universität innsbruck

SS 2024 lecture 8

Constraint Solving

René Thiemann and Fabian Mitterwallner

based on a previous course by Aart Middeldorp

Outline

Properties of DPLL(T) Simplex Algorithm

- termination ensured via Bland's rule:
  choose x_i and x_j for pivoting in a way that $(x_i, x_j) \in B \times N$ is lexicographically smallest

Properties of DPLL(T) Simplex Algorithm

- termination ensured via Bland's rule:
  choose x_i and x_j for pivoting in a way that $(x_i, x_j) \in B \times N$ is lexicographically smallest
- worst-case complexity is exponential, but often it runs in polynomial time

Properties of DPLL(T) Simplex Algorithm

- termination ensured via Bland's rule:
  choose x_i and x_j for pivoting in a way that $(x_i, x_j) \in B \times N$ is lexicographically smallest
- worst-case complexity is exponential, but often it runs in polynomial time
- provides incremental interface (activation flags for bounds) and unsatisfiable cores
  (Haskell: initSimplex, assert i, check, solution, checkpoint, backtrack cp)

Properties of DPLL(T) Simplex Algorithm

- termination ensured via Bland's rule:
  choose x_i and x_j for pivoting in a way that $(x_i, x_j) \in B \times N$ is lexicographically smallest
- worst-case complexity is exponential, but often it runs in polynomial time
- provides incremental interface (activation flags for bounds) and unsatisfiable cores
  (Haskell: initSimplex, assert i, check, solution, checkpoint, backtrack cp)
- Farkas' lemma: constraints $\bigwedge_i \ell_i \leq r_i$ are unsatisfiable iff a non-negative linear combination yields an obvious contradiction $\mathbb{Q} \ni \sum_i c_i \ell_i > \sum_i c_i r_i \in \mathbb{Q}$

Properties of DPLL(T) Simplex Algorithm

- termination ensured via Bland's rule:
  choose x_i and x_j for pivoting in a way that $(x_i, x_j) \in B \times N$ is lexicographically smallest
- worst-case complexity is exponential, but often it runs in polynomial time
- provides incremental interface (activation flags for bounds) and unsatisfiable cores
  (Haskell: initSimplex, assert i, check, solution, checkpoint, backtrack cp)
- Farkas' lemma: constraints $\bigwedge_i \ell_i \leq r_i$ are unsatisfiable iff a non-negative linear combination yields an obvious contradiction $\mathbb{Q} \ni \sum_i c_i \ell_i > \sum_i c_i r_i \in \mathbb{Q}$
- ranking functions for proving termination can be synthesized

Properties of DPLL(T) Simplex Algorithm

- termination ensured via Bland's rule:
  choose x_i and x_j for pivoting in a way that $(x_i, x_j) \in B \times N$ is lexicographically smallest
- worst-case complexity is exponential, but often it runs in polynomial time
- provides incremental interface (activation flags for bounds) and unsatisfiable cores
  (Haskell: initSimplex, assert i, check, solution, checkpoint, backtrack cp)
- Farkas' lemma: constraints $\bigwedge_i \ell_i \leq r_i$ are unsatisfiable iff a non-negative linear combination yields an obvious contradiction $\mathbb{Q} \ni \sum_i c_i \ell_i > \sum_i c_i r_i \in \mathbb{Q}$
- ranking functions for proving termination can be synthesized
- DPLL(T) simplex not well suited for linear programming, i.e., optimization problems

Outline

Example (Application of Linear Arithmetic: Termination Proving)

- last lecture

```
int factorial(int n) {
  int i = 1;
  int r = 1;
  while (i <= n) {
    r = r * i;
    i = i + 1;   }
  return r;          }
```

Example (Application of Linear Arithmetic: Termination Proving)

- last lecture

```
int factorial(int n) {
  int i = 1;
  int r = 1;
  while (i <= n) {
    r = r * i;
    i = i + 1;    }
  return r;          }
```

- remark: ranking function formula consists purely of $\leq$ inequalities
  - $\varphi := i \leq n \wedge n' = n \wedge i' = i + 1$
  - $\varphi \to e(i, n) \geq e(i', n') + d$
  - $\varphi \to e(i, n) \geq f$

Example (Application of Linear Integer Arithmetic: Termination Proving)

- consider another program

```
int log2(int x)      {
    int n := 0;
    while (x > 0) {
      x := x div 2;
      n := n + 1; }
    return n - 1;        }
```

Example (Application of Linear Integer Arithmetic: Termination Proving)

- consider another program

```
int log2(int x)       {
    int n := 0;
    while (x > 0) {
      x := x div 2;
      n := n + 1; }
    return n - 1;        }
```

- $\varphi := x > 0 \wedge 2x' \leq x \wedge x \leq 2x' + 1 \wedge n' = n + 1$ contains strict inequality

Example (Application of Linear Integer Arithmetic: Termination Proving)

- consider another program

```
int log2(int x)       {
    int n := 0;
    while (x > 0) {
      x := x div 2;
      n := n + 1; }
    return n - 1;        }
```

- $\varphi := x > 0 \wedge 2x' \leq x \wedge x \leq 2x' + 1 \wedge n' = n + 1$ contains strict inequality
- choose $e(x, n) = x$, $d = 1$ and $f = -1$; get two LIA problems that must be unsat
  - $\varphi \wedge x < x' + 1$ ($\neg$ decrease)
  - $\varphi \wedge x < -1$ ($\neg$ bounded)

Example (Application of Linear Integer Arithmetic: Termination Proving)

- consider another program

```
int log2(int x)      {
    int n := 0;
    while (x > 0) {
      x := x div 2;
      n := n + 1; }
    return n - 1;        }
```

- $\varphi := x > 0 \wedge 2x' \leq x \wedge x \leq 2x' + 1 \wedge n' = n + 1$ contains strict inequality
- choose $e(x, n) = x$, $d = 1$ and $f = -1$; get two LIA problems that must be unsat
  - $\varphi \wedge x < x' + 1$ $(\neg$ decrease)
  - $\varphi \wedge x < -1$ $(\neg$ bounded)
- $(\neg$ bounded) is unsatisfiable over $\mathbb{R}$

Example (Application of Linear Integer Arithmetic: Termination Proving)

- consider another program

```
int log2(int x)      {
    int n := 0;
    while (x > 0) {
      x := x div 2;
      n := n + 1; }
    return n - 1;          }
```

- $\varphi := x > 0 \wedge 2x' \leq x \wedge x \leq 2x' + 1 \wedge n' = n + 1$ contains strict inequality
- choose $e(x, n) = x$, $d = 1$ and $f = -1$; get two LIA problems that must be unsat
  - $\varphi \wedge x < x' + 1$ ($\neg$ decrease)
  - $\varphi \wedge x < -1$ ($\neg$ bounded)
- ($\neg$ bounded) is unsatisfiable over $\mathbb{R}$
- ($\neg$ decrease) is unsatisfiable over $\mathbb{Z}$, but not over $\mathbb{R}$ $\Longrightarrow$ require LIA solver

Example (Application of Linear Integer Arithmetic: Termination Proving)

- consider another program

```
int log2(int x)      {
    int n := 0;
    while (x > 0) {
      x := x div 2;
      n := n + 1; }
    return n - 1;       }
```

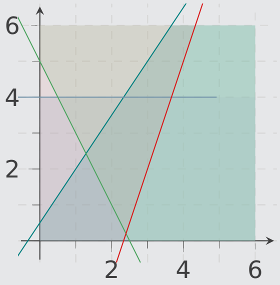
- $\varphi := x > 0 \land 2x' \leq x \land x \leq 2x' + 1 \land n' = n + 1$ contains strict inequality
- choose $e(x, n) = x$, $d = 1$ and $f = -1$; get two LIA problems that must be unsat
  - $\varphi \land x < x' + 1$ ($\neg$ decrease)
  - $\varphi \land x < -1$ ($\neg$ bounded)
- ($\neg$ bounded) is unsatisfiable over $\mathbb{R}$
- ($\neg$ decrease) is unsatisfiable over $\mathbb{Z}$, but not over $\mathbb{R}$ $\Longrightarrow$ require LIA solver
- remark: LIA reasoning is crucial, the problem is not wrong choice of expression e; program does not terminate when executed with real number arithmetic

