CS156: The Calculus of Computation

Zohar Manna

Winter 2008

Chapter 8: Quantifier-free Linear Arithmetic

Decision Procedures for Quantifier-free Fragments

For theory T with signature Σ and axioms $\mathcal{A}$, decide if

$$F[x_1, \ldots, x_n] \quad \text{or} \quad \exists x_1, \ldots, x_n.\ F[x_1, \ldots, x_n] \quad \text{is} \quad T\text{-satisfiable}$$

$$\left[\begin{array}{l}\text{Decide if} \\ \qquad F[x_1, \ldots, x_n] \quad \text{or} \quad \forall x_1, \ldots, x_n.\ F[x_1, \ldots, x_n] \quad \text{is} \quad T\text{-valid}\end{array}\right]$$

where F is quantifier-free and $\text{free}(F) = \{x_1, \ldots, x_n\}$

Note: no quantifier alternations

Conjunctive Quantifier-free Fragment

We consider only conjunctive quantifier-free Σ-formulae, i.e., conjunctions of Σ-literals (Σ-atoms or negations of Σ-atoms).

For given arbitrary quantifier-free Σ-formula F, convert it into DNF Σ-formula

$$F_1 \vee \ldots \vee F_k$$

where each F_i conjunctive.

F is T-satisfiable iff at least one F_i is T-satisfiable.

Preliminary Concepts

Vector

variable n-vector

$$\overline{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

n-vector $\overline{a} \in \mathbb{Q}^n$

$$\overline{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$$

transpose

$$\overline{a}^{\mathsf{T}} = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}$$

Matrix

$m \times n$-matrix

$A \in \mathbb{Q}^{m \times n}$

$$A = \begin{bmatrix} a_{11} \cdots a_{1n} \\ \vdots \quad \ddots \quad \vdots \\ a_{m1} \cdots a_{mn} \end{bmatrix}$$

transpose

$$A^{\mathsf{T}} = \begin{bmatrix} a_{11} \cdots a_{m1} \\ \vdots \quad \ddots \quad \vdots \\ a_{1n} \cdots a_{mn} \end{bmatrix}$$

column, row

$$\begin{bmatrix} & a_{1j} & \\ & \vdots & \\ a_{i1} \cdots & a_{ij} & \cdots a_{in} \\ & \vdots & \\ & a_{mj} & \end{bmatrix}$$

Multiplication I

vector-vector

$$\overline{a}^\mathsf{T}\overline{b} = [a_1 \ \cdots \ a_n]\begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = \sum_{i=1}^{n} a_i b_i$$

matrix-vector

$$A\overline{x} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} a_{1i}x_i \\ \vdots \\ \sum_{i=1}^{n} a_{mi}x_i \end{bmatrix}$$

Multiplication II

matrix-matrix

$$\underbrace{\begin{bmatrix} & \vdots & \\ \cdots & a_{ik} & \cdots \\ & \vdots & \end{bmatrix}}_{A}\underbrace{\begin{bmatrix} & \vdots & \\ \cdots & b_{kj} & \cdots \\ & \vdots & \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} & \vdots & \\ \cdots & p_{ij} & \cdots \\ & \vdots & \end{bmatrix}}_{P}$$

where

$$p_{ij} = \overline{a}_i \overline{b}_j = \begin{bmatrix} a_{i1} & \cdots & a_{in} \end{bmatrix} \begin{bmatrix} b_{1j} \\ \vdots \\ b_{nj} \end{bmatrix} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

Special Vectors and Matrices

$\overline{0}$ - vector (column) of 0s

$\overline{1}$ - vector of 1s

Thus $\overline{1}^T \overline{x} = \sum_{i=1}^{n} x_i$

$$I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$$ <u>identity</u> matrix ($n \times n$)

Thus $IA = AI = A$, for $n \times n$ matrix A.

<u>unit vector</u> $e_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$ $\leftarrow$ ith (<u>Note</u>: matrix indices start at 1)

Vector Space - set S of vectors closed under addition and scaling of vectors. That is,

$$\text{if } \overline{v}_1, \ldots, \overline{v}_k \in S \quad \text{then} \quad \lambda_1 \overline{v}_1 + \cdots + \lambda_k \overline{v}_k \in S$$
$$\text{for } \lambda_1, \ldots, \lambda_n \in \mathbb{Q}$$

Linear Equation

$$F \;:\; A\overline{x} \;=\; \overline{b}$$

$m \times n$-matrix   variable n-vector m-vector

represents the $\Sigma_{\mathbb{Q}}$-formula

$$F : (a_{11}x_1 + \cdots + a_{1n}x_n = b_1) \wedge \cdots \wedge (a_{m1}x_1 + \cdots + a_{mn}x_n = b_m)$$

Gaussian Elimination

Find $\overline{x}$ s.t. $A\overline{x} = \overline{b}$ by elementary row operations

- Swap two rows
- Multiply a row by a nonzero scalar
- Add one row to another

Example 4 I

Solve

$$\begin{bmatrix} 3 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \\ 2 \end{bmatrix}$$

Construct the augmented matrix

$$\left[\begin{array}{ccc|c} 3 & 1 & 2 & 6 \\ 1 & 0 & 1 & 1 \\ 2 & 2 & 1 & 2 \end{array}\right]$$

Apply the row operations as follows:

Example 4 II

1. Add $-2\overline{a}_1 + 4\overline{a}_2$ to $\overline{a}_3$

$$\left[\begin{array}{ccc|c} 3 & 1 & 2 & 6 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & -6 \end{array}\right]$$

2. Add $-\overline{a}_1 + 2\overline{a}_2$ to $\overline{a}_2$

$$\left[\begin{array}{ccc|c} 3 & 1 & 2 & 6 \\ 0 & -1 & 1 & -3 \\ 0 & 0 & 1 & -6 \end{array}\right]$$

This augmented matrix is in triangular form.

Example 4 III

Solving

$$\begin{aligned} & & x_3 &= -6 \\ -x_2 + x_3 &= -3 &\Rightarrow\quad x_2 &= -3 \\ 3x_1 + x_2 + 2x_3 &= 6 &\Rightarrow\quad x_1 &= 7 \end{aligned}$$

The solution is $\overline{x} = \begin{bmatrix} 7 & -3 & -6 \end{bmatrix}^T$

Inverse Matrix

A^{-1} is the <u>inverse</u> matrix of square matrix A if

$$AA^{-1} = A^{-1}A = I$$

Square matrix A is nonsingular (invertible) if its inverse A^{-1} exists.

