Decision Procedures

Jochen Hoenicke

Software Engineering
Albert-Ludwigs-University Freiburg

Summer 2012

Quantifier-free Rationals

Conjunctive Quantifier-free Fragment

In the next lectures, we consider conjunctive quantifier-free Σ-formulae, i.e., conjunctions of Σ-literals (Σ-atoms or negations of Σ-atoms).

Conjunctive Quantifier-free Fragment

In the next lectures, we consider conjunctive quantifier-free Σ-formulae, i.e., conjunctions of Σ-literals (Σ-atoms or negations of Σ-atoms).

Remark 1: From this an algorithm for arbitrary quantifier-free formulae can be built.

Conjunctive Quantifier-free Fragment

In the next lectures, we consider conjunctive quantifier-free Σ-formulae, i.e., conjunctions of Σ-literals (Σ-atoms or negations of Σ-atoms).

Remark 1: From this an algorithm for arbitrary quantifier-free formulae can be built.
For given arbitrary quantifier-free Σ-formula F, convert it into DNF Σ-formula

$$F_1 \lor \ldots \lor F_k$$

where each F_i conjunctive.
F is T-satisfiable iff at least one F_i is T-satisfiable.

Conjunctive Quantifier-free Fragment

In the next lectures, we consider conjunctive quantifier-free Σ-formulae, i.e., conjunctions of Σ-literals (Σ-atoms or negations of Σ-atoms).

Remark 1: From this an algorithm for arbitrary quantifier-free formulae can be built.
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.

Remark 2: One can also combine a decision procedure for conjunctive fragment with DPLL.

Conjunctive Quantifier-free Fragment of Rationals

For $T_{\mathbb{Q}}$ a formula in the conjunctive fragment looks like this:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\
\wedge a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\
&\vdots \\
\wedge a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_m
\end{aligned}
$$

Conjunctive Quantifier-free Fragment of Rationals

For $T_{\mathbb{Q}}$ a formula in the conjunctive fragment looks like this:

$$\begin{array}{r}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\
\wedge a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \\
\vdots \quad \\
\wedge a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m
\end{array}$$

as vectors: $A \cdot \vec{x} \leq \vec{b}$.

Conjunctive Quantifier-free Fragment of Rationals

For $T_{\mathbb{Q}}$ a formula in the conjunctive fragment looks like this:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\ \wedge a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\ &\vdots \\ \wedge a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_m \end{aligned}$$

$$\text{as vectors: } A \cdot \vec{x} \leq \vec{b}.$$

Note: $x = b$ can be expressed as $x \leq b \wedge -x \leq -b$.
$\neg(x \leq b)$ can be expressed as $-x < -b$.
$x < b$ requires some additional handling (later).

Dutertre–de Moura Algorithm

- Presented 2006 by B. Dutertre and L. de Moura
- Based on Simplex algorithm
- Simpler; it doesn't optimize.

Nonbasic and Basic Variables

The set of variables in the formula is called $\mathcal{N}$ (set of non-basic variables).

Nonbasic and Basic Variables

The set of variables in the formula is called $\mathcal{N}$ (set of non-basic variables).

Additionally we introduce basic variables $\mathcal{B}$, one variable for each linear term in the formula:

$$y_i := a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n$$

The basic variables depend on the non-basic variables.

Note: The naming is counter-intuitive. Unfortunately it is the standard naming for Simplex algorithm.

Nonbasic and Basic Variables

The set of variables in the formula is called $\mathcal{N}$ (set of non-basic variables).

Additionally we introduce basic variables $\mathcal{B}$, one variable for each linear term in the formula:

$$y_i := a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n$$

The basic variables depend on the non-basic variables.

Note: The naming is counter-intuitive. Unfortunately it is the standard naming for Simplex algorithm.

