V. Senni, A. Pettorossi, M. Proietti

A FOLDING RULE FOR ELIMINATING EXISTENTIAL VARIABLES FROM CONSTRAINT LOGIC PROGRAMS

R. 08-03, 2008
Abstract

The existential variables of a clause in a constraint logic program are the variables which occur in the body of the clause and not in its head. The elimination of these variables is a transformation technique which is often used for improving program efficiency and verifying program properties. We consider a folding transformation rule which ensures the elimination of existential variables and we propose an algorithm for applying this rule in the case where the constraints are linear inequations over rational or real numbers. The algorithm combines techniques for matching terms modulo equational theories and techniques for solving systems of linear inequations. Through some examples we show that an implementation of our folding algorithm performs well in practice.

Key words: Program transformation, folding rule, variable elimination, constraint logic programming
1. Introduction

Constraint logic programming is a very expressive language for writing programs in a declarative way and for specifying and verifying properties of software systems [9]. When writing programs in a declarative style or writing specifications, one often uses existential variables, that is, variables which occur in the body of a clause and not in its head. However, the use of existential variables may give rise to inefficient or even nonterminating computations (and this may happen when an existential variable denotes an intermediate data structure or when an existential variable ranges over an infinite set). For this reason some transformation techniques have been proposed for eliminating those variables from logic programs and constraint logic programs [13, 14]. These techniques make use of the unfolding and folding rules which have been first proposed in the context of functional programming by Burstall and Darlington [5], and then extended to logic programming [17] and to constraint logic programming [3, 7, 8, 11].

For instance, let us consider the problem of checking whether or not a list $P$ such that the sum of all elements of $P$ is at least $M$. A constraint logic program that solves this problem is the following:

1. $\text{prefixsum}(L, M) ← N ≥ M ∧ \text{app}(P, S, L) ∧ \text{sum}(P, N)$
2. $\text{app}([], Y, Y) ←$
3. $\text{app}([H|X], Y, [H|Z]) ← \text{app}(X, Y, Z)$
4. $\text{sum}([], 0) ←$
5. $\text{sum}(H|X), N) ← N = H + R ∧ \text{sum}(X, R)$

When answering queries which are instances of the atom $\text{prefixsum}(L, M)$, the program computes values for the variables $P$, $S$, and $N$ which are the existential variables of clause 1 and, in fact, are not needed in the final answer. We can eliminate these existential variables and improve the efficiency of the program, by applying the unfolding and folding rules as follows. From clause 1, by applying the unfolding rule several times, we derive:

6. $\text{prefixsum}(L, M) ← 0 ≥ M$
7. $\text{prefixsum}([H|T], M) ← N ≥ M ∧ N = H + R ∧ \text{app}(P, S, T) ∧ \text{sum}(P, R)$

Now we fold clause 7 by using clause 1 and we derive:

8. $\text{prefixsum}([H|T], M) ← \text{prefixsum}(T, M - H)$

For this folding step we have used the fact that, in our theory of constraints, clause 7 is equivalent to the clause $\text{prefixsum}([H|T], M) ← R ≥ M - H ∧ \text{app}(P, S, T) ∧ \text{sum}(P, R)$, whose body is an instance of the body of clause 1. The final program, consisting of clauses 6 and 8, has no existential variables and, thus, does not construct unnecessary intermediate values for computing the relation $\text{prefixsum}$.

As shown in the above example the folding rule plays a particularly relevant role in the techniques for eliminating existential variables. (In particular, it would have been impossible to eliminate all existential variables from the clauses defining $\text{prefixsum}$ by using the unfolding rule only.) For that reason in this paper we focus our attention on the folding rule, which in the general case can be defined as follows.

Let (i) $H$ and $K$ be atoms, (ii) $c$ and $d$ be constraints, and (iii) $G$ and $B$ be goals (that is, conjunctions of literals). Given two clauses $\gamma$: $H ← c ∧ G$ and $δ$: $K ← d ∧ B$, if there exist a constraint $e$, a substitution $ϑ$, and a goal $R$ such that $H ← e ∧ G$ is equivalent (w.r.t. a given theory of constraints) to $H ← e ∧ (d ∧ B)ϑ ∧ R$, then $γ$ is folded into the clause $η$: $H ← e ∧ Kϑ ∧ R$. In order to use the folding rule to eliminate existential variables we also require that every variable occurring in $Kϑ$ also occurs in $H$. 

In the literature no algorithm is provided to determine whether or not, given a theory of constraints, the suitable $e$, $\vartheta$, and $R$ which are required for folding, do exist [3, 7, 8, 11]. In this paper we propose an algorithm based on linear algebra and term rewriting techniques for computing $e$, $\vartheta$, and $R$, if they exist, in the case when the constraints are linear inequations over the rational numbers. The techniques we will present are valid without relevant changes also when the inequations are over the real numbers. As an example of application of the folding algorithm, let us consider the following clauses:

$$
\gamma: \quad p(X_1, X_2, X_3) \leftarrow X_1 < 1 \land X_1 \geq Z_1 + 1 \land Z_2 > 0 \land q(Z_1, f(X_3), Z_2) \land r(X_2)
$$

$$
\delta: \quad s(Y_1, Y_2, Y_3) \leftarrow W_1 < 0 \land Y_1 - 3 > 2W_1 \land W_2 > 0 \land q(W_1, Y_3, W_2)
$$

and suppose that we want to fold $\gamma$ using $\delta$ for eliminating the existential variables $Z_1$ and $Z_2$ occurring in $\gamma$. Our folding algorithm $\text{FA}$ computes (see Examples 1–4 in Section 4): (i) the constraint $e: X_1 < 1$, (ii) the substitution $\vartheta: \{Y_1/2X_1+1, Y_2/a, Y_3/f(X_3), W_1/Z_1, W_2/Z_2\}$, where $a$ is an arbitrary new constant, and (iii) the goal $R: r(X_2)$, and the clause derived by folding $\gamma$ using $\delta$ is:

$$
\eta: \quad p(X_1, X_2, X_3) \leftarrow X_1 < 1 \land s(2X_1+1, a, f(X_3)) \land r(X_2)
$$

which has no existential variables. (The correctness of this folding step can easily be checked by unfolding $\eta$ w.r.t. $s(2X_1+1, a, f(X_3))$.) In general, a triple $(e, \vartheta, R)$ that satisfies the conditions for the applicability of the folding rule may not exist or may not be unique. For this reason, our folding algorithm is nondeterministic and in different runs it may compute different folded clauses.

The paper is organized as follows. In Section 2 we introduce some basic definitions concerning constraint logic programs. In Section 3 we present the folding rule which we use for eliminating existential variables. In Section 4 we describe our algorithm for applying the folding rule and we prove the soundness and completeness of this algorithm with respect to the declarative specification of the rule. In Section 5 we analyze the complexity of our folding algorithm. We also describe an implementation of that algorithm and we evaluate its performance by presenting some experimental results. Finally, in Section 6 we discuss the related work and we suggest some directions for future investigations.

2. Preliminary Definitions

In this section we recall some basic definitions concerning constraint logic programs, where the constraints are conjunctions of linear inequations over the rational numbers. As already mentioned, the results we will present in this paper are valid without relevant changes also when the constraints are conjunctions of linear inequations over the real numbers. For notions not defined here the reader may refer to [9, 10].

Let us consider a first order language $\mathcal{L}$ given by a set $\text{Var}$ of variables, a set $\text{Fun}$ of function symbols, and a set $\text{Pred}$ of predicate symbols. Let $+$ denote addition, $\cdot$ denote multiplication, and $\mathbb{Q}$ denote the set of rational numbers. We assume that $\{+, \cdot\} \cup \mathbb{Q} \subseteq \text{Fun}$ (in particular, every rational number is assumed to be a 0-ary function symbol). We also assume that the predicate symbols $\geq$ and $>$ denoting inequality and strict inequality, respectively, belong to $\text{Pred}$.

In order to distinguish terms representing rational numbers from other terms (which may be viewed as finite trees), we assume that $\mathcal{L}$ is a typed language [10] with two basic types: $\text{rat}$, which is the type of the rational numbers, and $\text{tree}$, which is the type of the finite trees. We also consider types constructed from basic types by the usual type constructors $\times$ and $\rightarrow$. A variable $X \in \text{Var}$ has either type $\text{rat}$ or type $\text{tree}$. We denote by $\text{Var}_{\text{rat}}$ and $\text{Var}_{\text{tree}}$ the set
of variables of type \texttt{rat} and \texttt{tree}, respectively. A predicate symbol of arity \(n\) and a function symbol of arity \(n\) in \(\mathcal{L}\) have types of the form \(\tau_1 \times \cdots \times \tau_n\) and \(\tau_1 \times \cdots \times \tau_n \rightarrow \tau_{n+1}\), respectively, for some types \(\tau_1, \ldots, \tau_n, \tau_{n+1} \in \{\texttt{rat, tree}\}\). In particular, the predicate symbols \(\geq\) and \(>\) have type \texttt{rat} \times \texttt{rat}, the function symbols \(+\) and \(-\) have type \texttt{rat} \times \texttt{rat} \rightarrow \texttt{rat}, and the rational numbers have type \texttt{rat}. The function symbols in \(\{+, \cdot\} \cup Q\) are the only symbols whose type is \(\tau_1 \times \cdots \times \tau_n \rightarrow \texttt{rat}\), for some types \(\tau_1, \ldots, \tau_n\), with \(n \geq 0\).

A term \(u\) is either a term of type \texttt{rat} or a term of type \texttt{tree}. A term \(p\) of type \texttt{rat} is a linear polynomial of the form \(a_1X_1 + \cdots + a_nX_n + a_{n+1}\), where \(a_1, \ldots, a_{n+1}\) are rational numbers and \(X_1, \ldots, X_n\) are variables in \(\text{Var}_{\text{rat}}\) (a monomial of the form \(aX\) stands for the term \(aX\)). A term \(t\) of type \texttt{tree} is either a variable \(X\) in \(\text{Var}_{\text{tree}}\) or a term of the form \(f(u_1, \ldots, u_n)\), where \(f\) is a function symbol of type \(\tau_1 \times \cdots \times \tau_n \rightarrow \text{tree}\), and \(u_1, \ldots, u_n\) are terms of type \(\tau_1, \ldots, \tau_n\), respectively.

An atomic constraint is a linear inequation of the form \(p_1 \geq p_2\) or \(p_1 > p_2\). A constraint is a conjunction \(c_1 \wedge \cdots \wedge c_n\), where \(c_1, \ldots, c_n\) are atomic constraints. When \(n = 0\) we write \(c_1 \wedge \cdots \wedge c_n\) as true. A constraint of the form \(p_1 \geq p_2 \wedge p_2 \geq p_1\) is abbreviated as the equation \(p_1 = p_2\) (which, thus, is not an atomic constraint).

An atom is of the form \(r(u_1, \ldots, u_n)\), where \(r\) is a predicate symbol, not in \(\{\geq, >\}\), of type \(\tau_1 \times \cdots \times \tau_n\) and \(u_1, \ldots, u_n\) are terms of type \(\tau_1, \ldots, \tau_n\), respectively. A literal is either an atom (called a positive literal) or a negated atom (called a negative literal). A goal is a conjunction \(L_1 \wedge \cdots \wedge L_n\) of literals, with \(n \geq 0\). The conjunction of 0 literals is denoted by true. A constrained goal is a conjunction \(c \wedge G\), where \(c\) is a constraint and \(G\) is a goal. A clause is of the form \(H \leftarrow c \wedge G\), where \(H\) is an atom and \(c \wedge G\) is a constrained goal. A constraint logic program is a set of clauses. A formula of the language \(\mathcal{L}\) is constructed as usual in first order logic from the symbols of \(\mathcal{L}\) by using the logical connectives \(\wedge, \vee, \neg, \rightarrow, \leftarrow, \leftrightarrow\), and the quantifiers \(\exists, \forall\).

If \(f\) is a term or a formula then by \(\text{Var}_{\text{rat}}(f)\) and \(\text{Var}_{\text{tree}}(f)\) we denote, respectively, the set of variables of type \texttt{rat} and of type \texttt{tree} occurring in \(f\). By \(\text{Var}(f)\) we denote the set of all variables occurring in \(f\), that is, \(\text{Var}_{\text{rat}}(f) \cup \text{Var}_{\text{tree}}(f)\). Similar notation will also be used for the variables occurring in sets of terms and sets of formulas. Given a clause \(\gamma: H \leftarrow c \wedge G\), by \(\text{EVars}(\gamma)\) we denote the set of the existential variables of \(\gamma\), which is defined to be \(\text{Vars}(c \wedge G) \setminus \text{Vars}(H)\). The constraint-local variables of \(\gamma\) are the variables in the set \(\text{Vars}(c) \setminus \text{Vars}([H, G])\). Given a set \(X = \{X_1, \ldots, X_n\}\) of variables and a formula \(\varphi\), by \(\forall X \varphi\) we denote the formula \(\forall X_1 \cdots \forall X_n \varphi\) and by \(\exists X \varphi\) we denote the formula \(\exists X_1 \cdots \exists X_n \varphi\). By \(\forall(\varphi)\) and \(\exists(\varphi)\) we denote the universal closure and the existential closure of \(\varphi\), respectively. In what follows we will use the notion of substitution as defined in [10] with the following extra type condition: given any substitution \(\{X_i/t_i, \ldots, X_n/t_n\}\), for \(i = 1, \ldots, n\), the type of \(X_i\) is equal to the type of \(t_i\).

Let \(\mathcal{L}_{\text{rat}}\) denote the sublanguage of \(\mathcal{L}\) given by the set \(\text{Var}_{\text{rat}}\) of variables, the set \(\{+, \cdot\} \cup Q\) of function symbols, and the set \(\{\geq, >\}\) of predicate symbols. We denote by \(Q\) the interpretation which assigns to every function symbol or predicate symbol of \(\mathcal{L}_{\text{rat}}\) the expected function or relation on \(\mathcal{Q}\). For a formula \(\varphi\) of \(\mathcal{L}_{\text{rat}}\) (in particular, for a constraint), the satisfaction relation \(\mathcal{Q} \models \varphi\) is defined as usual in first order logic. A \(Q\)-interpretation is an interpretation \(I\) for the typed language \(\mathcal{L}\) which agrees with \(Q\) for each formula \(\varphi\) of \(\mathcal{L}_{\text{rat}}\), that is, for each \(\varphi\) of \(\mathcal{L}_{\text{rat}}\), \(I \models \varphi\) iff \(\mathcal{Q} \models \varphi\). The definition of a \(Q\)-interpretation for typed languages is a straightforward extension of the one for untyped languages [9]. We say that a \(Q\)-interpretation \(I\) is a \(Q\)-model of a program \(P\) if for every clause \(\gamma \in P\) we have that \(I \models \forall(\gamma)\). Similarly to the case of logic programs, we can define stratified constraint logic programs and we have that every such program \(P\) has a perfect \(Q\)-model [8, 9, 11], denoted by \(M(P)\).
A solution of a set $C$ of constraints is a ground substitution $\sigma$ of the form \( \{X_1/a_1, \ldots, X_n/a_n\} \), where \( \{X_1, \ldots, X_n\} = \text{Vars}(C) \) and $a_1, \ldots, a_n \in \mathbb{Q}$, such that $\mathbb{Q} \models c \sigma$ for every $c \in C$. A set of constraints is said to be satisfiable if it has a solution.

We assume that we are given a function $\text{solve}$ that takes in input a set $C$ of constraints and returns in output a solution $\sigma$ of $C$, if $C$ is satisfiable, and $\text{fail}$ otherwise. The function $\text{solve}$ can be implemented, for instance, by using the Fourier-Motzkin algorithm or the Khachiyan algorithm [16]. We assume that we are also given a function $\text{project}$ such that for every constraint $c$ and for every finite set of variables $X \subseteq \text{Vars}_{\text{rat}}, \mathbb{Q} \models \forall X ((\exists Y \ c) \leftrightarrow \text{project}(c, X))$, where $Y = \text{Vars}(c) - X$ and $\text{Vars}(\text{project}(c, X)) \subseteq X$. The $\text{project}$ function can be implemented, for instance, by using the Fourier-Motzkin algorithm or the algorithm presented in [19].

A clause $\gamma : H \leftarrow c \land G$ is said to be in normal form if (i) every term of type $\text{rat}$ occurring in $G$ is a variable, (ii) each variable of type $\text{rat}$ occurs at most once in $G$, (iii) $\text{Vars}_{\text{rat}}(H) \cap \text{Vars}_{\text{rat}}(G) = \emptyset$, and (iv) $\gamma$ has no constraint-local variables. It is always possible to transform any clause $\gamma_1$ into a clause $\gamma_2$ such that $\gamma_2$ has the same $\mathbb{Q}$-models as $\gamma_1$ and $\gamma_2$ is in normal form. (In particular, the constraint-local variables of any given clause can be eliminated by applying the $\text{project}$ function.) The clause $\gamma_2$ is called a normal form of $\gamma_1$. Without loss of generality, when presenting the folding rule and the algorithm for its application, we will assume that the clauses are in normal form.