How to compute A^{-1} of A?

$$[A \mid I] \xrightarrow[\text{row operations}]{\text{elementary}} [I \mid A^{-1}]$$

How to compute kth column of A^{-1}?
Solve $A\bar{y} = e_k$, i.e.

$$\left[\begin{array}{c|c} & 0 \\ & \vdots \\ A & 1 \\ & \vdots \\ & 0 \end{array}\right] \xrightarrow[\text{solve using elementary row operations}]{} \begin{array}{c} \text{solve triangular matrix} \\ \bar{y} = \dots \\ (k\text{th column of } A^{-1}) \end{array}$$

Linear Inequalities I

Polyhedral Space

For $m \times n$-matrix A, variable n-vector $\overline{x}$, and m-vector $\bar{b}$, the $\Sigma_{\mathbb{Q}}$-formula

$$G : A\overline{x} \le \bar{b}, \quad \text{i.e.,} \quad G : \bigwedge_{i=1}^{m} a_{i1}x_1 + \cdots + a_{in}x_n \le b_i$$

describes a subset (space) of $\mathbb{Q}^n$, called a **polyhedron**.

Linear Inequalities II

Convex Space

An n-dimensional space $S \subseteq \mathbb{R}^n$ is **convex** if for all pairs of points $\bar{v}_1, \bar{v}_2 \in S$,

$$\lambda \bar{v}_1 + (1 - \lambda)\bar{v}_2 \in S \quad \text{for } \lambda \in [0, 1] \ .$$

$A\overline{x} \leq \bar{b}$ defines a **convex space**. For suppose $A\bar{v}_1 \leq \bar{b}$ and $A\bar{v}_2 \leq \bar{b}$; then also

$$A(\lambda \bar{v}_1 + (1 - \lambda)\bar{v}_2) \leq \bar{b} \ .$$

Linear Inequalities III

Vertex

Consider $m \times n$-matrix A where $m \geq n$.

An n-vector $\bar{v}$ is a **vertex** of $A\bar{x} \leq \bar{b}$ if there is

- a nonsingular $n \times n$-submatrix A_0 of A and
- corresponding n-subvector $\bar{b}_0$ of $\bar{b}$

such that

$$A_0 \bar{v} = \bar{b}_0 \ .$$

The rows a_{0_i} in A_0 and corresponding values b_{0_i} of $\bar{b}_0$ are the set of **defining constraints** of the vertex $\bar{v}$.

Two vertices are **adjacent** if they have defining constraint sets that differ in only one constraint.

Example I

Consider the linear inequality

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{-1} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{-1} & \mathbf{0} \\ 0 & 1 & 0 & -1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \\ z_1 \\ z_2 \end{bmatrix}}_{\overline{x}} \leq \underbrace{\begin{bmatrix} 0 \\ 0 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{3} \\ \mathbf{2} \\ 2 \end{bmatrix}}_{\overline{b}}$$

A is a 7×4-matrix, $\overline{b}$ is a 7-vector, and
$\overline{x}$ is a variable 4-vector representing the four variables $\{x, y, z_1, z_2\}$.

Example II

$\overline{v} = [2\ 1\ 0\ 0]^T$ is a vertex of the constraints. For the nonsingular submatrix A_0 (rows 3, 4, 5, 6 of A: defining constraints of $\overline{v}$),

$$\underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & -\mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & -\mathbf{1} & \mathbf{0} \end{bmatrix}}_{A_0} \underbrace{\begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{\overline{v}} = \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{3} \\ \mathbf{2} \end{bmatrix}}_{b_0}$$

Example III

Another vertex: $\overline{v}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$, since

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}}_{A_0} \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\overline{v}_0} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{b_0}$$

(rows 1,2,3,4 of A: defining constraints of $\overline{v}_0$)

Note: $\overline{v}$ and $\overline{v}_0$ are not adjacent; they are different in 2 defining constraints.

Linear Programming I

Optimization Problem

$$\begin{array}{lll} \textbf{max} & \overline{c}^{\mathsf{T}}\overline{x} & \ldots \text{objective function} \\ \textbf{subject to} & & \\ & A\overline{x} \leq \overline{b} & \ldots \text{constraints} \end{array}$$

$$\textit{Maximize} \sum_{i=1}^{n} c_i x_i$$

$$\textit{subject to} \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \leq \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Linear Programming II

Solution:

Find vertex $\overline{v}^*$ satisfying $A\overline{x} \leq \overline{b}$ and maximizing $\overline{c}^\mathsf{T}\overline{x}$.
That is,

$A\overline{v}^* \leq \overline{b}$ and
$\overline{c}^\mathsf{T}\overline{v}^*$ is maximal: $\overline{c}^\mathsf{T}\overline{v}^* \geq \overline{c}^\mathsf{T}\overline{u}$ for all $\overline{u}$ satisfying $A\overline{u} \leq \overline{b}$

- If $A\overline{x} \leq \overline{b}$ is unsatisfiable, then maximum is $-\infty$
- It's possible that the maximum is unbounded, then maximum is ∞

Example: Consider optimization problem:

$$\textbf{max} \quad \underbrace{\begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}}_{\overline{c}^{\mathsf{T}}} \underbrace{\begin{bmatrix} x \\ y \\ z_1 \\ z_2 \end{bmatrix}}_{\overline{x}}$$

subject to

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \\ z_1 \\ z_2 \end{bmatrix}}_{\overline{x}} \leq \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 3 \\ 2 \\ 2 \end{bmatrix}}_{\overline{b}}$$

Example (cont):

The objective function is

$$(x - z_1) + (y - z_2) \ .$$

The constraints are equivalent to the $\Sigma_{\mathbb{Q}}$-formula

$$\begin{aligned} x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z_1 \geq 0 \ \wedge \ z_2 \geq 0 \\ \wedge \ x + y \leq 3 \ \wedge \ x - z_1 \leq 2 \ \wedge \ y - z_2 \quad \leq \quad 2 \end{aligned}$$

Example: Linear Programming I

A company is producing two different products using three machines A, B, and C.