We need to find a solution for $y_1 \leq b_1, \ldots, y_m \leq b_m$

Computing Basic from Non-basic Variables

The basic variables can be computed by a simple Matrix computation:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

Computing Basic from Non-basic Variables

The basic variables can be computed by a simple Matrix computation:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

One can also use tableaux notation:

	x_1	$\dots$	x_n
y_1	a_{11}	$\dots$	a_{1n}
$\vdots$	$\vdots$		$\vdots$
y_m	a_{m1}	$\dots$	a_{mn}

Computing Basic from Non-basic Variables

The basic variables can be computed by a simple Matrix computation:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

One can also use tableaux notation:

	x_1	$\dots$	x_n
y_1	a_{11}	$\dots$	a_{1n}
$\vdots$	$\vdots$		$\vdots$
y_m	a_{m1}	$\dots$	a_{mn}

We start by setting all non-basic to 0 and computing the basic variables, denoted as $\beta_0(x) := 0$. The valuation β_s assigns values for the variables at step s.

Configuration

A configuration at step s of the algorithm consists of

- a partition of the variables into non-basic and basic variables

$$\mathcal{N}_s \cup \mathcal{B}_s = \{x_1, \ldots, x_n, y_1, \ldots y_m\},$$

- a tableaux A (a $m \times n$ matrix) where the columns correspond to non-basic and rows correspond to basic variables,
- and a valuation β_s, that assigns
  - $\beta_s(x_i) = 0$ for $x_i \in \mathcal{N}_s$,
  - $\beta_s(y_i) = b_i$ for $y_i \in \mathcal{N}_s$,
  - $\beta_s(z_i) = \sum_{z_j \in \mathcal{N}_s} a_{ij}\beta(z_j)$ for $z_i \in \mathcal{B}_s$.

  (Here z stands for either an x or a y variable.)

Configuration

A configuration at step s of the algorithm consists of

- a partition of the variables into non-basic and basic variables

$$\mathcal{N}_s \cup \mathcal{B}_s = \{x_1, \dots, x_n, y_1, \dots y_m\},$$

- a tableaux A (a $m \times n$ matrix) where the columns correspond to non-basic and rows correspond to basic variables,
- and a valuation β_s, that assigns
  - $\beta_s(x_i) = 0$ for $x_i \in \mathcal{N}_s$,
  - $\beta_s(y_i) = b_i$ for $y_i \in \mathcal{N}_s$,
  - $\beta_s(z_i) = \sum_{z_j \in \mathcal{N}_s} a_{ij}\beta(z_j)$ for $z_i \in \mathcal{B}_s$.

  (Here z stands for either an x or a y variable.)

The initial configuration is:

$$\mathcal{N}_0 = \{x_1, \dots, x_n\}, \mathcal{B}_0 = \{y_1, \dots, y_m\}, A_0 = A, \beta_0(x_i) = 0$$

In later steps variables from $\mathcal{N}$ and $\mathcal{B}$ are swapped.

Suppose β_s is not a solution for $y_1 \leq b_1, \ldots, y_m \leq b_m$.
Let y_i be a variable whose value $\beta_s(y_i) > b_i$.
Consider the row in the matrix:

$$y_i = a_{i1}z_1 + a_{i2}z_2 + \cdots + a_{in}z_n$$

Pivoting aka. Exchanging Basic and Non-basic Variables

Suppose β_s is not a solution for $y_1 \leq b_1, \dots, y_m \leq b_m$.
Let y_i be a variable whose value $\beta_s(y_i) > b_i$.
Consider the row in the matrix:

$$y_i = a_{i1}z_1 + a_{i2}z_2 + \cdots + a_{in}z_n$$

Idea: Choose a z_j, then solve z_j in the above equation.
Thus, z_j becomes non-basic variable, y_i becomes basic.
Then decrease $\beta(y_i)$ to b_i.
This will either decrease z_j (if $a_{ij} > 0$)
or increase z_j (if $a_{ij} < 0$, z_j must be a x-variable).