Outline

Example

$$3x - 2y \geq -1$$
$$y \leq 4$$
$$2x + y \geq 5$$
$$3x - y \leq 7$$

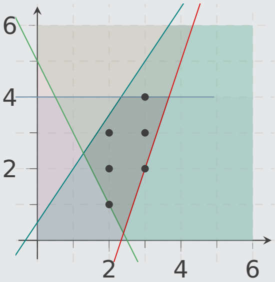
- looking for solution in $\mathbb{Z}^2$



Example

$$3x - 2y \geq -1$$
$$y \leq 4$$
$$2x + y \geq 5$$
$$3x - y \leq 7$$

- looking for solution in $\mathbb{Z}^2$
- infinite $\mathbb{R}^2$ solution space, six solutions in $\mathbb{Z}^2$



Example

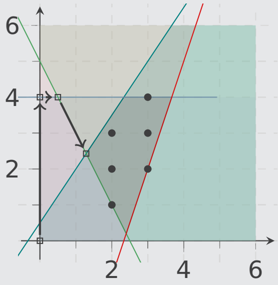
$3x - 2y \geq -1$

$y \leq 4$

$2x + y \geq 5$

$3x - y \leq 7$

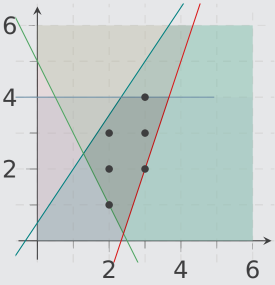
- looking for solution in $\mathbb{Z}^2$
- infinite $\mathbb{R}^2$ solution space, six solutions in $\mathbb{Z}^2$
- simplex returns $(\frac{9}{7}, \frac{17}{7})$



Example

$$3x - 2y \geq -1$$
$$y \leq 4$$
$$2x + y \geq 5$$
$$3x - y \leq 7$$

- looking for solution in $\mathbb{Z}^2$
- infinite $\mathbb{R}^2$ solution space, six solutions in $\mathbb{Z}^2$
- simplex returns $(\frac{9}{7}, \frac{17}{7})$



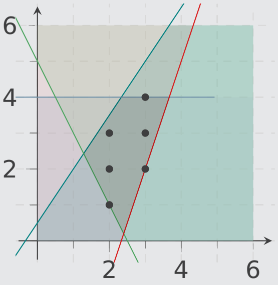
Branch and Bound, a Solver for LIA Formulas – Idea

- add constraints that exclude current solution in $\mathbb{R}^2 \setminus \mathbb{Z}^2$ but do not change solutions in $\mathbb{Z}^2$

Example

$$3x - 2y \geq -1$$
$$y \leq 4$$
$$2x + y \geq 5$$
$$3x - y \leq 7$$

- looking for solution in $\mathbb{Z}^2$
- infinite $\mathbb{R}^2$ solution space, six solutions in $\mathbb{Z}^2$
- simplex returns $(\frac{9}{7}, \frac{17}{7})$



Branch and Bound, a Solver for LIA Formulas – Idea

- add constraints that exclude current solution in $\mathbb{R}^2 \setminus \mathbb{Z}^2$ but do not change solutions in $\mathbb{Z}^2$
- in current solution $1 < x < 2$, so use simplex on two augmented problems:
  - $C \wedge x \leqslant 1$
  - $C \wedge x \geqslant 2$

Example

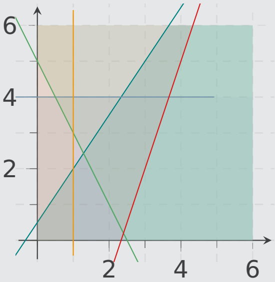
$3x - 2y \geq -1$

$y \leq 4$

$2x + y \geq 5$

$3x - y \leq 7$

- looking for solution in $\mathbb{Z}^2$
- infinite $\mathbb{R}^2$ solution space, six solutions in $\mathbb{Z}^2$
- simplex returns $(\frac{9}{7}, \frac{17}{7})$

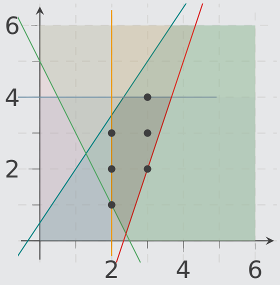


Branch and Bound, a Solver for LIA Formulas – Idea

- add constraints that exclude current solution in $\mathbb{R}^2 \setminus \mathbb{Z}^2$ but do not change solutions in $\mathbb{Z}^2$
- in current solution $1 < x < 2$, so use simplex on two augmented problems:
  - $C \wedge x \leqslant 1$ unsatisfiable
  - $C \wedge x \geqslant 2$

Example

$$3x - 2y \geq -1$$
$$y \leq 4$$
$$2x + y \geq 5$$
$$3x - y \leq 7$$



- looking for solution in $\mathbb{Z}^2$
- infinite $\mathbb{R}^2$ solution space, six solutions in $\mathbb{Z}^2$
- simplex returns $(\frac{9}{7}, \frac{17}{7})$

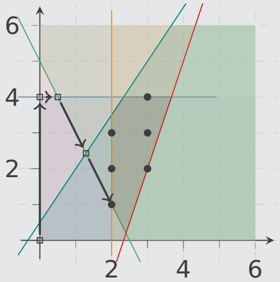
Branch and Bound, a Solver for LIA Formulas – Idea

- add constraints that exclude current solution in $\mathbb{R}^2 \setminus \mathbb{Z}^2$ but do not change solutions in $\mathbb{Z}^2$
- in current solution $1 < x < 2$, so use simplex on two augmented problems:
  - $C \wedge x \leqslant 1$ unsatisfiable
  - $C \wedge x \geqslant 2$ satisfiable,

Example

$$3x - 2y \geq -1$$
$$y \leq 4$$
$$2x + y \geq 5$$
$$3x - y \leq 7$$

- looking for solution in $\mathbb{Z}^2$
- infinite $\mathbb{R}^2$ solution space, six solutions in $\mathbb{Z}^2$
- simplex returns $(\frac{9}{7}, \frac{17}{7})$



Branch and Bound, a Solver for LIA Formulas – Idea

- add constraints that exclude current solution in $\mathbb{R}^2 \setminus \mathbb{Z}^2$ but do not change solutions in $\mathbb{Z}^2$
- in current solution $1 < x < 2$, so use simplex on two augmented problems:
  - $C \wedge x \leqslant 1$ unsatisfiable
  - $C \wedge x \geqslant 2$ satisfiable, simplex can return $(2, 1)$

Algorithm BranchAndBound(φ)

Input: LIA formula φ, a conjunction of linear inequalities
Output: unsatisfiable, or satisfying assignment

```
let res be result of deciding φ over ℝ                    ▷ e.g. by simplex
if res is unsatisfiable then
    return unsatisfiable
else if res is solution over ℤ then
    return res
else
    let x be variable assigned non-integer value q in res
    res = BranchAndBound(φ ∧ x ≤ ⌊q⌋)
    if res ≠ unsatisfiable then
        return res
    else
        return BranchAndBound(φ ∧ x ≥ ⌈q⌉)
```

Example (Termination Proof of log2, Continued)

- problematic formula (satisfiable over $\mathbb{R}$)

$$\psi := x > 0 \wedge 2x' \leq x \wedge x \leq 2x' + 1 \wedge x < x' + 1 \qquad (\neg \text{ decrease})$$

- execution of BranchAndBound on ψ (short notation: $BB(\psi)$)

Example (Termination Proof of log2, Continued)

- problematic formula (satisfiable over $\mathbb{R}$)

$$\psi := x > 0 \land 2x' \leq x \land x \leq 2x' + 1 \land x < x' + 1 \qquad (\neg \text{ decrease})$$

- execution of BranchAndBound on ψ (short notation: $BB(\psi)$)
  - simplex: $v(x) = 1$, $v(x') = \frac{1}{2}$

