Satisfiability Procedure for Arithmetic

- Symbols: $n, +, -, \geq, >, =$
- Like quantifier-free Presburger arithmetic.
- The sat. problem is to determine the sat. of a system of linear inequalities
- Equivalent to the linear programming problem
  → known to be in P
- If we add the constraint that ~~some~~ variables range over integers → integer programming → in NP (NP complete)
  → in practice LP algorithms are sound approximations of IP algorithms
- Pratt observed that most inequalities arising in program verification are of the form

  $$x - y \leq c \quad \text{or} \quad x \leq c \quad \text{or} \quad y \geq c$$

- 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 $x_i - x_j \leq c$

<u>Note</u> 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 $z - x \leq -c$

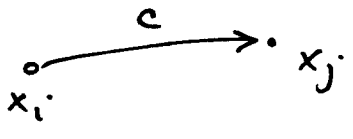
<u>Claim</u>: the satisfiability of a set of constraints is preserved by this move.

<u>Proof</u>: if ψ is a sat. interpretation for C then $\psi'(x) = \psi(x) + a$ 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.



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

δ_{ij} is well defined if only if there are no negative-weight cycles in the graph.

Assume there is a negative-weight cycle.

$$x_1 - x_2 \le c_1$$
$$x_2 - x_3 \le c_2$$
$$\vdots$$
$$x_n - x_1 \le c_n$$

$$0 \le c_1 + c_2 + \dots + c_n < 0$$

means that C is not satisfiable

Thus satisfiability can be decided by finding negative-weight cycles

See Bellman-Ford's algorithm that runs in $O(|\text{nr vars}| \cdot |\text{nr constraints}|)$

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

$$\delta_{ij} = \max_{\psi \Vdash C} \psi(x_i - x_j)$$

(defined as ∞ if no such maximum)

Proof

- for any $\psi \Vdash C$ (ψ 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. ψ such that $\psi(x_i - x_j) = \delta_{ij}$.

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

ψ satisfies C. Take $x_\ell - x_m \leq a \in C$

$$\psi(x_\ell - x_m) = \delta_{\ell j} - \delta_{mj} \leq \delta_{\ell m} \leq a$$

($\delta_{\ell j} - \delta_{mj} \leq \delta_{\ell m}$: triangle inequality; $\delta_{\ell m} \leq a$: a is the length of one path from ℓ to m)

→ 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_ℓ, or x_m are not predecessors of j in the graph

Lemma 2

- If C is satisfiable and δ_{ij} are the shortest path length

then

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

(This means that we can incrementally check satisfiability)

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

Thus $C \wedge x_i - x_j \leq a \Rightarrow 0 \leq \delta_{ji} + a$

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

$$C \Rightarrow x_i - x_j > a$$

$$C \Rightarrow x_j - x_i < -a$$

But $\max_{\psi \models C} \psi(x_j - x_i) = \delta_{ji}$ $\Big\} \Rightarrow \delta_{ji} < -a$

Lemma 3

- If C is satisfiable then

$$C \models x_i = x_j \quad \text{iff} \quad \delta_{ij} = \delta_{ji} = 0.$$

Prof easy using the max interpretation of δ_{ij}.

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

Lemma 4

If $C \underbrace{\sim}_{\delta}$ is satisfiable and $\overbrace{C \wedge x_i - x_j \leq a}^{\delta'}$ is satisfiable

then

$$\delta'_{kl} = \min(\delta_{kl}, \delta_{ki} + a + \delta_{jl})$$

Proof. simple given the shortest-path interpretation of δ_{kl}

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

→ complexity $O(n^2)$ for each step

Lemma 5

~~If~~ C is satisfiable in integers

iff

C is satisfiable in reals

- easy based on path interpretation of δ_{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 R !!

Algorithm

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

  if $\delta_{ij} < \infty$ then

  $P_{ij} = (x_k - x_\ell \leq a \ , \ prf)$ such that

  $\delta_{ij} = \delta_{ik} + a + \delta_{\ell j}$ and

  prf : pf $(x_k - x_\ell \leq a)$

  (P_{ij} tells us that the shortest path from i to j passes through k and ℓ)

assert $(x_i - x_j \leq a, prf)$

- addNode x_i and x_j if necessary
- if $\delta_{ij} \leq a$ then return
- if $\delta_{ji} + a < 0$ then
  raise Contra (mkPrattContraPrf (i, j, a, prf))
- for each $k \in$ Column(i), $l \in$ Line(j)
  if $\delta_{kl} > \delta_{ki} + a + \delta_{jl}$ then
  push $(k, l, \delta_{kl}, P_{kl})$ on the undoStack
  $\delta_{kl} \leftarrow \delta_{ki} + a + \delta_{jl}$
  $P_{kl} \leftarrow (x_i - x_j \leq a, prf)$