Pivoting aka. Exchanging Basic and Non-basic Variables

Suppose β_s is not a solution for $y_1 \le b_1, \ldots, y_m \le b_m$.
Let y_i be a variable whose value $\beta_s(y_i) > b_i$.
Consider the row in the matrix:

$$y_i = a_{i1}z_1 + a_{i2}z_2 + \cdots + a_{in}z_n$$

Idea: Choose a z_j, then solve z_j in the above equation.
Thus, z_j becomes non-basic variable, y_i becomes basic.
Then decrease $\beta(y_i)$ to b_i.
This will either decrease z_j (if $a_{ij} > 0$)
or increase z_j (if $a_{ij} < 0$, z_j must be a x-variable).
Solving z_j in the above equation gives:

$$z_j = \frac{a_{i1}}{-a_{ij}}z_1 + \frac{a_{i2}}{-a_{ij}}z_2 + \cdots + \frac{a_{in}}{-a_{ij}}z_n + \frac{1}{a_{ij}}y_i$$

Result of Pivoting

After pivoting y_i and z_j the matrix looks as follows:

$$\begin{array}{rcrcrcr}
y_1 & = & (a_{11} - \frac{a_{1j}a_{i1}}{a_{ij}})z_1 & + \cdots + & \frac{a_{1j}}{a_{ij}}y_i & + \cdots + & (a_{1n} - \frac{a_{1j}a_{in}}{a_{ij}})z_n \\
\vdots & & \vdots & & \vdots & & \vdots \\
z_j & = & -\frac{a_{i1}}{a_{ij}}z_1 & + \cdots + & \frac{1}{a_{ij}}y_i & + \cdots + & -\frac{a_{in}}{a_{ij}}z_n \\
\vdots & & \vdots & & \vdots & & \vdots \\
y_m & = & (a_{m1} - \frac{a_{mj}a_{i1}}{a_{ij}})z_1 & + \cdots + & \frac{a_{mj}}{a_{ij}}y_i & + \cdots + & (a_{mn} - \frac{a_{mj}a_{in}}{a_{ij}})z_n
\end{array}$$

Now, set $\beta_{s+1}(y_i)$ to b_i and recompute basic variables.

Detecting Conflicts

We may arrive at a configuration like:

$$y_i = 0 \cdot x_1 + \cdots + a_{ij_1} y_{j_1} + \cdots + a_{ij_k} y_{j_k} + 0 \cdot x_n$$

where the non-basic y variables are set to their bound:

$$\beta_s(y_{j_1}) = b_{j_1}, \ldots, \beta_s(y_{j_k}) = b_{j_k} \,,$$

coefficients of x variables are zero, coefficients $a_{ij_1}, \ldots, a_{ij_k} \leq 0$, and $\beta_s(y_i) > b_i$.

We may arrive at a configuration like:

$$y_i = 0 \cdot x_1 + \cdots + a_{ij_1} y_{j_1} + \cdots + a_{ij_k} y_{j_k} + 0 \cdot x_n$$

where the non-basic y variables are set to their bound:

$$\beta_s(y_{j_1}) = b_{j_1}, \ldots, \beta_s(y_{j_k}) = b_{j_k} \,,$$

coefficients of x variables are zero, coefficients $a_{ij_1}, \ldots, a_{ij_k} \le 0$, and $\beta_s(y_i) > b_i$.

Then, we have a conflict:

$$y_{j_1} \le b_{j_1} \wedge \cdots \wedge y_{j_k} \le b_{j_k} \rightarrow y_i > b_i \,.$$

The formula is not satisfiable.

Example

Consider the formula

$$F : x_1 + x_2 \geq 4 \wedge x_1 - x_2 \leq 1$$

Example

Consider the formula

$$F \,:\, x_1 + x_2 \geq 4 \wedge x_1 - x_2 \leq 1$$

We have two non-basic variables $\mathcal{N} = \{x_1, x_2\}$.
Define basic variables $\mathcal{B} = \{y_1, y_2\}$:

$$y_1 = -x_1 - x_2, \qquad y_1 \leq -4$$

$$y_2 = x_1 - x_2, \qquad y_2 \leq 1$$

We write the equation as a tableaux:

	x_1	x_2
y_1	-1	-1
y_2	1	-1

Example (cont.)

Tableaux:

	x_1	x_2
y_1	-1	-1
y_2	1	-1

Values:

$x_1 = x_2 = 0$

$\rightarrow y_1 = 0 > -4$ (!)