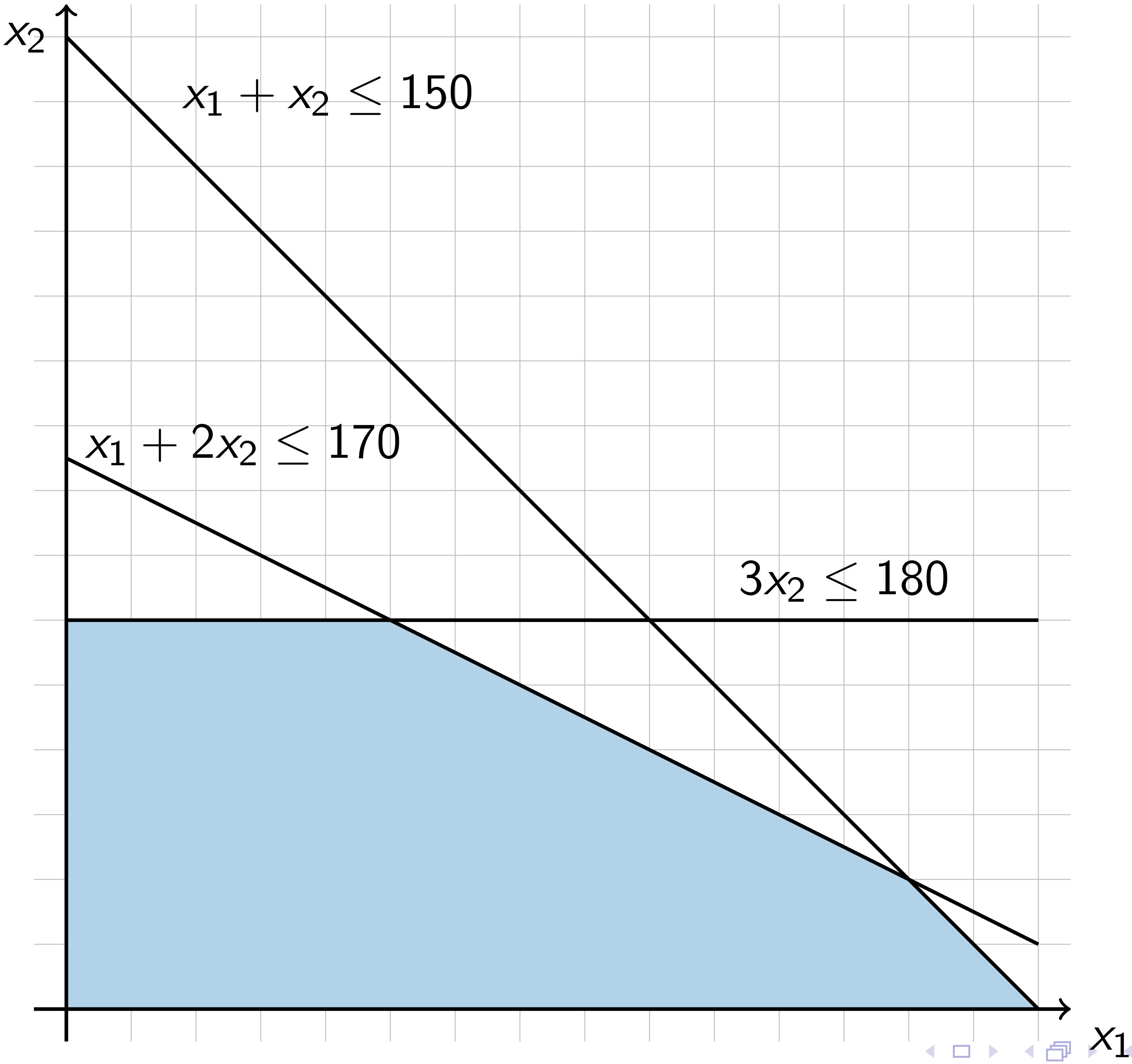
- Product 1 needs A for one, and B for one hour.
- Product 2 needs A for two, B for one, and C for three hours.
- Product 1 can be sold for \$300; Product 2 for \$500.
- Monthly availability of machines:
  A: 170 hours, B: 150 hours, C 180 hours.

Example: Linear Programming II

Let x_1 and x_2 denote the amount of product 1 and product 2, resp. We want to optimize $300x_1 + 500x_2$ subject to:

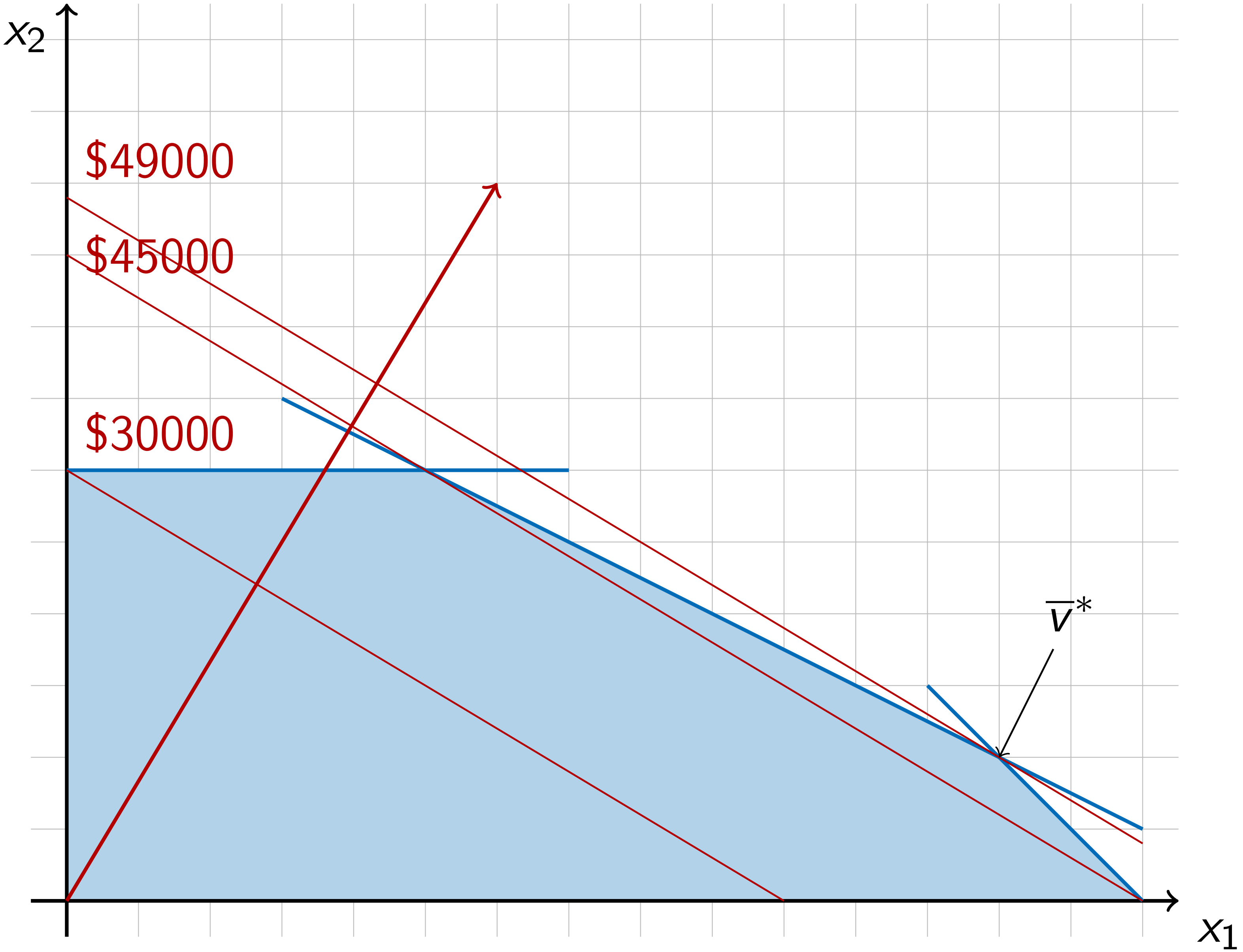
$$
\begin{aligned}
1x_1 + 2x_2 &\leq 170 && \text{Machine (A)} \\
1x_1 + 1x_2 &\leq 150 && \text{Machine (B)} \\
0x_1 + 3x_2 &\leq 180 && \text{Machine (C)} \\
x_1 \geq 0 \wedge x_2 &\geq 0
\end{aligned}
$$

Example: Linear Programming III



Example: Linear Programming IV

Optimize $300x_1 + 500x_2$:



Duality Theorem

For $m \times n$-matrix A, m-vector $\overline{b}$ and n-vector $\overline{c}$:

$$\max\{\overline{c}^{\mathsf{T}}\overline{x} \mid A\overline{x} \leq \overline{b} \ \wedge \ \overline{x} \geq \overline{0}\} = \min\{\overline{b}^{\mathsf{T}}\overline{y} \mid A^{\mathsf{T}}\overline{y} \geq \overline{c} \ \wedge \ \overline{y} \geq \overline{0}\}$$

if the constraints are satisfiable.

That is,

maximizing the function $c^{\mathsf{T}}\overline{x}$ over $A\overline{x} \leq \overline{b}$, $\overline{x} \geq \overline{0}$
(the <u>primal</u> form of the optimization problem)

is equivalent to

minimizing the function $\overline{b}^{\mathsf{T}}\overline{y}$ over $A^{\mathsf{T}}\overline{y} \geq \overline{c}$, $\overline{y} \geq \overline{0}$
(the <u>dual</u> form of the optimization problem)

<u>By convention</u>: when $A\overline{x} \leq b \ \wedge \ \overline{x} \geq 0$ unsatisfiable, the max is $-\infty$ and the min is ∞.

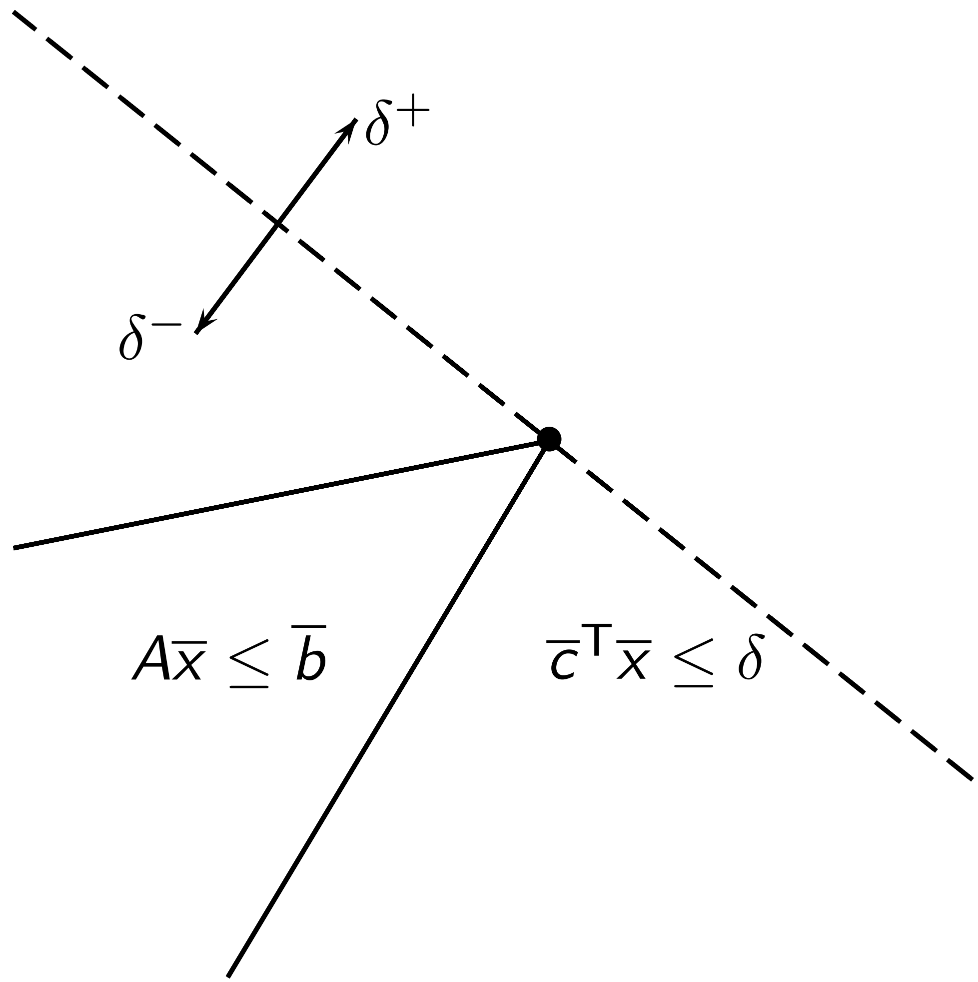


Figure: Visualization of the duality theorem

The region labeled $A\overline{x} \leq \overline{b}$ satisfies the inequality. The objective function $\overline{c}^\mathsf{T}\overline{x}$ is represented by the dashed line. Its value increases in the direction of the arrow labeled δ^+ and decreases in the direction of the arrow labeled δ^-.

