Efficient resolution of logical models

ENSTA-IA303

Alexandre Chapoutot and Sergio Mover

ENSTA Paris

2020-2021

Lecture 4: Decision Procedures for the Theory of Linear Real and Integer Arithmetic

Main goals for today

In class[1]:

- Understand the theories that can express **linear** arithmetic constraints.
- How to decide the $\mathcal{T}$-satisfiability for quantifier-free formula in the Theory of Linear Real Arithmetic (LRA) - $\mathcal{L}A(\mathbb{Q})$
- Same for the Theory of Linear Integers Arithmetic (LIA) - $\mathcal{L}A(\mathbb{Z})$

---

[1] Main references:

- The Calculus of Computation [Bradley and Manna, 2007], Chapter 9 (Section 9.1, 9.2, 9.3)

Main goals for today

In class[1]:

- Understand the theories that can express **linear** arithmetic constraints.
- How to decide the $\mathcal{T}$-satisfiability for quantifier-free formula in the Theory of Linear Real Arithmetic (LRA) - $\mathcal{L}A(\mathbb{Q})$
- Same for the Theory of Linear Integers Arithmetic (LIA) - $\mathcal{L}A(\mathbb{Z})$

In the tutorial (somehow disconnected from LRA and LIA):

- Encoding for the job shop problem
- This week: just one encoding (more for next week, together with verification)

[1] Main references:

- The Calculus of Computation [Bradley and Manna, 2007], Chapter 9 (Section 9.1, 9.2, 9.3)

1. Decision Procedures for the Theory of Linear Real and Integer Arithmetic
   - Linear Arithmetic Theories
   - A decision procedure for LRA ($\mathcal{LA}(\mathbb{Q})$)
   - A decision procedure for LIA ($\mathcal{LA}(\mathbb{Z})$)
   - Remarks

Linear Arithmetic Theories

- Formulas are a Boolean combinations of **linear** atoms in the form:

$$\sum_{i}^{n} c_i \cdot x_j \bowtie b$$

Linear Arithmetic Theories

- Formulas are a Boolean combinations of **linear** atoms in the form:

$$\sum_{i}^{n} c_i \cdot x_j \bowtie b$$

- In Linear Real Arithmetic ($\mathcal{L}A(\mathbb{Q})$):

$$x_j \text{ is a real-valued variable, } c_i, b \in \mathbb{Q} \text{ and } \bowtie \in \{<, \leq, =\}$$

- In Linear Integer Arithmetic ($\mathcal{L}A(\mathbb{Z})$):

$$x_j \text{ is an integer-valued variable, } c_i, b \in \mathbb{Z} \text{ and } \bowtie \in \{<, \leq, =\}$$

Why do we specialize to the linear case?

- Several problems just require linear constraints:

Why do we specialize to the linear case?

- Several problems just require linear constraints:
  - program verification: some classes of programs do not have "complex" math (e.g., device driver)

Why do we specialize to the linear case?

- Several problems just require linear constraints:
  - program verification: some classes of programs do not have "complex" math (e.g., device driver)
  - real time systems: linear constraints are usually sufficient to express time constraints

Why do we specialize to the linear case?

- Several problems just require linear constraints:
  - program verification: some classes of programs do not have "complex" math (e.g., device driver)
  - real time systems: linear constraints are usually sufficient to express time constraints
  - ...
  - linear constraints may be already a sufficient approximations of a problem (e.g., 1-norm instead of euclidian distance, convex polyhedra to approximate "non-linear sets" as circles)

Why do we specialize to the linear case?

- Several problems just require linear constraints:
  - program verification: some classes of programs do not have "complex" math (e.g., device driver)
  - real time systems: linear constraints are usually sufficient to express time constraints
  - ...
  - linear constraints may be already a sufficient approximations of a problem (e.g., 1-norm instead of euclidian distance, convex polyhedra to approximate "non-linear sets" as circles)
- Why don't we just use theories for non-linear arithmetic?
  - We pay a price from the increased expressibility!

Why do we specialize to the linear case?

- Several problems just require linear constraints:
  - program verification: some classes of programs do not have "complex" math (e.g., device driver)
  - real time systems: linear constraints are usually sufficient to express time constraints
  - ...
  - linear constraints may be already a sufficient approximations of a problem (e.g., 1-norm instead of euclidian distance, convex polyhedra to approximate "non-linear sets" as circles)
- Why don't we just use theories for non-linear arithmetic?
  - We pay a price from the increased expressibility!
  - Non-Linear Integer arithmetic: satisfiability is *undecidable*

Why do we specialize to the linear case?