$\rightarrow y_2 = 0 \leq 1$

Pivot y_1 against x_1: $x_1 = -y_1 - x_2$.

New Tableaux:

	y_1	x_2
x_1	-1	-1
y_2	-1	-2

Example (cont.)

Tableaux:

	y_1	x_2
x_1	-1	-1
y_2	-1	-2

Values:

$y_1 = -4, x_2 = 0$
$\rightarrow x_1 = 4$
$\rightarrow y_2 = 4 > 1$ (!)

y_2 cannot be pivoted with y_1, since -1 negative.
Pivot y_2 and x_2:

New Tableaux:

	y_1	y_2
x_1	-.5	.5
x_2	-.5	-.5

Tableaux:

	y_1	y_2
x_1	-.5	.5
x_2	-.5	-.5

Values:

$y_1 = -4, y_2 = 1$
$\rightarrow x_1 = 2.5$
$\rightarrow x_2 = 1.5$

We found a satisfying interpretation for:

$$F : x_1 + x_2 \geq 4 \land x_1 - x_2 \leq 1$$

Example

Now, consider the formula

$$F' : x_1 + x_2 \geq 4 \land x_1 - x_2 \leq 1 \land x_2 \leq 1$$

Example

Now, consider the formula

$$F' : x_1 + x_2 \geq 4 \land x_1 - x_2 \leq 1 \land x_2 \leq 1$$

We have two non-basic variables $\mathcal{N} = \{x_1, x_2\}$.
Define basic variables $\mathcal{B} = \{y_1, y_2, y_3\}$:

$$\begin{aligned} y_1 &= -x_1 - x_2, & y_1 &\leq -4 \\ y_2 &= x_1 - x_2, & y_2 &\leq 1 \\ y_3 &= x_2, & y_3 &\leq 1 \end{aligned}$$

We write the equation as tableaux:

	x_1	x_2
y_1	-1	-1
y_2	1	-1
y_3	0	1

The first two steps are identical:
pivot y_1 resp. y_2 and x_1 resp. x_2.

	y_1	y_2
x_1	-.5	.5
x_2	-.5	-.5
y_3	-.5	-.5

Example (cont.)

Tableaux:

	y_1	y_2
x_1	-.5	.5
x_2	-.5	-.5
y_3	-.5	-.5

Values:

$y_1 = -4, y_2 = 1$

$\rightarrow x_1 = 2.5$

$\rightarrow x_2 = 1.5$

$\rightarrow y_3 = 1.5 > 1$!

Now, y_3 cannot pivot, since all coefficients in that row are negative.
Conflict is $-x_1 - x_2 \leq -4 \wedge x_1 - x_2 \leq 1 \rightarrow x_2 > 1$.
Formula F' is unsatisfiable

Termination

To guarantee termination we need a fixed pivot selection rule.

Termination

To guarantee termination we need a fixed pivot selection rule.

The following rule works:

When choosing the basic variable (row) to pivot:

- Choose the y-variable with the smallest index, whose value exceeds the bound.
- If there is no such variable, return satisfiable

When choosing the non-basic variable (column) to pivot with:

- if possible, take a x-variable.
- Otherwise, take the y-variable with the smallest index, such that the corresponding coefficient in the matrix is positive.
- If there is no such variable, return unsatisfiable

Termination Proof

Assume we have an infinite computation of the algorithm.

Let y_j be the variable with the largest index, that is infinitely often pivoted.

Termination Proof

Assume we have an infinite computation of the algorithm.

Let y_j be the variable with the largest index, that is infinitely often pivoted.

Look at the step where y_j is pivoted to a non-basic variable and where for $k > j$, y_k is not pivoted any more. The (ordered) tableaux at the point of pivoting looks like this:

	x	$\cdots$	x	y	$\cdots$	y	y_j	y	$\cdots$
$\vdots$									
y_i	0	$\cdots$	0	$-/0$	$\cdots$	$-/0$	$+$	$\pm/0$	$\cdots$
$\vdots$									

($+$ denotes a positive coefficient, $-$ a negative coefficient)

Termination Proof

Assume we have an infinite computation of the algorithm.

