# Linear Arithmetic

I start with a syntax that already contains $-$, $\leq$, $<$, $\geq$, $\not\approx$ and $\mathbb{Q}$. All these functions and relations are indeed expressible by first-order forumulas over 0, 1, $\approx$, and $>$.

For the semantics there are two approaches. Either providing axioms, i.e., closed formulas, for the above symbols and then considering all algebras satisfying the axioms, or fixing one particular algebra or a class of algebras.

## 6.2.1 Definition (LA Syntax)

The syntax of LA is

$$\Sigma_{\mathsf{LA}} = (\{\mathsf{LA}\}, \{0, 1, +, -\} \cup \mathbb{Q}, \{\leq, <, \not\approx, >, \geq\})$$

where $-$ is unitary and all other symbols have the usual arities.

Terms and formulas over $\Sigma_{\mathsf{LA}}$ are built in the classical free first-order way, see Section 3.1. All first-order notions, i.e., terms, atoms, equations, literals, clauses, etc. carry over to LA formulas. The atoms and terms built over the LA signature are written in their standard infix notation, i.e., I write $3 + 5$ instead of $+(3, 5)$. Note that the signature does not contain multiplication. A term $3x$ is just an abbreviation for a term $x + x + x$.

### 6.2.2 Definition (Linear Rational Arithmetic Standard Semantics)

The $\Sigma_{\text{LA}}$ algebra $\mathcal{A}_{\text{LRA}}$ is defined by $\text{LA}^{\mathcal{A}_{\text{LRA}}} = \mathbb{Q}$ and all other signature symbols are assigned the standard interpretations over the rationals.

Due to the expressive LA language there is no need for negative literals, because $(\neg <)^{\mathcal{A}_{\text{LRA}}} = (\geq)^{\mathcal{A}_{\text{LRA}}}$, $(\neg >)^{\mathcal{A}_{\text{LRA}}} = (\leq)^{\mathcal{A}_{\text{LRA}}}$, and $(\neg \approx)^{\mathcal{A}_{\text{LRA}}} = (\not\approx)^{\mathcal{A}_{\text{LRA}}}$.

Note the difference between the above standard semantics over $\Sigma_{\mathsf{LA}}$ and the free first-order semantics over $\Sigma_{\mathsf{LA}}$, Definition 3.2.1. The equation $3 + 4 \approx 5$ has a model in the free first-order semantics, hence it is satisfiable, whereas in the standard model of linear rational arithmetic, Definition 6.2.2, the equation $3 + 4 \approx 5$ is false.

In addition, with respect to the standard LRA semantics the definitions of validity, satisfiability coincide with truth and the definition of unsatisfiability coincides with falsehood. This is the result of a single algebra semantics.

# Fourier-Motzkin Quantifier Elimination

It is decidable whether a first-order formula over $\Sigma_{\mathrm{LA}}$ is true or false in the standard LRAsemantics. This was first discovered in 1826 by J. Fourier and re-discovered by T. Motzkin in 1936 and is called FM for short. Note that validity of a $\Sigma_{\mathrm{LA}}$ formula with respect to the standard semantics is undecidable

Similar to Congruence Closure, Section 6.1, the starting point of the procedure is a conjunction of atoms without atoms of the form $\not\approx$. These will eventually be replaced by a disjunction, i.e., an atom $t \not\approx s$ is replaced by $t < s \vee t > s$.

Every atom over the variables $x, y_1, \ldots, y_n$ can be converted into an equivalent atom $x \circ t[\vec{y}]$ or $0 \circ t[\vec{y}]$, where $\circ \in \{<, >, \leq, \geq, \approx, \not\approx\}$ and $t[\vec{y}]$ has the form $\sum_i q_i \cdot y_i + q_0$ where $q_i \in \mathbb{Q}$.

In other words, a variable $x$ can be either isolated on one side of the atom or eliminated completely. This is the starting point of the FM calculus deciding a conjunction of LA atoms without $\not\approx$ modulo the isolation of variables and the reduction of ground formulas to $\top$, $\bot$.

