# The Simplex Algorithm

- Prominent algorithm for solving optimization problems over a set (conjunction) of linear inequations.
- For automated reasoning, optimization is not the focus, but solvability of a set of linear inequations.
- In this context the simplex algorithm is useful as well, due to its incremental nature.

# Incremental Nature

- Given a set $N$ of inequations where the simplex algorithm has already found a solution.
- Add an inequation $A$ to $N$.
- The algorithm needs not to start from scratch, but continues with the solution found for $N$.
- In practice, we only need a few steps to derive a solution for $N \cup \{A\}$ if it exists.

# Preview CDCL(T)

A *theory T* is the *ground conjunctive fragment* of a logic. CDCL(T) extends a solver for a theory T (*theory solver*) to the *complete ground fragment* of the logic. It even allows us to build a solver that combines different logics.

Important properties for a good theory solver for CDCL(T):

- good runtime
- produces an *assignment/model* in case of satisfiability
- good *incremental* behavior

# Drawbacks of Fourier-Motzkin

- worst case runtime $O(n^{2^m})$ (exponential runtime observed on relevant industrial problems)
- produces no *assignments* (would require additional bookkeeping)
- poor incremental behavior (would require additional expensive bookkeeping)

# The Simplex Algorithm

Idea: incrementally update a variable assignment until

a) the assignment is a solution, or

b) a conflict has been found

Advantages:

- worst case runtime single exponential (but very rare & not on relevant problems)
- provides an assignment or a conflict (with no overhead)
- good incremental behavior (just continue updating the assignment)

# The Input Problem

A set $N$ (conjunction) of (non-strict)[1] inequations over a set of variables $X$.

The inequations have the form:

$$(\sum_{x_j \in X} a_{i,j} x_j) \circ_i c_i,$$
$$\text{where } \circ_i \in \{\geq, \leq\} \text{ for all } i, \text{ and } \gcd\{a_{i,j} | x_j \in X\} = 1$$

Additional assumptions (without loss of generality):

- we assume that the $x_j$ are all different
- we assume that the variables $x_j \in X$ are totally ordered by some ordering $\prec$[2]

[1] We will later describe how to handle strict inequalities.

[2] The ordering $\prec$ will eventually guarantee termination of the algorithm.

# The Goal

Decide whether there exists an assignment $\beta$ from the $x_j$ into $\mathbb{Q}$ such that $\mathrm{LRA}(\beta) \models \bigwedge_i [(\sum_{x_j \in X} a_{i,j} x_j) \circ_i c_i]$, or equivalently, $\mathrm{LRA}(\beta) \models N$

So the $x_j$ are free variables, i.e., placeholders for concrete values, i.e., existentially quantified.

# First Step: Transforming $N$

The first step is to transform $N$ into two disjoint sets $E$, $B$ of equations and simple bounds, respectively.

Hence, we split every inequation $\sum_{x_j \in X} a_{i,j} x_j \circ_i c_i$ from $N$ into:

- an equation $y_i \approx \sum_{x_j \in X} a_{i,j} x_j$ (moved to $E$), where $y_i$ is a fresh variable[3],
- a (simple) bound $y_i \circ_i c_i$ (moved to $B$)

Optimized Transformation:

- Just move simple bounds $x_i \circ_i c_i$ from $N$ to $B$.
- Use the same variable/equation for inequations with the same left hand side

[3] The $y_i$ are also part of the total ordering $\prec$ on all variables!

# Equivalence of the Transformation

Clearly, for any assignment $\beta$ and its respective extension on the $y_i$, the two representations are equivalent:

$$\mathrm{LRA}(\beta) \models N$$

iff

$$\mathrm{LRA}(\beta[y_i \mapsto \beta(\textstyle\sum_{x_j \in X} a_{i,j} x_j)]) \models E$$

and

$$\mathrm{LRA}(\beta[y_i \mapsto \beta(\textstyle\sum_{x_j \in X} a_{i,j} x_j)]) \models B.$$

# (In)dependent Variables

Given $E$ and $B$ a variable $z$ is called *dependent* if it occurs on the left hand side of an equation in $E$, i.e., there is an equation $(z \approx \sum_{x_j \in X} a_{i,j} x_j) \in E$. Otherwise, $z$ is called *independent*.

By construction the initial $y_i$ are all dependent and do not occur on the right hand side of an equation.

Any assignment over the independent variables can be extended into an assignment over all variables that satisfies $E$.

Note: when we write $(x \approx ay + t)$ for some equation, we always assume that $y \notin \text{vars}(t)$.

# Update

Given:

- an assignment $\beta$,
- an independent variable $y$,
- a rational value $c$,
- a set of equations $E$

then the *update* of $\beta$ with respect to $y$, $c$, and $E$ is

$$\text{upd}(\beta, y, c, E) := \beta[y \mapsto c, \{x \mapsto \beta[y \mapsto c](t) \mid x \approx t \in E\}].$$

$\text{upd}(\beta, y, c, E)$ is a solution for $E$.