Let y_j be the variable with the largest index, that is infinitely often pivoted.

Look at the step where y_j is pivoted to a non-basic variable and where for $k > j$, y_k is not pivoted any more. The (ordered) tableaux at the point of pivoting looks like this:

	x	$\cdots$	x	y	$\cdots$	y	y_j	y	$\cdots$
$\vdots$									
y_i	0	$\cdots$	0	$-/0$	$\cdots$	$-/0$	$+$	$\pm/0$	$\cdots$
$\vdots$									

($+$ denotes a positive coefficient, $-$ a negative coefficient)

After pivoting the tableaux changes to:

	x	$\cdots$	x	y	$\cdots$	y	y_i	y	$\cdots$
$\vdots$									
y_j	0	$\cdots$	0	$+/0$	$\cdots$	$+/0$	$+$	$\mp/0$	$\cdots$
$\vdots$									

Termination Proof (cont.)

After pivoting the tableaux changes to:

	x	$\cdots$	x	y	$\cdots$	y	y_i	y	$\cdots$
$\vdots$									
y_j	0	$\cdots$	0	$+/0$	$\cdots$	$+/0$	$+$	$\mp/0$	$\cdots$
$\vdots$									

$$\sum_{k<j, y_k \in \mathcal{N}_s} a_k b_k + \sum_{k>j, y_k \in \mathcal{N}_s} a_k b_k = \beta_s(y_j) < b_j, \text{ where } a_k \geq 0 \text{ for } k < j.$$

Termination Proof (cont.)

After pivoting the tableaux changes to:

	x	$\cdots$	x	y	$\cdots$	y	y_i	y	$\cdots$
$\vdots$									
y_j	0	$\cdots$	0	$+/0$	$\cdots$	$+/0$	$+$	$\mp/0$	$\cdots$
$\vdots$									

$$\sum_{k<j, y_k \in \mathcal{N}_s} a_k b_k + \sum_{k>j, y_k \in \mathcal{N}_s} a_k b_k = \beta_s(y_j) < b_j, \text{ where } a_k \geq 0 \text{ for } k < j.$$

Now look at the step s' where y_j is pivoted back.
By the pivoting rule: $\beta_{s'}(y_k) \leq b_k$ for all $k < j$.
For $k > j$, the non-basic/basic variables do not change.

Termination Proof (cont.)

After pivoting the tableaux changes to:

	x	$\cdots$	x	y	$\cdots$	y	y_i	y	$\cdots$
$\vdots$									
y_j	0	$\cdots$	0	$+/0$	$\cdots$	$+/0$	$+$	$\mp/0$	$\cdots$
$\vdots$									

$$\sum_{k<j, y_k \in \mathcal{N}_s} a_k b_k + \sum_{k>j, y_k \in \mathcal{N}_s} a_k b_k = \beta_s(y_j) < b_j, \text{ where } a_k \geq 0 \text{ for } k < j.$$

Now look at the step s' where y_j is pivoted back.
By the pivoting rule: $\beta_{s'}(y_k) \leq b_k$ for all $k < j$.
For $k > j$, the non-basic/basic variables do not change.
Therefore, the value of y_j can only get smaller.

$$\beta_{s'}(y_j) = \sum_{k<j, y_k \in \mathcal{N}_s} a_k \cdot \beta_{s'}(y_k) + \sum_{k>j, y_k \in \mathcal{N}_s} a_k b_k < b_j$$

Termination Proof (cont.)