Example (Termination Proof of log2, Continued)

- problematic formula (satisfiable over $\mathbb{R}$)

$$\psi := x > 0 \wedge 2x' \leq x \wedge x \leq 2x' + 1 \wedge x < x' + 1 \qquad (\neg \text{ decrease})$$

- execution of BranchAndBound on ψ (short notation: $BB(\psi)$)
  - simplex: $v(x) = 1$, $v(x') = \frac{1}{2}$
  - invoke $BB(\psi \wedge x' \geq 1)$, simplex: unsatisfiable

Example (Termination Proof of log2, Continued)

- problematic formula (satisfiable over $\mathbb{R}$)

$$\psi := x > 0 \wedge 2x' \leq x \wedge x \leq 2x' + 1 \wedge x < x' + 1 \qquad (\neg \text{ decrease})$$

- execution of BranchAndBound on ψ (short notation: $BB(\psi)$)
  - simplex: $v(x) = 1$, $v(x') = \frac{1}{2}$
  - invoke $BB(\psi \wedge x' \geq 1)$, simplex: unsatisfiable
  - invoke $BB(\psi \wedge x' \leq 0)$, simplex: $v(x) = \frac{1}{2}, v(x') = -\frac{1}{4}$

Example (Termination Proof of log2, Continued)

- problematic formula (satisfiable over $\mathbb{R}$)

$$\psi := x > 0 \land 2x' \leq x \land x \leq 2x' + 1 \land x < x' + 1 \qquad (\neg \text{ decrease})$$

- execution of BranchAndBound on ψ (short notation: $BB(\psi)$)
  - simplex: $v(x) = 1$, $v(x') = \frac{1}{2}$
  - invoke $BB(\psi \land x' \geq 1)$, simplex: unsatisfiable
  - invoke $BB(\psi \land x' \leq 0)$, simplex: $v(x) = \frac{1}{2}, v(x') = -\frac{1}{4}$
    - invoke $BB(\psi \land x' \leq 0 \land x \geq 1)$, simplex: unsatisfiable

Example (Termination Proof of log2, Continued)

- problematic formula (satisfiable over $\mathbb{R}$)

$$\psi := x > 0 \land 2x' \leq x \land x \leq 2x' + 1 \land x < x' + 1 \qquad (\neg \text{ decrease})$$

- execution of BranchAndBound on ψ (short notation: $BB(\psi)$)
  - simplex: $v(x) = 1$, $v(x') = \frac{1}{2}$
  - invoke $BB(\psi \land x' \geq 1)$, simplex: unsatisfiable
  - invoke $BB(\psi \land x' \leq 0)$, simplex: $v(x) = \frac{1}{2}, v(x') = -\frac{1}{4}$
    - invoke $BB(\psi \land x' \leq 0 \land x \geq 1)$, simplex: unsatisfiable
    - invoke $BB(\psi \land x' \leq 0 \land x \leq 0)$, simplex: unsatisfiable

Example (Termination Proof of log2, Continued)

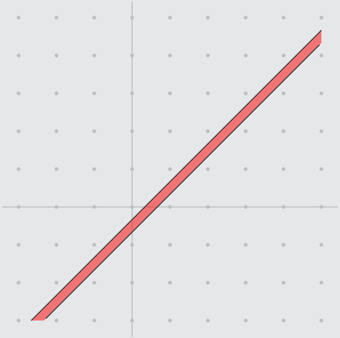
- problematic formula (satisfiable over $\mathbb{R}$)

$$\psi := x > 0 \wedge 2x' \leq x \wedge x \leq 2x' + 1 \wedge x < x' + 1 \qquad (\neg\ \text{decrease})$$

- execution of BranchAndBound on ψ (short notation: $BB(\psi)$)
  - simplex: $v(x) = 1$, $v(x') = \frac{1}{2}$
  - invoke $BB(\psi \wedge x' \geq 1)$, simplex: unsatisfiable
  - invoke $BB(\psi \wedge x' \leq 0)$, simplex: $v(x) = \frac{1}{2}, v(x') = -\frac{1}{4}$
    - invoke $BB(\psi \wedge x' \leq 0 \wedge x \geq 1)$, simplex: unsatisfiable
    - invoke $BB(\psi \wedge x' \leq 0 \wedge x \leq 0)$, simplex: unsatisfiable
  - return unsatisfiable

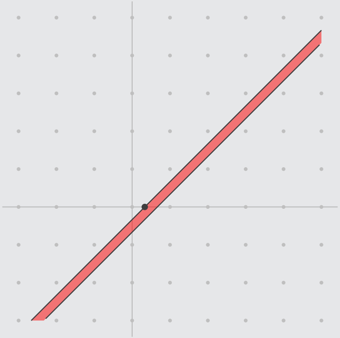
Example (Branch and Bound – Problem)

consider $\psi := 1 \leq 3x - 3y \wedge 3x - 3y \leq 2$



Example (Branch and Bound – Problem)

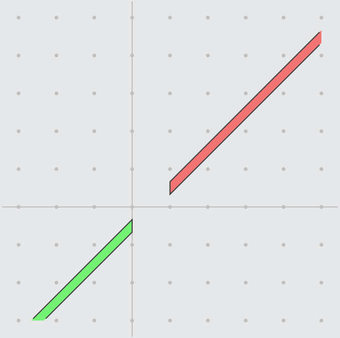
consider $\psi := 1 \leq 3x - 3y \wedge 3x - 3y \leq 2$



- $v(x) = \frac{1}{3}, v(y) = 0$

Example (Branch and Bound – Problem)

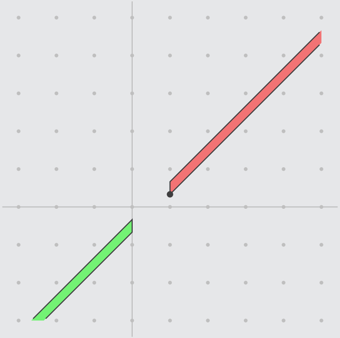
consider $\psi := 1 \leq 3x - 3y \wedge 3x - 3y \leq 2$



- $v(x) = \frac{1}{3}$, $v(y) = 0$, add $x \leq 0$ or $x \geq 1$

Example (Branch and Bound – Problem)

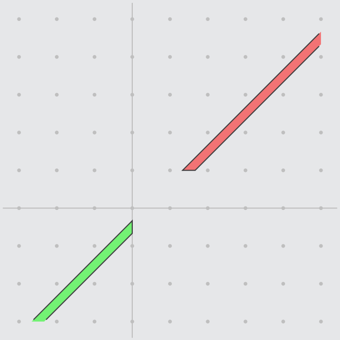
consider $\psi := 1 \leq 3x - 3y \wedge 3x - 3y \leq 2$



- $v(x) = \frac{1}{3}$, $v(y) = 0$, add $x \leq 0$ or $x \geq 1$
- for $\psi \wedge x \geq 1$: $v(x) = 1$, $v(y) = \frac{1}{3}$

Example (Branch and Bound – Problem)

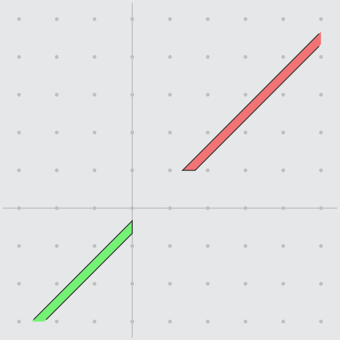
consider $\psi := 1 \leq 3x - 3y \wedge 3x - 3y \leq 2$



- $v(x) = \frac{1}{3}$, $v(y) = 0$, add $x \leq 0$ or $x \geq 1$
- for $\psi \wedge x \geq 1$: $v(x) = 1$, $v(y) = \frac{1}{3}$, add $y \leq 0$ or $y \geq 1$