Definition 2.1. Given two clauses $\gamma_1$ and $\gamma_2$, we write $\gamma_1 \equiv \gamma_2$ if there exist a normal form $H \leftarrow c_1 \land B_1$ of $\gamma_1$, a normal form $H \leftarrow c_2 \land B_2$ of $\gamma_2$, and a variable renaming $\rho$ such that: (1) $H = H_{\rho}$, (2) $B_1 =_{AC} B_2 \rho$, and (3) $\mathbb{Q} \models \forall (c_1 \leftrightarrow c_2 \rho)$, where $=_{AC}$ denotes equality modulo the equational theory of associativity and commutativity of conjunction. We will refer to this theory as the $AC \land$ theory [1].

Proposition 2.2. (i) The relation $\equiv$ is an equivalence relation. (ii) If $\gamma_1 \equiv \gamma_2$ then, for every $\mathbb{Q}$-interpretation $I$, $I \models \gamma_1$ iff $I \models \gamma_2$. (iii) If $\gamma_2$ is a normal form of $\gamma_1$ then $\gamma_1 \equiv \gamma_2$.

### 3. The Folding Rule

In this section we introduce our folding transformation rule which is a variant of the many folding rules considered in the literature [3, 7, 8, 11]. In particular, by using our variant of the folding rule we may replace a constrained goal occurring in the body of a clause where some existential variables occur, by an atom which has no existential variables in the folded clause.

Definition 3.1 (Folding Rule) Let $\gamma : H \leftarrow c \land G$ and $\delta : K \leftarrow d \land B$ be clauses in normal form without variables in common. Suppose also that there exist a constraint $e$, a substitution $\vartheta$, and a goal $R$ such that: (1) $\gamma \equiv H \leftarrow e \land d \vartheta \land B \vartheta \land R$; (2) for every variable $X$ in $\text{EVars}(\delta)$, the following conditions hold: (2.1) $X \vartheta$ is a variable not occurring in $\{H, e, R\}$, and (2.2) $X \vartheta$ does not occur in the term $Y \vartheta$, for every variable $Y$ occurring in $d \land B$ and different from $X$; (3) $\text{Vars}(K \vartheta) \subseteq \text{Vars}(H)$. By folding clause $\gamma$ using clause $\delta$ we derive the clause $\eta : H \leftarrow e \land K \vartheta \land R$.

Condition (3) ensures that no existential variable of $\eta$ occurs in $K \vartheta$. However, in $e$ or $R$ some existential variables may still occur. These variables may be eliminated by further folding steps using clause $\delta$ again or other clauses. In Theorem 3.2 below we establish the correctness of the folding rule w.r.t. the perfect model semantics. That correctness follows immediately from [3, 7, 8].
In order to state Theorem 3.2 we need the following notion. A transformation sequence is a sequence $P_0, \ldots, P_n$ of programs such that, for $k = 0, \ldots, n-1$, program $P_{k+1}$ is derived from program $P_k$ by an application of one of the following transformation rules: definition, unfolding, folding. For a detailed presentation of the definition and unfolding rules we refer to [8]. An application of the folding rule is defined as follows. For $k = 0, \ldots, n$, by $Defs_k$ we denote the set of clauses introduced by the definition rule during the construction of $P_0, \ldots, P_k$. Program $P_{k+1}$ is derived from program $P_k$ by an application of the folding rule if $P_{k+1} = (P_k - \{\gamma\}) \cup \{\eta\}$, where $\gamma$ is a clause in $P_k$, $\delta$ is a clause in $Defs_k$, and $\eta$ is the clause derived by folding $\gamma$ using $\delta$ as indicated in Definition 3.1.

**Theorem 3.2.** [8] Let $P_0$ be a stratified program and let $P_0, \ldots, P_n$ be a transformation sequence. Suppose that, for $k = 0, \ldots, n-1$, if $P_{k+1}$ is derived from $P_k$ by folding clause $\gamma$ using clause $\delta \in Defs_k$, then there exists $j$, with $0 < j < n$, such that $\delta \in P_j$ and $P_{j+1}$ is derived from $P_j$ by unfolding $\delta$ w.r.t. a positive literal in its body. Then $P_0 \cup Defs_n$ and $P_n$ are stratified and $M(P_0 \cup Defs_n) = M(P_n)$.

### 4. An Algorithm for Applying the Folding Rule

Now we will present an algorithm for determining whether or not a clause $\gamma: H \leftarrow c \land G$ can be folded using a clause $\delta: K \leftarrow d \land B$, according to Definition 3.1. The objective of our folding algorithm is to find a constraint $e$, a substitution $\vartheta$, and a goal $R$ such that Point (1) (that is, $\gamma \equiv H \leftarrow e \land d \vartheta \land B \vartheta \land R$), Point (2), and Point (3) of Definition 3.1 hold. Our algorithm computes $e$, $\vartheta$, and $R$, if they exist, by applying two procedures: (i) the goal matching procedure, called $GM$, which matches the goal $G$ against $B$ and returns a substitution $\alpha$ and a goal $R$ such that $G =_{AC} B \alpha \land R$, and (ii) the constraint matching procedure, called $CM$, which matches the constraint $c$ against $d \alpha$ and returns a substitution $\beta$ and a constraint $e$ such that $c$ is equivalent to $e \land d \alpha \beta$ in the theory of constraints. The substitution $\vartheta$ to be found is the composition of the substitutions $\alpha$ and $\beta$, denoted $\alpha \beta$. The output of the folding algorithm is either the clause $\eta: H \leftarrow e \land K \vartheta \land R$, or $fail$ if folding is not possible. Since Definition 3.1 does not determine $e$, $\vartheta$, and $R$ in a unique way, our folding algorithm is nondeterministic and, as already said, in different runs it may compute different folded clauses.

#### 4.1. Goal Matching

Let us now present the goal matching procedure $GM$. This procedure uses the notion of binding which is defined as follows: a binding is a pair of the form $e_1/e_2$, where $e_1$ and $e_2$ are either both goals or both terms. Thus, the notion of set of bindings is a generalization of the notion of substitution.

**Goal Matching Procedure: $GM$**

**Input:** two clauses in normal form without variables in common $\gamma: H \leftarrow c \land G$ and $\delta: K \leftarrow d \land B$.

**Output:** a substitution $\alpha$ and a goal $R$ such that: (1) $G =_{AC} B \alpha \land R$; (2) for every variable $X$ in $EVars(\delta)$, (2.1) $X \alpha$ is a variable not occurring in $\{H, R\}$, and (2.2) $X \alpha$ does not occur in the term $Y \alpha$, for every variable $Y$ occurring in $d \land B$ and different from $X$; (3) $Vars_{tree}(K \alpha) \subseteq Vars(H)$.

If such $\alpha$ and $R$ do not exist, then $fail$.

Consider a set $Bnds$ of bindings initialized to the singleton $\{(B \land T)/G\}$, where $T$ is a new symbol denoting a variable ranging over goals. Consider also the rewrite rules (i)–(x) listed...
below. When the left hand side of a rule is written as $Bnds_1 \cup Bnds_2 \implies \ldots$, we assume that $Bnds_1 \cap Bnds_2 = \emptyset$.

(i) \{$(L_1 \land B_1 \land T) / (G_1 \land L_2 \land G_2)\} \cup Bnds \implies \{L_1 / L_2, (B_1 \land T) / (G_1 \land G_2)\} \cup Bnds$

where: (1) $L_1$ and $L_2$ are either both positive or both negative literals and have the same predicate symbol with the same arity, and (2) $B_1$, $G_1$, and $G_2$ are goals (possibly, the empty conjunction true);

(ii) \{$\neg A_1 / \neg A_2\} \cup Bnds \implies \{A_1 / A_2\} \cup Bnds$

(iii) \{$a(s_1, \ldots, s_n) / a(t_1, \ldots, t_n)\} \cup Bnds \implies \{s_1/t_1, \ldots, s_n/t_n\} \cup Bnds$

(iv) \{$a(s_1, \ldots, s_m) / b(t_1, \ldots, t_n)\} \cup Bnds \implies \text{fail}$, if $a$ is different from $b$ or $m \neq n$;

(v) \{$a(s_1, \ldots, s_n) / X\} \cup Bnds \implies \text{fail}$, if $X \in \text{Vars}(\gamma)$;

(vi) \{$X/s\} \cup Bnds \implies \text{fail}$, if $X \in \text{Vars}(\delta)$ and $X/t \in Bnds$ for some $t$ syntactically different from $s$;

(vii) \{$X/s\} \cup Bnds \implies \text{fail}$, if $X \in \text{EVars}(\delta)$ and one of the following three conditions holds: (1) $s$ is not a variable, or (2) $s \in \text{Vars}(H)$, or (3) there exists $Y \in \text{Vars}(d \land B)$ different from $X$ such that: (3.1) $Y/t \in Bnds$, for some term $t$, and (3.2) $s \in \text{Vars}(t)$;

(viii) \{$X/s, T/G_1\} \cup Bnds \implies \text{fail}$, if $X \in \text{EVars}(\delta)$ and $s \in \text{Vars}(G_1)$;

(ix) \{$X/s\} \cup Bnds \implies \text{fail}$, if $X \in \text{Vars}_{\text{tree}}(K)$ and $\text{Vars}(s) \not\subseteq \text{Vars}(H)$;

(x) $Bnds \implies \{X/s\} \cup Bnds$, where $s$ is an arbitrary term of type $\text{tree}$ such that $\text{Vars}(s) \subseteq \text{Vars}(H)$, if $X \in \text{Vars}_{\text{tree}}(K) - \text{Vars}(B)$ and there is no term $t$ such that $X/t \in Bnds$.

If there exist a set of bindings $\alpha$ (which, by construction, is a substitution) and a goal $R$ such that: (c1) $(B \land T)/G \implies ^\ast \alpha \cup \{T/R\}$ (where $T/R \not\in \alpha$) and (c2) no $Bnds$ exists such that $\alpha \cup \{T/R\} \implies Bnds$ (that is, informally, $\alpha \cup \{T/R\}$ is a maximally rewritten, non-failing set of bindings derived from the singleton $(B \land T)/G$)

THEN return $\alpha$ and $R$ ELSE return fail.

Rule (i) associates each literal in $B$ with a literal in $G$ in a nondeterministic way. Rules (ii)–(vi) are a specialization to our case of the usual rules for matching [18]. Rules (vii)–(x) ensure that any pair $<\alpha, R>$ computed by $\text{GM}$ satisfies Conditions (2) and (3) of the folding rule, or if no such pair exists, then $\text{GM}$ returns fail.

Example 1. Let us apply the procedure $\text{GM}$ to the clauses $\gamma$ and $\delta$ presented in the Introduction, where the predicates $p$, $q$, $r$, and $s$ are of type $\text{rat} \times \text{tree} \times \text{tree}$, $\text{rat} \times \text{tree} \times \text{rat}$, $\text{tree}$, and $\text{rat} \times \text{tree} \times \text{tree}$, respectively, and the function $f$ is of type $\text{tree} \rightarrow \text{tree}$. The clauses $\gamma$ and $\delta$ are in normal form and have no variables in common. The procedure $\text{GM}$ performs the following rewritings, where the arrow $\Rightarrow$ denotes an application of the rewrite rule $r$:

$$
\{q(W_1, Y_3, W_2) \land T / \{q(Z_1, f(X_3), Z_2) \land r(X_2)\}\} \\
\downarrow \{q(W_1, Y_3, W_2) / q(Z_1, f(X_3), Z_2), \ T / r(X_2)\} \\
\implies \{W_1 / Z_1, \ Y_3 / f(X_3), \ W_2 / Z_2, \ T / r(X_2)\} \\
\Rightarrow \{W_1 / Z_1, \ Y_3 / f(X_3), \ W_2 / Z_2, \ Y_2 / a, \ T / r(X_2)\}
$$

In the final set of bindings, the term $a$ is an arbitrary constant of type $\text{tree}$. The output of $\text{GM}$ is the substitution $\alpha : \{W_1 / Z_1, \ Y_3 / f(X_3), \ W_2 / Z_2, \ Y_2 / a\}$ and the goal $R : r(X_2)$. 
The goal matching procedure is *sound* in the sense that if \( \text{GM} \) returns a substitution \( \alpha \) and a goal \( R \), then \( \alpha \) and \( R \) satisfy the output conditions of \( \text{GM} \). The goal matching procedure is also *complete* in the sense that if there exist a substitution \( \alpha \) and a goal \( R \) that satisfy the output conditions of \( \text{GM} \), then it does not return *fail*. The termination of the goal matching procedure can be shown via an argument based on the multiset ordering of the size of the bindings. Indeed, each of the rules (i)–(ix) replaces a binding by a finite number of smaller bindings, and rule (x) can be applied at most once for each variable occurring in the head of clause \( \delta \). A detailed proof of the soundness, completeness, and termination of \( \text{GM} \) can be found in the Appendix (see Theorem A.4).

### 4.2. Constraint Matching

Let us assume that given two clauses in normal form \( \gamma : H \leftarrow c \land G \) and \( \delta : K \leftarrow d \land B \), the goal matching procedure \( \text{GM} \) returns the substitution \( \alpha \) and the goal \( R \). By using \( \alpha \) and \( R \), we can construct the two clauses in normal form: \( H \leftarrow c \land B \alpha \land R \) and \( K \alpha \leftarrow d \alpha \land B \alpha \) such that \( G =_{AC} B \alpha \land R \). The constraint matching procedure \( \text{CM} \) takes in input these two clauses, which, for reasons of simplicity, we now rename as \( \gamma' : H' \leftarrow c \land B' \land R \) and \( \delta' : K' \leftarrow d' \land B' \), respectively, and returns in output a constraint \( e \) and a substitution \( \beta \) such that: (1) \( \gamma' \equiv H' \leftarrow e \land d' \land B' \land R \), (2) \( B' \beta = B' \), (3) \( \text{Vars}(K' \beta) \subseteq \text{Vars}(H) \), and (4) \( \text{Vars}(e) \subseteq \text{Vars}(\{H, R\}) \). If such \( e \) and \( \beta \) do not exist, then the procedure \( \text{CM} \) returns *fail*.

Let \( \tilde{e} \) denote the constraint \( \text{project}(c, X) \), where \( X = \text{Vars}(c) - \text{Vars}(B') \) (the definition of the \( \text{project} \) function is given in Section 2). By Lemma 4.1 below, the procedure \( \text{CM} \) does not lose any solution if it returns as constraint \( e \) the value of \( \tilde{e} \), and then compute a substitution \( \beta \) such that \( Q \models (c \leftarrow (\tilde{e} \land d' \beta)) \), \( B' \beta = B' \), and \( \text{Vars}(K' \beta) \subseteq \text{Vars}(H) \) hold.

**Lemma 4.1.** Let \( \gamma' : H \leftarrow c \land B' \land R \) and \( \delta' : K' \leftarrow d' \land B' \) be the input clauses of the constraint matching procedure. For every substitution \( \beta \), there exists a constraint \( e \) such that the following four conditions hold: (1) \( \gamma' \equiv H \leftarrow e \land d' \land B' \land R \), (2) \( B' \beta = B' \), (3) \( \text{Vars}(K' \beta) \subseteq \text{Vars}(H) \), and (4) \( \text{Vars}(e) \subseteq \text{Vars}(\{H, R\}) \) if\( f \). 

The following example illustrates the fact that if the procedure \( \text{CM} \) returns for the constraint \( e \) the value of \( \tilde{e} \), then \( \text{CM} \) may compute the substitution \( \beta \) by solving a set of constraints over the set \( Q \) of the rational numbers.

**Example 2.** Let us consider again the clauses \( \gamma \) and \( \delta \) of the Introduction. Let \( \alpha \) and \( r(X_2) \) be the substitution and the goal computed by applying the procedure \( \text{GM} \) to \( \gamma \) and \( \delta \) as shown in the above Example 1. Let us then consider the following clauses \( \gamma' : H \leftarrow c \land B' \land R \) and \( \delta' : K' \leftarrow d' \land B' \) which are equal to \( \gamma \) and \( \delta \alpha \), respectively:

\[
\begin{align*}
\gamma' : & \quad p(X_1, X_2, X_3) \leftarrow X_1 < 1 \land X_1 \geq Z_1 + 1 \land Z_2 > 0 \land q(Z_1, f(X_3), Z_2) \land r(X_2) \\
\delta' : & \quad s(Y_1, a, f(X_3)) \leftarrow Z_1 < 0 \land Y_1 - 3 \geq 2Z_1 \land Z_2 > 0 \land q(Z_1, f(X_3), Z_2)
\end{align*}
\]

Thus, the constraint \( c \) is \( X_1 < 1 \land X_1 \geq Z_1 + 1 \land Z_2 > 0 \) and the goal \( B' \) is \( q(Z_1, f(X_3), Z_2) \). Those two clauses \( \gamma' \) and \( \delta' \) are the input to the procedure \( \text{CM} \). The constraint \( \tilde{e} \) returned by the procedure \( \text{CM} \) is \( \text{project}((X_1 < 1 \land X_1 \geq Z_1 + 1 \land Z_2 > 0), \{X_1\}) \), which is equivalent to \( X_1 < 1 \).

