universität innsbruck

WS 2021 lecture 12

Constraint Solving

Cezary Kaliszyk René Thiemann

based on a previous course by Aart Middeldorp

Outline

1. **Summary of Previous Lecture**
2. **Cutting Planes**
3. **Difference Logic**
4. **Further Reading**

Example (Application of Linear Integer Arithmetic: Termination Proving)

- consider program
- model loop-iteration as formula φ using pre-variables $\vec{x}$ and post-variables $\vec{x}'$
- prove termination by choosing expression e and integer constant c and show that two LIA problems are unsatisfiable
  - $\varphi \wedge e(\vec{x}) < e(\vec{x}') + 1$ (¬ decrease)
  - $\varphi \wedge e(\vec{x}') < c$ (¬ bounded)
- for certain programs, reasoning over integers is essential

Branch-and-Bound Algorithm

- core idea for finding integral solution
  - simplex algorithm is used to find rational solution v or detect unsat in $\mathbb{Q}$
  - whenever $q := v(x) \notin \mathbb{Z}$, add $x \leq \lfloor q \rfloor \vee \lceil q \rceil \leq x$ and consider both possibilities
- small model property is required for termination: obtain finite search space

Theorem (Small Model Property)

if LIA formula ψ has solution over $\mathbb{Z}$ then it has a solution v with

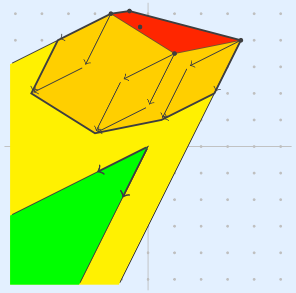
$$|v(x)| \leq bound(\psi) := (n+1)! \cdot c^n$$

for all x where

- *n: number of variables in ψ*
- *c: maximal absolute value of numbers in ψ*

Proof Idea of Small Model Property

1. convert conjunctive LIA formula ψ into form $A\vec{x} \leq \vec{b}$
2. represent polyhedron $\underbrace{\{\vec{x} \mid A\vec{x} \leq \vec{b}\}}_{\textit{yellow}}$ as polyhedron $P = \underbrace{hull(X)}_{\textit{red}} + \underbrace{cone(V)}_{\textit{green}}$
3. show that P has small integral solutions (orange), depending on X and V
4. approximate entries of vectors in X and V to obtain small model property



Outline

1. Summary of Previous Lecture

2. **Cutting Planes**

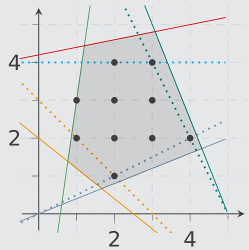
3. Difference Logic

4. Further Reading

Limitations of Branch-and-Bound

- global bounds for solution can be derived from formula, but are often too high for efficient practical procedures
- shape of case analysis is quite restricted: $x \leq c \vee c+1 \leq x$ for some variable x and $c \in \mathbb{Z}$
- $\Longrightarrow$ use cutting planes to restrict solution space more effectively

Example



Definition (Cut)

given solution v to problem over $\mathbb{R}^n$, **cut** is inequality $a_1x_1 + \cdots + a_nx_n \leq b$ which is not satisfied by v but by every $\mathbb{Z}^n$-solution

Method

like in Branch-and-Bound, keep adding cuts until integer solution found

Gomory Cuts: Assumptions

- DPLL(T) simplex returned solution v to

$$A\vec{x}_N = \vec{x}_B \qquad (1) \qquad\qquad -\infty \leq l_i \leq x_i \leq u_i \leq +\infty \qquad (2)$$

- for some $i \in B$ variable x_i is assigned $v(x_i) \notin \mathbb{Z}$
- for all $j \in N$ value $v(x_j)$ is l_j or u_j

Notation

- write $c = v(x_i) - \lfloor v(x_i) \rfloor$
- by assumption all nonbasic variables are assigned bounds, so we can split

$$L = \{\, j \in N \mid v(x_j) = l_j \,\} \qquad U = \{\, j \in N \mid v(x_j) = u_j \,\} \setminus L$$

$$L^+ = \{\, j \in L \mid A_{ij} \geq 0 \,\} \qquad U^+ = \{\, j \in U \mid A_{ij} \geq 0 \,\}$$