After pivoting the tableaux changes to:

	x	$\cdots$	x	y	$\cdots$	y	y_i	y	$\cdots$
$\vdots$									
y_j	0	$\cdots$	0	$+/0$	$\cdots$	$+/0$	$+$	$\mp/0$	$\cdots$
$\vdots$									

$$\sum_{k<j, y_k \in \mathcal{N}_s} a_k b_k + \sum_{k>j, y_k \in \mathcal{N}_s} a_k b_k = \beta_s(y_j) < b_j, \text{ where } a_k \geq 0 \text{ for } k < j.$$

Now look at the step s' where y_j is pivoted back.
By the pivoting rule: $\beta_{s'}(y_k) \leq b_k$ for all $k < j$.
For $k > j$, the non-basic/basic variables do not change.
Therefore, the value of y_j can only get smaller.

$$\beta_{s'}(y_j) = \sum_{k<j, y_k \in \mathcal{N}_s} a_k \cdot \beta_{s'}(y_k) + \sum_{k>j, y_k \in \mathcal{N}_s} a_k b_k < b_j$$

This contradicts $\beta_{s'}(y_j) > b_j$.

Therefore, assumption was wrong and algorithm terminates.

Strict Bounds

With strict bounds the formula looks like this:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\
&\vdots \\
\wedge a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n &\leq b_i \\
\wedge a_{(i+1)1}x_1 + a_{(i+1)2}x_2 + \cdots + a_{(i+1)n}x_n &< b_{i+1} \\
&\vdots \\
\wedge a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &< b_m
\end{aligned}
$$

Strict Bounds

With strict bounds the formula looks like this:
If the formula is satisfiable, then there is an $\varepsilon > 0$ with:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\
&\vdots \\
\wedge a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n &\leq b_i \\
\wedge a_{(i+1)1}x_1 + a_{(i+1)2}x_2 + \cdots + a_{(i+1)n}x_n &\leq b_{i+1} - \varepsilon \\
&\vdots \\
\wedge a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_m - \varepsilon
\end{aligned}
$$

Infinitesimal Numbers

We compute with ε symbolically. Our bounds are elements of

$$\mathbb{Q}_\varepsilon := \{a_1 + a_2\varepsilon \mid a_1, a_2 \in \mathbb{Q}\}$$

Infinitesimal Numbers

We compute with ε symbolically. Our bounds are elements of

$$\mathbb{Q}_\varepsilon := \{a_1 + a_2\varepsilon \mid a_1, a_2 \in \mathbb{Q}\}$$

The arithmetical operators and the ordering are defined as:

$$\begin{aligned}
(a_1 + a_2\varepsilon) + (b_1 + b_2\varepsilon) &= \\
a \cdot (b_1 + b_2\varepsilon) &= \\
a_1 + a_2\varepsilon \leq b_1 + b_2\varepsilon &\text{ iff}
\end{aligned}$$

Infinitesimal Numbers

We compute with ε symbolically. Our bounds are elements of

$$\mathbb{Q}_\varepsilon := \{a_1 + a_2\varepsilon \mid a_1, a_2 \in \mathbb{Q}\}$$

The arithmetical operators and the ordering are defined as:

$$\begin{aligned}
(a_1 + a_2\varepsilon) + (b_1 + b_2\varepsilon) &= (a_1 + b_1) + (a_2 + b_2)\varepsilon \\
a \cdot (b_1 + b_2\varepsilon) &= ab_1 + ab_2\varepsilon \\
a_1 + a_2\varepsilon \leq b_1 + b_2\varepsilon &\text{ iff } a_1 < b_1 \vee (a_1 = b_1 \wedge a_2 \leq b_2)
\end{aligned}$$

We compute with ε symbolically. Our bounds are elements of

$$\mathbb{Q}_\varepsilon := \{a_1 + a_2\varepsilon \mid a_1, a_2 \in \mathbb{Q}\}$$

The arithmetical operators and the ordering are defined as:

$$\begin{aligned}
(a_1 + a_2\varepsilon) + (b_1 + b_2\varepsilon) &= (a_1 + b_1) + (a_2 + b_2)\varepsilon \\
a \cdot (b_1 + b_2\varepsilon) &= ab_1 + ab_2\varepsilon \\
a_1 + a_2\varepsilon \leq b_1 + b_2\varepsilon &\text{ iff } a_1 < b_1 \vee (a_1 = b_1 \wedge a_2 \leq b_2)
\end{aligned}$$

Note: $\mathbb{Q}_\varepsilon$ is a two-dimensional vector space over $\mathbb{Q}$.