Now we will compute a substitution \( \beta \) such that: (i) \( Q \models (c \leftarrow (\tilde{e} \land d' \beta)) \) holds, and (ii) Conditions (2) and (3) as stated in Lemma 4.1, hold. These three conditions are as follows:

\[
Q \models \forall (X_1 < 1 \land X_1 \geq Z_1 + 1 \land Z_2 > 0) \leftarrow X_1 < 1 \land (Z_1 < 0 \land Y_1 - 3 \geq 2Z_1 \land Z_2 > 0) \beta
\]
q(Z_1, f(X_3), Z_2)\beta = q(Z_1, f(X_3), Z_2) \quad \text{(that is, } Z_1\beta = Z_1, \ X_3\beta = X_3, \ Z_2\beta = Z_2) \quad (2)

\text{Vars}(s(Y_1, a, f(X_3))\beta) \subseteq \{X_1, X_2, X_3\} \quad (3)

We have that Equation (f.0) holds if the following equivalences (f.1), (f.2), and (f.3), and implication (f.4) hold:

\begin{align*}
Q \models \forall (X_1 < 1 \leftrightarrow X_1 < 1) & \quad \text{(f.1)} \\
Q \models \forall (X_1 \geq Z_1 + 1 \leftrightarrow (Y_1 - 3 \geq 2Z_1)\beta) & \quad \text{(f.2)} \\
Q \models \forall (Z_2 > 0 \leftrightarrow (Z_2 > 0)\beta) & \quad \text{(f.3)} \\
Q \models \forall (X_1 < 1 \land X_1 \geq Z_1 + 1 \land Z_2 > 0 \rightarrow (Z_1 < 0)\beta) & \quad \text{(f.4)}
\end{align*}

Equivalence (f.1) trivially holds. Equivalence (f.2) can be reduced to an equation over the rational numbers because Equivalence (f.2) holds if there exists a rational number \( k > 0 \) such that

\[ Q \models \forall (k(X_1 - Z_1 - 1) = (Y_1 - 3 - 2Z_1)\beta) \]

holds. By Condition (2), the substitution \( \beta \) is the identity on \( Z_1 \) and, hence, the equation \( k(X_1 - Z_1 - 1) = (Y_1 - 3 - 2Z_1)\beta \) holds for any \( \beta \) such that

\[ Y_1\beta = (2 - k)Z_1 + kX_1 + 3 - k \]

Since by Condition (3) \( \text{Vars}(s(Y_1, a, f(X_3))\beta) \subseteq \{X_1, X_2, X_3\} \), we get that \( k = 2 \) and, thus, the binding \( Y_1/(2X_1 + 1) \) belongs to \( \beta \). We have that also Equivalences (f.3) and (f.4) hold for \( \beta = (Y_1/(2X_1 + 1)) \). We will see that, indeed, this substitution \( \beta \) is the one returned by the constraint matching procedure \text{CM} \ we will introduce below.

The crucial steps in Example 2 have been the following two: (i) the reduction of Equivalence (f.0) to a set of equivalences between atomic constraints (see (f.1)–(f.3)) or implications with atomic conclusions (see (f.4)), and (ii) the reduction of one of these equivalences, namely (f.2), to an equation over the rational numbers, via the introduction of the auxiliary rational parameter \( k \).

Now we introduce some notions and we state some properties (see Lemma 4.2 and Theorem 4.3) which will be exploited by the constraint matching procedure \text{CM} \ for performing in the general case those two crucial reduction steps. Indeed, the procedure \text{CM} \ is a set of rewrite rules which reduce the equivalence between \( c \) and \( \overline{c} \land d'\beta \) to a set of equations and inequations over the rational numbers, via the introduction of suitable auxiliary parameters. The properties we now state will also provide sufficient conditions under which we are guaranteed to find the desired substitution \( \beta \), if there exists one.

A conjunction \( a_1 \land \ldots \land a_m \) of (not necessarily distinct) atomic constraints is said to be redundant if there exists an atomic constraint \( a_i \), with \( 0 \leq i \leq m \), such that \( Q \models \forall((a_1 \land \ldots \land a_{i-1} \land a_{i+1} \land \ldots \land a_m) \rightarrow a_i) \). In this case we also say that \( a_i \) is redundant in \( a_1 \land \ldots \land a_m \). Thus, the empty conjunction true is non-redundant and an atomic constraint \( a \) is redundant iff \( Q \models \forall(a) \). Given a redundant constraint \( c \), we can always derive a non-redundant constraint \( c' \) which is equivalent to \( c \), that is, \( Q \models \forall(c \leftrightarrow c') \), by repeatedly eliminating from the constraint at hand an atomic constraint which is redundant in that constraint.

Without loss of generality, we can assume that any given constraint \( c \) is of the form \( p_1 \rho_1 0 \land \ldots \land p_m \rho_m 0 \), where \( m \geq 0 \) and \( p_1, \ldots, p_m \in \{\geq, >\} \). We define the interior of \( c \), denoted \( \text{interior}(c) \), to be the constraint \( p_1 > 0 \land \ldots \land p_m > 0 \).

A constraint \( c \) is said to be admissible if both \( c \) and \( \text{interior}(c) \) are satisfiable and non-redundant. For instance, the constraint \( c_1: X - Y \geq 0 \land Y \geq 0 \) is admissible, while the constraint \( c_2: X - Y \geq 0 \land Y \geq 0 \land X > 0 \) is not admissible (indeed, \( c_2 \) is non-redundant, but \( \text{interior}(c_2) \):
Let us consider an admissible constraint $a$ of the form $a_1 \land \ldots \land a_m$ and a constraint $b$ of the form $b_1 \land \ldots \land b_n$, where $a_1, \ldots, a_m, b_1, \ldots, b_n$ are atomic constraints (in particular, they are not equalities). We have that $Q \models (a \leftrightarrow b)$ holds iff there exists an injection $\mu : \{1, \ldots, m\} \rightarrow \{1, \ldots, n\}$ such that for $i = 1, \ldots, m$, $Q \models (a_i \leftrightarrow b_{\mu(i)})$ and for $j = 1, \ldots, n$, if $j \notin \{\mu(i) \mid 1 \leq i \leq m\}$, then $Q \models (a \rightarrow b_j)$.

In Lemma 4.2 above we have required that the constraint $a$ be admissible. This is a needed hypothesis as the following example shows. Let us consider the non-admissible constraint $c_2 : X - Y \geq 0 \land Y \geq 0 \land X > 0$ and the constraint $c_3 : X - Y \geq 0 \land Y \geq 0 \land X + Y > 0$. We have that $Q \models (c_2 \leftrightarrow c_3)$ and yet there is no injection $\mu$ which has the properties stated in Lemma 4.2.

Given the clauses $\gamma' : H \leftarrow c \land B' \land R$ and $\delta' : K' \leftarrow d' \land B'$ such that: (i) $c$ is an admissible constraint of the form $a_1 \land \ldots \land a_m$, and (ii) $\bar{c} \land d'$ is a constraint of the form $b_1 \land \ldots \land b_n$, where $\bar{c}$ is project$(c, \text{Vars}(c) - \text{Vars}(B'))$, the constraint matching procedure $CM$ may exploit Lemma 4.2 and compute a substitution $\beta$ which satisfies $Q \models (c \leftrightarrow (\bar{c} \land d' \beta))$ and Conditions (2) and (3) of Lemma 4.1, according to the following algorithm: first (1) $CM$ computes an injection $\mu$ from $\{1, \ldots, m\}$ to $\{1, \ldots, n\}$, (see rule (i) in the procedure $CM$ below) and then (2) it computes $\beta$ such that: (2.ii) for $i = 1, \ldots, m$, $Q \models (a_i \leftarrow b_{\mu(i)} \beta)$, and (2.ii) for $j = 1, \ldots, n$, if $j \notin \{\mu(i) \mid 1 \leq i \leq m\}$, then $Q \models (c \rightarrow b_j \beta)$ (see rules (ii)-(v) in the procedure $CM$ below).

By Lemma 4.2, one can show that if the constraint $c$ is admissible, the above algorithm for computing the substitution $\beta$ which satisfies $Q \models (c \leftrightarrow (\bar{c} \land d' \beta))$ and Conditions (2) and (3) of Lemma 4.1 is complete in the sense that it computes such a substitution $\beta$ if there exists one.

In order to compute $\beta$ satisfying Point (2.ii) above, the procedure $CM$ makes use of the following Property P1: given the satisfiable, non-redundant atomic constraints $p > 0$ and $q > 0$, we have that $Q \models (p > 0 \leftarrow q > 0)$ holds iff there exists a rational number $k > 0$ such that $Q \models (p k - q = 0)$ holds. Property P1 holds also if we consider $\forall (p > 0 \leftarrow q \geq 0)$, instead of $\forall (p > 0 \leftarrow q > 0)$.

In order to compute $\beta$ satisfying Point (2.ii) above, the procedure $CM$ makes use of the following Theorem 3.3 which is a generalization of the above Property P1 and it is an extension of Farkas’ Lemma to the case of systems of weak and strict inequalities [16], rather than weak inequalities only.

Theorem 4.3. Suppose that $p_1 \rho_1 0, \ldots, p_m \rho_m 0, p_{m+1} \rho_{m+1} 0$ are atomic constraints such that, for $i = 1, \ldots, m + 1$, $\rho_i \in \{\geq, >\}$ and $Q \models \exists(p_1 \rho_1 0 \land \ldots \land p_m \rho_m 0 \land p_{m+1} \rho_{m+1} 0)$ iff there exist $k_1 \geq 0, \ldots, k_{m+1} \geq 0$ such that: (i) $Q \models (k_1 p_1 + \cdots + k_m p_m + k_{m+1} = p_{m+1})$, and (ii) if $p_{m+1}$ is $> 0$ then $(\sum_{i \in I} k_i) > 0$, where $I = \{i \mid 1 \leq i \leq m+1, \rho_i \text{ is } >\}$.

As we will see below, the constraint matching procedure $CM$ may generate bilinear polynomials (see rules (i)-(iii)), that is, polynomials of a particular form, which we now define. Let $p$ be a polynomial and $(P_1, P_2)$ be a partition of a (proper or not) superset of $\text{Vars}(p)$. The polynomial $p$ is said to be bilinear in the partition $(P_1, P_2)$ if there exists a polynomial $q$ such that $Q \models (p = q)$ and $q$ is a sum of monomials, each of which is of the form: either (i) $k V U$, where $k$ is a rational number, $V \in P_1$, and $U \in P_2$, or (ii) $k U$, where $k$ is a rational number and $U \in P_1 \cup P_2$, or (iii) $k$, where $k$ is a rational number.
Let us consider a polynomial \( p \) which is bilinear in the partition \( \langle P_1, P_2 \rangle \), where \( P_2 = \{U_1, \ldots, U_m\} \). A normal form of \( p \), denoted \( nf(p) \), w.r.t. a given ordering \( U_1, \ldots, U_m \) of the variables in \( P_2 \), is any polynomial which is derived from \( p \) by: (i) computing a polynomial of the form \( r_1 U_1 + \cdots + r_m U_m + r_{m+1} \) such that: (i.1) \( Q \models \forall (r_1 = 0) \), and (i.2) \( r_1, \ldots, r_{m+1} \) are linear polynomials with variables in \( P_1 \), and (ii) erasing from that polynomial every summand \( r_i U_i \) such that \( Q \models \forall (r_i = 0) \).

Recall that in this paper any equation between polynomials of the form \( p_1 = p_2 \) is an abbreviation for the two inequations \( p_1 \geq p_2 \) and \( p_2 \geq p_1 \).

**Constraint Matching Procedure: CM**

*Input:* two clauses in normal form, possibly with variables in common, \( \gamma' : H \leftarrow c \land B' \land R \) and \( \delta : K' \leftarrow d' \land B' \).

*Output:* a constraint \( e \) and a substitution \( \beta \) such that: (1) \( \gamma' \equiv H \leftarrow e \land d' \land B' \land R \), (2) \( B' \land \beta = B' \), (3) \( Vars(K' \land \beta) \subseteq Vars(H) \), and (4) \( Vars(e) \subseteq Vars(\{H, R\}) \). If such \( e \) and \( \beta \) do not exist, then fail.

IF \( c \) is unsatisfiable THEN return an arbitrary unsatisfiable constraint \( e \) such that \( Vars(e) \subseteq Vars(\{H, R\}) \) and a substitution \( \beta \) of the form \( \{U_1/a_1, \ldots, U_s/a_s\} \), where \( \{U_1, \ldots, U_s\} = Vars_{rat}(K') \) and \( a_1, \ldots, a_s \) are arbitrary terms of type \texttt{rat} such that, for \( i = 1, \ldots, s \), \( Vars(a_i) \subseteq Vars(H) \).

ELSE proceed as follows.

Let \( X \) be the set \( Vars(c) - Vars(B') \), \( Y \) be the set \( Vars(d') - Vars(B') \), and \( Z \) be the set \( Vars_{rat}(B') \). Let \( e \) be the constraint \textit{project}(\( c, X \)). Without loss of generality, we may assume that: \( c \) is a constraint of the form \( p_1 p_1 0 \land \cdots \land p_m p_m 0 \), where for \( i = 1, \ldots, m \), \( p_i \) is a linear polynomial and \( p_i \in \{\geq, >\} \), and \( e \land d' \) is a constraint of the form \( q_1 \pi_1 0 \land \cdots \land q_n \pi_n 0 \), where for \( j = 1, \ldots, n \), \( q_j \) is a linear polynomial and \( \pi_j \in \{\geq, >\} \).

Let us consider the following rewrite rules (i)–(v) which are all of the form:

\[
\langle f_1 \leftarrow g_1, S_1, \sigma_1 \rangle \implies \langle f_2 \leftarrow g_2, S_2, \sigma_2 \rangle
\]

where: (1.1) \( f_1 \) and \( f_2 \) are constraints, (1.2) \( g_1 \) and \( g_2 \) are formulas of the form \( q \rho 0 \), where \( q \) is a bilinear polynomial and \( \rho \in \{\geq, >\} \), (2) \( S_1 \) and \( S_2 \) are sets of formulas of the form \( q \rho 0 \), where \( q \) is a bilinear polynomial and \( \rho \in \{\geq, >\} \), and (3) \( \sigma_1 \) and \( \sigma_2 \) are substitutions. The polynomials occurring in \( g_1, g_2, S_1, \) and \( S_2 \) are all bilinear in the partition \( \langle W, X \cup Y \cup Z \rangle \), where \( W \) is the set of the new variables introduced during the application of the rewrite rules (i)–(v). The normal forms of those bilinear polynomials are all defined w.r.t. any variable ordering of the form: \( Z_1, \ldots, Z_h, Y_1, \ldots, Y_k, X_1, \ldots, X_l \), where \( \{Z_1, \ldots, Z_h\} = Z \), \( \{Y_1, \ldots, Y_k\} = Y \), and \( \{X_1, \ldots, X_l\} = X \). In the rewrite rules (iv) and (v), whenever \( S_1 \) is written as \( A \cup B \), we assume that \( A \cap B = \emptyset \).