The calculus operates on a set of atoms $N$. The normal forms are conjunctions of atoms $s \circ t$ where $s, t$ do not contain any variables. These can be obviously eventually reduced to $\top$ or $\bot$. The FM calculus consists of two rules:

**Substitute** $N \uplus \{x \approx t\} \ \Rightarrow_{\mathrm{FM}} \ N\{x \mapsto t\}$
provided $x$ does not occur in $t$

**Eliminate** $N \uplus \bigcup_i \{x \circ_i^1 t_i\} \uplus \bigcup_j \{x \circ_j^2 s_j\} \ \Rightarrow_{\mathrm{FM}}$
$N \cup \bigcup_{i,j} \{t_i \circ_{i.j} s_j\}$
provided $x$ does not occur in $N$ nor in the $t_i, s_j$, $\circ_i^1 \in \{<, \leq\}$, $\circ_j^2 \in \{>, \geq\}$, and $\circ_{i,j} \ = >$ if $\circ_i^1 \ = <$ or $\circ_j^2 \ = >$, and $\circ_{i,j} \ = \geq$ otherwise

If all variablies in $N$ are implicitely existentially quantified, i.e., $N$ stands for $\exists\vec{x}.N$, then the above two rules constitute a sound and complete decision procedure for conjunctions of LA atoms without $\not\approx$.

### 6.2.3 Lemma (FM Termination on a Conjunction of Atoms)

FM terminates on a conjunction of atoms.

### 6.2.4 Lemma (FM Soundness and Completeness on a Conjunction of Atoms)

$N \Rightarrow^*_{\mathrm{FM}} \top$ iff $\mathcal{A}_{\mathrm{LRA}} \models \exists\vec{x}.N$.
$N \Rightarrow^*_{\mathrm{FM}} \bot$ iff $\mathcal{A}_{\mathrm{LRA}} \not\models \exists\vec{x}.N$.

The FM calculus on conjunctions of atoms can be extended to arbitrary closed LRA first-order formulas $\phi$. I always assume that different quantifier occurrences in $\phi$ bind different variables. This can always be obtained by renaming one variable.

The first step is to eliminate $\top$, $\bot$ from $\phi$ and to transform $\phi$ in negation normal form, see Section 3.9. The resulting formula only contains the operators $\forall$, $\exists$, $\wedge$, $\vee$, $\neg$, where all negation symbols occur in front of atoms.

The following rule can be used to remove the negation symbols as well:

**ElimNeg** $\chi[\neg\, s \circ^1 t]_p \;\Rightarrow_{\mathrm{FM}}\; \chi[s \circ^2 t]_p$

where the pairs $(\circ_1, \circ_2)$ are given by pairs $(<, \geq), (\leq, >), (\approx, \not\approx)$ and their symmetric variants

The above two FM rules on conjunctions cannot cope with atoms $s \not\approx t$, so they are eliminated as well:

**Elim**$\not\approx$ $\chi[s \not\approx t]_p \;\Rightarrow_{\mathrm{FM}}\; \chi[s < t \lor s > t]_p$

The next step is to compute a *Prenex Normal Form*, a formula $\{\exists, \forall\}x_1 \ldots \{\exists, \forall\}x_n.\phi$ where $\phi$ does not contain any quantifiers. This can be done by simply applying the mini-scoping rules, see Section 3.9, in the opposite direction:

**Prenex1** $\chi[(\forall x.\psi_1) \circ \psi_2]_p \;\Rightarrow_{\mathrm{FM}}\; \chi[\forall x.(\psi_1 \circ \psi_2)]_p$
provided $\circ \in \{\wedge, \vee\}$, $x \notin \mathrm{fvars}(\psi_2)$

**Prenex2** $\chi[(\exists x.\psi_1) \circ \psi_2]_p \;\Rightarrow_{\mathrm{FM}}\; \chi[\exists x.(\psi_1 \circ \psi_2)]_p$
provided $\circ \in \{\wedge, \vee\}$, $x \notin \mathrm{fvars}(\psi_2)$

**Prenex3** $\chi[(\forall x.\psi_1) \land (\forall y.\psi_2)]_p \Rightarrow_{\mathrm{FM}}$
$\chi[\forall x.(\psi_1 \land \psi_2\{y \mapsto x\})]_p$

**Prenex4** $\chi[(\exists x.\psi_1) \lor (\exists y.\psi_2)]_p \Rightarrow_{\mathrm{FM}}$
$\chi[\exists x.(\psi_1 \lor \psi_2\{y \mapsto x\})]_p$

where Prenex3 and Prenex4 are preferred over Prenex1 and Prenex2.