Infinitesimal Numbers

We compute with ε symbolically. Our bounds are elements of

$$\mathbb{Q}_\varepsilon := \{a_1 + a_2\varepsilon \mid a_1, a_2 \in \mathbb{Q}\}$$

The arithmetical operators and the ordering are defined as:

$$\begin{aligned}(a_1 + a_2\varepsilon) + (b_1 + b_2\varepsilon) &= (a_1 + b_1) + (a_2 + b_2)\varepsilon \\ a \cdot (b_1 + b_2\varepsilon) &= ab_1 + ab_2\varepsilon \\ a_1 + a_2\varepsilon \leq b_1 + b_2\varepsilon &\text{ iff } a_1 < b_1 \vee (a_1 = b_1 \wedge a_2 \leq b_2)\end{aligned}$$

Note: $\mathbb{Q}_\varepsilon$ is a two-dimensional vector space over $\mathbb{Q}$.
Changes to the configuration:

- β gives values for variables in $\mathbb{Q}_\varepsilon$.
- The tableaux does not contain ε. It is still a $\mathbb{Q}^{m \times n}$ matrix.

Example

$$F_1 : 3x_1 + 2x_2 < 5 \land 2x_1 + 3x_2 < 1 \land x_1 + x_2 > 1$$

Example F_1

Step 1:

	x_1	x_2	β	b_i	
β	0	0			
y_1	3	2	0	$5 - \varepsilon$	
y_2	2	3	0	$1 - \varepsilon$	
y_3	-1	-1	0	$-1 - \varepsilon$	(!)

Step 2:

	y_3	x_2	β	b_i	
β	$-1 - \varepsilon$	0			
y_1	-3	-1	$3 + 3\varepsilon$	$5 - \varepsilon$	
y_2	-2	1	$2 + 2\varepsilon$	$1 - \varepsilon$	(!)
x_1	-1	-1	$1 + 1\varepsilon$		

Step 3:

	y_3	y_2	β	b_i
β	$-1 - \varepsilon$	$1 - \varepsilon$		
y_1	-5	-1	$4 + 6\varepsilon$	$5 - \varepsilon$
x_2	2	1	$-1 - 3\varepsilon$	
x_1	-3	-1	$2 + 4\varepsilon$	

$\beta(y_1) = 4 + 6\varepsilon \leq 5 - \varepsilon$ (for $0 < \varepsilon \leq 1/7$).

Solution ($\varepsilon = 0.1$): $x_1 = 2.4$, $x_2 = -1.3$.

Example

$$F_2 : 3x_1 + 2x_2 < 5 \land 2x_1 - x_2 > 1 \land x_1 + 3x_2 > 4$$

Example F_2

Step 1:

	x_1	x_2	β	b_i	
β	0	0			
y_1	3	2	0	$5-\varepsilon$	
y_2	-2	1	0	$-1-\varepsilon$	(!)
y_3	-1	-3	0	$-4-\varepsilon$	(!)

Step 2:

	x_1	y_2	β	b_i	
β	0	$-1-\varepsilon$			
y_1	7	2	$-2-2\varepsilon$	$5-\varepsilon$	
x_2	2	1	$-1-\varepsilon$		
y_3	-7	-3	$3+3\varepsilon$	$-4-\varepsilon$	(!)

Step 3:

	y_3	y_2	β	b_i	
β	$-4-\varepsilon$	$-1-\varepsilon$			
y_1	-1	-1	$5+2\varepsilon$	$5-\varepsilon$	(!)
x_2	-2/7	1/7	$1+1/7\varepsilon$		
x_1	-1/7	-3/7	$1+4/7\varepsilon$		

Now $5+2\varepsilon > 5-\varepsilon$ but all coefficients in first row negative.

Unsatisfiable.

Correctness of the Algorithm

Theorem (Sound and Complete)

Quantifier-free conjunctive $\Sigma_{\mathbb{Q}}$-formula F is $T_{\mathbb{Q}}$-satisfiable iff the Dutertre-de-Moura algorithm returns satisfiable.