(i) \( \langle p \rho 0 \land f \leftarrow g_1 \land q \rho 0 \land g_2, S, \sigma \rangle \implies \langle f \leftarrow g_1 \land g_2, \{nf(V p - q) = 0, V \geq 0\} \cup S, \sigma \rangle \)

where \( V \) is a new variable and \( \rho \in \{\geq, >\} \);

(ii) \( \langle true \leftarrow q \geq 0 \land g, S, \sigma \rangle \implies \langle true \leftarrow g, \{nf(V_1 p_1 + \cdots + V_m p_m + V_{m+1} - q) = 0, V_1 \geq 0, \ldots, V_{m+1} \geq 0\} \cup S, \sigma \rangle \)

where \( V_1, \ldots, V_{m+1} \) are new variables;
(iii) $\langle \text{true} \leftarrow q > 0 \land g, \ S, \ \sigma \rangle \implies$

$\langle \text{true} \leftarrow g, \ \{nf(V_1p_1 + \ldots + V_mp_m + V_{m+1} - q) = 0, \ V_1 \geq 0, \ldots, V_{m+1} \geq 0, \ \{\sum_{i \in I} V_i > 0\} \cup S, \ \sigma\rangle$

where $V_1, \ldots, V_{m+1}$ are new variables and $I = \{i \mid 1 \leq i \leq m+1, \ \rho_i \text{ is } > \}$;

(iv) $\langle f \leftarrow g, \ \{pU + q = 0\} \cup S, \ \sigma \rangle \implies \langle f \leftarrow g, \ \{p = 0, q = 0\} \cup S, \ \sigma\rangle$

if $U \in X \cup Z$;

(v) $\langle f \leftarrow g, \ \{aU + q = 0\} \cup S, \ \sigma \rangle \implies$

$\langle f \leftarrow (g\{U - \frac{s}{2}\}), \ \{nf(p\{U - \frac{s}{2}\}) \rho p0 \mid p \rho 0 \in S\}, \ \sigma\{U - \frac{s}{2}\}\rangle$

if $U \in Y$, $Vars(q) \cap Vars(R) = \emptyset$, and $a \in (Q - \{0\})$;

If there exist a set $C$ of atomic constraints and a substitution $\sigma_C$ such that: (c1) $\langle c \leftarrow e \land d' \rangle, \ \emptyset, \ \emptyset$ $\implies*$ $\langle \text{true} \leftarrow \text{true}, \ C, \ \sigma_Y \rangle$, (c2) for every $f \in C$, we have that $f$ is of the form $pp0$, where $p$ is a linear polynomial and $\rho \in \{\geq, >\}$, and $Vars(f) \subseteq W$, where $W$ is the set of the new variables introduced during the rewriting steps from $\langle c \leftarrow e \land d', \ \emptyset, \ \emptyset \rangle$ to $\langle \text{true} \leftarrow \text{true}, \ C, \ \sigma_Y \rangle$, and (c3) $C$ is satisfiable and $\text{solve}(C) = \sigma_w$.

Then construct a ground substitution $\sigma_G$ of the form $\{U_1/a_1, \ldots, U_s/a_s\}$, where $\{U_1, \ldots, U_s\} = Vars_{\text{rat}}(K'\sigma_Y \sigma_w) - Vars(H)$ and $a_1, \ldots, a_s$ are arbitrary terms of type $\text{rat}$ such that, for $i = 1, \ldots, s$, $Vars(a_i) \subseteq Vars(H)$, and return the constraint $e$ and the substitution $\beta = \varphi_Y \sigma_G$, where $\varphi_Y$ is the substitution $\sigma_Y \sigma_w$ restricted to the set $Y$.

ELSE return $\text{fail}$.

Note that: (1) the procedure $CM$ is nondeterministic (in particular, rule (i) associates a constraint in $c$ with a constraint in $e \land d'$ in a nondeterministic way), and (2) in order to apply rules (iv) and (v), $pU$ and $aU$ should be the leftmost monomials in the bilinear polynomials $pU + q$ and $aU + q$, respectively.

The procedure $CM$ is sound in the sense that if it returns the constraint $e$ and the substitution $\beta$, then $e$ and $\beta$ satisfy the output Conditions (1)–(4) of $CM$. We now sketch the proof of soundness. A detailed proof is given in the Appendix (see Theorem A.13). By Lemma 4.1, it is enough to show that, for $e = \text{project}(c, X)$, $Q \models \forall (c \leftarrow e \land d' \beta)$ and the output Conditions (2) and (3) hold. By the definition of the sets $X, Y$, and $Z$ of variables, without loss of generality, we may assume that $X\beta = X, Z\beta = Z$, and $Z \cap Vars(Y\beta) = \emptyset$ (for a proof of these facts, see Theorem A.13 in the Appendix). Hence, it is enough to show that the substitution $\beta$ is such that $Q \models \forall (c \leftarrow (e \land d')\beta)$ (where $\beta$ is applied also to the constraint $e$) and Conditions (2) and (3) hold.

The procedure $CM$ starts from the initial triple $\langle c \leftarrow e \land d', \emptyset, \emptyset \rangle$ and nondeterministically constructs a sequence of rewritings by applying the rewrite rules (i)–(v), until Conditions (c1)–(c3) are satisfied. If no such sequence of rewritings exists, $CM$ returns $\text{fail}$. We will say that a substitution $\beta$ satisfies a triple $\langle f \leftarrow g, \ S, \ \sigma \rangle$ if there exists a value for the variables in the set $W$ such that $Q \models \forall X \forall Z (f \leftarrow g\beta)$, $Q \models \forall X \forall Z (S\beta)$, and, for every variable $U \in Y$, $Q \models \forall (U\sigma\beta = U\beta)$ (note that the variables in the set $W$ may occur in the formula $g$, in the set $S$, and in the substitution $\sigma$).

Now we show that each rewrite rule which rewrites a triple $\langle f_1 \leftarrow g_1, \ S_1, \ \sigma_1 \rangle$ into a triple of the form $\langle f_2 \leftarrow g_2, \ S_2, \ \sigma_2 \rangle$, is sound in the sense that, for all substitutions $\beta$, if $\beta$ satisfies the
triple \( \langle f_2 \leftrightarrow g_2, S_2, \sigma_2 \rangle \) then \( \beta \) satisfies also the triple \( \langle f_1 \leftrightarrow g_1, S_1, \sigma_1 \rangle \). Moreover, if \( \beta \) satisfies the initial triple \( \langle c \leftrightarrow e \land d', \emptyset, \emptyset \rangle \) then \( \beta \) is a correct output substitution.

Let us now consider an application of each of the rewrite rules (i)–(v). For an application of the rewrite rule (i), \( \mathbf{CM} \) selects an atomic constraint \( pp0 \) from \( f_1 \) and an atomic constraint \( qp0 \) from \( f_2 \). Thus, by a sequence of applications of rule (i) which rewrites the initial triple into a triple of the form \( \langle \text{true} \leftrightarrow g, S_2, \sigma_2 \rangle \), \( \mathbf{CM} \) constructs an injective mapping from the atomic constraints in \( c \) to the atomic constraints in \( e \land d' \). If such injective mapping does not exist, \( \mathbf{CM} \) returns \text{fail}. Rule (i) discards the selected atomic constraint \( pp0 \) and \( qp0 \) and adds to the second component of the triple the equation \( nf(Vp - q) = 0 \) and the constraint \( V > 0 \). The soundness of rule (i) follows from Property P1, which ensures that \( Q \models \forall (pp0 \leftrightarrow (qp0)\beta) \) if there exists a rational number \( V > 0 \) such that \( Q \models \forall (nf(Vp - q) = 0) \).

Rules (ii) and (iii) are applied when the first component of the triple at hand is of the form \( \text{true} \leftrightarrow q \), that is, the atomic constraints in \( q \) do not belong to the image of the injection computed by rule (i). Every application of rules (ii) and (iii) discards an atomic constraint \( qp0 \) from \( q \) and adds to the second component of the triple the equation \( nf(Vp - q) = 0 \) and a set \( \{V_1 \geq 0, \ldots, V_{m+1} \geq 0\} \) of constraints (with an additional constraint of the form \( \sum_{i \in I} V_i > 0 \) in case of rule (iii)). The soundness of rules (ii) and (iii) follows from the fact that \( c \) is a constraint of the form \( p_1 p_1 \lor \ldots \lor p_m p_m \) and, by Theorem 4.3, we have that \( Q \models \forall (nf(Vp_1 + \ldots + V_{m+m+1} = q) = 0) \) (together with the constraint \( \sum_{i \in I} V_i > 0 \) in case of rule (iii)).

The soundness of rules (iv) and (v) is based on the following Property P2:

\[
Q \models \forall ((pU + q = 0) \leftrightarrow (p = 0 \land q = 0) \lor (p \neq 0 \land U = -Q / p))
\]

Rule (iv) replaces an equation \( pU + q = 0 \), where \( U \in X \cup Z \), by the two equations \( p = 0 \) and \( q = 0 \). The soundness of this rule follows from the fact that, for any value of the variables \( V_1, \ldots, V_r \in W \), \( Q \models \forall ((pU + q) = 0) \iff Q \models \forall (p = 0) \) and \( Q \models \forall (q = 0) \). This equivalence follows from Property P2, by observing that: (1) \( pU = pU \) because \( U \in X \cup Z \) and \( pU + q \) is bilinear in \( (W, X \cup Y \cup Z) \) and, therefore, \( Vars(p) \subseteq W \), and (2) the case where \( Q \models \forall (U = -Q / p) \) is impossible because, for any \( \beta, U \not\in Vars(q(\beta)) \) (indeed: (2.1) since \( pU + q \) is in normal form, we have that \( U \not\in Vars(q) \), (2.2) since \( Z \cap Vars(Y\beta) = \emptyset \) if \( U \in Z \) then we have that \( U \not\in Vars(q(\beta)) \), and (2.3) since by the variable ordering we use for computing normal forms no variable in the set \( Y \) occurs in \( pU + q \) to the right of a variable in the set \( X \), if \( U \in X \) then we have that \( Y \cap Vars(q) = \emptyset \) and, thus, \( q(\beta) = q \).

Rule (v) discards an equation \( aU + q = 0 \), where \( U \in Y \), \( Vars(q) \cap Vars(R) = \emptyset \), and \( a \in Q - \{0\} \), and applies the substitution \( \{U - Q / a \} \) to all components of the triple at hand. (Note that \( U \) does not occur in \( f \).) The soundness of this rule follows from the fact that, for any value of the variables \( V_1, \ldots, V_r \in W \), \( Q \models \forall ((aU + q) = 0) \iff Q \models \forall (U(\beta = -Q / p)) \). This equivalence follows from Property P2, because \( a \in Q - \{0\} \). (Note that the condition \( a \in Q - \{0\} \) is always satisfied because \( aU + q = 0 \) is in normal form and it can be shown that, by construction, \( Vars(a) = \emptyset \) whenever \( U \in Y \). Note also that the condition \( Vars(q) \cap Vars(R) = \emptyset \) is required to satisfy the output Condition (3) of \( \mathbf{CM} \).

If the rewriting process terminates and from the initial triple \( \langle c \leftrightarrow e \land d', \emptyset, \emptyset \rangle \) we derive, by a sequence of applications of rules (i)–(v), a new triple \( \langle \text{true} \leftrightarrow \text{true}, C, \sigma \rangle \) such that Conditions (c1)–(c3) of \( \mathbf{CM} \) hold, then no rule can be applied to the triple \( \langle \text{true} \leftrightarrow \text{true}, C, \sigma \rangle \) and, hence, in the set \( C \) there is no occurrence of a variable in \( X \cup Y \cup Z \). Moreover, the second component \( C \) of this triple will be a set of constraints on the variables \( V_1, \ldots, V_r \), where \( W = \)
Let us consider again the clauses \( \land \leftrightarrow \land - \langle \). We also consider the clauses \( \sigma \). The substitution computed by applying the procedure \( CM \) is complete, in the sense that if there exist a constraint \( c \) and a substitution \( \beta \) that satisfy the output conditions of \( CM \) then it does not return \( \text{ fail} \) (see Theorem A.13 in the Appendix).

The termination of the constraint matching procedure is a consequence of the following facts: (1) each application of rules (i), (ii), and (iii) reduces the number of atomic constraints occurring in the formula \( g \) in the first component of the triple \( \langle f \leftarrow g, S, \sigma \rangle \) at hand; (2) each application of rule (iv) does not modify the first component of the triple \( \langle f \leftarrow g, S, \sigma \rangle \) at hand, does not introduce any new variables, and reduces the number of occurrences in \( S \) of the variables in the set \( X \cup Z \); (3) each application of rule (v) does not modify the number of atomic constraints in the first component of the triple \( \langle f \leftarrow g, S, \sigma \rangle \) at hand and eliminates all occurrences in \( S \) of a variable in the set \( Y \). Thus, the termination of \( CM \) can be proved by a suitable lexicographic ordering on the number of the atomic constraints and variables. The details of the termination proof can be found in the Appendix (see the proof of Theorem A.13).

The following example illustrates an execution of the procedure \( CM \).

**Example 3.** Let us consider again the clauses \( \gamma \) and \( \delta \) of the Introduction and let \( \alpha \) be the substitution computed by applying the procedure \( GM \) to \( \gamma \) and \( \delta \) as shown in Example 1. Let us also consider the clauses \( \gamma' \) and \( \delta' \), where \( \gamma' \) is \( \gamma \) and \( \delta' \) is \( \delta \alpha \), that is,

\[
\gamma': \quad p(X_1, X_2, X_3) \leftarrow X_1 < 1 \land X_1 \geq Z_1 + 1 \land Z_2 > 0 \land q(Z_1, f(X_3), Z_2) \land r(X_2)
\]

\[
\delta': \quad s(Y_1, a, f(X_3)) \leftarrow Z_1 < 0 \land Y_1 - 3 \geq 2 Z_1 \land Z_2 > 0 \land q(Z_1, f(X_3), Z_2)
\]

Now we apply the procedure \( CM \) to clauses \( \gamma' \) and \( \delta' \). The constraint \( X_1 < 1 \land X_1 \geq Z_1 + 1 \land Z_2 > 0 \) occurring in \( \gamma' \) is satisfiable. The procedure \( CM \) starts off by computing the constraint \( e \) as follows:

\[
e = \text{ project}(X_1 < 1 \land X_1 \geq Z_1 + 1 \land Z_2 > 0, \{X_1\}) = X_1 < 1
\]

Now \( CM \) performs the following rewritings, where: (i) all polynomials are bilinear in the partition \( \langle \{V_1, \ldots, V_7\}, \{X_1, Y_1, Z_1, Z_2\} \rangle \), (ii) their normal forms are computed w.r.t. the variable ordering \( Z_1, Z_2, Y_1, X_1 \), and (iii) \( \text{r}^k \) denotes \( k \) applications of rule \( r \). (We have underlined the constraints that are rewritten by an application of a rule. Note also that the atomic constraints occurring in the initial triple are the ones in \( \gamma' \) and \( \delta' \), rewritten into the form \( p > 0 \) or \( p \geq 0 \).)

\[
\langle 1 - X_1 > 0 \land X_1 - Z_1 - 1 > 0 \land Z_2 > 0 \rangle \leftarrow \langle 1 - X_1 > 0 \land - Z_1 > 0 \land Y_1 - 3 - 2 Z_1 > 0 \land Z_2 > 0 \rangle, \quad \emptyset, \quad \emptyset
\]

\[
\overset{1}{\overset{\text{r}}{\vdash}} \langle (X_1 - Z_1 - 1 \geq 0 \land Z_2 > 0) \leftarrow (-Z_1 > 0 \land Y_1 - 3 - 2 Z_1 > 0 \land Z_2 > 0),
\{1 - V_1\} X_1 + V_1 - 1 = 0, \quad V_1 > 0, \quad \emptyset \rangle
\]

\[
\overset{1}{\overset{\text{r}}{\vdash}} \langle Z_2 > 0 \leftrightarrow (-Z_1 > 0 \land Z_2 > 0),
\{1 - V_1\} X_1 + V_1 - 1 = 0, \quad V_1 > 0, \quad (2 - V_2) Z_1 - Y_1 + V_2 X_1 - V_2 + 3 = 0, \quad V_2 > 0 \}, \quad \emptyset \rangle
\]

\[
\overset{1}{\overset{\text{r}}{\vdash}} \langle \text{ true} \leftrightarrow - Z_2 > 0,
\{1 - V_1\} X_1 + V_1 - 1 = 0, \quad V_1 > 0, \quad (2 - V_2) Z_1 - Y_1 + V_2 X_1 - V_2 + 3 = 0, \quad V_2 > 0,
(V_3 - 1) Z_2 = 0, \quad V_3 > 0 \}, \quad \emptyset \rangle
\]
4.3. The Folding Algorithm

The Folding Algorithm: FA

Input: two clauses in normal form without variables in common

Output: the clause \( \eta: H \leftarrow e \land K \land \delta \land R \), if it is possible to fold \( \gamma \) using \( \delta \) according to Definition 3.1, and fail, otherwise.

IF there exist a substitution \( \alpha \) and a goal \( R \) which are the output of an execution of the procedure GM when clauses \( \gamma \) and \( \delta \) are given in input to GM

AND there exist a constraint \( e \) and a substitution \( \beta \) which are the output of an execution of the procedure CM when clauses \( \gamma': H \leftarrow c \land B \alpha \land R \) and \( \delta': K \alpha \leftarrow da \land B \alpha \) are given in input to CM

THEN return the clause \( \eta: H \leftarrow e \land K \alpha \beta \land R \) ELSE return fail.