- Several problems just require linear constraints:
  - program verification: some classes of programs do not have "complex" math (e.g., device driver)
  - real time systems: linear constraints are usually sufficient to express time constraints
  - ...
  - linear constraints may be already a sufficient approximations of a problem (e.g., 1-norm instead of euclidian distance, convex polyhedra to approximate "non-linear sets" as circles)
- Why don't we just use theories for non-linear arithmetic?
  - We pay a price from the increased expressibility!
  - Non-Linear Integer arithmetic: satisfiability is *undecidable*
  - Non-Linear Real Arithmetic (i.e., polynomial inequalities, semi-algebraic sets): decidable, but the complexity is *doubly exponential*

1. Decision Procedures for the Theory of Linear Real and Integer Arithmetic
   - Linear Arithmetic Theories
   - A decision procedure for LRA ($\mathcal{L}A(\mathbb{Q})$)
   - A decision procedure for LIA ($\mathcal{L}A(\mathbb{Z})$)
   - Remarks

A decision procedure for LRA ($\mathcal{LA}(\mathbb{Q})$)

"Usual" settings (we obtain this with a pre-processing of the formula)

- Conjunction of inequalities of the form (over rational numbers)

$$\sum_{i}^{n} c_i \cdot x_j \bowtie b \qquad \bowtie = \{\geq, \leq\}$$

A decision procedure for LRA ($\mathcal{LA}(\mathbb{Q})$)

"Usual" settings (we obtain this with a pre-processing of the formula)

- Conjunction of inequalities of the form (over rational numbers)

$$\sum_{i}^{n} c_i \cdot x_j \bowtie b \qquad \bowtie = \{\geq, \leq\}$$

- What about equalities?

$x + 3y = 3$ can be rewritten as $x + 3y \leq 3 \wedge x + 3y \geq 3$

A decision procedure for LRA ($\mathcal{LA}(\mathbb{Q})$)

"Usual" settings (we obtain this with a pre-processing of the formula)

- Conjunction of inequalities of the form (over rational numbers)

$$\sum_i^n c_i \cdot x_j \bowtie b \qquad \bowtie = \{\geq, \leq\}$$

- What about equalities?

$$x + 3y = 3 \text{ can be rewritten as } x + 3y \leq 3 \wedge x + 3y \geq 3$$

- What about strict inequalities?
  Can be dealt with an extension of the algorithm we see here,
  see [Dutertre and De Moura, 2006]

A decision procedure for LRA ($\mathcal{LA}(\mathbb{Q})$)

"Usual" settings (we obtain this with a pre-processing of the formula)

- Conjunction of inequalities of the form (over rational numbers)

$$\sum_{i}^{n} c_i \cdot x_j \bowtie b \qquad \bowtie = \{\geq, \leq\}$$

- What about equalities?

$$x + 3y = 3 \text{ can be rewritten as } x + 3y \leq 3 \wedge x + 3y \geq 3$$

- What about strict inequalities?
  Can be dealt with an extension of the algorithm we see here,
  see [Dutertre and De Moura, 2006]

What algorithm can we use to find a solution (to a conjunction of $\mathcal{LA}(\mathbb{Q})$ inequalities)?

- Generalized Simplex from linear programming

$$\vec{A}\vec{x} = 0$$

$$l \leq x \leq u, \text{ for all } x_i \in \{m\}$$

From a conjunction of inequalities to the general simplex

- Our input is a set of m inequalities and we have a total of n variables

$$\sum_i^n c_{i,1} \cdot x_i \bowtie b_1$$

$$\dots$$

$$\sum_i^n c_{i,m} \cdot x_i \bowtie b_m$$

- We introduce m equalities (e.g., $\vec{A}\vec{x} = 0$) and m additional bound constraints from every j-inequality ($j \in \{1, \dots, m\}$):

$$\sum_i^n c_i \cdot x_j \bowtie b$$

1. We introduce a new slack variable s_j
2. We add the equality $\sum_i^n c_i \cdot x_j - s_j = 0$
3. We introduce the bound constraint:
   - $-\infty \leq s_j \leq b$, if $\bowtie = \leq$
   - $b \leq s_j \leq \infty$, if $\bowtie = \geq$

Example of general form

We can rewrite the set of constraints:

$$
\begin{aligned}
x + y &\geq 2 \\
2x - y &\geq 0 \\
-x + 2y &\geq 1
\end{aligned}
$$

In the general form:

$$
\begin{aligned}
x + y - \mathbf{s_1} &= 0 \\
2x - y - \mathbf{s_2} &= 0 \\
-x + 2y - \mathbf{s_3} &= 0 \\
\mathbf{s_1} &\geq 2 \\
\mathbf{s_2} &\geq 0 \\
\mathbf{s_3} &\geq 1
\end{aligned}
$$

s_1, s_2, s_3 are **additional** variables, x, y are **problem** variables.
The two problems are "equivalent"