Example: A Dual Problem

What is the value of a machine hour?
Let y_A, y_B, y_C be the values of machine A, B, and C.
The value of the machine hours to produce something $\geq$ the value of the product ($>$ if that product should not be produced).

$$
\begin{aligned}
y_A \geq 0 \ \wedge \ y_B \geq 0 \ \wedge \ y_C \geq 0 \\
1y_A + 1y_B + 0y_C \geq 300 \\
2y_A + 1y_B + 3y_C \geq 500
\end{aligned}
$$

We minimize the value $170y_A + 150y_B + 180y_C$ to get the value of a machine hour:

$$
\begin{aligned}
y_A = 200 \ \wedge \ y_B = 100 \ \wedge \ y_C = 0 \\
170y_A + 150y_B + 180y_C = 49000
\end{aligned}
$$

This is the dual problem. It has the same optimal value.

The Simplex Method

Consider linear program

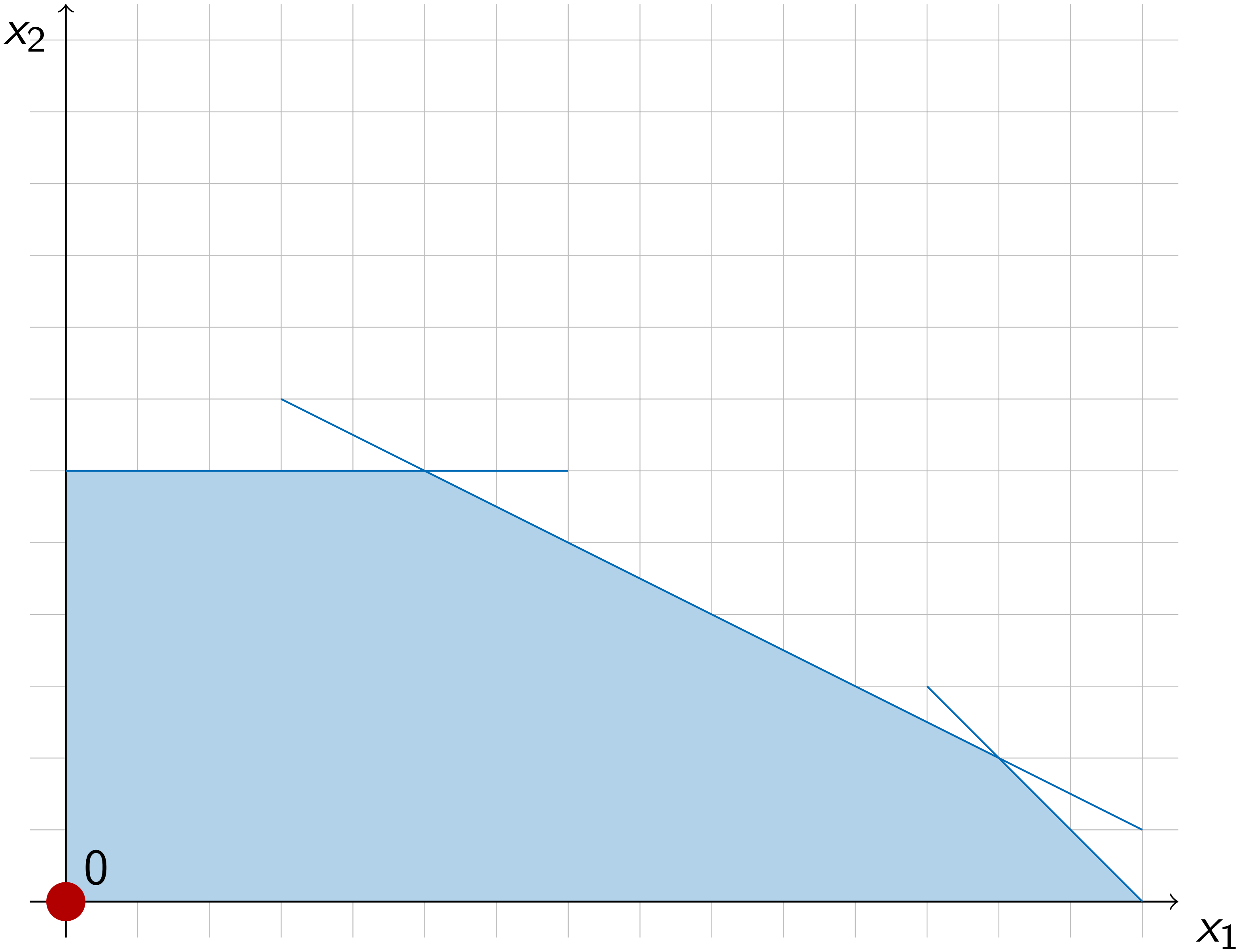
$$
\begin{aligned}
M \quad : \quad & \textbf{max } \bar{c}^{\mathsf{T}}\bar{x} \\
& \textbf{subject to } G : A\bar{x} \leq \bar{b}
\end{aligned}
$$

The **simplex method** solves the linear program in two main steps:

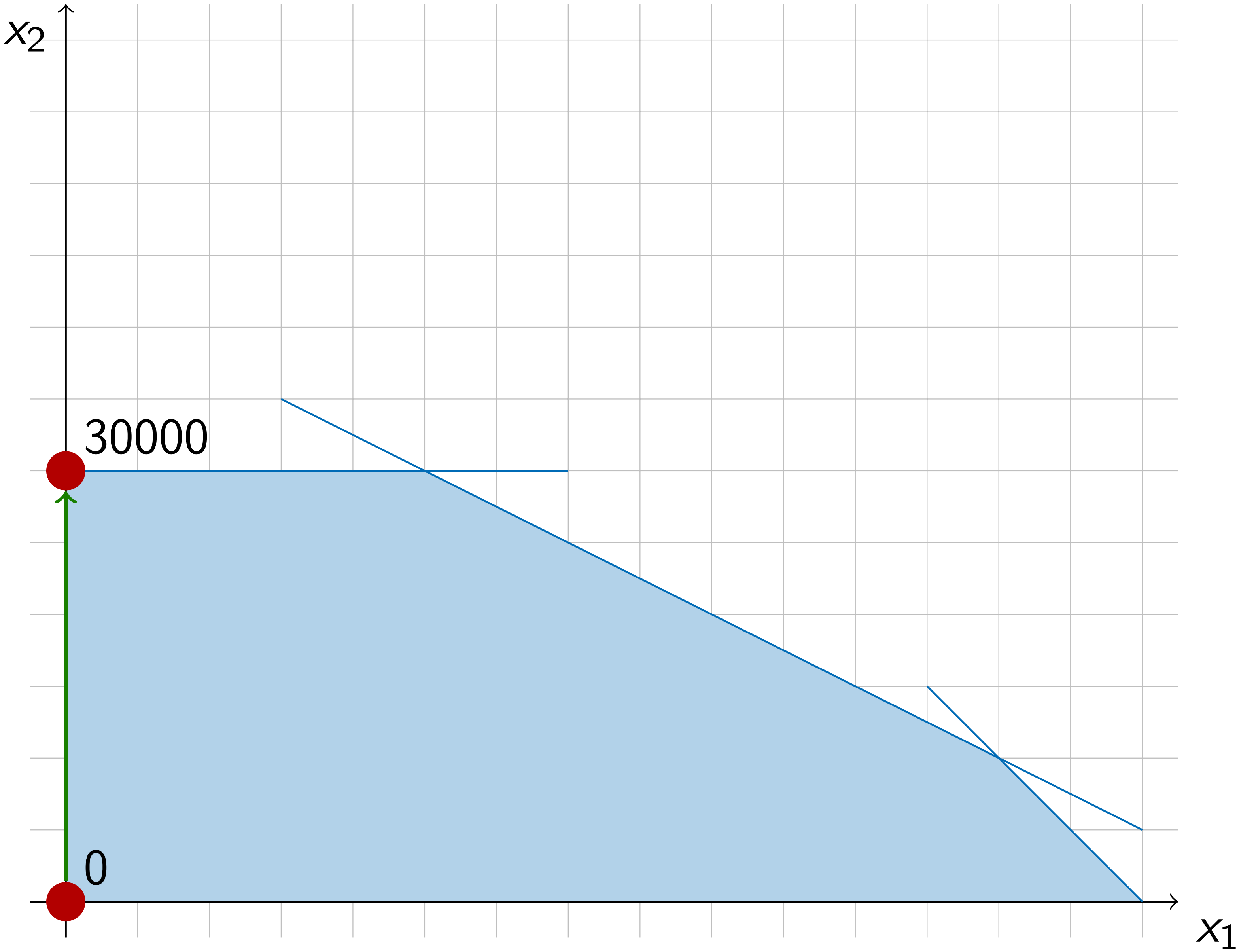
1. Obtain an initial vertex $\bar{v}_1$ of $A\bar{x} \leq \bar{b}$.
2. Iteratively traverse the vertices of $A\bar{x} \leq \bar{b}$, beginning at $\bar{v}_1$, in search of the vertex that maximizes $\bar{c}^{\mathsf{T}}\bar{x}$. On each iteration determine if $\bar{c}^{\mathsf{T}}\bar{v}_i > \bar{c}^{\mathsf{T}}\bar{v}_i'$ for the vertices $\bar{v}_i'$ adjacent to $\bar{v}_i$:
   - If not, move to one of the adjacent vertices $\bar{v}_i'$ with a greater objective value.
   - If so, halt and report $\bar{v}_i$ as the optimum point with value $\bar{c}^{\mathsf{T}}\bar{v}_i$.

The final vertex $\bar{v}_i$ is a **local optimum** since its adjacent vertices have lesser objective values. But because the space defined by $A\bar{x} \leq \bar{b}$ is convex, $\bar{v}_i$ is also the **global optimum**: it is the highest value attained by any point that satisfies the constraints.