Example (Branch and Bound – Problem)

consider $\psi := 1 \leq 3x - 3y \wedge 3x - 3y \leq 2$



- $v(x) = \frac{1}{3}$, $v(y) = 0$, add $x \leq 0$ or $x \geq 1$
- for $\psi \wedge x \geq 1$: $v(x) = 1$, $v(y) = \frac{1}{3}$, add $y \leq 0$ or $y \geq 1$
- ... *BranchAndBound* is not terminating, since search space is unbounded

Theorem (Small Model Property of LIA)

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

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

for all x where

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

Theorem (Small Model Property of LIA)

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

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

for all x where

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

Consequences and Remarks

- satisfiability of ψ for LIA formula is in NP

Theorem (Small Model Property of LIA)

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

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

for all x where

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

Consequences and Remarks

- satisfiability of ψ for LIA formula is in NP
- invoke

$$BranchAndBound\left(\psi \wedge \bigwedge_{x \in vars(\psi)} -bound(\psi) \leq x \leq bound(\psi)\right)$$

  to decide solvability of ψ over $\mathbb{Z}$

Theorem (Small Model Property of LIA)

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

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

for all x where

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

Consequences and Remarks

- satisfiability of ψ for LIA formula is in NP
- invoke

$$BranchAndBound\left(\psi \wedge \bigwedge_{x \in vars(\psi)} -bound(\psi) \leq x \leq bound(\psi)\right)$$

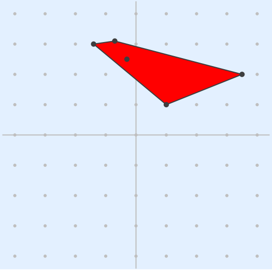
  to decide solvability of ψ over $\mathbb{Z}$
- bound is quite tight: $c \leq x_1 \wedge c \cdot x_1 \leq x_2 \wedge \ldots \wedge c \cdot x_{n-1} \leq x_n$ implies $x_n \geq c^n$

Outline

Geometric Objects

- polytope: convex hull of finite set of points X

$$hull(X) = \{\lambda_1 \vec{v}_1 + \ldots + \lambda_m \vec{v}_m \mid \{\vec{v}_1, \ldots, \vec{v}_m\} \subseteq X \wedge \lambda_1, \ldots, \lambda_m \geq 0 \wedge \sum \lambda_i = 1\}$$



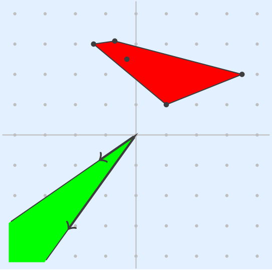
Geometric Objects

- polytope: convex hull of finite set of points X

$$hull(X) = \{\lambda_1 \vec{v}_1 + \ldots + \lambda_m \vec{v}_m \mid \{\vec{v}_1, \ldots, \vec{v}_m\} \subseteq X \wedge \lambda_1, \ldots, \lambda_m \geq 0 \wedge \sum \lambda_i = 1\}$$

- finitely generated cone: non-negative linear combinations of finite set of vectors V

$$cone(V) = \{\lambda_1 \vec{v}_1 + \ldots + \lambda_m \vec{v}_m \mid \{\vec{v}_1, \ldots, \vec{v}_m\} \subseteq V \wedge \lambda_1, \ldots, \lambda_m \geq 0\}$$



Geometric Objects

- polytope: convex hull of finite set of points X

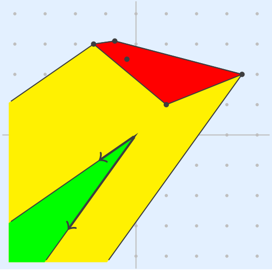
$$hull(X) = \{\lambda_1\vec{v}_1 + \ldots + \lambda_m\vec{v}_m \mid \{\vec{v}_1, \ldots, \vec{v}_m\} \subseteq X \wedge \lambda_1, \ldots, \lambda_m \geq 0 \wedge \sum \lambda_i = 1\}$$

- finitely generated cone: non-negative linear combinations of finite set of vectors V

$$cone(V) = \{\lambda_1\vec{v}_1 + \ldots + \lambda_m\vec{v}_m \mid \{\vec{v}_1, \ldots, \vec{v}_m\} \subseteq V \wedge \lambda_1, \ldots, \lambda_m \geq 0\}$$

- polyhedron: polytope + finitely generated cone

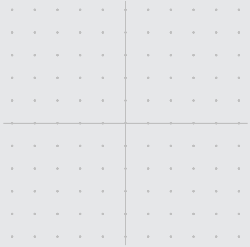
$$hull(X) + cone(V) = \{\vec{x} + \vec{v} \mid \vec{x} \in hull(X) \wedge \vec{v} \in cone(V)\}$$



More Geometric Objects

- C is polyhedral cone iff $C = \{\vec{x} \mid A\vec{x} \leq \vec{0}\}$ for some matrix A
  iff C is intersection of finitely many half-spaces

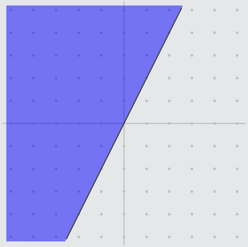
Example



More Geometric Objects

- C is polyhedral cone iff $C = \{\vec{x} \mid A\vec{x} \leq \vec{0}\}$ for some matrix A
  iff C is intersection of finitely many half-spaces

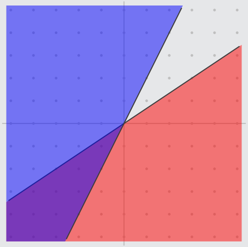
Example



More Geometric Objects

- C is polyhedral cone iff $C = \{\vec{x} \mid A\vec{x} \leq \vec{0}\}$ for some matrix A
  iff C is intersection of finitely many half-spaces

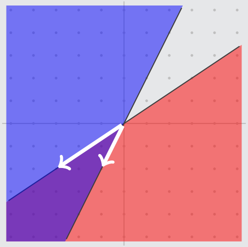
Example



More Geometric Objects

- C is polyhedral cone iff $C = \{\vec{x} \mid A\vec{x} \leq \vec{0}\}$ for some matrix A
  iff C is intersection of finitely many half-spaces

Example



Theorem (Farkas, Minkowski, Weyl)

A cone is polyhedral iff it is finitely generated.

Theorem (Farkas, Minkowski, Weyl)

A cone is polyhedral iff it is finitely generated.

Theorem (Farkas, Minkowski, Weyl)

A cone is polyhedral iff it is finitely generated.

Theorem (Decomposition Theorem for Polyhedra)

A set $P \subseteq \mathbb{R}^n$ can be described as a polyhedron $P = hull(X) + cone(V)$ for finite X and V iff $P = \{\vec{x} \mid A\vec{x} \leq \vec{b}\}$ for some matrix A and vector $\vec{b}$.
Moreover, given X and V one can compute A and $\vec{b}$, and vice versa.

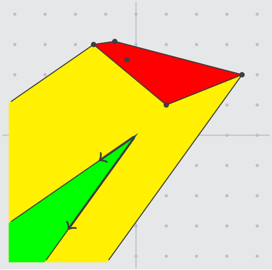
Theorem (Farkas, Minkowski, Weyl)

A cone is polyhedral iff it is finitely generated.

Theorem (Decomposition Theorem for Polyhedra)

A set $P \subseteq \mathbb{R}^n$ can be described as a polyhedron $P = \mathit{hull}(X) + \mathit{cone}(V)$ for finite X and V iff $P = \{\vec{x} \mid A\vec{x} \leq \vec{b}\}$ for some matrix A and vector $\vec{b}$.
Moreover, given X and V one can compute A and $\vec{b}$, and vice versa.

