$$\begin{array}{ll}
 & \exists x_2\, \exists x_1.\ \phi[x_1, x_2, \vec{y}] \\
\leftrightarrow & \exists x_2.\ \big(\bigvee_{t_1 \in T_1} \phi[x_1, x_2, \vec{y}]\, \{x_1 \mapsto t_1\}\big) \\
\leftrightarrow & \bigvee_{t_1 \in T_1} (\exists x_2.\ \phi[x_1, x_2, \vec{y}]\, \{x_1 \mapsto t_1\}) \\
\leftrightarrow & \bigvee_{t_1 \in T_1} \bigvee_{t_2 \in T_2} (\phi[x_1, x_2, \vec{y}]\, \{x_1 \mapsto t_1\}\, \{x_2 \mapsto t_2\})
\end{array}$$

A sequence of m quantifier alternations $\exists\forall\exists\forall \ldots \exists$ turns the top-level disjunction after moving the inner negation into a top-level conjunction. An existential quantifier does not distribute over a conjunction, so the procedure needs $O(n^2)$ runtime for each step, therefore doubly exponential runtime in sum, $O(n^{2^m})$.

I

The formulas resulting from one virtual substitution step are typically highly redundant, because $\top$, $\bot$ are introduced into the formula. In any implementation, redundancy elimination, even beyond the elimination of $\top$, $\bot$ is mandatory for good performance. For example, an equation can also in virtual substitution be eliminated by a substitution, see FM, if the boolean structure and quantification enables it.

As a prerequisite, also virtual substitution needs the initial transformation into negation normal form. In case of the occurrence of logical equivalences, this can already result in an exponential blow up in size of the formula. Renaming, Section 3.9, is not an option because the LA language does not contain free predicate symbols.

C

The application of a test point set to a formula results in an equivalent formula. Actually, it is typically the goal of a quantifier elimination procedure to produce an equivalent formula with one quantifier less. If the goal is only to test whether a formula true, then a respective procedure gains more freedom. For example, instead of expanding the overall test point set disjunctions, it can just choose one test point and continue. Resulting in a less space consuming depth first procedure, rather then the breadth first search performed by virtual substitution.

The virtual substitution method can also be extended to non-linear arithmetic over the reals. It is recognized to be an efficient procedure in case of polynomials with low degrees.

6.2.4 Cooper's Algorithm for Linear Integer Arithmetic

At the beginning of the 20th century, mathematicians had the intuition that a problem will typically get easier if the domain under consideration gets smaller. A prominent example was thinking that (linear) rational (or real) arithmetic is more complicated than linear integer arithmetic. In this and the following section I show that it is the other way round by first presenting Cooper's [16] quantifier elimination algorithm in LIA and then a Branch & Bound approach for conjunctions of LIA atoms.

The first obvious difference between LRA and LIA is the restriction of the domain and coefficients to the integers. A first consequence of the integer domain is that strict and non-strict inequations can be easily exchanged, e.g., $t \leq c$ iff $t < c + 1$. Now, following the development of the FM procedure from Section 6.2.1, I consider a conjunction of inequations with atoms over $\{<, >\}$. Then the first FM step isolates some variable x, i.e., transforms all inequations containing x into $x \circ_i t_i$ where t_i does not contain x and $\circ_i \in \{<, >\}$. However, in LIA this is not possible. For example, an inequation $5x - 2y > 5$ can only be equivalently transformed into $x > \frac{2}{5}y + 1$ if x is to be isolated. But this inequation contains rational coefficients. Therefore, the isolation of x has to be done in two steps. Firstly, isolating x keeping the positive integer coefficients of x resulting in inequations $c_i x \circ_i t_i$ where $c_i \in \mathbb{N}$. Negative coefficients of x can be removed by multiplication of the inequation with -1. Now in order to combine the inequations and eliminate x, the c_i have to be all equal, i.e., the respective inequations have to be multiplied by some integer constant such that they have the form $dx \circ_i t'_i$ where d is the *least common multiple* (lcm) of all c_i. Following the FM idea, solvability of the $dx < t'_i$, $dx > s'_j$ is equivalent to solvability of the $s'_j < t'_i$. This is not the case for LIA, in general. A hint is already contained in the proof of Lemma 6.2.4, where the respective value for x in case is satisfiability of the $s'_j < t'_i$ is computed by $\frac{1}{2}(\min(\cup_i\{t'_i\}) + \max(\cup_j\{s'_j\}))$ which is not necessarily an integer.