# Pivot

Given:

- a dependant variable $x$,
- an independent variable $y$,
- a set of equations $E$, and
- the defining equation $(x \approx ay + t) \in E$ of $x$ with $a \neq 0$,

then the *pivot* operation exchanges the roles of $x$, $y$ in $E$, i.e., $x$ becomes independent and $y$ dependent.

Let $E'$ be $E$ without the defining equation of $x$. Then

$$\mathrm{piv}(E, x, y) := \left\{ y \approx \frac{1}{a}x + \frac{1}{-a}t \right\} \cup E' \left\{ y \mapsto \left( \frac{1}{a}x + \frac{1}{-a}t \right) \right\}.$$

$E$ and $\mathrm{piv}(E, x, y)$ are equivalent.

# A Simplex State

A Simplex problem state is a quintuple $(E; B; \beta; S; s)$ where:

- $E$ is a set of equations,
- $B$ a set of simple bounds,
- $\beta$ an assignment to all variables in $E$, $B$,
- $S$ a set of derived bounds, and
- $s$ the status of the problem with $s \in \{\top, \mathrm{IV}, \mathrm{DV}, \bot\}$.

# The Status $s$

Given a state $(E; B; \beta; S; s)$:

- $s = \top$ indicates that $\text{LRA}(\beta) \models S$;
- $s = \text{IV}$ indicates that potentially $\text{LRA}(\beta) \not\models x \circ c$ for some independent variable $x$, $x \circ c \in S$;
- $s = \text{DV}$ indicates that $\text{LRA}(\beta) \models x \circ c$ for all independent variables $x$, $x \circ c \in S$, but potentially $\text{LRA}(\beta) \not\models x' \circ c'$ for some dependent variable $x'$, $x' \circ c' \in S$;
- $s = \bot$ indicates that the problem is unsatisfiable

# Start and Final States

| | |
|---|---|
| $(E; B; \beta_0; \emptyset; \top)$ | is the start state for $N$ and its transformation into $E$, $B$, and assignment $\beta_0(x) := 0$ for all $x \in \text{vars}(E \cup B)$ |
| $(E; \emptyset; \beta; S; \top)$ | is a final state, where $\text{LRA}(\beta) \models E \cup S$ and hence the problem is solvable |
| $(E; B; \beta; S; \bot)$ | is a final state, where $E \cup B \cup S$ has no model |

# Invariants

The important invariants of the simplex algorithm are:

i) for every dependent variable there is exactly one equation in $E$ defining the variable

ii) dependent variables do not occur on the right hand side of an equation

iii) $\text{LRA}(\beta) \models E$

iv) Any assignment satisfying $N$ can be extended to an assignment satisfying $E \cup B \cup S$

v) Any assignment satisfying $E \cup B \cup S$ is an assignment satisfying $N$

These invariants hold initially and are maintained by a pivot (piv) or an update (upd) operation.

# Rough Draft

The simplex algorithm:

1. $\top$: moves one bound from $B$ to $S$
2. IV: repair $\beta$ for all bounds in $S$ over independent variables (update)
3. DV: repair $\beta$ for all bounds in $S$ over dependent variables (pivot & update)
4. repeat

**FailBounds**

$$(E; B; \beta; S; \top) \ \Rightarrow_{\text{SIMP}} \ (E; B; \beta; S; \bot)$$

if there are two contradicting bounds $x \leq c_1$ and $x \geq c_2$ in $B \cup S$ for some variable $x$

**EstablishBound**

$$(E; B \uplus \{x \circ c\}; \beta; S; \top) \;\Rightarrow_{\text{SIMP}}\; (E; B; \beta; S \cup \{x \circ c\}; \text{IV})$$

**AckBounds**

$$(E; B; \beta; S; V) \;\Rightarrow_{\text{SIMP}}\; (E; B; \beta; S; \top)$$

if $\text{LRA}(\beta) \models S$, $V \in \{\text{IV}, \text{DV}\}$

**FixIndepVar**

$$(E; B; \beta; S; \mathrm{IV}) \Rightarrow_{\mathrm{SIMP}} (E; B; \mathrm{upd}(\beta, x, c, E); S; \mathrm{IV})$$

if $(x \circ c) \in S$, $\mathrm{LRA}(\beta) \not\models x \circ c$, $x$ independent

**AckIndepBound**

$$(E; B; \beta; S; \mathrm{IV}) \ \Rightarrow_{\mathrm{SIMP}} \ (E; B; \beta; S; \mathrm{DV})$$

if $\mathrm{LRA}(\beta) \models x \circ c$, for all independent variables $x$ with bounds $x \circ c$ in $S$