Example



Proof Idea of Small Model Property

1. convert conjunctive LIA formula ψ into form $A\vec{x} \leq \vec{b}$

Proof Idea of Small Model Property

1. convert conjunctive LIA formula ψ into form $A\vec{x} \leq \vec{b}$
2. represent polyhedron $\{\vec{x} \mid A\vec{x} \leq \vec{b}\}$ as polyhedron $P = \mathit{hull}(X) + \mathit{cone}(V)$

Proof Idea of Small Model Property

1. convert conjunctive LIA formula ψ into form $A\vec{x} \leq \vec{b}$
2. represent polyhedron $\{\vec{x} \mid A\vec{x} \leq \vec{b}\}$ as polyhedron $P = hull(X) + cone(V)$
3. show that P has small integral solutions, depending on X and V

Proof Idea of Small Model Property

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

Proof Idea of Small Model Property

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

Remark

- given ψ, one can compute X and V instead of using approximations
- however, this would be expensive: decomposition theorem requires exponentially many steps (in n, m) for input $A \in \mathbb{Z}^{m \times n}$ and $\vec{b} \in \mathbb{Z}^m$

Step 1: Conjunctive LIA Formula into Matrix Form $A\vec{x} \leq \vec{b}$

- (variable renamed) formula

$$x_1 > 0 \qquad 2x_2 \leq x_1 \qquad x_1 \leq 2x_2 + 1 \qquad x_1 < x_2 + 1$$

Step 1: Conjunctive LIA Formula into Matrix Form $A\vec{x} \leq \vec{b}$

- (variable renamed) formula

$$x_1 > 0 \qquad 2x_2 \leq x_1 \qquad x_1 \leq 2x_2 + 1 \qquad x_1 < x_2 + 1$$

- eliminate strict inequalities (only valid in LIA)

$$x_1 \geq 0 + 1 \qquad 2x_2 \leq x_1 \qquad x_1 \leq 2x_2 + 1 \qquad x_1 + 1 \leq x_2 + 1$$

Step 1: Conjunctive LIA Formula into Matrix Form $A\vec{x} \leq \vec{b}$

- (variable renamed) formula

$$x_1 > 0 \qquad 2x_2 \leq x_1 \qquad x_1 \leq 2x_2 + 1 \qquad x_1 < x_2 + 1$$

- eliminate strict inequalities (only valid in LIA)

$$x_1 \geq 0 + 1 \qquad 2x_2 \leq x_1 \qquad x_1 \leq 2x_2 + 1 \qquad x_1 + 1 \leq x_2 + 1$$

- normalize (only $\leq$, constant to the right-hand-side)

$$-x_1 \leq -1 \qquad -x_1 + 2x_2 \leq 0 \qquad x_1 - 2x_2 \leq 1 \qquad x_1 - x_2 \leq 0$$

Step 1: Conjunctive LIA Formula into Matrix Form $A\vec{x} \leq \vec{b}$

- (variable renamed) formula

$$x_1 > 0 \qquad 2x_2 \leq x_1 \qquad x_1 \leq 2x_2 + 1 \qquad x_1 < x_2 + 1$$

- eliminate strict inequalities (only valid in LIA)

$$x_1 \geq 0 + 1 \qquad 2x_2 \leq x_1 \qquad x_1 \leq 2x_2 + 1 \qquad x_1 + 1 \leq x_2 + 1$$

- normalize (only $\leq$, constant to the right-hand-side)

$$-x_1 \leq -1 \qquad -x_1 + 2x_2 \leq 0 \qquad x_1 - 2x_2 \leq 1 \qquad x_1 - x_2 \leq 0$$

- matrix form

$$\begin{pmatrix} -1 & 0 \\ -1 & 2 \\ 1 & -2 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} -1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Step 3: Small Integral Solutions of Polyhedrons

- consider finite sets $X \subseteq \mathbb{R}^n$ and $V \subseteq \mathbb{Z}^n$
- define

$$B = \{\lambda_1 \vec{v}_1 + \ldots + \lambda_n \vec{v}_n \mid \{\vec{v}_1, \ldots, \vec{v}_n\} \subseteq V \wedge 1 \geq \lambda_1, \ldots, \lambda_n \geq 0\} \subseteq cone(V)$$

Step 3: Small Integral Solutions of Polyhedrons

- consider finite sets $X \subseteq \mathbb{R}^n$ and $V \subseteq \mathbb{Z}^n$
- define

$$B = \{\lambda_1 \vec{v_1} + \ldots + \lambda_n \vec{v}_n \mid \{\vec{v}_1, \ldots, \vec{v}_n\} \subseteq V \wedge 1 \geq \lambda_1, \ldots, \lambda_n \geq 0\} \subseteq cone(V)$$

Theorem

$(hull(X) + cone(V)) \cap \mathbb{Z}^n = \emptyset \longleftrightarrow (hull(X) + B) \cap \mathbb{Z}^n = \emptyset$

Step 3: Small Integral Solutions of Polyhedrons

- consider finite sets $X \subseteq \mathbb{R}^n$ and $V \subseteq \mathbb{Z}^n$
- define

$$B = \{\lambda_1 \vec{v_1} + \ldots + \lambda_n \vec{v}_n \mid \{\vec{v}_1, \ldots, \vec{v}_n\} \subseteq V \wedge 1 \geq \lambda_1, \ldots, \lambda_n \geq 0\} \subseteq cone(V)$$

Theorem

$(hull(X) + cone(V)) \cap \mathbb{Z}^n = \emptyset \longleftrightarrow (hull(X) + B) \cap \mathbb{Z}^n = \emptyset$

Corollary

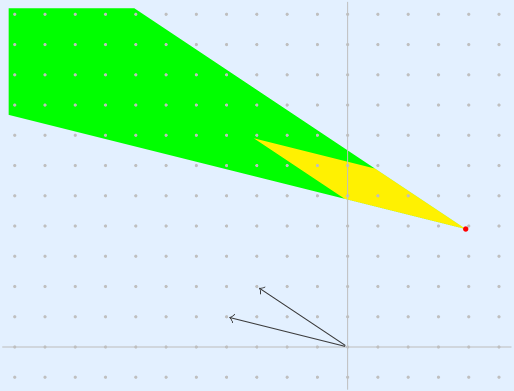
Assume $|c| \leq b \in \mathbb{Z}$ for all entries c of all vectors in $X \cup V$.
Define $Bnd := (n+1) \cdot b$. Then

$$(hull(X) + cone(V)) \cap \mathbb{Z}^n = \emptyset$$
$$\longleftrightarrow (hull(X) + cone(V)) \cap \{-Bnd, \ldots, Bnd\}^n = \emptyset$$

Theorem

$$(hull(X) + cone(V)) \cap \mathbb{Z}^n = \emptyset \longleftrightarrow (hull(X) + B) \cap \mathbb{Z}^n = \emptyset$$

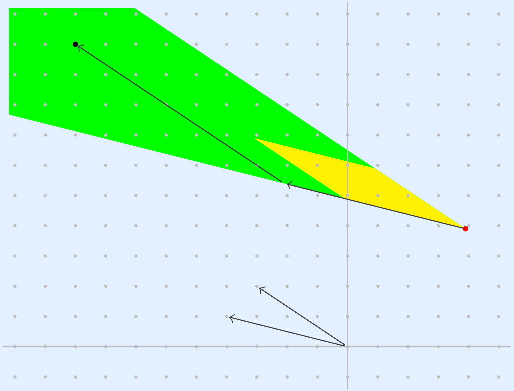
Proof



Theorem

$$(hull(X) + cone(V)) \cap \mathbb{Z}^n = \emptyset \longleftrightarrow (hull(X) + B) \cap \mathbb{Z}^n = \emptyset$$

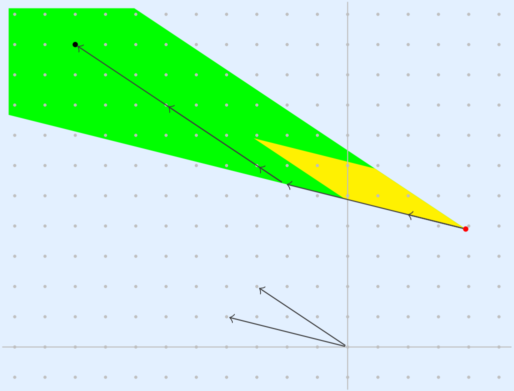
Proof



Theorem

$$(hull(X) + cone(V)) \cap \mathbb{Z}^n = \emptyset \longleftrightarrow (hull(X) + B) \cap \mathbb{Z}^n = \emptyset$$