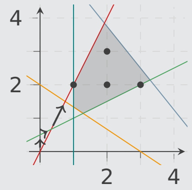
$$L^- = \{\, j \in L \mid A_{ij} < 0 \,\} \qquad U^- = \{\, j \in U \mid A_{ij} < 0 \,\}$$

Lemma (Gomory Cut)

cut is given by inequality

$$\frac{1}{1-c} \cdot \sum_{j \in L^+} A_{ij}(x_j - l_j) - \frac{1}{1-c} \cdot \sum_{j \in U^-} A_{ij}(u_j - x_j) - \frac{1}{c} \cdot \sum_{j \in L^-} A_{ij}(x_j - l_j) + \frac{1}{c} \cdot \sum_{j \in U^+} A_{ij}(u_j - x_j) \geq 1$$

Example



$$-2x - 3y \leq -6$$
$$-2x + y \leq 0$$
$$x - 2y \leq -1$$
$$5x + 4y \leq 25$$

- infinite $\mathbb{R}^2$-solution space
- four solutions in $\mathbb{Z}^2$
- Simplex solution search

$$\begin{array}{c|cc} & x & y \\ s_1 & -2 & -3 \\ s_2 & -2 & 1 \\ s_3 & 1 & -2 \\ s_4 & 5 & 4 \end{array} \quad \begin{array}{l} s_1 \leq -6 \\ s_2 \leq 0 \\ s_3 \leq -1 \\ s_4 \leq 25 \end{array} \quad \longrightarrow \quad \begin{array}{c|cc} & s_2 & s_1 \\ s_3 & -\frac{7}{8} & \frac{3}{8} \\ x & -\frac{3}{8} & -\frac{1}{8} \\ y & \frac{1}{4} & -\frac{1}{4} \\ s_4 & -\frac{7}{8} & -\frac{13}{8} \end{array} \quad \begin{array}{ll} x = \frac{3}{4} & s_1 = -6 \\ y = \frac{3}{2} & s_2 = 0 \\ & s_3 = -2\frac{1}{4} \\ & s_4 = 9\frac{3}{4} \end{array}$$

initial tableau — final tableau — solution

- nonbasic variables $s_2 = 0$ and $s_1 = -6$ at bounds, basic x is assigned $\frac{3}{4} \notin \mathbb{Z}$
- from $c = \frac{3}{4}$ obtain Gomory cut $1/(1 - \frac{3}{4}) \cdot (\frac{3}{8}(0 - s_2) + \frac{1}{8}(-6 - s_1)) \geq 1$
- corresponds to $-3(-2x + y) - (-2x - 3y) \geq 8$, simplified $x \geq 1$

$$A\vec{x}_N = \vec{x}_B \qquad (1) \qquad\qquad -\infty \leq l_i \leq x_i \leq u_i \leq +\infty \qquad (2)$$

Proof (1)

- consider potential integer solution $\vec{x}$ to (1) and (2)
- $\vec{x}$ satisfies i-th row of (1):

$$x_i = \sum_{j \in N} A_{ij} x_j \qquad (3)$$

- because v is solution have

$$v(x_i) = \sum_{j \in N} A_{ij} v(x_j) \qquad (4)$$

- subtract (4) from (3):

$$x_i - v(x_i) = \sum_{j \in N} A_{ij}(x_j - v(x_j))$$

$$= \sum_{j \in L} A_{ij}(x_j - l_j) - \sum_{j \in U} A_{ij}(u_j - x_j) \qquad (5)$$

Proof (2)

- have

$$x_i - v(x_i) = \underbrace{\sum_{j \in L} A_{ij}(x_j - l_j)}_{\mathcal{L}} - \underbrace{\sum_{j \in U} A_{ij}(u_j - x_j)}_{\mathcal{U}} \qquad (5)$$