General Form - problem

A system is in general form if is such that

$$\vec{A}\vec{x} = 0$$
$$l \leq s_j \leq u, \text{ for all } s_j \in \{1, \ldots, m\}$$

The size of $\vec{A}$ is $m \times (n+m)$ (with n problem variables and m additional variables)

$$\begin{aligned} x + y - \mathbf{s_1} &= 0 \\ 2x - y - \mathbf{s_2} &= 0 \\ -x + 2y - \mathbf{s_3} &= 0 \\ \mathbf{s_1} &\geq 2 \\ \mathbf{s_2} &\geq 0 \\ \mathbf{s_3} &\geq 1 \end{aligned}$$

$$\vec{A} = \begin{pmatrix} 1 & 1 & -1 & 0 & 0 \\ 2 & -1 & 0 & -1 & 0 \\ -2 & 2 & 0 & 0 & -1 \end{pmatrix}$$

Tableau and Bounds Constraints

Recall the matrix $\vec{A}$ has always the following form (with a diagonal sub-matrix):

$$\vec{A} = \begin{pmatrix} 1 & 1 & -1 & 0 & 0 \\ 2 & -1 & 0 & -1 & 0 \\ -2 & 2 & 0 & 0 & -1 \end{pmatrix}$$

The simplex algorithm represents the problem with a **tableau**, **bound constraints**, and **assignments**:

Tableau	Bounds	Assignment
$\begin{pmatrix} 1 & 1 \\ 2 & -1 \\ -1 & 2 \end{pmatrix}$	$2 \leq s_1 \leq \infty$ $0 \leq s_2 \leq \infty$ $1 \leq s_3 \leq \infty$	$x \rightarrow 0$ $y \rightarrow 0$ $s_1 \rightarrow 0$ $s_2 \rightarrow 0$ $s_3 \rightarrow 0$

Variables in the columns are **non-basic** while the variables in the rows are **basic**

The algorithm

The algorithm maintains the following invariants:

- $\vec{A}\vec{x} = 0$
- candidate solution α always consistent with the tableau
  Initially: α satisfies $\vec{A}\vec{x} = 0$

The algorithm checks if the bound for the **basic** variables is satisfied:

- If yes, then α is a feasible assignment (satisfiable!)
- If not, the algorithm update α with a pivoting operation
- From pivoting we can either infer a conflict (terminate, unsatisfiable!) or iterate checking the bound of the basic variables.

Example of Pivoting

Tableau	Bounds	Assignment
		$x \to 0$
$s_1 = 1x + 1y$	$2 \leq s_1 \leq \infty$	$y \to 0$
$s_2 = 2x - 1y$	$0 \leq s_2 \leq \infty$	$s_1 \to 0$
$s_3 = -1x + 2y$	$1 \leq s_3 \leq \infty$	$s_1 \to 0$

Example of Pivoting

Tableau	Bounds	Assignment
		$x \rightarrow 0$
$s_1 = 1x + 1y$	$2 \leq s_1 \leq \infty$	$y \rightarrow 0$
$s_2 = 2x - 1y$	$0 \leq s_2 \leq \infty$	$s_1 \rightarrow 0$
$s_3 = -1x + 2y$	$1 \leq s_3 \leq \infty$	$s_1 \rightarrow 0$

- $\alpha(s_1) = 0$, violates bound constraint $2 \leq s_1$

Example of Pivoting

Tableau	Bounds	Assignment
		$x \rightarrow 0$
$s_1 = 1x + 1y$	$2 \leq s_1 \leq \infty$	$y \rightarrow 0$
$s_2 = 2x - 1y$	$0 \leq s_2 \leq \infty$	$s_1 \rightarrow 0$
$s_3 = -1x + 2y$	$1 \leq s_3 \leq \infty$	$s_1 \rightarrow 0$

- $\alpha(s_1) = 0$, violates bound constraint $2 \leq s_1$
- We can select x as *pivot column* - Pivoting: "swap" s_i and x_j

Example of Pivoting

Tableau	Bounds	Assignment
		$x \to 0$
$s_1 = 1x + 1y$	$2 \leq s_1 \leq \infty$	$y \to 0$
$s_2 = 2x - 1y$	$0 \leq s_2 \leq \infty$	$s_1 \to 0$
$s_3 = -1x + 2y$	$1 \leq s_3 \leq \infty$	$s_1 \to 0$