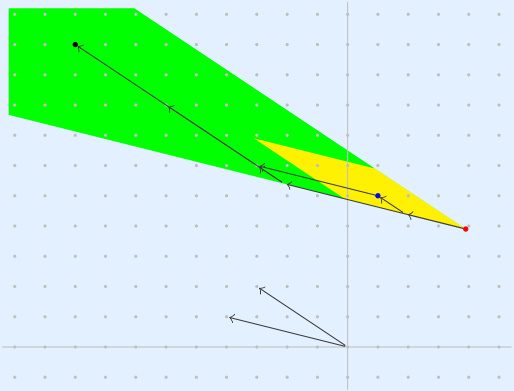
Proof



Theorem

$$(hull(X) + cone(V)) \cap \mathbb{Z}^n = \emptyset \longleftrightarrow (hull(X) + B) \cap \mathbb{Z}^n = \emptyset$$

Proof



Step 2a: Decomposing Polyhedron $P = \{\vec{u} \mid A\vec{u} \leq \vec{b}\}$ into $hull(X) + cone(V)$

1. use FMW to convert polyhedral cone of $\left\{ \vec{v} \;\middle|\; \begin{pmatrix} A & -\vec{b} \\ \vec{0} & -1 \end{pmatrix} \vec{v} \leq \vec{0} \right\}$ into $cone(C)$ for integral vectors $C = \left\{ \begin{pmatrix} \vec{y}_1 \\ \tau_1 \end{pmatrix}, \dots, \begin{pmatrix} \vec{y}_\ell \\ \tau_\ell \end{pmatrix}, \begin{pmatrix} \vec{z}_1 \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} \vec{z}_k \\ 0 \end{pmatrix} \right\}$ with $\tau_i > 0$ for all $1 \leq i \leq \ell$

Step 2a: Decomposing Polyhedron $P = \{\vec{u} \mid A\vec{u} \leq \vec{b}\}$ into $hull(X) + cone(V)$

1. use FMW to convert polyhedral cone of $\left\{ \vec{v} \;\middle|\; \begin{pmatrix} A & -\vec{b} \\ \vec{0} & -1 \end{pmatrix} \vec{v} \leq \vec{0} \right\}$ into $cone(C)$ for integral vectors $C = \left\{ \begin{pmatrix} \vec{y}_1 \\ \tau_1 \end{pmatrix}, \dots, \begin{pmatrix} \vec{y}_\ell \\ \tau_\ell \end{pmatrix}, \begin{pmatrix} \vec{z}_1 \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} \vec{z}_k \\ 0 \end{pmatrix} \right\}$ with $\tau_i > 0$ for all $1 \leq i \leq \ell$
2. define $\vec{x}_i := \frac{1}{\tau_i}\vec{y}_i$

Step 2a: Decomposing Polyhedron $P = \{\vec{u} \mid A\vec{u} \leq \vec{b}\}$ into $hull(X) + cone(V)$

1. use FMW to convert polyhedral cone of $\left\{ \vec{v} \;\middle|\; \begin{pmatrix} A & -\vec{b} \\ \vec{0} & -1 \end{pmatrix} \vec{v} \leq \vec{0} \right\}$ into $cone(C)$ for integral vectors $C = \left\{ \begin{pmatrix} \vec{y}_1 \\ \tau_1 \end{pmatrix}, \dots, \begin{pmatrix} \vec{y}_\ell \\ \tau_\ell \end{pmatrix}, \begin{pmatrix} \vec{z}_1 \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} \vec{z}_k \\ 0 \end{pmatrix} \right\}$ with $\tau_i > 0$ for all $1 \leq i \leq \ell$
2. define $\vec{x}_i := \frac{1}{\tau_i}\vec{y}_i$
3. return $X := \{\vec{x}_1, \dots, \vec{x}_\ell\}$ and $V := \{\vec{z}_1, \dots, \vec{z}_k\}$

Step 2a: Decomposing Polyhedron $P = \{\vec{u} \mid A\vec{u} \leq \vec{b}\}$ into $hull(X) + cone(V)$

1. use FMW to convert polyhedral cone of $\left\{ \vec{v} \;\middle|\; \begin{pmatrix} A & -\vec{b} \\ \vec{0} & -1 \end{pmatrix} \vec{v} \leq \vec{0} \right\}$ into $cone(C)$ for integral vectors $C = \left\{ \begin{pmatrix} \vec{y}_1 \\ \tau_1 \end{pmatrix}, \dots, \begin{pmatrix} \vec{y}_\ell \\ \tau_\ell \end{pmatrix}, \begin{pmatrix} \vec{z}_1 \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} \vec{z}_k \\ 0 \end{pmatrix} \right\}$ with $\tau_i > 0$ for all $1 \leq i \leq \ell$
2. define $\vec{x}_i := \frac{1}{\tau_i}\vec{y}_i$
3. return $X := \{\vec{x}_1, \dots, \vec{x}_\ell\}$ and $V := \{\vec{z}_1, \dots, \vec{z}_k\}$

Theorem

$P = hull(X) + cone(V)$

Step 2a: Decomposing Polyhedron $P = \{\vec{u} \mid A\vec{u} \leq \vec{b}\}$ into $hull(X) + cone(V)$

1. use FMW to convert polyhedral cone of $\left\{ \vec{v} \;\middle|\; \begin{pmatrix} A & -\vec{b} \\ \vec{0} & -1 \end{pmatrix} \vec{v} \leq \vec{0} \right\}$ into $cone(C)$ for integral vectors $C = \left\{ \begin{pmatrix} \vec{y}_1 \\ \tau_1 \end{pmatrix}, \dots, \begin{pmatrix} \vec{y}_\ell \\ \tau_\ell \end{pmatrix}, \begin{pmatrix} \vec{z}_1 \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} \vec{z}_k \\ 0 \end{pmatrix} \right\}$ with $\tau_i > 0$ for all $1 \leq i \leq \ell$
2. define $\vec{x}_i := \frac{1}{\tau_i}\vec{y}_i$
3. return $X := \{\vec{x}_1, \dots, \vec{x}_\ell\}$ and $V := \{\vec{z}_1, \dots, \vec{z}_k\}$

Theorem

$P = hull(X) + cone(V)$

Bounds

- the absolute values of the numbers in $X \cup V$ are all bounded by the absolute values of the numbers in C
- hence, bounds on C can be reused to bound vectors in $X \cup V$

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

First direction: finitely generated implies polyhedral

- consider *cone* (V) for $V = \{\vec{v}_1, \ldots, \vec{v}_m\} \subseteq \mathbb{Z}^n$

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

First direction: finitely generated implies polyhedral

- consider $\mathit{cone}(V)$ for $V = \{\vec{v}_1, \ldots, \vec{v}_m\} \subseteq \mathbb{Z}^n$
- consider every set $W \subseteq V$ of linearly independent vectors with $|W| = n - 1$

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

First direction: finitely generated implies polyhedral

- consider *cone* (V) for $V = \{\vec{v}_1, \ldots, \vec{v}_m\} \subseteq \mathbb{Z}^n$
- consider every set $W \subseteq V$ of linearly independent vectors with $|W| = n - 1$
- obtain integral normal vector $\vec{c}$ of hyper-space spanned by W

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

First direction: finitely generated implies polyhedral

- consider *cone* (V) for $V = \{\vec{v}_1, \ldots, \vec{v}_m\} \subseteq \mathbb{Z}^n$
- consider every set $W \subseteq V$ of linearly independent vectors with $|W| = n - 1$
- obtain integral normal vector $\vec{c}$ of hyper-space spanned by W
- next check whether V is contained in hyper-space $\{\vec{v} \mid \vec{v} \cdot \vec{c} \leq 0\}$ or $\{\vec{v} \mid \vec{v} \cdot (-\vec{c}) \leq 0\}$
  - if $\vec{v}_i \cdot \vec{c} \leq 0$ for all i, then add $\vec{c}$ as row to A
  - if $\vec{v}_i \cdot \vec{c} \geq 0$ for all i, then add $-\vec{c}$ as row to A