- for $c = v(x_i) - \lfloor v(x_i) \rfloor$ have $0 < c < 1$, can write $v(x_i) = \lfloor v(x_i) \rfloor + c$, so

$$x_i - \lfloor v(x_i) \rfloor = c + \mathcal{L} - \mathcal{U} \qquad (6)$$

- for integer solution $\vec{x}$ left-hand side must be integer, so also right-hand side
- abbreviate

$$\mathcal{L}^+ = \sum_{j \in L^+} A_{ij}(x_j - l_j) \qquad \mathcal{U}^+ = \sum_{j \in U^+} A_{ij}(u_j - x_j)$$

$$\mathcal{L}^- = \sum_{j \in L^-} A_{ij}(x_j - l_j) \qquad \mathcal{U}^- = \sum_{j \in U^-} A_{ij}(u_j - x_j)$$

  so $\mathcal{L} = \mathcal{L}^+ + \mathcal{L}^-$ and $\mathcal{U} = \mathcal{U}^+ + \mathcal{U}^-$
- have $\mathcal{L}^+ \geq 0, \mathcal{U}^+ \geq 0$ and $\mathcal{L}^- \leq 0, \mathcal{U}^- \leq 0$
- distinguish $\mathcal{L} \geq \mathcal{U}$ or $\mathcal{L} < \mathcal{U}$

Proof (3)

- both sides are integer in equation

$$x_i - \lfloor v(x_i) \rfloor = c + \mathcal{L} - \mathcal{U} \quad (6)$$

since $\mathcal{L}^+ \geq \mathcal{L}$ and $\mathcal{U}^- \leq \mathcal{U}$

- if $\mathcal{L} \geq \mathcal{U}$
  - have $c + \mathcal{L} - \mathcal{U} \geq 1$ because integer, so $\mathcal{L} - \mathcal{U} \geq 1 - c$
  - in particular $\mathcal{L}^+ - \mathcal{U}^- \geq 1 - c$
  - $$\frac{1}{1-c}(\mathcal{L}^+ - \mathcal{U}^-) \geq 1 \quad (7)$$

since $\mathcal{U}^+ \geq \mathcal{U}$ and $\mathcal{L}^- \leq \mathcal{L}$

- if $\mathcal{L} < \mathcal{U}$
  - have $c + \mathcal{L} - \mathcal{U} \leq 0$ because integer, so $\mathcal{U} - \mathcal{L} \geq c$
  - in particular $\mathcal{U}^+ - \mathcal{L}^- \geq c$
  - $$\frac{1}{c}(\mathcal{U}^+ - \mathcal{L}^-) \geq 1 \quad (8)$$

- terms $\mathcal{L}^+, \mathcal{U}^+, -\mathcal{L}^-$ and $-\mathcal{U}^-$ always non-negative, as well as c and $1 - c$

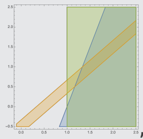
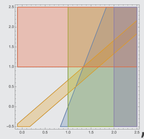
desired inequality!

- add (7) and (8) to obtain cut $$\frac{1}{1-c}(\mathcal{L}^+ - \mathcal{U}^-) + \frac{1}{c}(\mathcal{U}^+ - \mathcal{L}^-) \geq 1$$ □

Gomory Cuts Assumptions

for some $i \in B$ variable x_i is assigned $v(x_i) \notin \mathbb{Z}$, and for all $j \in N$ value $v(x_j)$ is l_j or u_j

Example (tableau $s = 3x - 3y$ and bounds $1 \leq s \leq 2$)

- simplex: $B = \{x\}$, $N = \{y, s\}$ and $v(x) = 1/3, v(y) = 0, v(s) = 1$, y has no bounds
- consequence: branch-and-cut, combine cutting planes with branch-and-bound
- branch-and-bound: add $x \geq 1$ (and later on try $x \leq 0$)
- simplex: $B = \{y\}$, $N = \{x, s\}$ and $v(x) = 1, v(y) = 2/3, v(s) = 1$
- now Gomory assumptions are satisfied; compute $c = 2/3$, $L^+ = \{x\}$, $L^- = \{s\}$, $U = \emptyset$ and add cut which can be simplified to $s + 6x \geq 9$, i.e., $3x - y \geq 3$
- add new slack variable t, tableau equation $t = s + 6x$ and bound $t \geq 9$ with $v(t) = 7$
- have , next , . . . still not terminating without global bounds