- $\alpha(s_1) = 0$, violates bound constraint $2 \leq s_1$
- We can select x as *pivot column* - Pivoting: "swap" s_i and x_j
  - Ok, because $1 > 0$ and $\alpha(x) \leq \infty$

Example of Pivoting

Tableau	Bounds	Assignment
		$x \rightarrow 0$
$s_1 = 1x + 1y$	$2 \leq s_1 \leq \infty$	$y \rightarrow 0$
$s_2 = 2x - 1y$	$0 \leq s_2 \leq \infty$	$s_1 \rightarrow 0$
$s_3 = -1x + 2y$	$1 \leq s_3 \leq \infty$	$s_1 \rightarrow 0$

- $\alpha(s_1) = 0$, violates bound constraint $2 \leq s_1$
- We can select x as *pivot column* - Pivoting: "swap" s_i and x_j
  - Ok, because $1 > 0$ and $\alpha(x) \leq \infty$
  - Solve equation for x: $x = s_1 - y$

Example of Pivoting

Tableau	Bounds	Assignment
		$x \to 0$
$x = 1s_1 - 1y$	$2 \leq s_1 \leq \infty$	$y \to 0$
$s_2 = 2x - 1y$	$0 \leq s_2 \leq \infty$	$s_1 \to 0$
$s_3 = -1x + 2y$	$1 \leq s_3 \leq \infty$	$s_1 \to 0$

- $\alpha(s_1) = 0$, violates bound constraint $2 \leq s_1$
- We can select x as *pivot column* - Pivoting: "swap" s_i and x_j
  - Ok, because $1 > 0$ and $\alpha(x) \leq \infty$
  - Solve equation for x: $x = s_1 - y$
  - Replace $s1 = 1x + 1y$ with $x = s_1 - y$

Example of Pivoting

Tableau	Bounds	Assignment
		$x \to 0$
$x = 1s_1 - 1y$	$2 \leq s_1 \leq \infty$	$y \to 0$
$s_2 = 2s_1 - 3y$	$0 \leq s_2 \leq \infty$	$s_1 \to 0$
$s_3 = -s_1 + 3y$	$1 \leq s_3 \leq \infty$	$s_1 \to 0$

- $\alpha(s_1) = 0$, violates bound constraint $2 \leq s_1$
- We can select x as *pivot column* - Pivoting: "swap" s_i and x_j
  - Ok, because $1 > 0$ and $\alpha(x) \leq \infty$
  - Solve equation for x: $x = s_1 - y$
  - Replace $s1 = 1x + 1y$ with $x = s_1 - y$
  - Replace the new x with s_1 in the other rows

Example of Pivoting

Tableau	Bounds	Assignment
		$x \rightarrow 0$
$x = 1s_1 - 1y$	$2 \leq s_1 \leq \infty$	$y \rightarrow 0$
$s_2 = 2s_1 - 3y$	$0 \leq s_2 \leq \infty$	$s_1 \rightarrow 0$
$s_3 = -s_1 + 3y$	$1 \leq s_3 \leq \infty$	$s_1 \rightarrow 0$

- $\alpha(s_1) = 0$, violates bound constraint $2 \leq s_1$
- We can select x as *pivot column* - Pivoting: "swap" s_i and x_j
  - Ok, because $1 > 0$ and $\alpha(x) \leq \infty$
  - Solve equation for x: $x = s_1 - y$
  - Replace $s1 = 1x + 1y$ with $x = s_1 - y$
  - Replace the new x with s_1 in the other rows
  - Compute new bounds:

Example of Pivoting

Tableau	Bounds	Assignment
		$x \rightarrow 0$
$x = 1s_1 - 1y$	$2 \leq s_1 \leq \infty$	$y \rightarrow 0$
$s_2 = 2s_1 - 3y$	$0 \leq s_2 \leq \infty$	$s_1 \rightarrow 2$
$s_3 = -s_1 + 3y$	$1 \leq s_3 \leq \infty$	$s_1 \rightarrow 0$

- $\alpha(s_1) = 0$, violates bound constraint $2 \leq s_1$
- We can select x as *pivot column* - Pivoting: "swap" s_i and x_j
  - Ok, because $1 > 0$ and $\alpha(x) \leq \infty$
  - Solve equation for x: $x = s_1 - y$
  - Replace $s1 = 1x + 1y$ with $x = s_1 - y$
  - Replace the new x with s_1 in the other rows
  - Compute new bounds:
    - ⋆ $\alpha(s_1) = 2$ (in this case, the lower bound)