Finally, for the resulting formula $\{\exists, \forall\}x_1 \ldots \{\exists, \forall\}x_n.\phi$ in prenex normal form the FM algorithm computes a DNF of $\phi$ by exhaustively applying the rule PushConj, Section 2.5.2.

The result is a formula $\{\exists, \forall\}x_1 \ldots \{\exists, \forall\}x_n.\phi$ where $\phi$ is a DNF of atoms without containing an atom of the form $s \not\approx t$.

Then FM on formulas considers the quantifiers iteratively in an innermost way. For the formula $\{\exists,\forall\}x_1 \ldots \{\exists,\forall\}x_n.\phi$ always the innermost quantifier $\{\exists,\forall\}x_n$ is considered.

If it is an existential quantifier, $\exists x_n$, then the FM rules Substitute, Eliminate are applied to the variable $x_n$ for each conjunct $C_i$ of $\phi = C_1 \vee \ldots \vee C_n$. The result is a formula $\{\exists,\forall\}x_1 \ldots \{\exists,\forall\}x_{n-1}.(C'_1 \vee \ldots \vee C'_n)$ which is again in prenex DNF. Furthermore, by Lemma 6.2.4 it is equivalent to $\{\exists,\forall\}x_1 \ldots \{\exists,\forall\}x_n.\phi$.

If the innermost quantifier is a universal quantifier $\forall x_n$, then the formula is replaced by $\{\exists, \forall\}x_1 \ldots \{\exists, \forall\}x_{n-1}\neg\exists x_n.\neg\phi$ and the above steps for negation normal form and DNF are repeated for $\neg\phi$ resulting in an equivalent formula $\{\exists, \forall\}x_1 \ldots \{\exists, \forall\}x_{n-1}\neg\exists x_n.\phi'$ where $\phi'$ is in DNF and does not contain negation symbols nor atoms $s \not\approx t$.

Then the FM rules Substitute, Eliminate are applied to the variable $x_n$ for each conjunct $C_i$ of $\phi' = C_1 \vee \ldots \vee C_n$. The result is an equivalent formula $\{\exists, \forall\}x_1 \ldots \{\exists, \forall\}x_{n-1}.\neg(C'_1 \vee \ldots \vee C'_n)$. Finally, the above steps for negation normal form and DNF are repeated for $\neg(C'_1 \vee \ldots \vee C'_n)$ resulting in an equivalent formula $\{\exists, \forall\}x_1 \ldots \{\exists, \forall\}x_{n-1}.\phi''$ where $\phi'$ is in DNF and does not contain negation symbols nor atoms $s \not\approx t$. This completes for FM decision procedure for LRA formulas.

Every LRA formula can by reduced to $\top$ or $\bot$ via the FM decision procedure. Therefore LRA is called a *complete* theory, i.e., every closed formula over the signature of LRA is either true or false.

LA formulas over the rationals and over the reals are indistinguishable by first-order formulas over the signature of LRA. These properties do not hold for extended signatures, e.g., then additional free symbols are introduced. Furthermore, FM is no decision procedures over the integers, even if the LA syntax is restricted to integer constants.

# FM Complexity

The complexity of the FM calculus depends mostly on the quantifier alternations in $\{\exists, \forall\}x_1 \ldots \{\exists, \forall\}x_n.\phi$.

In case an existential quantifier $\exists$ is eliminated, the formula size grows worst-case quadratically, therefore $O(n^2)$ runtime. For $m$ quantifiers $\exists \ldots \exists$: a naive implementation needs worst-case $O(n^{2^m})$ runtime. It is not known whether an optimized implementation with simply exponential runtime is possible.

If there are $m$ quantifier alternations $\exists\forall\exists\forall\ldots\exists\forall$, a CNF to DNF conversion is required after each step. Each conversion has a worst-case exponential run time, see Section 2.5. Therefore, the overall procedure has a worst-case non-elementary runtime.