Summary on LIA-solving

- branch-and-bound
  - additional constraints are trivial: $x \leq \lfloor r \rfloor \vee \lceil r \rceil \leq x$ for $v(x) = r \notin \mathbb{Z}$
  - pruning of search space is limited
- cutting planes via Gomory cuts
  - calculation of cut is more complex, but still a simple algorithm
  - more effective pruning of search space, no branching
  - disadvantage: simplex invocations become more and more costly, since every cut increases size of tableau
- combination: branch-and-cut
  - run branch-and-bound with in-between additions of cuts
- further methods
  - unit cube test (Bromberger, Weidenbach)
  - improved bounds for small model property, depending on constraint structure (Bromberger)
  - mixed integer linear arithmetic where only some variables have to be integral

Outline

1. Summary of Previous Lecture
2. Cutting Planes
3. **Difference Logic**
4. Further Reading

Difference Logic

conjunction of constraints of the form

- $x - y \leqslant c$
- $x - y < c$

Remarks

- difference logic is fragment of linear arithmetic; advantage: faster decision procedure
- domains: rational numbers (polynomial time) and integers (polynomial time)
- $x - y = c \quad \Longleftrightarrow \quad x - y \leqslant c \ \wedge \ y - x \leqslant -c$
- $x - y \geqslant c \quad \Longleftrightarrow \quad y - x \leqslant -c$
- $x - y > c \quad \Longleftrightarrow \quad y - x < -c$
- $x < c \quad \Longleftrightarrow \quad x - x_0 < c$ where x_0 is fresh variable that must be assigned 0

Example Job-Shop Scheduling

- m machines $(M_1, \dots, M_m)$ and n jobs $(J_1, \dots, J_n)$
- each job J_i is sequence $(M^i_1, d^i_1), \dots, (M^i_{n_i}, d^i_{n_i})$ of operations consisting of machine and duration (rational number; $\tau(M, d) = d$)
- O is multiset of all operations from all jobs
- schedule is function S that defines for each operation $v \in O$ its starting time $S(v)$ on machine specified by v
- schedule S is feasible if

$$S(v) \geqslant 0 \qquad \text{for all } v \in O$$

$$S(v_i) + \tau(v_i) \leqslant S(v_j) \qquad \text{for all consecutive } v_i, v_j \text{ in same job}$$

$$S(v_i) + \tau(v_i) \leqslant S(v_j) \ \vee \ S(v_j) + \tau(v_j) \leqslant S(v_i)$$

for every pair of different operations v_i, v_j scheduled on same machine

- length of schedule S is $\max\{S(v) + \tau(v) \mid v \in O\}$

Definition Inequality Graph

conjunction φ of nonstrict difference constraints

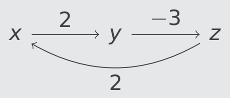
- inequality graph of φ contains edge from x to y with weight c for every constraint $x - y \leqslant c$ in φ

Theorem

conjunction φ of nonstrict difference constraints is satisfiable $\Longleftrightarrow$ inequality graph of φ has no negative cycle

Example

$$x - y \leqslant 2$$
$$y - z \leqslant -3$$
$$z - x \leqslant 2$$



satisfiable

Theorem

conjunction φ of nonstrict difference constraints is satisfiable $\Longleftrightarrow$ inequality graph of φ has no negative cycle

Proof

$\Rightarrow$ negative cycle $\quad x_1 \xrightarrow{k_1} x_2 \xrightarrow{k_2} x_3 \longrightarrow \cdots \longrightarrow x_n \xrightarrow{k_n} x_1$

in inequality graph of φ corresponds to conjuction

$$x_1 - x_2 \leqslant k_1 \ \wedge \ x_2 - x_3 \leqslant k_2 \ \wedge \ \cdots \ \wedge \ x_n - x_1 \leqslant k_n$$

adding these literals gives

$$0 \leqslant k_1 + k_2 + \cdots + k_n$$

with $k_1 + k_2 + \cdots + k_n < 0$ ↯

Theorem

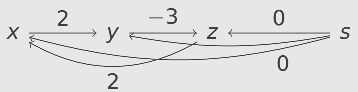
conjunction φ of nonstrict difference constraints is satisfiable $\iff$ inequality graph of φ has no negative cycle