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

First direction: finitely generated implies polyhedral

- consider $cone\,(V)$ for $V = \{\vec{v}_1, \ldots, \vec{v}_m\} \subseteq \mathbb{Z}^n$
- consider every set $W \subseteq V$ of linearly independent vectors with $|W| = n - 1$
- obtain integral normal vector $\vec{c}$ of hyper-space spanned by W
- next check whether V is contained in hyper-space $\{\vec{v} \mid \vec{v} \cdot \vec{c} \leq 0\}$ or $\{\vec{v} \mid \vec{v} \cdot (-\vec{c}) \leq 0\}$
  - if $\vec{v}_i \cdot \vec{c} \leq 0$ for all i, then add $\vec{c}$ as row to A
  - if $\vec{v}_i \cdot \vec{c} \geq 0$ for all i, then add $-\vec{c}$ as row to A
- $cone\,(V) = \{\vec{x} \mid A\vec{x} \leq \vec{0}\}$

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

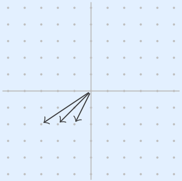
First direction: finitely generated implies polyhedral

- consider $cone\,(V)$ for $V = \{\vec{v}_1, \ldots, \vec{v}_m\} \subseteq \mathbb{Z}^n$
- consider every set $W \subseteq V$ of linearly independent vectors with $|W| = n - 1$
- obtain integral normal vector $\vec{c}$ of hyper-space spanned by W
- next check whether V is contained in hyper-space $\{\vec{v} \mid \vec{v} \cdot \vec{c} \leq 0\}$ or $\{\vec{v} \mid \vec{v} \cdot (-\vec{c}) \leq 0\}$
  - if $\vec{v}_i \cdot \vec{c} \leq 0$ for all i, then add $\vec{c}$ as row to A
  - if $\vec{v}_i \cdot \vec{c} \geq 0$ for all i, then add $-\vec{c}$ as row to A
- $cone\,(V) = \{\vec{x} \mid A\vec{x} \leq \vec{0}\}$
- bounds
  - each normal vector $\vec{c}$ can be computed via determinants

$\Longrightarrow$ obtain bound on numbers in $\vec{c}$ by using bounds on determinants

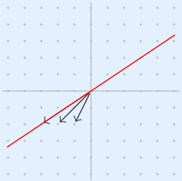
Example: Construction of Polyhedral Cone from Finitely Generated Cone

$$V = \left\{ \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\}$$



$$A = \begin{pmatrix} \quad \end{pmatrix}$$

Example: Construction of Polyhedral Cone from Finitely Generated Cone



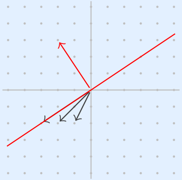
$$V = \left\{ \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\}$$

$$A = \begin{pmatrix} \quad \end{pmatrix}$$

- pick $W = \{\vec{w}\}$, $\vec{w} = \begin{pmatrix} -3 \\ -2 \end{pmatrix}$ and consider $span\, W$

Example: Construction of Polyhedral Cone from Finitely Generated Cone

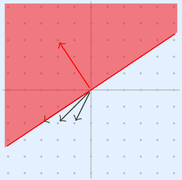
$$V = \left\{ \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\}$$



$$A = \begin{pmatrix} \quad \end{pmatrix}$$

- pick $W = \{\vec{w}\}$, $\vec{w} = \begin{pmatrix} -3 \\ -2 \end{pmatrix}$ and consider $span\,W$
- compute normal vector $\vec{c} = \begin{pmatrix} -2 & 3 \end{pmatrix}$

Example: Construction of Polyhedral Cone from Finitely Generated Cone

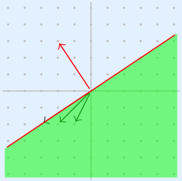


$$V = \left\{ \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\}$$

$$A = \begin{pmatrix} \\ \end{pmatrix}$$

- pick $W = \{\vec{w}\}$, $\vec{w} = \begin{pmatrix} -3 \\ -2 \end{pmatrix}$ and consider $span\, W$
- compute normal vector $\vec{c} = \begin{pmatrix} -2 & 3 \end{pmatrix}$
- if V is in same half-space, add $\pm\vec{c}$ to A

Example: Construction of Polyhedral Cone from Finitely Generated Cone



$$V = \left\{ \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\}$$

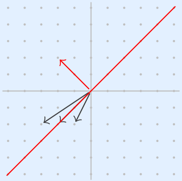
$$A = \begin{pmatrix} -2 & 3 \\ & \end{pmatrix}$$

- pick $W = \{\vec{w}\}$, $\vec{w} = \begin{pmatrix} -3 \\ -2 \end{pmatrix}$ and consider $span\, W$
- compute normal vector $\vec{c} = \begin{pmatrix} -2 & 3 \end{pmatrix}$
- if V is in same half-space, add $\pm\vec{c}$ to A

Example: Construction of Polyhedral Cone from Finitely Generated Cone

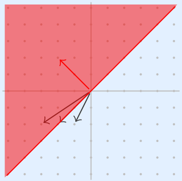
$$V = \left\{ \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\}$$

$$A = \begin{pmatrix} -2 & 3 \\ & \end{pmatrix}$$



- pick $W = \{\vec{w}\}$, $\vec{w} = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$ and consider $span\, W$
- compute normal vector $\vec{c} = \begin{pmatrix} -2 & 2 \end{pmatrix}$
- if V is in same half-space, add $\pm\vec{c}$ to A

Example: Construction of Polyhedral Cone from Finitely Generated Cone

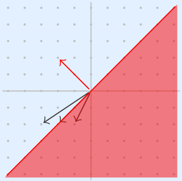


$$V = \left\{ \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\}$$

$$A = \begin{pmatrix} -2 & 3 \\ & \end{pmatrix}$$

- pick $W = \{\vec{w}\}$, $\vec{w} = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$ and consider *span* W
- compute normal vector $\vec{c} = \begin{pmatrix} -2 & 2 \end{pmatrix}$
- if V is in same half-space, add $\pm\vec{c}$ to A

Example: Construction of Polyhedral Cone from Finitely Generated Cone

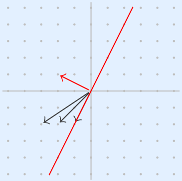


$$V = \left\{ \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\}$$

$$A = \begin{pmatrix} -2 & 3 \\ & \end{pmatrix}$$

- pick $W = \{\vec{w}\}$, $\vec{w} = \begin{pmatrix} -2 \\ -2 \end{pmatrix}$ and consider $span\, W$
- compute normal vector $\vec{c} = \begin{pmatrix} -2 & 2 \end{pmatrix}$
- if V is in same half-space, add $\pm\vec{c}$ to A

Example: Construction of Polyhedral Cone from Finitely Generated Cone

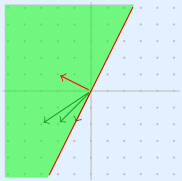


$$V = \left\{ \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\}$$

$$A = \begin{pmatrix} -2 & 3 \\ & \end{pmatrix}$$

- pick $W = \{\vec{w}\}$, $\vec{w} = \begin{pmatrix} -1 \\ -2 \end{pmatrix}$ and consider $span\, W$
- compute normal vector $\vec{c} = \begin{pmatrix} -2 & 1 \end{pmatrix}$
- if V is in same half-space, add $\pm\vec{c}$ to A

Example: Construction of Polyhedral Cone from Finitely Generated Cone



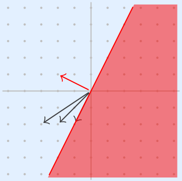
$$V = \left\{ \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\}$$

$$A = \begin{pmatrix} -2 & 3 \\ 2 & -1 \end{pmatrix}$$

- pick $W = \{\vec{w}\}$, $\vec{w} = \begin{pmatrix} -1 \\ -2 \end{pmatrix}$ and consider $\mathit{span}\, W$
- compute normal vector $\vec{c} = \begin{pmatrix} -2 & 1 \end{pmatrix}$
- if V is in same half-space, add $\pm\vec{c}$ to A