For example, consider the two inequations $3x > 4$ and $2x < 3$. The lcm of $2, 3$ is 6, so the tranformed equations are $6x > 8$ and $6x < 9$ resulting after elimination of x in $8 < 9$ which is obviously true. In LRA the solution for $6x$ is then $\frac{8+9}{2} = \frac{17}{2}$, i.e., a solution for x is $\frac{17}{12}$, following the proof of Lemma 6.2.4. In fact, this is a solution to the initial inequations. However, it is not an integer solution. The two initial inequations do not have in integer solution because $x = 2$ satisfies the first but not the second, for $x = 1$ it is the other way round, and also for all other integers either the first or second inequation is true. Something is missing. For the example, the condition is missing that between 8 and 9 there is an integer that is divisible by 6. This can be expressed by a divisibility constraints $c \,|\, t$ expressing that the positive integer c divides t. Then the conjunction $6x > 8 \;\wedge\; 6x < 9$ is replaced with the LIA equivalent conjunction $x > 8 \;\wedge\; x < 9 \;\wedge\; 6 \,|\, x$. This conjunction has a solution if there is a value for x strictly greater 8 and smaller than 9 that is dividable by 6. This can be expressed by the finite disjunction

$$\bigvee_{i=1}^{i\leq 6} (8 + i > 8 \;\wedge\; 8 + i < 9 \;\wedge\; 6 \,|\, 8 + i)$$

where the first atom $8 + i > 8$ is obsolete, because it is true by construction. All of these six disjuncts are false in LIA, hence the initial inequations don't have a solution.

Definition 6.2.6 (LIA Syntax)**.** The syntax of LIA is

$$\Sigma_{\text{LIA}} = (\{\text{LIA}\}, \{0, 1, +, -\} \cup \mathbb{Z}, \{\leq, <, \,|\,, \not|, >, \geq\})$$

where $-$ is unitary and all other symbols have the usual arities. The bar $\mid$ is the devisability operator between a positive integer constant d and a term t, i.e., it generates atoms of the form $d \mid t$.

Definition 6.2.7 (Linear Integer Arithmetic Standard Semantics)**.** The Σ_{LIA} algebra $\mathcal{A}_{\text{LIA}}$ is defined by $\text{LA}^{\mathcal{A}_{\text{LIA}}} = \mathbb{Z}$ and all other signature symbols are assigned the standard interpretations over the integers.

In particular, $a \mid b$ for $a, b \in \mathbb{Z}, a > 0$ if there is some $c \in \mathbb{Z}$ such that $a * c = b$. Obviously, the difference between the respective syntax, Definition 6.2.1, and semantics, Definition 6.2.2, of rational integer arithmetic is the restriction to integer coefficients and to the integer domain plus the additional divisibility operator.

Cooper [15] showed that given two strict inequations $x < t$ and $x > s$, and a divisibility constraint $d \mid x$ the variable x can be eliminated in the above described way.

$$\exists x.(x < t \wedge x > s \wedge d \mid x) \quad \text{iff} \quad \bigvee_{i=1}^{i \leq d} (s + i < t \wedge d \mid s + i)$$

This needs to be further generalized to cope with $\not\mid$, multiple inequations, and divisibility constraints for some variable x. The actual procedure is then similar to virtual substitution, Section 6.2.3. Note that virtual substitution was invented after Cooper's algorithm for variable elimination over the integers.

Let $\exists x.\phi$ be a formula of LIA, where ϕ is in negation normal form, ϕ does not contain any quantifiers nor negation symbols, and the LIA relations occurring in ϕ are $\{<, >, \mid, \not\mid\}$. Any LIA formula can be effectively transformed into this form, see the discussion above and Section 6.2.1, and the rule ElimNeg. Furthermore, for all inequations $cx \circ t$ and divisibility atoms $a \circ' bx + s$, $\circ \in \{<, >\}$, $\circ' \in \{\mid, \not\mid\}$, I assume $c = 1$, $b = 1$. If c is negative for some inequation it is multiplied by -1 and then transformed into its strict form. If b is negative, for divisibility atoms it is sufficient to multiply the right hand side by -1. If there are atoms $c_i x \circ_i t_i$, $a_j \circ'_j b_j x + s_j$, in ϕ with $c_i > 1$ or $b_j > 1$ for some i, j, then the lcm d of the c_i, b_j is computed, the atoms are first replaced by $dx \circ_i \frac{d}{c_i} t_i$, $a_j \circ'_j dx + \frac{d}{b_j} s_j$, respectively, and finally they are replaced by $x \circ_i \frac{d}{c_i} t_i$, $a_j \circ'_j x + \frac{d}{b_j} s_j$, respectively, and the divisibility constraint $d \mid x$ is added conjunctively to ϕ.