Example of Pivoting

Tableau	Bounds	Assignment
		$x \rightarrow 2$
		$y \rightarrow 0$
$x = 1s_1 - 1y$	$2 \leq s_1 \leq \infty$	$s_1 \rightarrow 2$
$s_2 = 2s_1 - 3y$	$0 \leq s_2 \leq \infty$	$s_1 \rightarrow 0$
$s_3 = -s_1 + 3y$	$1 \leq s_3 \leq \infty$	$s_2 \rightarrow 0$
		$s_3 \rightarrow 0$

- $\alpha(s_1) = 0$, violates bound constraint $2 \leq s_1$
- We can select x as *pivot column* - Pivoting: "swap" s_i and x_j
  - Ok, because $1 > 0$ and $\alpha(x) \leq \infty$
  - Solve equation for x: $x = s_1 - y$
  - Replace $s1 = 1x + 1y$ with $x = s_1 - y$
  - Replace the new x with s_1 in the other rows
  - Compute new bounds:
    - $\alpha(s_1) = 2$ (in this case, the lower bound)
    - Increase $\alpha(x)$ by $\alpha(s_1 - y) = 2 - 0 = 2$, so $\alpha(x) = 2$

Example of Pivoting

Tableau	Bounds	Assignment
		$x \to 2$
		$y \to 0$
$x = 1s_1 - 1y$	$2 \leq s_1 \leq \infty$	$s_1 \to 2$
$s_2 = 2s_1 - 3y$	$0 \leq s_2 \leq \infty$	$s_1 \to 0$
$s_3 = -s_1 + 3y$	$1 \leq s_3 \leq \infty$	$s_2 \to 4$
		$s_3 \to -2$

- $\alpha(s_1) = 0$, violates bound constraint $2 \leq s_1$
- We can select x as *pivot column* - Pivoting: "swap" s_i and x_j
  - Ok, because $1 > 0$ and $\alpha(x) \leq \infty$
  - Solve equation for x: $x = s_1 - y$
  - Replace $s1 = 1x + 1y$ with $x = s_1 - y$
  - Replace the new x with s_1 in the other rows
  - Compute new bounds:
    - $\alpha(s_1) = 2$ (in this case, the lower bound)
    - Increase $\alpha(x)$ by $\alpha(s_1 - y) = 2 - 0 = 2$, so $\alpha(x) = 2$
    - Update bounds for s_2 and s_3

Pivoting

- Find a basic variable x_i that violates its bound (suppose, $\alpha(x_i) > u_i$)
- Find a non-basic variable x_j that make $\alpha(x_i)$ satisfy the bound:
  - Can increase the value of x_i: $c_{i,j} > 0$ and $\alpha x_j < u_j$.
  - Can decrease the value of x_i: $c_{i,j} < 0$ and $\alpha x_j > l_j$.
  - If such variable does not exist, return unsat
- Pivoting:
  - $c_{i,j}$ is the pivot element
  - x_j is the pivot column
  - x_i is the pivot row
  - "Swap" x_j with x_i

Pivoting x_i and x_j

Same step as in Gaussian elimination:

- Replace i row:
  - i-th row in the tableau: $x_i = c_{i,j}x_j + \sum_{k \neq j} c_{i,k}x_k$
  - Becomes: $x_j = \frac{1}{c_{i,j}}x_i + \sum_{k \neq j} \frac{-c_{i,j}}{c_{i,k}}x_k$
- Replace x_j with $\frac{1}{c_{i,j}}x_i + \sum_{k \neq j} \frac{-c_{i,j}}{c_{i,k}}x_k$ in all the other columns

Update assignment α:

- $\alpha(x_i)$ is the upper (lower) bound (the bound that was violated).
- $\alpha(x_j)$ is incremented by $\frac{u_i - \alpha(x_i)}{c_{i,j}}$ (or decremented, depending on the violated bound)
- Update the other basic variables
- x_j becomes basic, so it may violate some bounds.

Example of Simplex (cont)

Tableau	Bounds	Assignment
		$x \to 2$
$x = 1s_1 - 1y$	$2 \le s_1 \le \infty$	$y \to 0$
$s_2 = 2s_1 - 3y$	$0 \le s_2 \le \infty$	$s_1 \to 2$
$s_3 = -s_1 + 3y$	$1 \le s_3 \le \infty$	$s_2 \to 4$
		$s_3 \to -2$

Example of Simplex (cont)

Tableau	Bounds	Assignment
		$x \to 2$
$x = 1s_1 - 1y$	$2 \leq s_1 \leq \infty$	$y \to 0$
$s_2 = 2s_1 - 3y$	$0 \leq s_2 \leq \infty$	$s_1 \to 2$
$s_3 = -s_1 + 3y$	$1 \leq s_3 \leq \infty$	$s_2 \to 4$
		$s_3 \to -2$

