Automated Reasoning 2018

Lecture 11: Theory of linear rational arithmetic (LRA)

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2018-08-21

Topic 11.1

Theory of rational linear arithmetic

Rational linear arithmetic (LRA)

Formulas with structure $\Sigma = (\{+/2, 0, 1, \dots\}, \{< /2\})$ with a set of axioms
Note: We have seen the axioms in the third lecture.

Example 11.1

The following formulas are in the quantifier-free fragment of the theory (QF_LRA), where x, y, and z are the rationals.

- $x \geq 0 \lor y + z \approx 5$
- $x < 300 \land x - z \not\approx 5$

Exercise 11.1

There is no $\leq$ in the signature. How can we use the symbol?

Proof system for QF_LRA

Due to the Farkas lemma, the following proof rule is complete for the reasoning over QF_LRA.

$$[\textsc{Comb}]\frac{t_1 \leq 0 \quad t_2 \leq 0}{t_1\lambda_1 + t_2\lambda_2 - \lambda_3 \leq 0}\lambda_1, \lambda_2, \lambda_3 \geq 0$$

Example 11.2

The following is an instance of the proof step

$$\frac{2x - y \leq 1 \quad 4y - 2x \leq 6}{x + y \leq 5}\lambda_1 = 1, \lambda_2 = 0.5, \lambda_3 = 1$$

Example 11.3

The following is an another instance of the proof step that derives false.

$$\frac{x + y \leq -2 \quad -x \leq 0 \quad -y \leq 1}{0 \leq -1}$$

Flattened rule instances

Theory solver for rational linear arithmetic

We will discuss the following method to find satisfiability of conjunction of linear inequalities.

- Simplex

We may cover some of the following methods in the next lecture.

- Fourier-Motzkin
- Loos-Weispfenning quantifier elimination
- Omega test method
- Elliposid method
- Kermakar's method

We present the above methods using non-strict linear inequalities. However, they are extendable to strict inequalities, equalities, dis-equalities.

Topic 11.2

Simplex

Simplex

Simplex was originally designed for linear optimization problems, e.g.,

$$max\{cx|Ax \leq b\}.$$

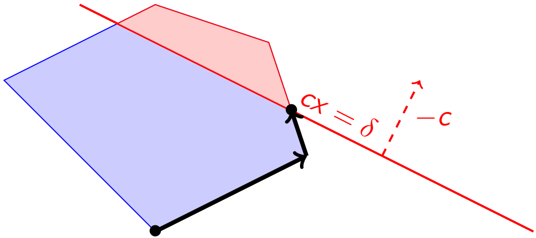
A simplex variation is used to check satisfiablity, called incremental simplex.

Commentary: In fact, there are several design choices for implementing simplex. The presentation here is one version of simplex.

Incremental simplex

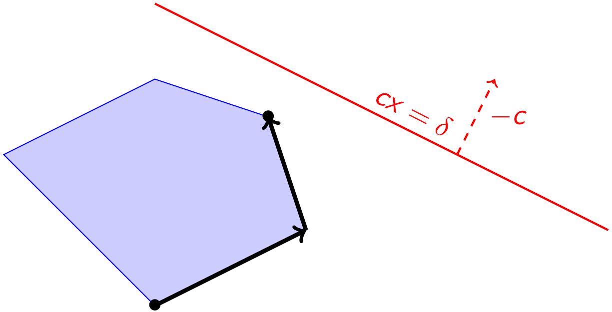
Incremental simplex

- takes atoms one by one,
- maintains a current assignment that satisfies the atoms seen so far, and
- after receiving a new atom $cx \leq \delta$,
  - attempts to move the assignment in the direction of $-c$ (optimization like operation)



Incremental simplex: unsatisfiable input

Simplex may fail to reach $cx = \delta$ and the input is unsatisfiable



Exercise 11.2

Who is responsible for the unsatisfiability?

Incremental simplex as theory solver

Recall the expected interface for SMT solver:

- push(): add new atom to the simplex state.
- pop(): inexpensive operation
- unsatCore(): again inexpensive operation

Notation

Consider the conjunction of linear inequalities in matrix form

$$Ax \leq b,$$

where A is a $m \times n$ matrix.

By introducing fresh variables, we transform the above into

$$\begin{bmatrix} -\mathtt{I} & A \end{bmatrix} \begin{bmatrix} s \\ x \end{bmatrix} = 0 \text{ and } s \leq b.$$

s are called slack variables. Since there is no reason to distinguish x and s in simplex, A will refer to $\begin{bmatrix} -\mathtt{I} \; A \end{bmatrix}$ and x will refer to $\begin{bmatrix} s \\ x \end{bmatrix}$.

Notation (contd.)