The following theorem, which is proven in the Appendix, states that the folding algorithm FA terminates (Point 1), FA is sound (Point 2), and, if the constraint \( c \) is admissible, then FA is complete (Point 3).

**Theorem 4.4 (Termination, Soundness, and Completeness of FA)** Let the input of the algorithm FA be two clauses \( \gamma \) and \( \delta \) in normal form without variables in common. Then:

1. FA terminates;
2. if FA returns a clause \( \eta \), then \( \eta \) can be derived by folding \( \gamma \) using \( \delta \) according to Definition 3.1;
3. if it is possible to fold \( \gamma \) using \( \delta \) according to Definition 3.1 and the constraint occurring in \( \gamma \) is either unsatisfiable or admissible, then FA does not return fail.

Let \( C \) be the set of constraints occurring in the second component of the final triple of the above sequence of rewritings. We have that \( C \) is satisfiable and has a unique solution given by the following substitution: \( \sigma_w = solve(C) = \{ V_1/1, V_2/2, V_3/1, V_4/1, V_5/0, V_6/0, V_7/0 \} \). The substitution \( \sigma_y \) computed in the third component of the final triple of the above sequence of rewritings is \( \{ Y_1/V_2 X_1 - V_2 + 3 \} \). Hence, the substitution \( \varphi_y \), which is defined as \( \sigma_y \sigma_w \) restricted to \( \{ Y_1 \} \), is \( \{ Y_1/2X_1 + 1 \} \). Since we have that \( Vars_{rat}(s(Y_1, a, f(X_3))\sigma_y \sigma_w) - Vars(H) = \{ X_1, X_3 \} - \{ X_1, X_2, X_3 \} = \emptyset \), the substitution \( \sigma_G \) is the identity. Thus, the output of the procedure CM is the constraint \( e = X_1 < 1 \) and the substitution \( \beta = \varphi_y \sigma_G = \{ Y_1/2X_1 + 1 \} \).

Now we are ready to present our folding algorithm.
Example 4. Let us consider the clause
\[ \gamma: p(X_1, X_2, X_3) \leftarrow X_1 < 1 \land X_1 \geq Z_1 + 1 \land Z_2 > 0 \land q(Z_1, f(X_3), Z_2) \land r(X_2) \]
and the clause
\[ \delta: s(Y_1, Y_2, Y_3) \leftarrow W_1 < 0 \land Y_1 - 3 \geq 2W_1 \land W_2 > 0 \land q(W_1, Y_3, W_2) \]
of the Introduction. Let the substitution \( \alpha: \{W_1/Z_1, Y_3/f(X_3), W_2/Z_2, Y_2/a\} \) and the goal \( R: r(X_2) \) be the result of applying the procedure \( \text{GM} \) to \( \gamma \) and \( \delta \) as shown in Example 1, and let the constraint \( e: X_1 < 1 \) and the substitution \( \beta: \{Y_1/2X_1 + 1\} \) be the result of applying the procedure \( \text{CM} \) to \( \gamma \) and \( \delta \) as shown in Example 3. Then, the output of the folding algorithm \( \text{FA} \) is the clause \( \eta: p(X_1, X_2, X_3) \leftarrow e \land s(Y_1, Y_2, Y_3) \alpha \beta \land R \), that is:
\[ \eta: p(X_1, X_2, X_3) \leftarrow X_1 < 1 \land s(2X_1 + 1, a, f(X_3)) \land r(X_2). \]

5. Complexity of the Algorithm and Experimental Results

Let us first analyze the time complexity of our folding algorithm \( \text{FA} \) by assuming that: (A1) each rule application during the goal matching procedure \( \text{GM} \) and the constraint matching procedure \( \text{CM} \) takes constant time, and (A2) each computation of the functions \( \eta, \text{solve}, \) and \( \text{project} \) takes constant time. In these hypotheses our algorithm \( \text{FA} \) is in NP (w.r.t. the number of occurrences of symbols in the input clauses, including \( \land \) and \( \lnot \)). To show this fact, it is sufficient to show that both the goal matching procedure \( \text{GM} \) and the constraint matching procedure \( \text{CM} \) are in NP.

First we show that, under the assumption (A1), \( \text{GM} \) is in NP w.r.t. the number \( N \) of occurrences of symbols in the input clauses \( \gamma \) and \( \delta \). Let us consider a sequence \( s \) of applications of the rewrite rules (i)–(x) of \( \text{GM} \) starting from the initial set \( \{(B \land T)/G\} \) of bindings, where \( B \) and \( G \) are the goals occurring in the body of \( \delta \) and \( \gamma \), respectively. Each application of one of the rules (i)–(ix) reduces by at least one the number of occurrences of symbols. Rule (x) can be applied at most \( M \) times, where \( M \) is the number of variables occurring in the head of clause \( \delta \). Thus, the length of the sequence \( s \) is linear in \( N \). Finally, by a single application of a rule, any set of bindings can be rewritten into at most \( K \) different new sets of bindings, where \( K \) is the number of occurrences of literals in \( G \) (see, in particular, rule (i)). Thus, \( \text{GM} \) is in NP w.r.t. \( N \).

Now we show that, under the assumptions (A1) and (A2), also \( \text{CM} \) is in NP w.r.t. the number \( N \) of occurrences of symbols in the initial triple \( \{c \leftrightarrow e \land d', \emptyset, \emptyset\} \). We have the following property: for every maximal sequence \( s_1 \) of the form \( D \Rightarrow \cdots \Rightarrow E \) constructed by applications of the rewrite rules (i)–(v) of \( \text{CM} \), there exists a sequence \( s_2 \) of the form \( D \Rightarrow \cdots \Rightarrow E \) such that the length of \( s_1 \) is equal to the length of \( s_2 \) and in \( s_2 \) every application of rules (i), (ii), and (iii) occurs before all applications of rules (iv) and (v). Thus, for the time complexity analysis of \( \text{CM} \) we may restrict ourselves to maximal sequences where every application of rules (i), (ii), and (iii) occurs before all applications of rules (iv) and (v). Since each application of rules (i), (ii), and (iii) reduces the number of constraints occurring in the first component of the triple at hand, in a given sequence we may have at most \( N \) applications of these three rules. Moreover, each application of rules (i), (ii), and (iii) introduces at most \( m + 1 \) new variables, where \( m + 1 \leq N \). Hence, at most \( N^2 \) new variables are introduced. Since every application of rules (i), (ii), and (iii) occurs before all applications of rules (iv) and (v), and we consider maximal sequences, when rules (iv) and (v) are applied, we have that the first component of the triple at hand is of the form \( c' \rightarrow \text{true} \). Rule (iv) can be applied at most \( M \) times, where \( M \) is the number of variable occurrences in the second component of the triple at hand. Each application of rule (v) eliminates all occurrences of one variable in \( Y \), which is a subset of the variables occurring in
the input triple and, therefore, this rule can be applied at most $N$ times. Note that, for each application of rule (v), the first component does not change, because we are assuming that it is $c' \leftarrow true$. Moreover, the cardinality of the second component of the triple at hand does not change and the number of variable occurrences in each constraint in that second component is bounded by a polynomial of the cardinality of $X \cup Y \cup Z$ (which is at most $N$), because the number of variables occurring in a polynomial in normal form is at most $|X \cup Y \cup Z| \times |W|$. Thus, $M$ is bounded by a polynomial of $N$. Finally, by a single application of a rule, any triple can be rewritten into at most $K$ different new triples, where $K$ is a polynomial of $N$ and, thus, $\text{GM}$ is in NP w.r.t. $N$.

A more detailed time complexity analysis of our folding algorithm $\text{FA}$ where we do not assume that the functions $\text{nf}$, $\text{solve}$, and $\text{project}$ are computed in constant time, is as follows. (i) $\text{nf}$ takes polynomial time in the size of its argument, (ii) $\text{solve}$ takes polynomial time in the number of variables of its argument, by using Khachiyan’s method [16], and (iii) $\text{project}$ takes $O(2^v)$ time, where $v = |\text{Vars}(c) \cap \text{Vars}(B')|$ (see [19] for the complexity of variable elimination from linear constraints). Since the $\text{project}$ function is applied only once at the beginning of the procedure $\text{CM}$, we get that the computation of our algorithm $\text{FA}$ requires nondeterministic polynomial time plus $O(2^v)$ time.

Note that since matching modulo the equational theory $\text{AC}_\wedge$ is NP-complete [2], there is no folding algorithm whose asymptotic time complexity is significantly better than our algorithm $\text{FA}$.

In the following Table 1 we report some experimental results concerning our algorithm $\text{FA}$, implemented in SICStus Prolog 3.12, on a Pentium IV 3GHz. Each element of Table 1 refers to an example: Example $D 0$ of the Introduction, four Examples $D 1$–$D 4$ for which folding can be done in one way only ($\text{Number of Foldings} = 1$), and four Examples $N 1$–$N 4$ for which folding can be done in more than one way ($\text{Number of Foldings} > 1$).

The row named $\text{Number of Variables}$ indicates the number of variables in clause $\gamma$ (to be folded) plus the number of variables in clause $\delta$ (used for folding). The row named $\text{Time}$ indicates the seconds required for finding the folded clause (or the first folded clause, in examples $N 1$–$N 4$). The row named $\text{Total-Time}$ indicates the seconds required for finding all folded clauses. (Note that even when there exists one folded clause only, $\text{Total-Time}$ is greater than $\text{Time}$ because, after the folded clause has been found, $\text{FA}$ checks whether or not other folded clauses can be computed.)

In Example $D 1$ clause $\gamma$ is $p(A) \leftarrow A < 1 \land A \geq B + 1 \land q(B)$ and clause $\delta$ is $r(C) \leftarrow D < 0 \land C - 3 \geq 2D \land q(D)$. In Example $N 1$ clause $\gamma$ is $p \leftarrow A > 1 \land 3 > A \land B > 1 \land 3 > B \land q(A) \land q(B)$ and clause $\delta$ is $r \leftarrow C > 1 \land 3 > C \land D > 1 \land 3 > D \land q(C) \land q(D)$. Similar clauses (with more variables) have been used in the other examples.

Our algorithm $\text{FA}$ performs reasonably well in practice. However, when the number of variables (and, in particular, the number of variables of type $\text{rat}$) increases, its performance rapidly deteriorates.

6. Related Work and Conclusions

The elimination of existential variables from logic programs and constraint logic programs is a program transformation technique which has been proposed for improving program performance [14] and for proving program properties [13]. This technique makes use of the definition, unfolding, and folding rules [3, 7, 8, 11, 17]. In this paper we have considered constraint logic programs, where the constraints are linear inequations over the rational (or real) numbers, and
we have studied the problem of automating the application of the folding rule. Indeed, the applicability conditions of the many folding rules for transforming constraint logic programs which have been proposed in the literature [3, 7, 8, 11, 13], are specified in a declarative way and no algorithm is given to determine whether or not, given a clause \( \gamma \) to be folded by using a clause \( \delta \), one can actually perform that folding step. The problem of checking the applicability conditions of the folding rule is not trivial (see, for instance, the example presented in the Introduction).

In this paper we have considered a folding rule which is a variant of the rules proposed in the literature, and we have given an algorithm, called \textsc{fa}, for checking its applicability conditions. To the best of our knowledge, ours is the first algorithmic presentation of the folding rule. The applicability conditions of our rule consist of the usual conditions (see, for instance, [8]) together with the extra condition that, after folding, the existential variables should be eliminated. Thus, our algorithm \textsc{fa} is an important step forward for the full automation of the above mentioned program transformation techniques [13, 14] which improve program efficiency or prove program properties by eliminating existential variables.

We have proved the termination and the soundness of our folding algorithm \textsc{fa}. We have also proved that if the constraint appearing in the clause \( \gamma \) to be folded is admissible, then \textsc{fa} is complete, that is, it does not return \textsf{fail} whenever folding is possible. The class of admissible constraints is quite large. We have also implemented the folding algorithm and our experimental results show that it performs reasonably well in practice.

Our algorithm \textsc{fa} consists of two procedures: (i) the \textit{goal matching} procedure, and (ii) the \textit{constraint matching} procedure. The \textit{goal matching} procedure solves a problem similar to the problem of matching two terms modulo an associative, commutative equational theory (AC theory, for short) [2]. However, in our case we have the extra conditions that: (i.1) the matching substitution should be consistent with the types (either rational numbers or trees), and (i.2) after folding, the existential variables should be eliminated. Thus, we could not directly use the AC-matching algorithms available in the literature [6].

The \textit{constraint matching} procedure solves a generalized form of the matching problem, modulo the equational theory \( \text{LIN}_\mathbb{Q} \) of linear inequations over the rational numbers. That problem can be seen as a \textit{restricted unification} problem [4]. In [4] it is described how to obtain, under certain conditions, an algorithm for solving a restricted unification problem from an algorithm that solves the corresponding unrestricted unification problem. To the best of our knowledge, for the theory \( \text{LIN}_\mathbb{Q} \) of constraints an algorithm is provided neither for the restricted unification problem nor for the unrestricted one. Moreover, one cannot apply the so called combination methods [15]. These methods consist in constructing a matching algorithm for a given theory which is the combination of simpler theories, starting from the matching algorithms for those simpler theories. Unfortunately, as we said, we cannot use these combination methods for the theory \( \text{LIN}_\mathbb{Q} \) because some applicability conditions are not satisfied and, in particular, \( \text{LIN}_\mathbb{Q} \) is neither \textit{collapse-free} nor \textit{regular} [15].

<table>
<thead>
<tr>
<th>Example</th>
<th>( D_0 )</th>
<th>( D_1 )</th>
<th>( D_2 )</th>
<th>( D_3 )</th>
<th>( D_4 )</th>
<th>( N_1 )</th>
<th>( N_2 )</th>
<th>( N_3 )</th>
<th>( N_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Foldings</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Number of Variables</td>
<td>10</td>
<td>4</td>
<td>8</td>
<td>12</td>
<td>16</td>
<td>4</td>
<td>8</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>Time (seconds)</td>
<td>0.01</td>
<td>0.01</td>
<td>0.08</td>
<td>3.03</td>
<td>306</td>
<td>0.02</td>
<td>0.08</td>
<td>0.23</td>
<td>1.09</td>
</tr>
<tr>
<td>Total-Time (seconds)</td>
<td>0.02</td>
<td>0.02</td>
<td>0.14</td>
<td>4.89</td>
<td>431</td>
<td>0.03</td>
<td>49</td>
<td>1016</td>
<td>11025</td>
</tr>
</tbody>
</table>

Table 1: Execution times of the folding algorithm \textsc{fa} for various examples.
In the future we plan to adapt our folding algorithm FA to other constraint domains such as the linear inequations over the integers. We will also perform a more extensive experimentation of our folding algorithm using the MAP program transformation system [12].

Acknowledgements

We thank the anonymous referees for helpful suggestions. We also thank John Gallagher for comments on a draft of this paper.
A. Appendix

In this Appendix we provide the proofs of the results presented in the paper. In order to show the termination, the soundness, and the completeness of the algorithm $\text{FA}$ we first prove Theorems A.4 and A.13 that state the termination, the soundness, and the completeness of the goal matching procedure $\text{GM}$ and of the constraint matching procedure $\text{CM}$, respectively.

A.1. Termination, Soundness and Completeness of the Goal Matching Procedure

In the following, we will refer to the restriction of a substitution $\vartheta$ to a set of variables $V$, denoted by $\vartheta |_V$, as the substitution $\{X/s \in \vartheta \mid X \in V\}$.

**Definition A.1.** A GM-redex is either fail or a finite set of bindings of the form $\{t_1/u_1, \ldots, t_n/u_n, (G_1 \land T)/G_2\}$, where $n \geq 0$, for $i = 1, \ldots, n$, $t_i$ and $u_i$ are either both literals or both terms, $T$ is a variable ranging over goals, and $G_1, G_2$ are goals (possibly, the empty conjunction true).

It follows directly from the definition that if $D$ is a GM-redex and $D \Rightarrow E$, where $\Rightarrow$ is the rewriting relation defined in the procedure $\text{GM}$, then $E$ is a GM-redex.

**Definition A.2.** Let $D$ be a GM-redex, $\alpha$ a substitution, and $R$ a goal. Then we say that $D(\alpha, R)$ holds if (i) $D$ is of the form $\{t_1/u_1, \ldots, t_n/u_n, (G_1 \land T)/G_2\}$, for $n \geq 0$, (ii) for $i = 1, \ldots, n$, $t_i\alpha = u_i$, and (iii) $G_1\alpha \land R = AC G_2$.