- $\alpha(s_3) = 0$, violates bound constraint $1 \leq s_3$

Example of Simplex (cont)

Tableau	Bounds	Assignment
		$x \rightarrow 2$
$x = 1s_1 - 1y$	$2 \leq s_1 \leq \infty$	$y \rightarrow 0$
$s_2 = 2s_1 - 3y$	$0 \leq s_2 \leq \infty$	$s_1 \rightarrow 2$
$s_3 = -s_1 + 3y$	$1 \leq s_3 \leq \infty$	$s_2 \rightarrow 4$
		$s_3 \rightarrow -2$

- $\alpha(s_3) = 0$, violates bound constraint $1 \leq s_3$
- We can seletc y as *pivot column*

Example of Simplex (cont)

Tableau	Bounds	Assignment
		$x \to 2$
$x = 1s_1 - 1y$	$2 \leq s_1 \leq \infty$	$y \to 0$
$s_2 = 2s_1 - 3y$	$0 \leq s_2 \leq \infty$	$s_1 \to 2$
$s_3 = -s_1 + 3y$	$1 \leq s_3 \leq \infty$	$s_2 \to 4$
		$s_3 \to -2$

- $\alpha(s_3) = 0$, violates bound constraint $1 \leq s_3$
- We can seletc y as *pivot column*
  - Ok, because $3 > 0$ and $\alpha(y) \leq \infty$

Example of Simplex (cont)

Tableau	Bounds	Assignment
		$x \to 2$
$x = 1s_1 - 1y$	$2 \leq s_1 \leq \infty$	$y \to 0$
$s_2 = 2s_1 - 3y$	$0 \leq s_2 \leq \infty$	$s_1 \to 2$
$s_3 = -s_1 + 3y$	$1 \leq s_3 \leq \infty$	$s_2 \to 4$
		$s_3 \to -2$

- $\alpha(s_3) = 0$, violates bound constraint $1 \leq s_3$
- We can seletc y as *pivot column*
  - Ok, because $3 > 0$ and $\alpha(y) \leq \infty$
  - Solve equation for y: $y = \frac{1}{3}s_3 + \frac{1}{3}x$

Example of Simplex (cont)

Tableau	Bounds	Assignment
$x = 1s_1 - 1y$	$2 \leq s_1 \leq \infty$	$x \to 2$
$s_2 = 2s_1 - 3y$	$0 \leq s_2 \leq \infty$	$y \to 0$
$y = \frac{1}{3}s_3 + \frac{1}{3}s_1$	$1 \leq s_3 \leq \infty$	$s_1 \to 2$
		$s_2 \to 4$
		$s_3 \to -2$

- $\alpha(s_3) = 0$, violates bound constraint $1 \leq s_3$
- We can seletc y as *pivot column*
  - Ok, because $3 > 0$ and $\alpha(y) \leq \infty$
  - Solve equation for y: $y = \frac{1}{3}s_3 + \frac{1}{3}x$
  - Replace $s_3 = -1x + 3y$ with $y = \frac{1}{3}s_1 + \frac{1}{3}s_3$

Example of Simplex (cont)

Tableau	Bounds	Assignment
		$x \to 2$
$x = \frac{2}{3}s_1 - \frac{1}{3}s_3$	$2 \leq s_1 \leq \infty$	$y \to 0$
$s_2 = s_1 - s_3$	$0 \leq s_2 \leq \infty$	$s_1 \to 2$
$y = \frac{1}{3}s_1 + \frac{1}{3}s_3$	$1 \leq s_3 \leq \infty$	$s_2 \to 4$
		$s_3 \to -2$

- $\alpha(s_3) = 0$, violates bound constraint $1 \leq s_3$
- We can seletc y as *pivot column*
  - Ok, because $3 > 0$ and $\alpha(y) \leq \infty$
  - Solve equation for y: $y = \frac{1}{3}s_3 + \frac{1}{3}x$
  - Replace $s_3 = -1x + 3y$ with $y = \frac{1}{3}s_1 + \frac{1}{3}s_3$
  - Replace the new y with s_3 in the other rows

Example of Simplex (cont)

Tableau	Bounds	Assignment
$x = \frac{2}{3}s_1 - \frac{1}{3}s_3$		$x \rightarrow 2$
$s_2 = s_1 - s_3$	$2 \leq s_1 \leq \infty$	$y \rightarrow 0$
$y = \frac{1}{3}s_1 + \frac{1}{3}s_3$	$0 \leq s_2 \leq \infty$	$s_1 \rightarrow 2$
	$1 \leq s_3 \leq \infty$	$s_2 \rightarrow 4$
		$s_3 \rightarrow -2$