In general, the constraints will be denoted by

$$Ax = 0 \text{ and } \bigwedge_{i=1}^{m+n} l_i \leq x_i \leq u_i.$$

l_i and u_i are $+\infty$ and $-\infty$ if there is no lower and upper bound, respectively.

- A is $m \times (m+n)$ matrix.
- Since $Ax = 0$ defines an n-dim subspace in $(m+n)$-dim space, if we choose values of n variables then we fix values of the other m variables.
- We will refer to ith column of A as the column corresponding to x_i.

Example: notation

Example 11.4

Consider: $-x + y \leq -2 \wedge x \leq 3$

We introduce slack variables s_1 and s_2 for each inequality.

In matrix form,

$$\left[\begin{array}{cc:cc} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{array}\right] \left[\begin{array}{c} s_1 \\ s_2 \\ \hdashline x \\ y \end{array}\right] = 0 \qquad \begin{array}{c} s_1 \leq -2 \\ s_2 \leq 3 \end{array}$$

Basic and nonbasic variables

Definition 11.1

Simplex assumes all the columns of $-\mathtt{I}$ (of size $m \times m$) occur in A.

- *The variables corresponding to the columns are called basic variables.*
- *Others are called nonbasic variables.*

$$-\mathtt{I}$$

$$\begin{bmatrix} \vdots & 0 & \vdots & -1 & \vdots & 0 \\ \dots & -1 & \dots & 0 & \dots & 0 \\ \vdots & 0 & \vdots & 0 & \vdots & -1 \end{bmatrix}$$

Exercise 11.3

What are the numbers of basic and nonbasic variables ?

Example: Basic and nonbasic variables

Definition 11.2

Let B *be the set of indexes for the basic variables and* $NB \triangleq 1..(m+n) - B$*.*
For $j \in B$*, let* k_j *be a row such that* $A_{k_j j} = -1$ *and we may write*

$$x_j = \sum_{i \in NB} a_{k_j i} x_i,$$

which is called the definition of x_j*.*

Example 11.5

$$\begin{bmatrix} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{array}{c} s_1 \leq -2 \\ s_2 \leq 3 \end{array}$$

Currently, s_1 *and* s_2 *are basic and* x *and* y *are nonbasic.*
$B = \{1, 2\}$*,* $NB = \{3, 4\}$*,* $k_1 = 1$*, and* $k_2 = 2$*.*
The definition of s_1 *is* $-x + y$*.*

Exercise 11.4 *What is the definition of the other basic variable?*

Current assignment

Definition 11.3

Simplex maintains current assignment $v : x \to \mathbb{Q}$ *s.t.*

- $Av = 0$,
- *nonbasic variables satisfy their bounds, and,*
- *consequently values for basic variables in* v *are fixed and* v *may* violate *a bound of* at most *one basic variable.*

Explained later why "at most" one

Example 11.6

$$\begin{bmatrix} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0$$

$s_1 \leq -2$ (Currently violated)

$s_2 \leq 3$

Initially, $v = \{\underbrace{x \mapsto 0, y \mapsto 0}, s_1 \mapsto 0, s_2 \mapsto 0\}$

Choose values for nonbasic variables, others follow!

State

Simplex ensures the following invariant.
For variable $i \in NB$,

- if x_i is unbounded then $v(x_i) = 0$ and
- otherwise $v(x_i)$ is equal to one of the existing bounds of x_i

We will mark the active bounds by *.

Definition 11.4

A bound on x_i is called active if $v(x_i)$ is equal to the bound.

Definition 11.5

The NB set and bound activity defines the current state of simplex.

Example 11.7

$$\begin{bmatrix} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_1 \leq -2 \\ s_2 \leq 3 \end{matrix}$$

Since all nonbasic variables have no bounds, no bound is marked active.

Pivot operation

If v violates a bound of a basic variable, then simplex corrects it by pivoting.

Definition 11.6

Let us suppose x_j is basic, column j has -1 at row k, and x_i is nonbasic. A pivot operation between i and j exchanges the roll between x_i and x_j. The pivoting is row operations until column i has a single nonzero entry -1 at row k.

$$
\begin{array}{c} \\ \\ k \longrightarrow \\ \\ \end{array}
\begin{array}{c} \quad j \quad\quad\quad i \\ \quad\downarrow \quad\quad\quad \downarrow \\
\begin{bmatrix} \vdots & 0 & \vdots & a & \vdots \\ \dots & -1 & \dots & b & \dots \\ \vdots & 0 & \vdots & c & \vdots \end{bmatrix}\end{array}
=>
\begin{array}{c}\text{After pivot operation between } i \text{ and } j \\
\begin{bmatrix} \vdots & \frac{a}{b} & \vdots & 0 & \vdots \\ \dots & \frac{1}{b} & \dots & -1 & \dots \\ \vdots & \frac{c}{b} & \vdots & 0 & \vdots \end{bmatrix}\end{array}
$$

Variables for pivot operations

Three variables are involved in the pivoting

1. the violated basic variable
2. nonbasic variable for pivot
3. basic variable for pivot

The violated basic variable does not participate in pivoting.