**Lemma A.3.** Let the relation $\Rightarrow$ be defined as in the procedure $\text{GM}$ and let $D$ be a GM-redex. For every substitution $\alpha$ and goal $R$, $D(\alpha, R)$ holds iff either $D$ is of the form $\{T/R'\}$, where $T/R' \notin \alpha$, $\alpha \subseteq \alpha$, and $R' = AC R$, or there exists a GM-redex $E$ such that: (i) $D \Rightarrow E$, and (ii) $E(\alpha, R)$ holds.

**Proof.** (If part) Assume that $D$ is of the form $\alpha' \cup \{T/R'\}$, for $\alpha' \subseteq \alpha$ and $R' = AC R$, then for every binding $t/u \in \alpha'$ we have $t\alpha = u$ and, thus, $D(\alpha, R)$ holds. Now, assume that there exists $E$ such that $D \Rightarrow E$ and $E(\alpha, R)$ holds. Since $E$ is a GM-redex, by Definition A.1 it is a set of bindings of the form $\{t_1/u_1, \ldots, t_n/u_n, (G_1 \land T)/G_2\}$. We proceed by considering the rules that can be used to rewrite $D$ into $E$. Suppose that we have obtained $E$ from $D$ by applying rule (i). Then, without loss of generality, we can assume that $D$ is of the form $\{t_2/u_2, \ldots, t_n/u_n, (t_1 \land G_1 \land T)/(u_1 \land G_2)\}$, where $t_1$ and $u_1$ are both positive or both negative literals and they have the same predicate symbol and arity. By hypothesis, we have $t_1\alpha = u_1$ and $G_1\alpha \land R = AC G_2$, and, therefore, we have $t_1\alpha \land G_1\alpha \land R = AC u_1 \land G_2$. Thus, $D(\alpha, R)$ holds. Suppose that we have obtained $E$ from $D$ by applying rule (ii). Then, without loss of generality, we can assume that $D$ is of the form $\{\lnot t_1, \ldots, t_n, u_n, (G_1 \land T)/G_2\}$. Since $E(\alpha, R)$ holds, also $D(\alpha, R)$ holds. Suppose that we have obtained $E$ from $D$ by applying rule (iii). Then, without loss of generality, we can assume that $D$ is of the form $\{a(t_1, \ldots, t_k)/u_1, \ldots, u_k, t_{k+1}/u_{k+1}, \ldots, t_n/u_n, (G_1 \land T)/G_2\}$, where $k \leq n$. Since $E(\alpha, R)$ holds, also $D(\alpha, R)$ holds. Note that we cannot obtain $E$ from $D$ by applying rules (iv)–(ix) because $E(\alpha, R)$ holds and, therefore, $E$ is different from fail. Finally, suppose that we have obtained $E$ from $D$ by applying rule (x). Then, without loss of generality, we can assume that $D$ is of the form $\{t_2/u_2, \ldots, t_n/u_n, (G_1 \land T)/G_2\}$. Also in this case, since $E(\alpha, R)$ holds, then $D(\alpha, R)$ holds.

(Only If part) Assume that $D(\alpha, R)$ holds. $D$ is a GM-redex and, thus, by definition, it is a set of bindings of the form $\{t_1/u_1, \ldots, t_n/u_n, (G_1 \land T)/G_2\}$. $D \Rightarrow E$, then $D$ is of the form $\{t_1/u_1, \ldots, t_n/u_n, (G_1 \land T)/G_2\}$.
\( \alpha' \cup \{T/R'\} \), where \( \alpha' \subseteq \alpha \) and \( R' =_{AC} R \). Let us assume that there is no \( E \) such that \( D \implies E \) and let us consider each of the rules (i)–(x). Since \( D \) cannot be rewritten we have the following properties. We cannot apply rule (i), thus there is no literal \( L \) which occurs as a conjunct in both \( G_1 \alpha \) and \( G_2 \). Since \( D(\alpha, R) \) holds, every literal occurring as a conjunct in \( G_1 \alpha \) also occurs as a conjunct in \( G_2 \) and, hence, \( G_1 \alpha \) is the empty conjunction. We cannot apply rule (ii) and, since \( D(\alpha, R) \) holds, for all \( i = 1, \ldots, n \) we have that \( t_i \) and \( u_i \) are both atoms or both terms. We cannot apply rule (iii) and, thus, there is no binding in \( D \) of the form \( a(r_1, \ldots, r_k)/a(s_1, \ldots, s_k) \), for some predicate or function symbol \( a \) and some terms \( r_1, \ldots, r_k \) and \( s_1, \ldots, s_k \). We cannot apply rule (iv) and thus, there is no binding in \( D \) of the form \( a(r_1, \ldots, r_k)/b(s_1, \ldots, s_m) \), for a syntactically different from \( b \) and some terms \( r_1, \ldots, r_k \) and \( s_1, \ldots, s_m \). Finally, we cannot apply rule (v) and thus, since \( D(\alpha, R) \) holds, there is no binding in \( D \) of the form \( t/X \), where \( t \) is a term and \( X \) is a variable. As a consequence of the non-applicability of rules (i)–(v) we have that \( D \) is a GM-redex of the form \( \{X_1/u_1, \ldots, X_n/u_n, T/G_2\} \), where \( X_1, \ldots, X_n \) are variables and \( u_1, \ldots, u_n \) are terms. Also, we cannot apply rule (vi), which entails that \( X_1, \ldots, X_n \) are distinct variables. Therefore, \( \{X_1/u_1, \ldots, X_n/u_n\} \) is a substitution. Since by hypothesis \( D(\alpha, R) \) holds, for \( i = 1, \ldots, n \), we have that \( X_i \alpha = u_i \) and \( R =_{AC} G_2 \). That is, \( D \) is of the form \( \alpha' \cup \{T/R'\} \), where \( \alpha' \subseteq \alpha \) and \( R' =_{AC} R \).

Now we prove that if \( D(\alpha, R) \) holds and \( D \) is not of the form \( \alpha' \cup \{T/R'\} \), where \( \alpha' \subseteq \alpha \) and \( R' =_{AC} R \), then there exists a GM-redex \( E \) such that \( D \implies E \) and \( E(\alpha, R) \) holds. Let us assume that \( D \) is not of the form \( \alpha' \cup \{T/R'\} \), for some \( \alpha' \subseteq \alpha \) and \( R' =_{AC} R \). Since \( D \) is in general of the form \( \{t_1/u_1, \ldots, t_n/u_n, (G_1 \land T)/G_2\} \), we have the following cases: either (a) \( \{t_1/u_1, \ldots, t_n/u_n\} \) is not a substitution, or (b) it is a substitution and \( \{t_1/u_1, \ldots, t_n/u_n\} \not\subseteq \alpha \), or (c) \( G_1 \) is not the empty conjunction \( true \), and \( R \not=_{AC} G_2 \). By hypothesis, \( D(\alpha, R) \) holds and, thus, for \( i = 1, \ldots, n \), \( t_i \alpha = u_i \) and \( G_1 \alpha \land R =_{AC} G_2 \). As a consequence, case (b) is impossible, because \( t_1, \ldots, t_n \) are distinct variables and if there exists \( i \in \{1, \ldots, n\} \) such that \( t_i/u_i \notin \alpha \) then \( t_i \alpha \neq u_i \), which contradicts the hypothesis. Also case (d) is impossible because, by hypothesis, \( G_1 \alpha \land R =_{AC} G_2 \). We now want to show that the remaining cases (a) and (c) entail that there exists a GM-redex \( E \) such that \( D \implies E \) and \( E(\alpha, R) \) holds. In case (a) we have that either \( t_1, \ldots, t_n \) are non-distinct variables, which is impossible (because it would imply that two bindings in \( D \) are identical whereas \( D \) is a set), or there exists \( i \in \{1, \ldots, n\} \) such that \( t_i \) is not a variable. Without loss of generality, we can assume that \( i = 1 \). Then, \( t_1 \) is either a literal of the form \( \neg A_1 \) or a term (or an atom) of the form \( a(r_1, \ldots, r_k) \). Hence, \( u_1 \) cannot be a variable because \( t_1 \alpha = u_1 \). Thus, \( u_1 \) must be a literal of the form \( \neg A_2 \) or a term (or atom) of the form \( a(s_1, \ldots, s_k) \), respectively. Let us first consider the case where both \( u_1 \) and \( t_1 \) are literals. Then, there exists a GM-redex \( E \), which can be obtained by applying rule (ii), such that \( D \implies E \). In particular, \( E \) is of the form \( \{A_1/A_2, t_2/u_2, \ldots, t_n/u_n, G_1 \land T/G_2\} \). Since \( D(\alpha, R) \) holds, also \( E(\alpha, R) \) holds. If we consider the case where both \( t_1 \) and \( u_1 \) are terms (or atoms), there exists a GM-redex \( E \), which can be obtained by applying rule (iii), such that \( D \implies E \). The GM-redex \( E \) is of the form \( \{r_1/s_1, \ldots, r_k/s_k, t_2/u_2, \ldots, t_n/u_n, G_1 \land T/G_2\} \). Again, since \( D(\alpha, R) \) holds, also \( E(\alpha, R) \) holds. Let us now consider case (c), where \( G_1 \) is not the empty conjunction \( true \). Since \( G_1 \alpha \land R =_{AC} G_2 \), we have that \( G_1 \alpha \) is of the form \( L_1 \alpha \land G_1' \) and \( G_2 \) is of the form \( G_2' \land L_2 \land G_2'' \). Thus, \( L_1 \) and \( L_2 \) are both positive or both negative literals and they have the same predicate symbol and arity. As a consequence, there exists a GM-redex \( E \), that can be obtained by applying rule (i), such that \( D \implies E \). In particular, \( E \) is of the form \( \{L_1/L_2, t_1/u_1, \ldots, t_n/u_n, G_1' \land T/G_2' \land G_2'' \} \) and \( E(\alpha, R) \) holds. \( \blacksquare \)

**Theorem A.4 (Termination, Soundness, and Completeness of GM)** Let \( \gamma : H \leftarrow c \land G \)
and $\delta : K \leftarrow d \land B$ be two clauses in normal form and without variables in common. Let $\gamma$ and $\delta$ be the input of the goal matching procedure GM. The following properties hold:

(a) GM terminates, that is: (1) given a GM-redex $D_0$ and the rewriting relation $\Rightarrow$ defined in the procedure GM, every sequence $D_0 \Rightarrow D_1 \Rightarrow \ldots$ is finite and (2) for every GM-redex $D$, there are finitely many GM-redexes $E_1, \ldots, E_n$ such that, for $i = 1, \ldots, n$, $D \Rightarrow E_i$;

(b) For every substitution $\alpha$ and goal $R$, if $\alpha$ and $R$ are the output of GM, then: (1) $G = AC B\alpha \land R$, (2) for every variable $X$ in $Evars(\delta)$, the following conditions hold: (2.1) $X\alpha$ is a variable not occurring in $\{H, R\}$, and (2.2) for every variable $Y$ occurring in $d \land B$ and different from $X$, $X\alpha$ does not occur in the term $Y\alpha$, and (3) $\forallvars_{\text{tree}}(\alpha\kappa) \subseteq \forallvars(H)$, and (4) the clauses $\gamma' : H \leftarrow c \land B\alpha \land G$ and $\delta' : K\alpha \leftarrow \alpha \land B\alpha$ are in normal form;

(c) For every substitution $\alpha$ and goal $R$, if (1) $G = AC B\alpha \land R$, (2) for every variable $X$ in $Evars(\delta)$, the following conditions hold: (2.1) $X\alpha$ is a variable not occurring in $\{H, R\}$, and (2.2) for every variable $Y$ occurring in $d \land B$ and different from $X$, $X\alpha$ does not occur in the term $Y\alpha$, and (3) $\forallvars_{\text{tree}}(\alpha\kappa) \subseteq \forallvars(H)$, then there exist a substitution $\alpha'$ and a goal $R'$ such that: (4) $\alpha'$ and $R'$ are the output of GM, (5) $\alpha'\big|_V = \alpha\big|_V$, where $V$ is the set $\forallvars(B) \cup \forallvars_{\text{tree}}(K)$ of variables, and (6) $R' = AC R$.

Proof. (a) We first prove that, given a GM-redex $D_0$, every sequence $D_0 \Rightarrow D_1 \Rightarrow \ldots$ is finite. Let us introduce some notions on well-founded orders on multisets, which will be necessary below. A multiset $S$ is represented as $\{x_1, \ldots, x_n\}$, where $x_1, \ldots, x_n$ are the elements (with, possibly, multiple occurrences) of $S$. In this proof, we will use $\forallmultiset$ to denote multiset union, $\emptyset$ to denote the empty multiset, and $S(x)$ to denote the number of occurrences of an element $x$ in a multiset $S$. Let us consider the well-founded set $(\forallmultiset(\forallmultiset(\mathbb{N})), \gg)$, where $\forallmultiset(\forallmultiset(\mathbb{N}))$ is the set of all finite multisets of elements of $\mathbb{N}$, and, for all $S_1, S_2 \in \forallmultiset(\forallmultiset(\mathbb{N}))$, $S_1 \gg S_2$ iff $S_1 \neq S_2$ and, for every $x \in \mathbb{N}$, if $S_2(x) > S_1(x)$ then there exists $y \in \mathbb{N}$ such that $y > x$ and $S_1(y) > S_2(y)$. For every GM-redex $D$ let us define $kvars(D)$ to be the cardinality of the following set $\{V \in \forallvars_{\text{tree}}(K) - \forallvars(B) \mid \neg \exists t \in D\}$. In the following, given a term or goal $a$, we will denote by $|a|$ the number of symbols in $a$. (In particular, $||T|| = 1$, if $T$ is a variable ranging over goals, $||V|| = 1$, if $V$ is a variable of type rat or tree, and $||\text{true}|| = 1$.) Let us now introduce the termination function $\xi$, that maps GM-redexes to elements of $\forallmultiset(\forallmultiset(\mathbb{N}))$. Let $D$ be a GM-redex, then $\xi(D) = \emptyset$, if $D$ is fail, and $\xi(D) = \{||t_1|| + kvars(D) \mid t_1/t_2 \in D\}$ otherwise. Note that, by definition of GM-redex, if $D$ is a GM-redex different from fail then the multiset $\xi(D)$ is not the empty multiset. Now we want to show that if $\xi(D) = \emptyset$ then $\xi(D) \gg \xi(E)$. Let us consider the case where $D \Rightarrow E$ by using rule (i). Let $D$ be the GM-redex $\{L_1 \land B_1 \land T\} / (G_1 \land G_2) \cup Bnds$ and $E$ the GM-redex $\{L_1 / L_2, (B_1 \land T) / (G_1 \land G_2)\} \cup Bnds$ where $B_1$, $G_1$, and $G_2$ are goals, possibly the empty conjunction true, and $L_1$, $L_2$ are literals. We have that $\xi(D) = \{||L_1|| + kvars(Bnds)\} \cup \forallmultiset(Bnds)$ and $\xi(E) = \{||L_1|| + kvars(Bnds), ||B_1 \land T|| + kvars(Bnds)\} \cup \forallmultiset(Bnds)$. Since $||L_1 \land B_1 \land T|| > ||L_1||$ and $||L_1 \land B_1 \land T|| > ||B_1 \land T||$, we get that $\xi(D) \gg \xi(E)$. Similarly we can show that $\xi(D) \gg \xi(E)$ in the case where $D \Rightarrow E$ by using rule (ii) or rule (iii). Since $\xi(\text{fail}) = \emptyset$, if $D \Rightarrow E$ by using one among rules (iv)–(ix) then $\xi(D) \gg \xi(E)$, because $\xi(D)$ is not the empty multiset. Let us now consider the case where $D \Rightarrow E$ by using rule (x). Then, $E$ is the GM-redex $\{X/s\} \cup D$, for some variable $X$ in $\forallvars_{\text{tree}}(K) - \forallvars(B)$ such that there is no binding $X/t \in D$. Let $\xi(D)$ be the multiset $\{m_1 + kvars(D), \ldots, m_k + kvars(D)\}$, where $k \geq 1$ because, by hypothesis, $\xi(D)$ is not the empty multiset, and, by definition, for $i = 1, \ldots, k$, $m_k \geq 1$. As a consequence, $\xi(E)$ is the multiset $\{m_1 + (kvars(D) - 1), \ldots, m_k + (kvars(D) - 1), kvars(D)\}$, where $\xi(\{X/s\}) = kvars(D)$, and, thus, $\xi(D) \gg \xi(E)$. Since $(\forallmultiset(\forallmultiset(\mathbb{N})), \gg)$ is a well founded set,
we have that, given a GM-redex $D_0$, every sequence $D_0 \Longrightarrow D_1 \Longrightarrow \ldots$ is finite.