**FixDepVar**$\geq$

$$(E; B; \beta; S; \mathrm{DV}) \ \Rightarrow_{\mathrm{SIMP}} \ (E'; B; \mathrm{upd}(\beta, x, c, E'); S; \mathrm{DV})$$

if $(x \geq c) \in S$, $x$ dependent, $\mathrm{LRA}(\beta) \not\models x \geq c$, there is an independent variable $y$ and equation $(x \approx ay + t) \in E$ where
$(a > 0$ and $\beta(y) < c'$ for all $(y \leq c') \in S)$ or
$(a < 0$ and $\beta(y) > c'$ for all $(y \geq c') \in S)$ and $E' := \mathrm{piv}(E, x, y)$

**FixDepVar**$\leq$

$$(E; B; \beta; S; \mathrm{DV}) \ \Rightarrow_{\mathrm{SIMP}} \ (E'; B; \mathrm{upd}(\beta, x, c, E'); S; \mathrm{DV})$$

if $(x \leq c) \in S$, $x$ dependent, $\mathrm{LRA}(\beta) \not\models x \leq c$, there is an independent variable $y$ and equation $(x \approx ay + t) \in E$ where $(a < 0$ and $\beta(y) < c'$ for all $(y \leq c') \in S)$ or $(a > 0$ and $\beta(y) > c'$ for all $(y \geq c') \in S)$ and $E' := \mathrm{piv}(E, x, y)$

**FailDepVar**$\leq$

$$(E; B; \beta; S; \mathrm{DV}) \ \Rightarrow_{\mathrm{SIMP}} \ (E; B; \beta; S; \perp)$$

if $(x \leq c) \in S$, $x$ dependent, $\mathrm{LRA}(\beta) \not\models x \leq c$ and there is no independent variable $y$ and equation $(x \approx ay + t) \in E$ where $(a < 0$ and $\beta(y) < c'$ for all $(y \leq c') \in S)$ or $(a > 0$ and $\beta(y) > c'$ for all $(y \geq c') \in S)$

**FailDepVar$\geq$**

$$(E; B; \beta; S; \mathrm{DV}) \;\Rightarrow_{\mathrm{SIMP}}\; (E; B; \beta; S; \bot)$$

if $(x \geq c) \in S$, $x$ dependent, $\beta \not\models_{\mathrm{LA}} x \geq c$ and there is no independent variable $y$ and equation $(x \approx ay + t) \in E$ where
$(a > 0$ and $\beta(y) < c'$ for all $(y \leq c') \in S)$ or
$(a < 0$ and $\beta(y) > c'$ for all $(y \geq c') \in S)$

## 6.2.5 Definition (Reasonable Strategy)

A *reasonable* strategy prefers FailBounds over EstablishBounds and the FixDepVar rules select minimal variables $x$, $y$ in the ordering $\prec$.

What does this mean?

- We apply the FixDepVar rules only to the smallest dependent variable $x$ in the ordering $\prec$ with a violated bound
- For the FixDepVar rules, we select the smallest variable $y$ in the ordering $\prec$ that can be used to fix $x$

## 6.2.6 Theorem (Simplex Soundness, Completeness & Termination)

Given a reasonable strategy and initial set $N$ of inequations and its separation into $E$ and $B$ :
(i) $\Rightarrow_{\text{SIMP}}$ terminates on $(E; B; \beta_0; \emptyset; \top)$,
(ii) if $(E; B; \beta_0; \emptyset; \top) \Rightarrow^*_{\text{SIMP}} (E'; B'; \beta; S; \bot)$, then $N$ has no solution,
(iii) if $(E; B; \beta_0; \emptyset; \top) \Rightarrow^*_{\text{SIMP}} (E'; \emptyset; \beta; B; \top)$ and $(E; \emptyset; \beta; B; \top)$ is a normal form, then $\text{LRA}(\beta) \models N$,
(iv) all final states $(E'; B'; \beta; S; V)$ match either (ii) or (iii).

# Strict Bounds

Introduce an infinitesimal small constant $\delta > 0$ and replace the strict bound by a non-strict one. For example, a bound $x < c$ is replaced by $x \leq c - \delta$ and $x > c$ is replaced by $x \geq c + \delta$.

Now $\delta$ is treated symbolically through the overall computation, i.e., we extend $\mathbb{Q}$ to $\mathbb{Q}_\delta$ with new pairs $(q, k)$ with $q, k \in \mathbb{Q}$ where $(q, k)$ represents $q + k\delta$ and the operations, relations on $\mathbb{Q}$ are lifted to $\mathbb{Q}_\delta$:

$$
\begin{aligned}
(q_1, k_1) + (q_2, k_2) &:= (q_1 + q_2, k_1 + k_2) \\
p(q, k) &:= (pq, pk) \\
(q_1, k_1) \leq (q_2, k_2) &:= (q_1 < q_2) \vee (q_1 = q_2 \wedge k_1 \leq k_2)
\end{aligned}
$$