Example: Construction of Polyhedral Cone from Finitely Generated Cone

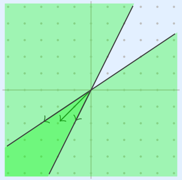
$$V = \left\{ \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\}$$



$$A = \begin{pmatrix} -2 & 3 \\ 2 & -1 \end{pmatrix}$$

- pick $W = \{\vec{w}\}$, $\vec{w} = \begin{pmatrix} -1 \\ -2 \end{pmatrix}$ and consider *span* W
- compute normal vector $\vec{c} = \begin{pmatrix} -2 & 1 \end{pmatrix}$
- if V is in same half-space, add $\pm\vec{c}$ to A

Example: Construction of Polyhedral Cone from Finitely Generated Cone



$$V = \left\{ \begin{pmatrix} -3 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\}$$

$$A = \begin{pmatrix} -2 & 3 \\ 2 & -1 \end{pmatrix}$$

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

Second direction: polyhedral implies finitely generated

- consider $\{\vec{x} \mid A\vec{x} \leq \vec{0}\}$

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

Second direction: polyhedral implies finitely generated

- consider $\{\vec{x} \mid A\vec{x} \leq \vec{0}\}$
- define W as the set of row vectors of A

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

Second direction: polyhedral implies finitely generated

- consider $\{\vec{x} \mid A\vec{x} \leq \vec{0}\}$
- define W as the set of row vectors of A
- by first direction obtain integral matrix B such that $cone(W) = \{\vec{x} \mid B\vec{x} \leq \vec{0}\}$

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

Second direction: polyhedral implies finitely generated

- consider $\{\vec{x} \mid A\vec{x} \leq \vec{0}\}$
- define W as the set of row vectors of A
- by first direction obtain integral matrix B such that $cone\,(W) = \{\vec{x} \mid B\vec{x} \leq \vec{0}\}$
- define V as the set of row vectors of B

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

Second direction: polyhedral implies finitely generated

- consider $\{\vec{x} \mid A\vec{x} \leq \vec{0}\}$
- define W as the set of row vectors of A
- by first direction obtain integral matrix B such that $cone(W) = \{\vec{x} \mid B\vec{x} \leq \vec{0}\}$
- define V as the set of row vectors of B
- $\{\vec{x} \mid A\vec{x} \leq \vec{0}\} = cone(V)$

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

Second direction: polyhedral implies finitely generated

- consider $\{\vec{x} \mid A\vec{x} \leq \vec{0}\}$
- define W as the set of row vectors of A
- by first direction obtain integral matrix B such that $cone(W) = \{\vec{x} \mid B\vec{x} \leq \vec{0}\}$
- define V as the set of row vectors of B
- $\{\vec{x} \mid A\vec{x} \leq \vec{0}\} = cone(V)$
- bounds carry over from first direction

Step 2b: Theorem of Farkas, Minkowski, Weyl

A cone is polyhedral iff it is finitely generated.

Second direction: polyhedral implies finitely generated

- consider $\{\vec{x} \mid A\vec{x} \leq \vec{0}\}$
- define W as the set of row vectors of A
- by first direction obtain integral matrix B such that $cone\,(W) = \{\vec{x} \mid B\vec{x} \leq \vec{0}\}$
- define V as the set of row vectors of B
- $\{\vec{x} \mid A\vec{x} \leq \vec{0}\} = cone\,(V)$
- bounds carry over from first direction

Step 4: Theorem of Farkas, Minkowski, Weyl (bounded version)

Let $C \subseteq \mathbb{R}^n$ be a polyhedral cone, given via an integral matrix A. Let b be a bound for all matrix entries, $b \geq |A_{ij}|$. Then C is generated by a finite set of integral vectors V whose entries are at most $\pm\sqrt{(n-1)^{n-1}} \cdot b^{n-1}$.

Theorem (Hadamard's Inequality)

- *Let A be a square matrix of dimension n such that $|A_{i,j}| \leq b$ for all i, j. Then $|det(A)| \leq \sqrt{n^n} \cdot b^n$.*

Theorem (Hadamard's Inequality)

- *Let A be a square matrix of dimension n such that $|A_{i,j}| \leq b$ for all i, j. Then $|det(A)| \leq \sqrt{n^n} \cdot b^n$.*
- *Whenever $n = 2^k$, then the bound is tight, i.e., there exists a matrix A of dimension n such that $det(A) = \sqrt{n^n} \cdot b^n = n^{n/2} \cdot b^n$.*

Theorem (Hadamard's Inequality)

- *Let A be a square matrix of dimension n such that $|A_{i,j}| \leq b$ for all i, j. Then $|det(A)| \leq \sqrt{n^n} \cdot b^n$.*
- *Whenever $n = 2^k$, then the bound is tight, i.e., there exists a matrix A of dimension n such that $det(A) = \sqrt{n^n} \cdot b^n = n^{n/2} \cdot b^n$.*

Proof

- uses results about Gram matrices

Theorem (Hadamard's Inequality)

- *Let A be a square matrix of dimension n such that $|A_{i,j}| \leq b$ for all i, j. Then $|det(A)| \leq \sqrt{n^n} \cdot b^n$.*
- *Whenever $n = 2^k$, then the bound is tight, i.e., there exists a matrix A of dimension n such that $det(A) = \sqrt{n^n} \cdot b^n = n^{n/2} \cdot b^n$.*

Proof

- uses results about Gram matrices
- construct matrices $A_0, A_1, A_2, \ldots, A_k$ of dimensions $2^0, 2^1, 2^2, \ldots, 2^k$ as follows:

$$A_0 = \begin{pmatrix} b \end{pmatrix}, A_1 = \begin{pmatrix} b & b \\ -b & b \end{pmatrix} = \begin{pmatrix} A_0 & A_0 \\ -A_0 & A_0 \end{pmatrix}, A_2 = \begin{pmatrix} A_1 & A_1 \\ -A_1 & A_1 \end{pmatrix}, \ldots$$

Theorem (Hadamard's Inequality)

- *Let A be a square matrix of dimension n such that $|A_{i,j}| \leq b$ for all i, j. Then $|det(A)| \leq \sqrt{n^n} \cdot b^n$.*
- *Whenever $n = 2^k$, then the bound is tight, i.e., there exists a matrix A of dimension n such that $det(A) = \sqrt{n^n} \cdot b^n = n^{n/2} \cdot b^n$.*

Proof

- uses results about Gram matrices
- construct matrices $A_0, A_1, A_2, \ldots, A_k$ of dimensions $2^0, 2^1, 2^2, \ldots, 2^k$ as follows:

$$A_0 = \begin{pmatrix} b \end{pmatrix}, A_1 = \begin{pmatrix} b & b \\ -b & b \end{pmatrix} = \begin{pmatrix} A_0 & A_0 \\ -A_0 & A_0 \end{pmatrix}, A_2 = \begin{pmatrix} A_1 & A_1 \\ -A_1 & A_1 \end{pmatrix}, \ldots$$

  obtain desired equality $det(A_k) = (2^k)^{2^k/2} \cdot b^{2^k}$ by induction on k:

$$det(A_{k+1}) = det(2 \cdot A_k \cdot A_k) = 2^{2^k} \cdot det(A_k)^2 = 2^{2^k} \cdot ((2^k)^{2^k/2} \cdot b^{2^k})^2 = (2^{k+1})^{2^{k+1}/2} \cdot b^{2^{k+1}}$$

Example Hadamard Matrix

$$\det \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{pmatrix} = 4096 = 8^4 \cdot 1^8$$

Outline

Kröning and Strichmann

- Section 5.3

Further Reading

Alexander Schrijver
Theory of linear and integer programming, Chapters 7, 16, 17, and 24
Wiley, 1998.

Important Concepts

- branch-and-bound
- cone (finitely generated or polyhedral)
- decomposition theorem for polyhedra
- Farkas–Minkowski–Weyl theorem
- Hadamard's inequality
- polyhedron
- small model property of LIA