Now we prove that, for every GM-redex $D$, there are finitely many GM-redexes $E_1, \ldots, E_n$ such that, for $i = 1, \ldots, n$, $D \Longrightarrow E_i$. Let $D$ be of the form \{\frac{t_1}{u_1}, \ldots, \frac{t_n}{u_n}, (G_1 \land T)/G_2\}. Since $G_2$ is a finite conjunction of literals, there are finitely many GM-redexes $E_1, \ldots, E_n$ such that, for $i = 1, \ldots, n$, $D \Longrightarrow E_i$, by using rule (i). In the case where $D$ is rewritten by using one of rules (ii)–(ix), we can use arguments similar to the ones for the case of rule (i) because, by definition of GM-redex, $D$ is a finite set. Finally, since at rule (x) we can choose an arbitrary term $s$ of type $\text{tree}$ such that $\text{Vars}(s) \subseteq \text{Vars}(H)$, we can assume that the term $s$ is a constant of type $\text{tree}$ fixed in advance, that is, for any input $\gamma$ and $\delta$ of $\text{GM}$, therefore, since the set $\text{Vars}_{\text{tree}}(K) - \text{Vars}(B)$ is finite, there are finitely many GM-redexes $E_1, \ldots, E_n$ such that, for $i = 1, \ldots, n$, $D \Longrightarrow E_i$. Thus, we get the thesis.

(b) Assume that $\gamma$ and $\delta$ are the input of $\text{GM}$. We want to show that if $\alpha$ and $R$ are the output of $\text{GM}$ then Conditions (b.1)–(b.4) hold.

(b.1) Assume that $\alpha$ and $R$ are the output of $\text{GM}$. Thus, by Condition (c1) of $\text{GM}$, there exists $n \geq 0$ such that $\{B \land T/G\} \Longrightarrow^n \alpha \cup \{T/R\}$ and, by Condition (c2) of $\text{GM}$, the set $\alpha \cup \{T/R\}$ of bindings cannot be further rewritten. By definition, $\alpha \cup \{T/R\}(\alpha, R)$ holds. By induction on $n$ and by (the If part of) Lemma A.3, we have that $\{B \land T/G\}(\alpha, R)$ holds and, thus, $G =_{AC} B\alpha \land R$. Therefore, Condition (b.1) holds.

(b.2) Since the GM-redex $\alpha \cup \{T/R\}$ cannot be rewritten to $\text{fail}$, the conditions for the application of rules (vii) and (viii) are not satisfied and thus, (recalling that rule (x) does not affect the variables occurring in $B$) $\alpha$ satisfies Condition (b.2).

(b.3) Let us consider $X \in \text{Vars}_{\text{tree}}(K)$ and assume that $X \in \text{Vars}(B)$. Then, by construction, there exists a term $t$ such that $X/t \in \alpha$. Since the conditions for the application of rule (ix) are not satisfied by $\alpha \cup \{T/R\}$, we have that $\text{Vars}(t) \subseteq \text{Vars}(H)$. Now, assume that $X \notin \text{Vars}(B)$. Then, by Condition (c2) of $\text{GM}$ rule (x) cannot be applied to the GM-redex $\alpha \cup \{T/R\}$, we have that $\text{Vars}(X\alpha) \subseteq \text{Vars}(H)$. It follows that if $X \in \text{Vars}_{\text{tree}}(K)$ then $\text{Vars}_{\text{tree}}(X\alpha) \subseteq \text{Vars}(H)$. Since no term of type $\text{rat}$ can have a subterm of type $\text{tree}$, we get Condition (b.3).

(b.4) Let us first show that the clause $\gamma': H \leftarrow c \land B\alpha \land R$ is in normal form. Indeed, every term of type $\text{rat}$ in $B\alpha \land R$ is a variable, because $R$ is a subgoal of $G$ and $\gamma$ is in normal form, every term of type $\text{rat}$ in $B$ is a variable, because $\delta$ is in normal form, and, by (b.2), if $X \in \text{Vars}(B)$ then $X\alpha$ is a variable. Also, every variable of type $\text{rat}$ in $B\alpha \land R$ occurs at most once in $B\alpha \land R$, because, by (b.2), if $X \in \text{Vars}_{\text{rat}}(B)$ then $X\alpha$ does not occur in $R$ and for all $Y \in \text{Vars}_{\text{rat}}(B)$ different from $X$ we have $\text{Vars}(X\alpha) \cap \text{Vars}(Y\alpha) = \emptyset$. By hypothesis, $\text{Vars}(R) \cap \text{Vars}(H) = \emptyset$ and, by (b.2), if $X \in \text{Vars}(B)$ then $X\alpha$ does not occur in $H$. Thus, $\text{Vars}_{\text{rat}}(H) \cap \text{Vars}_{\text{rat}}(B\alpha \land R) = \emptyset$. Finally, since by (b.1) $G =_{AC} B\alpha \land R$, we have that $\text{Vars}(B\alpha \land R) = \text{Vars}(G)$ and that $c$ has no constraint-local variables in $\gamma'$. Let us now show that also clause $\delta': K\alpha \leftarrow d\alpha \land Ba$ is in normal form. Indeed, by using arguments similar to those given above, we can show that every term of type $\text{rat}$ in $B\alpha$ is a variable and occurs at most once in $B\alpha$. Since, by construction and by the hypothesis that $\delta$ is in normal form, $X\alpha \neq X$ if $X \in \text{Vars}(B) \cup \text{Vars}_{\text{tree}}(K)$, we have that if $Y \in \text{Vars}_{\text{rat}}(K)$ then $Y\alpha = Y$. By construction, $\text{Vars}_{\text{rat}}(B\alpha) \subseteq \text{Vars}(\gamma)$). Therefore, by the hypothesis that $\gamma$ and $\delta$ have no variables in common, we have that $\text{Vars}_{\text{rat}}(K\alpha) \cap \text{Vars}_{\text{rat}}(B\alpha) = \emptyset$. Finally, since $\text{Vars}(d) \subseteq \text{Vars}_{\text{rat}}(B) \cup \text{Vars}_{\text{rat}}(K)$, we have that $\delta'$ has no constraint-local variables. Therefore, Condition (b.4) holds.

(c) Assume that $\gamma$ and $\delta$ are the input of $\text{GM}$ and there exist a substitution $\alpha$ and a goal $R$ such that Conditions (c.1)–(c.3) hold. We want to show that Conditions (c.4)–(c.6) hold. We have that $\{B \land T/G\}$ is a GM-redex and, by (c.1), $\{B \land T/G\}(\alpha, R)$ holds. By (the Only If part
of) Lemma A.3, we can construct a maximal sequence \( S \) of GM-redexes \( D_1 \Rightarrow D_2 \Rightarrow \ldots \) such that \( D_1 = \{ B \land T(G) \} \) and, if \( D_i \) occurs in the sequence \( S \) then either \( D_i \) is of the form \( \alpha' \cup \{ T/R' \} \), where \( T/R' \notin \alpha' \), \( \alpha' \subseteq \alpha \), and \( R' = AC \) \( R \), or \( D_i \Rightarrow D_{i+1} \) and \( D_{i+1}(\alpha, R) \) holds.  

Since, by Condition (a) of this theorem we have proved that GM terminates, \( S \) is finite, that is, there exists \( n \geq 0 \) such that \( S = D_1 \Rightarrow D_2 \Rightarrow \ldots \Rightarrow D_n \), where \( D_n \) is of the form \( \alpha' \cup \{ T/R' \} \), where \( T/R' \notin \alpha' \), \( \alpha' \subseteq \alpha \), and \( R' = AC \) \( R \), and \( D_n \) cannot be rewritten. As a consequence, (c.4) and (c.6) hold. By Condition (b.1) of this theorem, we have also that \( G = AC \) \( Bo(\alpha \land R') \) and since, by hypothesis, \( G = AC Bo(\alpha \land R) \), we have that \( Bo(\alpha \land R') \) holds. \( \alpha \land R' \) is of the form \( \alpha \land \beta \) of GM-redexes and, thus, we can assume \( \alpha' \subseteq Vars(H) \) and, thus, we can assume \( \alpha' \subseteq \{ X \} \). Hence, Condition (c.5) holds. 

### A.2. Termination, Soundness and Completeness of the Constraint Matching Procedure

First we prove Lemma 4.1, which has been presented in Section 4.2. The following lemma will be used in the proof of Lemma 4.1.

**Lemma A.5.** Let \( \gamma_1: H \leftarrow c \land B \) and \( \gamma_2: H \leftarrow d \land B \) be clauses in normal form. Then \( \gamma_1 \equiv \gamma_2 \) iff \( Q \models \forall (c \leftarrow d) \).

**Proof.** Since \( \gamma_1 \) and \( \gamma_2 \) are in normal form, it follows directly from the definitions that \( \gamma_1 \equiv \gamma_2 \) iff there exists a variable renaming \( \rho \) such that: (1) \( H = H \rho \), (2) \( B = B \rho \), and (3) \( Q \models \forall (c \leftarrow d \rho) \). Since there are no constraint-local variables in \( \gamma_2 \), we have that \( Vars(d) \subseteq Vars(\{ H, B \}) \) and, thus, \( d \rho = d \). 

**Proof of Lemma 4.1.**

By hypothesis, \( \gamma' \) and \( \delta' \) are in normal form. Now we show that also \( \gamma'': H \leftarrow e \land \delta' \land B' \land R \) is in normal form. The validity of Conditions (i)–(iii) of the definition of normal form (see Section 2) for \( \gamma'' \) directly follows from the validity of these conditions for \( \gamma' \). For \( \gamma'' \), Condition (iv) of the definition of normal form (that is, \( \gamma'' \) has no constraint-local variables) can be written as: \( Vars(e \land \delta' \land B' \land R) \subseteq Vars(\{ H, B', B', \}) \), and it can be proved as follows. Since \( \delta' \) is in normal form, \( Vars(d') \subseteq Vars(\{ K', B' \}) \). Therefore, by hypotheses (2) and (3) of this lemma we have that \( Vars(d') \subseteq Vars(\{ H, B', B' \}) \) and, by hypothesis (4), we get that Condition (iv) holds for \( \gamma'' \). Thus, by applying Lemma A.5 we have that Conditions (1)–(4) hold iff \( Q \models \forall (c \leftarrow (e \land \delta')) \) and Conditions (2)–(4) hold.

Now, assume that \( Q \models \forall (c \leftarrow (\bar{e} \land \bar{d'})) \) and Conditions (2) and (3) hold. Since \( Vars(\bar{e}) \subseteq Vars(\{ H, R \}) \), we get that there exists a constraint \( e \) such that \( Q \models \forall (c \leftarrow (e \land \bar{d'})) \) and Conditions (2)–(4) hold.

Finally, assume that \( Q \models \forall (c \leftarrow (e \land \bar{d'})) \) and Conditions (2), (3), and (4) hold. Thus, (i) \( Q \models \forall (c \leftarrow c) \), (ii) \( Q \models \forall (c \leftarrow \bar{d'}) \), and (iii) \( Q \models \forall (e \land \bar{d'} \land c) \). In order to show that \( Q \models \forall (c \leftarrow (\bar{e} \land \bar{d'})) \), it suffices to show: (iv) \( Q \models \forall (c \leftarrow \bar{e}) \) and (v) \( Q \models \forall (\bar{e} \land \bar{d'}) \). Since \( \bar{e} = \text{project}(c, X) \), where \( X = Vars(c) \setminus Vars_{\text{rat}}(B') \), by the definition of the project function we have that \( Q \models \forall (\bar{e} \leftarrow \exists Z c) \), where \( Z = Vars(c) \cap Vars_{\text{rat}}(B') \). Hence, (iv) holds. By (4), \( Vars(c) \subseteq Vars(\{ H, R \}) \) and, since \( \gamma' \) is in normal form, \( Vars(e) \cap Z = \emptyset \). Thus, by (i), \( Q \models \forall (\exists Z c \leftarrow c) \) holds and, by (iii), we get (v).

Now we prove Lemma 4.2, which has been presented in Section 4.2.

The closure of a constraint \( c \), denoted \( \text{closure}(c) \), is defined as follows: let \( c \) be a constraint of the form \( p_1 \rho_1 0 \land \ldots \land p_m \rho_m 0 \), where, for \( i = 1, \ldots, n \), \( \rho_i \in \{ \geq, > \} \), then \( \text{closure}(c) \) is the
constraint \( p_1 \geq 0 \land \ldots \land p_m \geq 0 \). In order to prove Lemma 4.2 we now show the following result which characterizes the equivalence of two conjunctions of strict inequations.

**Lemma A.6.** Let \( a \) and \( b \) be two satisfiable, non-redundant constraints of the form \( a_1 \land \ldots \land a_m \) and \( b_1 \land \ldots \land b_n \), respectively, where each constraint \( a_i \in \{a_1, \ldots, a_m, b_1, \ldots, b_n\} \) is of the form \( p_i > 0 \), for some linear polynomial \( p_i \). Then \( Q \models \forall(a \leftrightarrow b) \) holds iff \( m = n \) and there exists a bijection \( \mu: \{1, \ldots, m\} \to \{1, \ldots, n\} \) such that for \( i = 1, \ldots, m \), \( Q \models \forall(a_i \leftrightarrow b_{\mu(i)}) \) holds.

**Proof.** (Sketch) (If part) Trivial. (Only If part) We will identify constraints with polytopes in \( \mathbb{Q}^k \), where \( k \) is the number of distinct variables occurring in \( a \) or \( b \). Equivalence of constraints will be identified with equality of polytopes. Let us consider the atomic constraint \( a_i \), for some \( i \in \{1, \ldots, m\} \). By hypothesis, \( a_i \) is of the form \( p_i > 0 \), for some linear polynomial \( p_i \). We have that \( a \subseteq a_i \) and, since \( Q \models \forall(a \leftrightarrow b) \), we also have that \( b \subseteq a_i \). Now we have three cases: (i) \( a_i \) is external to \( b \), that is, no vertex of \( \text{closure}(b) \) satisfies the equation \( p_i = 0 \). For \( j = 1, \ldots, n \), there exists \( a_i \equiv b_j \), that is, \( a_i \) is tangent to \( b_j \) and for some \( j \in \{1, \ldots, n\} \), \( a_i = b_j \), that is, \( a_i \) is tangent to \( b_j \) and for some \( j \in \{1, \ldots, n\} \), \( a_i = b_j \), that is, \( h \) vertices of \( \text{closure}(b) \), with \( 1 \leq h < k \), satisfy the equation \( p_i = 0 \). Thus, \( a_i \equiv b_j \).

Similarly, it can be shown that, for all \( j \in \{1, \ldots, n\} \), there exists \( i \in \{1, \ldots, m\} \) such that \( Q \models \forall(a_i \equiv b_j) \). We have that \( j = \mu(i) \), because \( Q \models \forall(b_j \equiv b_{\mu(i)}) \) and \( b \) is non-redundant. Thus, \( m = n \) and \( \mu \) is a bijection from \( \{1, \ldots, m\} \) onto itself such that, for \( i = 1, \ldots, m \), \( Q \models \forall(a_i \equiv b_{\mu(i)}) \).

**Lemma A.7.** If \( a \) is an admissible constraint, \( b \) is a non-redundant constraint, and \( Q \models \forall(a \rightarrow b) \), then \( \text{interior}(b) \) is non-redundant.

**Proof.** (Sketch) As in Lemma A.6, we will identify constraints with polytopes in \( \mathbb{Q}^k \), where \( k \) is the number of distinct variables occurring in \( a \) or \( b \). Equivalence of constraints will be identified with equality of polytopes. Assume that \( b \) is a constraint of the form \( b_1 \land \ldots \land b_n \), where, for \( i = 1, \ldots, n \), \( b_i \) is an atomic constraint and let \( \text{interior}(b) \) be the constraint \( \overline{b}_1 \land \ldots \land \overline{b}_n \), where, for \( i = 1, \ldots, n \), \( \overline{b}_i \) is \( \text{interior}(b_i) \). Assume, by contradiction, that \( \text{interior}(b) \) is redundant, that is, there exists \( \overline{b}_i \in \{\overline{b}_1, \ldots, \overline{b}_n\} \) such that \( Q \models \forall(b_i' \rightarrow \overline{b}_i) \), where \( b_i' \) is the constraint \( \overline{b}_1 \land \ldots \land \overline{b}_{i-1} \land \overline{b}_{i+1} \land \ldots \land \overline{b}_n \). Let \( \overline{b}_i \) be of the form \( p_i > 0 \). Since \( b' \subseteq \overline{b}_i \), we have three cases: (i) \( \overline{b}_i \) is external to \( b' \), that is, no vertex of \( \text{closure}(b') \) satisfies the equation \( p_i = 0 \), (ii) \( \overline{b}_i \) is tangent to \( b' \) and, for \( j = 1, \ldots, n \), \( j \neq i \), \( \overline{b}_i \neq \overline{b}_j \), that is, \( h \) vertices of \( \text{closure}(b') \), with \( 1 \leq h < k \), satisfy the equation \( p_i = 0 \), and (iii) \( \overline{b}_i \) is tangent to \( b' \) and for some \( j \in \{1, \ldots, n\} \) with \( j \neq i \), \( \overline{b}_i = \overline{b}_j \), that is, \( h \) vertices of \( \text{closure}(b') \), with \( h \geq k \), satisfy the equation \( p_i = 0 \). Case (i) entails that \( Q \models \forall(b_1 \land \ldots \land b_{i-1} \land b_{i+1} \land \ldots \land b_n \rightarrow \overline{b}_i) \), which contradicts the hypothesis that \( b \) is non-redundant. Now let us consider Case (ii). We first define a set of points that belong to the intersection between the polytope \( b_1 \land \ldots \land b_{i-1} \land b_{i+1} \land \ldots \land b_n \) and the hyperplane \( p_i = 0 \) (these points can be seen as the tangency points of the hyperplane \( p_i = 0 \) with the given polytope).