- $\alpha(s_3) = 0$, violates bound constraint $1 \leq s_3$
- We can seletc y as *pivot column*
  - Ok, because $3 > 0$ and $\alpha(y) \leq \infty$
  - Solve equation for y: $y = \frac{1}{3}s_3 + \frac{1}{3}x$
  - Replace $s_3 = -1x + 3y$ with $y = \frac{1}{3}s_1 + \frac{1}{3}s_3$
  - Replace the new y with s_3 in the other rows
  - Compute new bounds:

Example of Simplex (cont)

Tableau	Bounds	Assignment
		$x \to 2$
$x = \frac{2}{3}s_1 - \frac{1}{3}s_3$	$2 \leq s_1 \leq \infty$	$y \to 0$
$s_2 = s_1 - s_3$	$0 \leq s_2 \leq \infty$	$s_1 \to 2$
$y = \frac{1}{3}s_1 + \frac{1}{3}s_3$	$1 \leq s_3 \leq \infty$	$s_2 \to 4$
		$s_3 \to -2$

- $\alpha(s_3) = 0$, violates bound constraint $1 \leq s_3$
- We can seletc y as *pivot column*
  - Ok, because $3 > 0$ and $\alpha(y) \leq \infty$
  - Solve equation for y: $y = \frac{1}{3}s_3 + \frac{1}{3}x$
  - Replace $s_3 = -1x + 3y$ with $y = \frac{1}{3}s_1 + \frac{1}{3}s_3$
  - Replace the new y with s_3 in the other rows
  - Compute new bounds:
    - $\alpha(s_3) = 1$ (in this case, the lower bound)

Example of Simplex (cont)

Tableau	Bounds	Assignment
$x = \frac{2}{3}s_1 - \frac{1}{3}s_3$	$2 \leq s_1 \leq \infty$	$x \to 1$
$s_2 = s_1 - s_3$	$0 \leq s_2 \leq \infty$	$y \to 1$
$y = \frac{1}{3}s_1 + \frac{1}{3}s_3$	$1 \leq s_3 \leq \infty$	$s_1 \to 2$
		$s_2 \to 1$
		$s_3 \to 1$

- $\alpha(s_3) = 0$, violates bound constraint $1 \leq s_3$
- We can seletc y as *pivot column*
  - Ok, because $3 > 0$ and $\alpha(y) \leq \infty$
  - Solve equation for y: $y = \frac{1}{3}s_3 + \frac{1}{3}x$
  - Replace $s_3 = -1x + 3y$ with $y = \frac{1}{3}s_1 + \frac{1}{3}s_3$
  - Replace the new y with s_3 in the other rows
  - Compute new bounds:
    - $\alpha(s_3) = 1$ (in this case, the lower bound)
    - Increase $\alpha(y)$ by $\alpha(\frac{1}{3}s_1 + \frac{1}{3}s_3) = \frac{2}{3} + \frac{1}{3} = 1$, so $\alpha(y) = 1$

Example of Simplex (cont)

Tableau	Bounds	Assignment
$x = \frac{2}{3}s_1 - \frac{1}{3}s_3$	$2 \leq s_1 \leq \infty$	$x \to 1$
$s_2 = s_1 - s_3$	$0 \leq s_2 \leq \infty$	$y \to 1$
$y = \frac{1}{3}s_1 + \frac{1}{3}s_3$	$1 \leq s_3 \leq \infty$	$s_1 \to 2$
		$s_2 \to 1$
		$s_3 \to 1$

- $\alpha(s_3) = 0$, violates bound constraint $1 \leq s_3$
- We can seletc y as *pivot column*
  - Ok, because $3 > 0$ and $\alpha(y) \leq \infty$
  - Solve equation for y: $y = \frac{1}{3}s_3 + \frac{1}{3}x$
  - Replace $s_3 = -1x + 3y$ with $y = \frac{1}{3}s_1 + \frac{1}{3}s_3$
  - Replace the new y with s_3 in the other rows
  - Compute new bounds:
    - $\alpha(s_3) = 1$ (in this case, the lower bound)
    - Increase $\alpha(y)$ by $\alpha(\frac{1}{3}s_1 + \frac{1}{3}s_3) = \frac{2}{3} + \frac{1}{3} = 1$, so $\alpha(y) = 1$
    - Update $\alpha(x) = 1$ and $\alpha(s_2) = 1$

Simplex - some remarks

- Generalized simplex: n variables, m equations
- In practice: a sequence of pivot steps to find a feasible bound
- The algorithm runtime can be exponential (in the number of variables) in the worst case. However, good performance in practice