Similar to the arguments for composing the virtual substitution test points, solutions for $\exists x.\phi$ can be considered from $-\infty$ to ∞ or the other way round. I explain the former, the latter is then a standard exercise. Let $x < t_i$, $x > s_j$, $a_k \mid x + r_k$, $b_h \not\mid x + l_h$ be all atoms in ϕ containing x where the t_i, s_j, r_k, l_h do not contain x. Let $p_1, \ldots, p_n$ be the positions of the atoms $x < t_i$ in ϕ and $q_1, \ldots, q_o$ be the positions of the atoms $x > s_j$ in ϕ. Let d be the lcm of the a_k, b_h. Then

$$\begin{gathered}\mathcal{A}_{\text{LIA}}(\beta) \models \exists x.\phi \\ \text{iff} \\ \mathcal{A}_{\text{LIA}}(\beta) \models \bigvee_{m=1}^{m\leq d} \phi[\top]_{p_1,\ldots,p_n}[\bot]_{q_1,\ldots,q_o}\{x \mapsto m\} \;\vee\; \bigvee_{m=1}^{m\leq d}\bigvee_{s_j} \phi\{x \mapsto s_j + m\}\end{gathered}$$

The first formula

$$\bigvee_{m=1}^{m\leq d} \phi[\top]_{p_1,\ldots,p_n}[\bot]_{q_1,\ldots,q_o}\{x \mapsto m\}$$

expresses the virtual substitution of $-\infty$ for x. In this case all atoms $x < t_i$ are true and all atoms $x > s_j$ are false for a sufficiently small value of x. For the divisibility constraints it is sufficient to find a satisfying assignment between 1 and d as a representative for a sufficiently small value of x.

The second formula

$$\bigvee_{m=1}^{m\leq d}\bigvee_{s_j} \phi\{x \mapsto s_j + m\}$$

represents the elimination of x by considering all solutions of combinations $s_j < t_i$ plus the requirement of the existence of a value for x that can be divided by d. This is implemented by substituting $s_j + m$ for x for all lower bounds s_j and all values of m between 1 and d. Of course, all resulting atoms $s_j + m > s_j$ are true for all j.

The worst-case complexity of Cooper's variable elimination procedure is immense. The lcm d computed for each eliminated variable x grows worst-case exponentially in the size of ϕ. Thus also the formula after eliminating x is exponentially larger then ϕ. The overall runtime is again non-elementary, similar to FM quantifier elimination. The formulas resulting from the test points $-\infty$ and $s_j + m$ contain a lot of redundancy that can be eliminated afterwards. However, even if the formula size can be drastically reduced through redundancy elimination, in each step the exponentially growing coefficients m remain.

Due to its inherent complexity, Cooper's elimination procedure is rarely used in practice. It mainly serves as a theoretical background for more practical procedures. The next section, Section 6.2.5, introduces a search procedure for conjunctions of LIA inequations where variables can be a priori singly exponentially bound.

There are two operations on divisibility atoms that can be used for simplification. Given two divisibility constraints $c_1 \mid a_1x + t_1$ and $c_2 \mid a_2x + t_2$ the variable x can be eliminated from one of the constraints by

$$\begin{gathered} c_1 \mid a_1x + t_1 \;\wedge\; c_2 \mid a_2x + t_2 \\ \text{iff} \\ c_1c_2 \mid dx + t_1c_2p + t_2c_1q \;\wedge\; d \mid t_1a_2 - t_2a_1 \end{gathered}$$

where $d = \gcd(a_1c_2, a_2c_1)$ and $d = pa_1c_2 + qa_2c_1$ for integers p, q. If Euclid's algorithm is used for the gcd computation, then p, q also result from the algorithm as a by-product.

A divisibility atom $c \mid ax+t$ has a solution iff $d \mid t$ where $d = \gcd(c, a)$. In case there is a solution, they are $x = -p\frac{t}{d} + i\frac{c}{d}$ for all integers i where $d = pa + qc$ for integers p, q. Note that p, q always exists because $d = \gcd(c, a)$.

6.2.5 Branch and Bound for LIA

Historic and Bibliographic Remarks