□

acc $\leftarrow$ nil

- for each k, l
  if $\delta_{kl} = \delta_{lk} = 0$ then
  acc = acc $\cup$ $\{(x_l = x_k, \text{mkPrattEqPrf}(k, l))\}$
- return acc

addNode x_i

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

undo

pop (add x_i) from undoStack
  - remove x_i

pop (k, l, δ_{kl}, p) from undoStack
  $\delta_{kl} \leftarrow a$
  $P_{kl} \leftarrow p$.

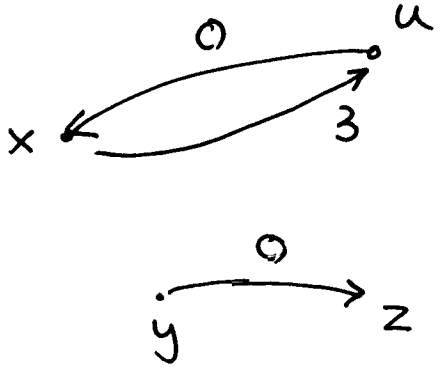
Example with Pratt

Consider the set of inequalities

$x \geq u$, $y \leq 0$, $u+3 \geq x$, $x+1 \leq y$, $u+1 \geq 0$

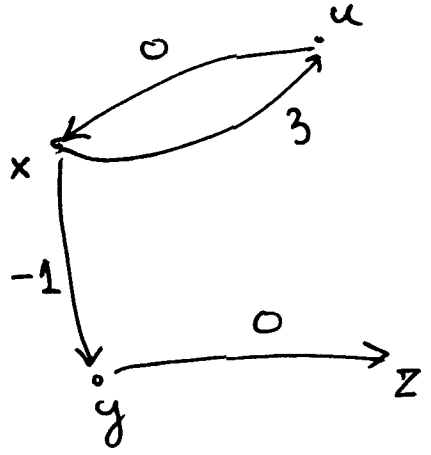
$v+2 \geq 0$ $v \leq u-2$

After asserting $x \geq u$, $y \leq 0$, $u+3 \leq x$ we have



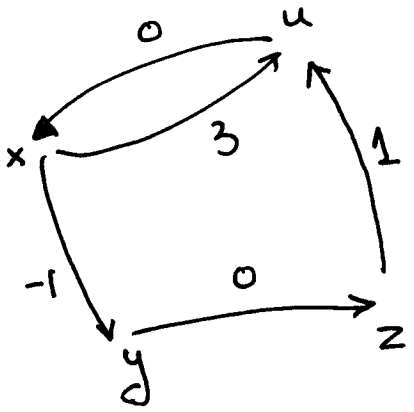
	x	u	y	z
x	0	0		
u	0	0		
y			0	0
z				0

After asserting $x+1 \leq y$ (Ok, since $\delta_{yx}+1 \geq 0$)



	x	u	y	z
x	0	3	-1	-1
u	0	0	-1	-1
y			0	0
z				0

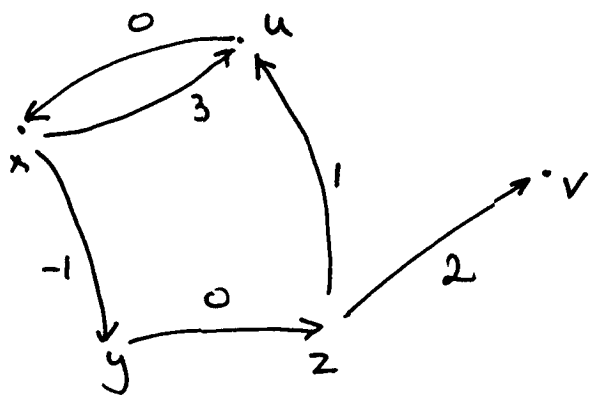
After asserting $u+1 \geq 0$ (Ok since $\delta_{uz}+1 \geq 0$)



	x	u	y	z
x	0	$\not\geq 0$	-1	-1
u	0	0	-1	-1
y	1	1	0	0
z	1	1	0	0

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

After asserting $v+2 \geq 0$ (Ok since $\delta_{vz}+2 \geq 0$)



	x	u	y	z	v
x	0	0	-1	-1	1
u	0	0	-1	-1	1
y	1	1	0	0	2
z	1	1	0	0	2
v					0

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

Contra since $\delta_{uv}+-2 < 0$