Commentary: In a special case, the violated basic variable may participate in pivoting. Otherwise, the violated variable remains basic variables after pivot.

Violated basic variable

Wlog, let $1 \in B$, $k_1 = 1$, and $v(x_1)$ violates u_1.

We need to decrease $v(x_1)$.

We call $v(x_1) - u_1$ violation difference.

Exercise 11.5
Write other cases that are ignored due to "wlog"

Choosing nonbasic column for pivot

Since $x_1 = \sum_{i \in NB} a_{1i}x_i$, we need to change $v(x_i)$ of some x_i s.t. $a_{1i}x_i$ decreases

Definition 11.7

A column $i \in NB$ is suitable if

- *x_i is unbounded,*
- *$x_i = u_i$ and $a_{1i} > 0$, or*
- *$x_i = l_i$ and $a_{1i} < 0$.*

i is selected suitable column if i is the smallest suitable column.

Example 11.8

$$\begin{bmatrix} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_1 \leq -2 \\ s_2 \leq 3 \end{matrix}$$

Column 3 and 4 are suitable.

Choosing basic column for pivot

v satisfies all bounds except u_1. Change in $v(x_i)$ may lead to violations, because x_i appears in the definitions of basic variables.

Wlog, let x_i is unbounded and $a_{1i} < 0$. Therefore, we need to increase $v(x_i)$.

Definition 11.8

We need to find the maximum allowed change.

$$ch := min \bigcup_{j \in B} \{ \frac{v(x_j) - u_j}{a_{k_j i}} | a_{k_j i} < 0 \} \cup \{ \frac{v(x_j) - l_j}{a_{k_j i}} | a_{k_j i} > 0 \}$$

We choose the smallest j for which the above min is attained.

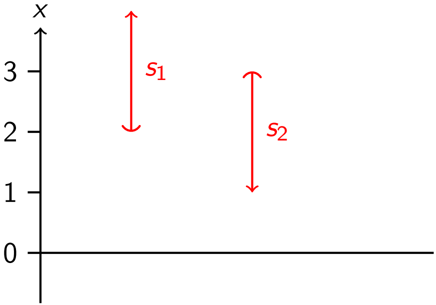
Exercise 11.6

What are the other cases in the without loss of generality?

Example: choosing basic column for pivot

Example 11.9

We change x(selected suitable column) to reduce violation difference.
Since $v(y) = 0$ and we are varying x, $s_1 = -x$ and $s_2 = x$.
The bounds on basic variables are $s_1 \leq -2$, and $s_2 \leq 3$.



Therefore, s_1 allows $2 \leq x$ and s_2 allows $x \leq 3$.
Clearly, $ch = 3$ and $j = 2$.

Simplex - pivoting operation to reduce violation difference

We carry ch and j from the last slide. Wlog, $ch = \frac{v(x_j) - u_j}{a_{k_j i}}$.

Now there are three possibilities

1. If $ch = u_i = +\infty$, pivot between i and 1 and activate u_1
2. If $ch > (u_i - l_i)$, we assign $v(x_i) = l_i$ and no pivoting
3. Otherwise, we apply pivoting between nonbasic i and basic j. We activate u_j bound on variable x_j.

If the violation persists, we apply further pivot operations.

Theorem 11.1

Pivoting operation never increases violation difference

Example: pivoting

Example 11.10

Our running example, s_1 is in violation, chosen nonbasic column is 3 and chosen basic column is 2

$$\begin{bmatrix} -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_1 \leq -2 \\ s_2 \leq 3 \end{matrix}$$

After pivoting between 3 and 2.

$$\begin{bmatrix} -1 & -1 & 0 & 1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_1 \leq -2 \\ s_2 \leq 3^* \end{matrix}$$

Now v is satisfying.

Exercise 11.7

What is v?

Incremental simplex and single violation assumption

Before adding next atom, simplex has a solution of atoms added so far.

New atom $cx \leq \delta$ is added in the following steps.

- A fresh slack variable s is introduced
- $s = cx$ is added as a row in A and $s \leq \delta$ is added in the bounds
- The new row may have non-zeros in basic columns. They are removed by row operations on the new row.
- s is added to B, declaring it to be a basic variable.

Therefore, the current assignment can only violate the bound of s.

The above strategy is called eager pivoting. We may lazily remove the violations, without breaking the correctness.

