) = d_{ji}$ \quad \Rightarrow \quad d_{ji} < -a$

Lemma 3

- If $C$ is satisfiable then

$C \Rightarrow x_i = x_j$ if $d_{ij} = d_{ji} = 0.$

Proof easy using the max interpretation of $d_{ij}$.

(This gives us an easy way to detect all equalities)
Lemma 4

If $C$ is satisfiable and $C \land x_i = x_j \land a$ is satisfiable then

$$d_{ki}^{' min} = \min (d_{ke}, d_{ki} + a + d_{je})$$

Proof: simple given the shortest-path interpretation of $d_{ke}$

(This means that an incremental step has an easy way to recompute $S$)

$\Rightarrow$ complexity $O(n^2)$ for each step

Lemma 5

$C$ is satisfiable in integers

$C$ is satisfiable in reals

Easy based on path interpretation of $d_{ij}$

This is a special case when a SAT proc for $R$ is also one for $\mathbb{N}$

$!!$ But the theory is only convex in $\mathbb{R} !!$
Algorithm

- we use an undoStack to allow undo
- we use a data structure to store $d_{ij}$
  (sparse array with easy access to the line and column of $i$)
- we use a data structure $P_{ij}$ with invariant

  \[
  \text{if } d_{ij} < \infty \text{ then } \\
  P_{ij} = (x_k - x_e \leq a, \text{pref}) \text{ such that } \\
  d_{ij} = d_{ik} + a + d_{ej} \text{ and } \\
  \text{pref: pref} (x_k - x_e \leq a)
  \]

  ($P_{ij}$ tells us that the shortest path from $i$ to $j$ passes through $k$ and $l$)
assert \((x_i-x_j \leq a, \text{prf})\)
  - addNode \(x_i\) and \(x_j\) if necessary
  - if \(d_{ij} \leq a\) then return
  - if \(d_{ji} + a < 0\) then
    raise Contra \((\text{mkPrattCondPrf}(i, j, a, \text{prf}))\)
  - for each \(k \in \text{Column}(i), l \in \text{Line}(j)\)
    if \(d_{ke} > d_{ki} + a + d_{ji}\) then
      push \((k, l, d_{ke}, \text{Prf})\) on the undoStack
      \(d_{ke} \leftarrow d_{ki} + a + d_{ji}\)
      \(\text{Prf} \leftarrow (x_i-x_j \leq a, \text{prf})\)

\[=\]

\(\text{acc} \leftarrow \text{nil}\)
  - for each \(k, l\)
    if \(d_{ke} = d_{lk} = 0\) then
      \(\text{acc} = \text{acc} \cup \{(x_k=x_l, \text{mkPrattEqPrf}(k, l))\}\)

  - return \(\text{acc}\)

addNode \(x_i\)
  - set \(d_{ii} = 0\), \(d_{ij} = d_{ji} = \infty\) for \(j \neq i\)
  - push \((\text{add } x_i)\) on the undoStack

undo
  pop \((\text{add } x_i)\) from undoStack
  remove \(x_i\)
  pop \((k, l, d_{ke}, \text{Prf})\) from undoStack
  \(d_{ke} \leftarrow a\)
  \(\text{Prf} \leftarrow \text{Prf}\).
Example with Pratt

Consider the set of inequalities:

\[ x \geq u, \quad y \leq 0, \quad u + 3 \geq x, \quad x + 1 \leq y, \quad u + 1 \geq 0 \quad v + 2 \geq 0 \quad v - 3 \leq u - 2 \]

After asserting \[ x \geq u, \quad y \leq 0, \quad u + 3 \leq x \] we have

\[
\begin{array}{cccc}
\times & u & y & z \\
\hline
x & 0 & \theta & 0 \\
u & 0 & 0 & 0 \\
y & 0 & 0 & 0 \\
z & 0 & 0 & 0 \\
\end{array}
\]

After asserting \[ x + 1 \leq y \] (OK, since \( \delta_{yx} + 1 \geq 0 \))

\[
\begin{array}{cccc}
\times & u & y & z \\
\hline
x & 0 & 3 & -1 & -1 \\
u & 0 & 0 & -1 & -1 \\
y & 0 & 0 & 0 & 0 \\
z & 0 & 0 & 0 & 0 \\
\end{array}
\]
After asserting $u + 1 \geq 0$ (Ok since $5u_2 + 1 \geq 0$)

\[
\begin{array}{cccc}
\text{X} & \text{M} & \text{Y} & \text{Z} \\
\hline
\text{X} & 0 & 0 & -1 & -1 \\
\text{Y} & 0 & 0 & -1 & -1 \\
\text{Z} & 1 & 1 & 0 & 0 \\
\text{V} & 1 & 1 & 0 & 0 \\
\end{array}
\]

Detects equalities $x = u$ and $y = z$

After asserting $v + 2 \geq 0$ (Ok since $5v_2 + 2 \geq 0$)

\[
\begin{array}{cccc}
\text{X} & \text{U} & \text{Y} & \text{Z} & \text{V} \\
\hline
\text{X} & 0 & 0 & -1 & -1 & 1 \\
\text{U} & 0 & 0 & -1 & -1 & 1 \\
\text{Y} & 1 & 1 & 0 & 0 & 0 \\
\text{Z} & 1 & 1 & 0 & 0 & 2 \\
\text{V} & 1 & 1 & 0 & 0 & 2 \\
\end{array}
\]

Now try to add $x \leq u - 2$.

Contrary since $5u_2 + -2 < 0$