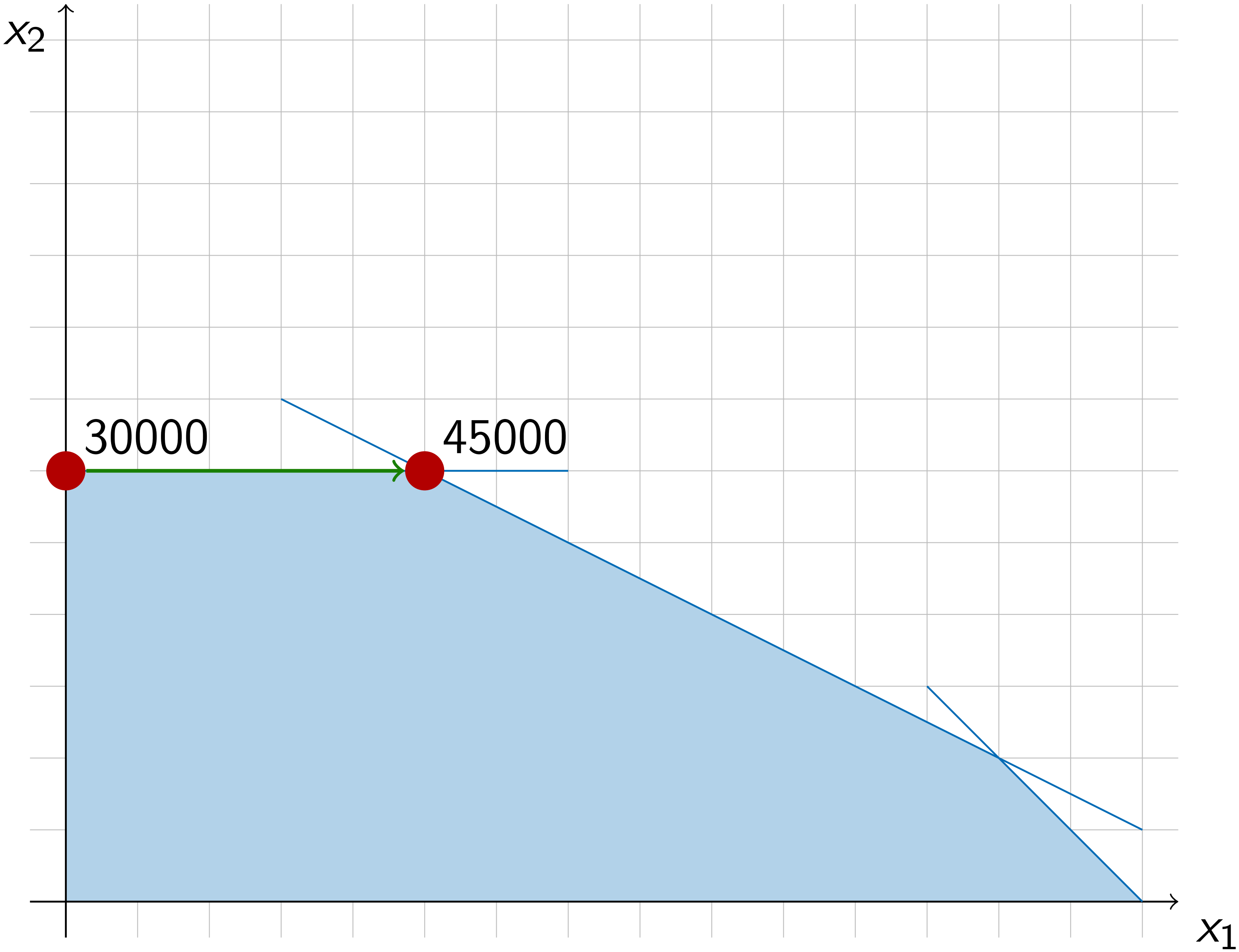
Example



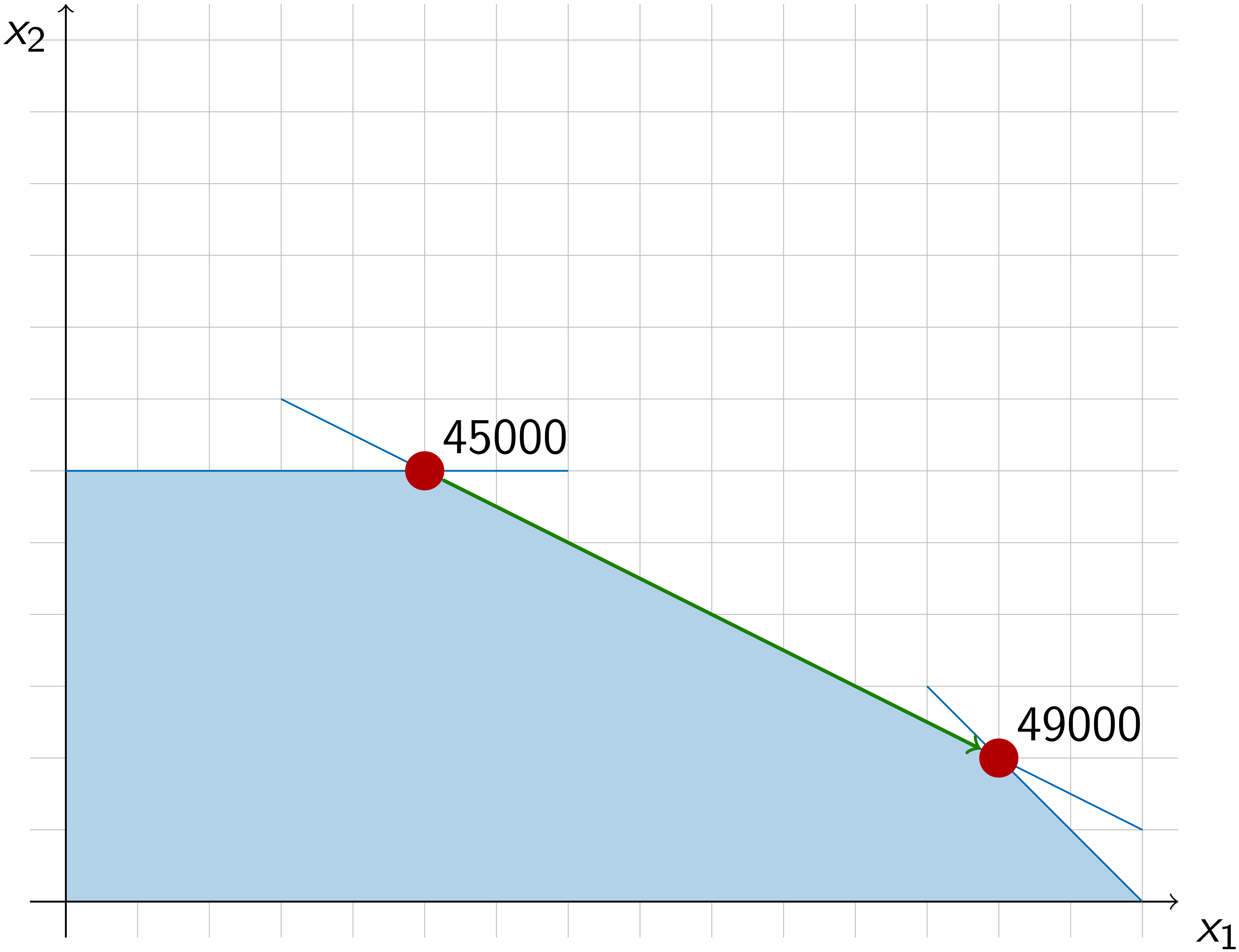
Example



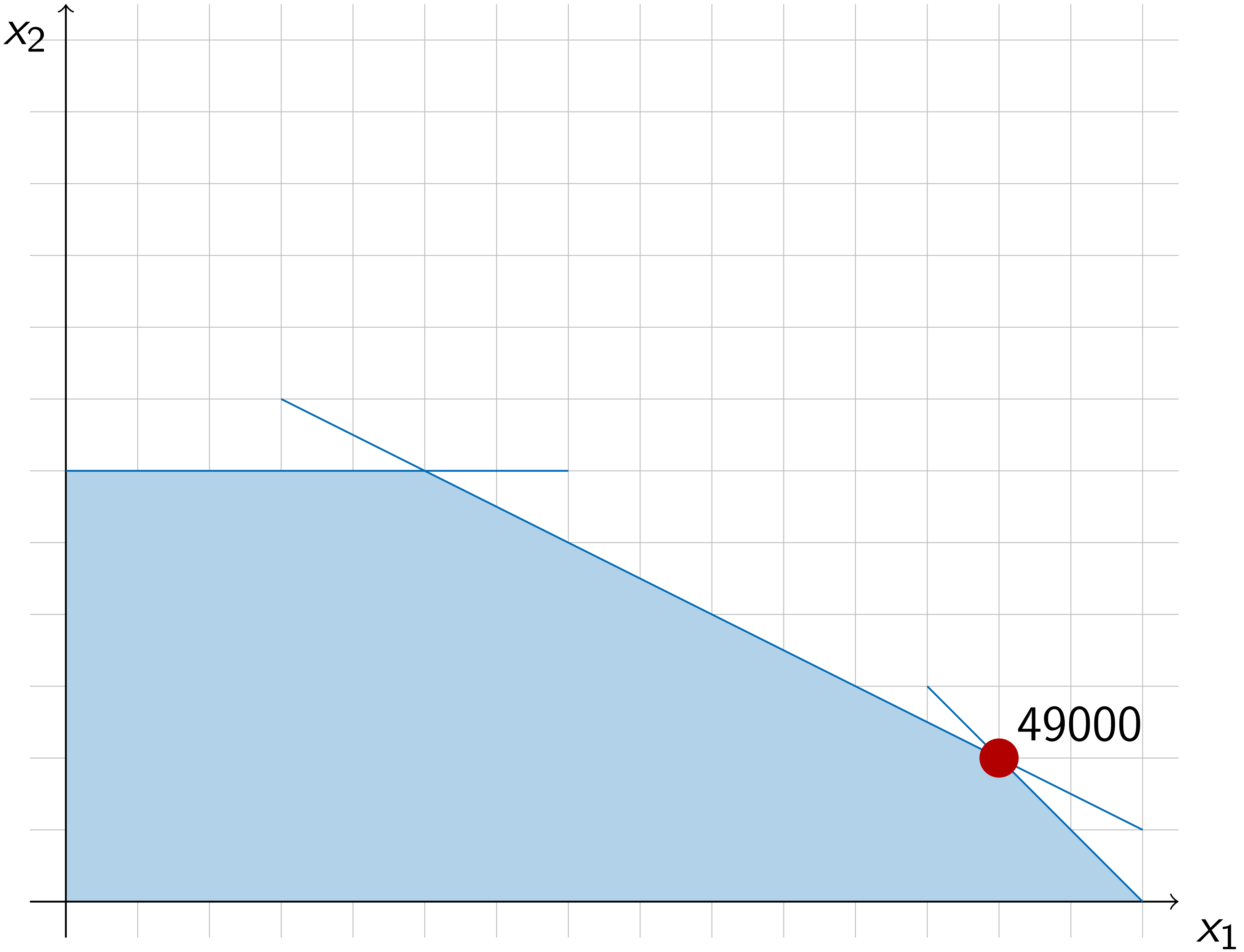
Example



Example



Example



How do we use optimization to determine satisfiability?

We are not interested in an *optimal* solution $\overline{x}$ such that

$$F : A\overline{x} \leq \overline{b} \ ;$$

we want *some* solution. However, this hard to find.

Idea: Transform F into an *optimization* problem with an initial (not-optimal) vertex $\overline{v}_1$ and a desired optimum v_F.

Apply the Simplex Method until an optimal vertex $\overline{v}^*$ is obtained.

The optimum value for $\overline{v}^*$ is v_F iff $F : Ax \leq b$ is satisfiable.

The solution can be computed from the optimal solution $\overline{x}$ of the optimization problem.

Outline of the Algorithm I

Determine if $\Sigma_{\mathbb{Q}}$-formula

$$F: \quad \bigwedge_{i=1}^{m} a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i$$
$$\wedge \quad \bigwedge_{i=1}^{\ell} \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n < \beta_i$$

is satisfiable.

Note: Equations

$$a_{i1}x_1 + \ldots + a_{in}x_n = b_i$$

are allowed; break them into two inequalities:

$$\begin{aligned} a_{i1}x_1 + \ldots + a_{in}x_n &\leq b_i \\ -a_{i1}x_1 + \ldots + -a_{in}x_n &\leq -b_i \end{aligned}$$

Outline of the Algorithm II

F is $T_{\mathbb{Q}}$-equivalent to the $\Sigma_{\mathbb{Q}}$-formula

$$
\begin{aligned}
F' : \quad & \bigwedge_{i=1}^{m} a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i \\
\wedge \quad & \bigwedge_{i=1}^{\ell} \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + z \leq \beta_i \\
\wedge \quad & z > 0
\end{aligned}
$$

Outline of the Algorithm III

To decide the $T_{\mathbb{Q}}$-satisfiability of F', solve the linear program

max z

subject to

$$\bigwedge_{i=1}^{m} a_{i1}x_1 + \ldots + a_{in}x_n \quad \leq \quad b_i$$

$$\bigwedge_{i=1}^{\ell} \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + z \quad \leq \quad \beta_i$$

F' is $T_{\mathbb{Q}}$-satisfiable iff the optimum is positive.

Outline of the Algorithm IV

When F does not contain any strict inequality literals, the corresponding linear program

max 1
subject to

$$\bigwedge_{i=1}^{m} a_{i1}x_1 + \ldots + a_{in}x_n \quad \leq \quad b_i$$

has optimum $-\infty$ iff the constraints are $T_{\mathbb{Q}}$-unsatisfiable,
1 iff the constraints are $T_{\mathbb{Q}}$-satisfiable.