1. Decision Procedures for the Theory of Linear Real and Integer Arithmetic
   - Linear Arithmetic Theories
   - A decision procedure for LRA ($\mathcal{L}A(\mathbb{Q})$)
   - A decision procedure for LIA ($\mathcal{L}A(\mathbb{Z})$)
   - Remarks

A decision procedure for LIA ($\mathcal{LA}(\mathbb{Z})$)

- conjunction of inequalities of the form (over Integer numbers)

$$\sum_{i}^{n} c_i \cdot x_j \bowtie b \qquad \bowtie = \{\geq, \leq\}$$

A decision procedure for LIA ($\mathcal{LA}(\mathbb{Z})$)

- conjunction of inequalities of the form (over Integer numbers)

$$\sum_{i}^{n} c_i \cdot x_j \bowtie b \qquad \bowtie = \{\geq, \leq\}$$

- satisfiability for $\mathcal{LA}(\mathbb{Z})$ is NP-COMPLETE

A decision procedure for LIA ($\mathcal{LA}(\mathbb{Z})$)

- conjunction of inequalities of the form (over Integer numbers)

$$\sum_{i}^{n} c_i \cdot x_j \bowtie b \qquad \bowtie = \{\geq, \leq\}$$

- satisfiability for $\mathcal{LA}(\mathbb{Z})$ is NP-COMPLETE
- main idea: use simplex on a problem relaxation, then use branch and bound

A decision procedure for LIA ($\mathcal{LA}(\mathbb{Z})$)

- conjunction of inequalities of the form (over Integer numbers)

$$\sum_{i}^{n} c_i \cdot x_j \bowtie b \qquad \bowtie = \{\geq, \leq\}$$

- satisfiability for $\mathcal{LA}(\mathbb{Z})$ is NP-COMPLETE
- main idea: use simplex on a problem relaxation, then use branch and bound
  - If the relaxation is unsatisfiable, then the original problem is unsatisfiable

A decision procedure for LIA ($\mathcal{L}A(\mathbb{Z})$)

- conjunction of inequalities of the form (over Integer numbers)

$$\sum_{i}^{n} c_i \cdot x_j \bowtie b \qquad \bowtie = \{\geq, \leq\}$$

- satisfiability for $\mathcal{L}A(\mathbb{Z})$ is NP-COMPLETE
- main idea: use simplex on a problem relaxation, then use branch and bound
  - If the relaxation is unsatisfiable, then the original problem is unsatisfiable
  - If the relaxation is satisfiable:
    - Select a non-integral value r for a variable x_i found in the relaxation
    - Force the integral value $v \leq \lfloor r \rfloor$, and search for a new solution
    - 2nd branch: force the integral value $v \geq \lceil r \rceil$, and search for a new solution

Branch and Bound for LIA (using the LRA relaxation)

procedure Branch-and-bound-LIA(S)
  ISSAT, α := simplex(relaxed(S))
  if $\neg$ ISSAT **then**
    return unsatisfiable
  else
    if α is integral **then**
      return satisfiable
    else
      select a variable x_i such that $\alpha(x_i) = r$ is not integral
      floor := Branch-and-bound-LIA($S \cup (v \leq \lfloor r \rfloor)$)
      ceil := Branch-and-bound-LIA($S \cup (v \geq \lceil r \rceil)$)
      return floor $\vee$ ceil

1 Decision Procedures for the Theory of Linear Real and Integer Arithmetic
- Linear Arithmetic Theories
- A decision procedure for LRA ($\mathcal{LA}(\mathbb{Q})$)
- A decision procedure for LIA ($\mathcal{LA}(\mathbb{Z})$)
- Remarks

Satisfiability is not enough . . .

The implementation of an efficient theory solver should also:

- be incremental: stack-based interface (*assert*, *backtrack*), adding and removing conjunctions (e.g., for $\mathcal{T}$-deduction and early pruning)
- Produce a small set of conflicts
- Correct under numerical errors (another difference with LP implementation):
  - Use an infinite precision representation for numbers (i.e., no floating point!), eventually
  - More implementation "tricks" to scale (e.g., use integers number, faster, in simplex first, and switch to rationals when needed)

To sum up

What did we see today?

- A decision procedure for $\mathcal{L}A(\mathbb{Q})$ (simplex algorithm)
- A decision procedure $\mathcal{L}A(\mathbb{Z})$

Next week: Application of SMT solvers to verification

References I

Bradley, A. R. and Manna, Z. (2007).
The calculus of computation - decision procedures with applications to verification.
Springer.

Dutertre, B. and De Moura, L. (2006).
A fast linear-arithmetic solver for dpll (t).
In *International Conference on Computer Aided Verification*, pages 81–94.
Springer.