Proof

$\Leftarrow$ assume inequality graph of φ has no negative cycle
construct satisfying assignment for φ as follows

- add additional starting node s in graph, add edges $s \to x$ with weight 0 for all variables x
- define $v(x) = -\text{distance}(s, x)$; well-defined, since there are no negative cycles
- v satisfies φ

Example



$v(x) = 1, v(y) = 0, v(z) = 3$

Algorithms for Distance Computation and Negative Cycle Detection

- Dijkstra
  - computes distances from a single source to all other nodes
  - complexity: $\mathcal{O}(|V| \cdot log(|V|) + |E|)$
  - restriction: no negative cycles allowed
- Bellman-Ford
  - computes distances from a single source to all other nodes
  - complexity: $\mathcal{O}(|V| \cdot |E|)$
  - can also detect negative cycles
- Floyd-Warshall
  - computes distances between all nodes
  - complexity: $\mathcal{O}(|V|^3)$
  - can also detect negative cycles

$\Longrightarrow$ use Bellman-Ford algorithm for difference logic

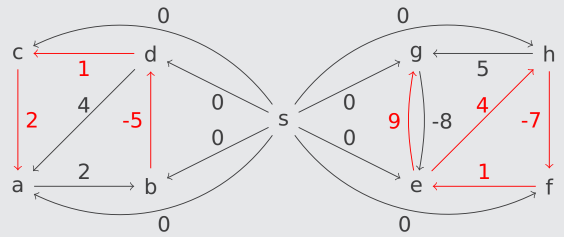
Bellman-Ford Algorithm for Inequality Graphs

Input inequality graph (V, E, w) with fresh starting node s

Output $\exists$ negative cycle or distances to node s

1. distance[v] $:= 0$ for all nodes $v \in V$ (this step is special for inequality graphs)
2. repeat $|V| - 1$ times
   for all $(u, v) \in E$ do
     if distance[v] $>$ distance[u] $+ w(u, v)$ then
       distance[v] $:=$ distance[u] $+ w(u, v)$
       predecessor[v] $:= u$
3. for all $(u, v) \in E$ do
   if distance[v] $>$ distance[u] $+ w(u, v)$ then
     return "$\exists$ negative cycle"
       which can be reconstructed using predecessor array
4. return distance array, shortest paths available via predecessor array

Example Bellman-Ford in Action



iteration	a	b	c	d	e	f	g	h
0	0	0	0	0	0	0	0	0
1	0	0	0	-5	-8	-7	0	0
2	-1	0	-4	-5	-8	-7	0	-4
3	-2	0	-4	-5	-8	-11	0	-4
4	-2	0	-4	-5	-10	-11	0	-4
5	-2	0	-4	-5	-10	-11	-1	-6

. . . 2 more iterations, then negative cycle is detected

Outline

1. Summary of Previous Lecture
2. Cutting Planes
3. Difference Logic
4. **Further Reading**

Kröning and Strichmann

- Sections 5.3 and 5.7

Further Reading

Martin Bromberger
A Reduction from Unbounded Linear Mixed Arithmetic Problems into Bounded Problems
Proc. IJCAR 2018, volume 10900 of LNCS, pages 329-–345, 2018

Martin Bromberger and Christoph Weidenbach
New techniques for linear arithmetic: cubes and equalities
Formal Methods in System Design, volume 51, pages 433-–461, 2017

Bruno Dutertre and Leonardo de Moura
Integrating Simplex with DPLL(T)
Technical Report SRI–CSL–06–01, SRI International, 2006

Important Concepts

- Bellman-Ford algorithm
- cutting planes
- difference logic
- Gomory cut
- inequality graph