Example: inserting a new atom

Example 11.11

Let us add atom $-2x - y \leq -8$ in our running example
We add a new slack variable s_3 and a corresponding row.

$$\begin{bmatrix} -1 & 0 & 0 & -2 & -1 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_3 \\ s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_3 \leq -8 \\ s_1 \leq -2 \\ s_2 \leq 3^* \end{matrix}$$

After removing basic variables ($\{s_1, x\}$) from the top row

$$\begin{bmatrix} -1 & 0 & -2 & 0 & -1 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_3 \\ s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{matrix} s_3 \leq -8 \\ s_1 \leq -2 \\ s_2 \leq 3^* \end{matrix}$$

Exercise 11.8

Now s_3 is violated. Pivot if possible.

Simplex - iterations

Simplex is a sequence of pivot operations

- If a state is reached without violation then v is a satisfying assignment.
- If there are no suitable columns to rpair a violation then input is unsat.

Example 11.12

s_3 is still in violation.

$$\begin{bmatrix} -1 & -1 & -3 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_3 \\ s_1 \\ s_2 \\ x \\ y \end{bmatrix} = 0 \qquad \begin{array}{c} s_3 \leq -8 \\ s_1 \leq -2^* \\ s_2 \leq 3^* \end{array}$$

Now, we can not find a suitable column.
Therefore, the constraints are unsat.

Example 11.13

Run simplex on $x_1 \leq 5 \wedge 4x_1 + x_2 \leq 25 \wedge -2x_1 - x_2 \leq -25$

After push of the first atom

$$\begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ x_1 \end{bmatrix} = 0 \qquad s_1 \leq 5 \qquad v = \{x_1 \mapsto 0, s_1 \mapsto 0\}$$

After push of the second atom

$$\begin{bmatrix} -1 & 0 & 4 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_2 \\ s_1 \\ x_1 \\ x_2 \end{bmatrix} = 0 \qquad \begin{array}{c} s_1 \leq 5 \\ s_2 \leq 25 \end{array} \qquad v = \{\_ \mapsto 0\}$$

After push of the last atom

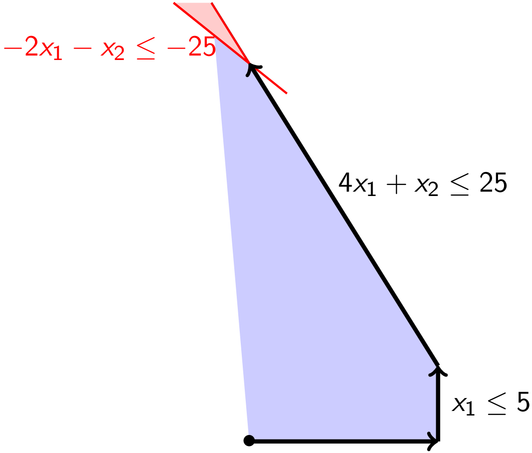
$$\begin{bmatrix} -1 & 0 & 0 & -2 & -1 \\ 0 & -1 & 0 & 4 & 1 \\ 0 & 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_3 \\ s_2 \\ s_1 \\ x_1 \\ x_2 \end{bmatrix} = 0 \qquad \begin{array}{c} s_1 \leq 5 \\ s_2 \leq 25 \\ s_3 \leq -25 \end{array} \qquad v = \{\_ \mapsto 0\}$$

Exercise 11.9

Finish the run

An example of worst case Simplex

The previous example is the case of exponential number of pivots.



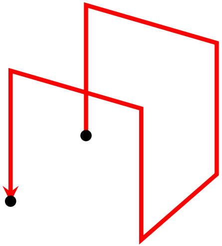
Worst case Simplex

For n dimensional problem, we can make simplex do $2^n - 1$ pivots, i.e., walking on $2^n - 1$ edges of a n-dimensional cuboid.

Example 11.14

Problem input for three dimensions

$x_1 \leq 5 \land 4x_1 + x_2 \leq 25 \land 8x_1 + 4x_2 + x_3 \leq 125 \land -4x_1 - 2x_2 - x_3 \leq -125$



Simplex complexity

Simplex is average time linear and worst case exponential.

In practice, none of the above complexities are observed

Ellipsoid method is a polynomial time algorithm for linear constraints. In practice, simplex performs better in many classes of problems.

Theory solver interface pop()

If we want to remove some atom from simplex state, we

- make the corresponding slack variable x_i basic variable and
- remove the corresponding row k_i and bound constraints on x_i

Cost: one pivot operation

Theory solver interface UnsatCore()

If input is unsat, there must be a violated basic variable x_j

- we collect the slack variables that appear in the row k_j
- the atoms corresponding to the slack variables are part of unsat core

Cost: zero.

However, we used the simplex design that excessively uses slack variables.

Commentary: Some times slack variables can be avoided. For example, input atom is equality. We can solve the constraints without introducing slack variables.

End of Lecture 11