Now, we distinguish between the following two Cases (ii.A) and (ii.B). In Case (ii.A), we have that \( T \) is the empty set, that is, the polytope \( b_1 \land \ldots \land b_{i-1} \land b_{i+1} \land \ldots \land b_n \) and the hyperplane
$p_i = 0$ have an empty intersection. Hence, we have that $Q \models \forall (b_1 \land \ldots \land b_{i-1} \land b_{i+1} \land \ldots \land b_n \rightarrow b_i)$, which contradicts the hypothesis that $b$ is non-redundant. In Case (ii), we have that $T$ is not the empty set. If $b_i$ is of the form $p_i \geq 0$ then $Q \models \forall (b_1 \land \ldots \land b_{i-1} \land b_{i+1} \land \ldots \land b_n \rightarrow b_i)$, which contradicts the hypotheses. Otherwise, if $b_i$ is of the form $p_i > 0$, since, by hypothesis, the atomic constraint $b_i$ is non-redundant in $b$ and $Q \models \forall (a \leftrightarrow b)$, we have $T \cap a = \emptyset$. Hence, there exists a constraint $a_j$ (not necessarily equivalent to $b_j$) in $a$ such that $a_j$ of the form $q_j > 0$ and $T \subseteq (q_j = 0)$. Thus, $a_j$ is non-redundant in $a$, while $\text{interior}(a_j)$ is redundant in $\text{interior}(a)$ and this is contradicts the hypothesis that $a$ is admissible. Finally, we consider Case (iii). There exists $b_j$ such that $b_j = \overline{b}_j$. Assume that $b_j$ is of the form $p_j > 0$, for some linear polynomial $p_j$. As a consequence, $Q \models \forall (b_1 \land \ldots \land b_{i-1} \land b_{i+1} \land \ldots \land b_n \rightarrow b_j)$. Now assume that $b_j$ is of the form $p_j \geq 0$. Then, we have $Q \models \forall (b_1 \land \ldots \land b_{i-1} \land b_{j+1} \land \ldots \land b_n \rightarrow b_j)$. Both cases contradict the hypothesis that $b$ is non-redundant. Thus, in each of Cases (i), (ii), and (iii), the assumption that the constraint $\text{interior}(b)$ is redundant leads to a contradiction and we conclude that the constraint $\text{interior}(b)$ is non-redundant.

**Proof of Lemma 4.2.**

(If part) Trivial. (Only If part) Without loss of generality, we assume that there exists a constraint $b_1 \land \ldots \land b_k$, with $k \leq n$, that is non-redundant and such that $Q \models \forall (b_1 \land \ldots \land b_k)$. Since by transitivity $Q \models \forall (a \leftrightarrow b_1 \land \ldots \land b_k)$, we have that $Q \models \forall (\text{interior}(a) \leftrightarrow \text{interior}(b_1) \land \ldots \land \text{interior}(b_k))$ (because if two, not necessarily closed, polytopes are equal then also the corresponding open polytopes obtained by removing the facets, are equal). By Lemma A.7 we have that the constraint $\text{interior}(b_1) \land \ldots \land \text{interior}(b_k)$ is non-redundant. Finally, by Lemma A.6, $m = k$ and there exists a bijection $\mu : \{1, \ldots, m\} \rightarrow \{1, \ldots, k\}$ such that, for $i = 1, \ldots, m$, $Q \models \forall (\text{interior}(a_i) \leftrightarrow \text{interior}(b_{\mu(i)}))$. For every $a_i \in \{a_1, \ldots, a_m\}$ we have the following cases: (i) $a_i$ is of the form $t > 0$ and $b_{\mu(i)}$ is of the form $t \geq 0$, (ii) $a_i$ is of the form $t \geq 0$ and $b_{\mu(i)}$ is of the form $t > 0$, (iii) $a_i$ is of the form $t > 0$ and $b_{\mu(i)}$ is of the form $t > 0$, (iv) $a_i$ is of the form $t \geq 0$ and $b_{\mu(i)}$ is of the form $t \geq 0$. Case (i) leads to a contradiction because it entails $Q \models (\neg (a_1 \land \ldots \land a_m) \land b_1 \land \ldots \land b_k)$. Similarly, Case (ii) leads to a contradiction. The remaining Cases (iii) and (iv) imply that $Q \models \forall (a_i \leftrightarrow b_{\mu(i)})$. By the assumptions of non-redundancy of $a$ and $b_1 \land \ldots \land b_k$ the function $\mu$ is an injection from $\{1, \ldots, m\}$ to $\{1, \ldots, n\}$, and $Q \models \forall (b_1 \land \ldots \land b_k \rightarrow b_j)$, for all $j \in \{1, \ldots, n\}$ such that $j \notin \{\mu(i) \mid 1 \leq i \leq m\}$. Thus, we get the thesis. □

Next we prove Theorem 4.3. We need the following lemma.

**Lemma A.8.** Let $k_1, \ldots, k_{m+1} \in Q$ and suppose that

$$Q \models \forall X_1 \ldots X_m (X_1 \rho_1 0 \land \ldots \land X_m \rho_m 0 \rightarrow (k_1 X_1 + \ldots + k_m X_m + k_{m+1}) \rho_{m+1} 0)$$

where $\rho_1, \ldots, \rho_{m+1} \in \{>, \geq\}$. We have that $k_1 \geq 0, \ldots, k_{m+1} \geq 0$, and if $\rho_{m+1} >$ then

$$\left(\sum_{i \in I} k_i \right) > 0, \text{ where } I = \{i \mid 1 \leq i \leq m+1, \rho_i \text{ is }>\}. \quad (i)$$

**Proof.** We proceed by cases.

(Case 1) Let $\rho_{m+1}$ be $\geq$. By hypothesis we have that for all $X_1, \ldots, X_m \in Q$, if $X_1 \rho_1 0, \ldots, X_m \rho_m 0$, then $k_{m+1} \geq (-k_1 X_1) + \ldots + (-k_m X_m)$. Suppose, by contradiction, that there exists $i \in \{1, \ldots, m\}$, such that $k_i < 0$. Without loss of generality, we may assume that $i = 1$. For all $r \in Q$, by taking $X_1 \geq (-k_2 X_2) + \ldots + (-k_m X_m) - r$, we get that $k_{m+1} \geq r$. Thus, for all $r \in Q$, $k_{m+1} \geq r$, which is a contradiction. Therefore, $k_1 \geq 0, \ldots, k_{m+1} \geq 0$. Moreover, from $k_{m+1} \geq (-k_1 X_1) + \ldots + (-k_m X_m)$, where the $k_i$‘s are all non negative and $X_1, \ldots, X_m$ can be taken to be arbitrarily small positive numbers, it follows that for all negative $r \in Q$, $k_{m+1} \geq r$ and, thus, $k_{m+1} \geq 0$. 


(Case 2) Let \( \rho_{m+1} > \). By hypothesis we have that for all \( X_1, \ldots, X_n \in \mathbb{Q} \), if \( X_1 \rho_1 0, \ldots, X_m \rho_m 0 \), then \( k_{m+1} > (k_1 X_1) + \ldots + (k_m X_m) \). Similarly to Case (1), we have that \( k_1 \geq 0, \ldots, k_m \geq 0 \). Without loss of generality, we may assume that for \( i = 1, \ldots, \ell \), with \( 0 \leq \ell \leq m \), \( \rho_i \) is \( > \) and for \( i = \ell + 1, \ldots, m \), \( \rho_i \) is \( \geq 0 \). If \( \ell = 0 \) then for \( X_1 = \ldots = X_m = 0 \), we have that \( k_{m+1} > 0 \).

If \( \ell > 0 \) then, similarly to Case (1), we have that \( k_{m+1} \geq 0 \). It remains to show that if \( \ell > 0 \) then \((\dagger)\) holds. Suppose, by contradiction, that for \( i = 1, \ldots, \ell \), \( \rho_i = 0 \) and \( k_{m+1} = 0 \). Then for \( X_1 = \ldots = X_m = 0 \), from \( k_{m+1} > (k_1 X_1) + \ldots + (k_m X_m) \) we get \( 0 > 0 \).

**Proof of Theorem 4.3.**

If part Assume that \( k_1 p_1 + \ldots + k_m p_m + k_{m+1} = p_{m+1} \), for some \( k_1 \geq 0, \ldots, k_{m+1} \geq 0 \). The proof proceeds by cases.

(Case 1) Let \( \rho_{m+1} \geq 0 \). Since \( \rho_i \in \{\geq, >\} \), for \( i = 1, \ldots, m \), if \( t_1 \rho_1 0 \land \ldots \land t_m \rho_m 0 \) then \( k_1 t_1 + \ldots + k_m t_m + k_{m+1} \geq 0 \).

(Case 2) Let \( \rho_{m+1} > \rho_1, \ldots, \rho_\ell \) be >, for \( 0 \leq \ell \leq m \), and \((\sum_{i=1}^\ell k_i) > 0 \), where \( I = \{i \mid 1 \leq i \leq m+1, \rho_i >\} \). Then, either there exists \( i \in \{1, \ldots, \ell\} \) such that \( k_i > 0 \) or \( k_{m+1} > 0 \). Therefore \( k_1 p_1 + \ldots + k_m p_m + k_{m+1} > 0 \).

(Only If part) Assume that \( Q \models \forall(p_1 \rho_1 0 \land \ldots \land p_m \rho_m 0 \rightarrow p_{m+1} \rho_{m+1} 0) \). Without loss of generality, we can also assume that the set \( \{p_1 = 0, \ldots, p_i = 0\} \subseteq \{p_1 = 0, \ldots, p_m = 0\} \), with \( l \leq m \), is a maximal set of linearly independent equations. Let us define the following affine transformation \( \{X_1 = p_1, \ldots, X_i = p_i\} \), where the variables \( X_1, \ldots, X_i \) are of type \( \text{rat} \) and do not occur in \( p_1, \ldots, p_m, p_{m+1} \). By applying this transformation we obtain \( Q \models \forall(X_1 \rho_1 0 \land \ldots \land X_i \rho_i 0 \land f_1(X_1, \ldots, X_i) \rho_{i+1} 0 \land \ldots \land f_{m-\ell}(X_1, \ldots, X_i) \rho_m 0 \rightarrow g(X_1, \ldots, X_i, V) \rho_{m+1} 0) \), where the linear polynomials \( p_{k+1}, \ldots, p_m \) have been transformed into the linear polynomials \( f_1(X_1, \ldots, X_i), \ldots, f_{m-\ell}(X_1, \ldots, X_i) \), and the linear polynomial \( p_{m+1} \) has been transformed into the linear polynomial \( g(X_1, \ldots, X_i, V) \), where \( V = \text{vars}(p_{m+1}) - \text{vars}(p_1, \ldots, p_m) \). Since \( V \cap \{X_1, \ldots, X_i\} = \emptyset \), we have \( Q \models \forall(X_1 \rho_1 0 \land \ldots \land X_i \rho_i 0 \land f_1(X_1, \ldots, X_i) \rho_{i+1} 0 \land \ldots \land f_{m-\ell}(X_1, \ldots, X_i) \rho_m 0 \rightarrow \forall V (g(X_1, \ldots, X_i, V) \rho_{m+1} 0)) \). Let us show that \( V = \emptyset \). Suppose, by contradiction, that the set \( V \) is not empty. Without loss of generality, we can assume that \( g(X_1, \ldots, X_i, V) \) is of the form \( a Y + h(X_1, \ldots, X_i, V - \{Y\}) \), where \( a \neq 0, h \) is a linear polynomial, and \( Y \in V \), otherwise all the variables in \( V \) can be eliminated from \( g(X_1, \ldots, X_i, V) \). As a consequence, the formula \( \forall V (g(X_1, \ldots, X_i, V) \rho_{m+1} 0) \) is equivalent to \( false \) in \( Q \), and this contradicts the hypothesis that \( Q \models \exists(p_1 \rho_1 0 \land \ldots \land p_m \rho_m 0) \). This entails that \( V = \emptyset \) and we will write \( g(X_1, \ldots, X_i, V) \) as \( g(X_1, \ldots, X_i) \). Thus, we have \( Q \models \forall(X_1 \rho_1 0 \land \ldots \land X_i \rho_i 0 \land f_1(X_1, \ldots, X_i) \rho_{i+1} 0 \land \ldots \land f_{m-\ell}(X_1, \ldots, X_i) \rho_m 0 \rightarrow g(X_1, \ldots, X_i) \rho_{m+1} 0) \).

A straightforward consequence is that \( g(X_1, \ldots, X_i) \) is equivalent to

\[
\sum_{k=1}^{m+1} q_{k} X_k
\]

for some \( k_1, \ldots, k_{m+1} \) (where \( k_{l+1} = 0, \ldots, k_m = 0 \)). Hence, by Lemma A.8, we have that \( k_1 \geq 0, \ldots, k_{m+1} \geq 0 \) and if \( \rho_{m+1} \) is \( > \) then \((\sum_i k_i) > 0 \), where \( I = \{i \mid 1 \leq i \leq m+1, \rho_i >\} \).

In the following we prove that Property P1 stated in Section 4.2 is a consequence of Theorem 4.3.

**Property P1.** Let \( p \) and \( q \) be two linear polynomials, and \( p > 0 \) and \( q > 0 \) be two satisfiable, non-redundant constraints. \( Q \models \forall(p > 0 \leftrightarrow q > 0) \) if there exists a rational number \( k > 0 \) such that \( Q \models \forall(k p - q = 0) \).

**Proof.** (If part) By hypothesis we have \( Q \models \forall(k p - q = 0) \) for some \( k > 0 \). Thus, the thesis follows from the fact that for every \( k > 0 \), \( Q \models \forall(p > 0 \leftrightarrow k p) \).

(Only If part) Assume that \( Q \models \forall(p > 0 \leftrightarrow q > 0) \). By Theorem 4.3 we have that \( Q \models \forall(p > 0 \leftrightarrow q > 0) \) for exist \( k_1 \geq 0 \) and \( k_2 \geq 0 \) such that \( Q \models \forall(k_1 p + k_2 q = 0) \) and \( k_1 + k_2 > 0 \). Moreover,
we have that \( k_1 \neq 0 \) because, otherwise, \( \mathcal{Q} \models k_2 = q \) and the constraint \( q > 0 \) would be either unsatisfiable (if \( k_2 = 0 \)) or redundant if \( k_2 > 0 \). Thus, \( k_1 > 0 \) and \( k_2 \geq 0 \). Analogously, from \( \mathcal{Q} \models \forall (q > 0 \rightarrow p > 0) \) we get \( \mathcal{Q} \models \forall (k_3 q + k_4 = p) \), for \( k_3 > 0 \) and \( k_4 \geq 0 \). By substituting \( k_1 p + k_2 \) for \( q \), we have \( \mathcal{Q} \models \forall (k_3 (k_1 p + k_2) + k_4 = p) \), which entails \( k_3 k_2 + k_4 = 0 \). Since \( k_3 > 0 \), \( k_2 \geq 0 \), and \( k_4 \geq 0 \), we get \( k_2 = k_4 = 0 \). Thus, from \( \mathcal{Q} \models (k_1 p + k_2 = q) \) we get \( \mathcal{Q} \models (k_1 p = q) \), and the thesis follows.

The proof of Property P1 where the constraints \( p > 0 \) and \( q > 0 \) have been replaced by \( p \geq 0 \) and \( q \geq 0 \), respectively, is similar.

Now we introduce some notions that will be used in the proof of Theorem A.13 below. We will say that a substitution \( \alpha \) is for variables of type \( \mathsf{rat} \) if for every binding \( V/t \in \alpha \) we have that \( V \) is a variable of type \( \mathsf{rat} \) and \( t \) is a term of type \( \mathsf{rat} \). We will also say that \( \alpha \) is for the set \( S \) of variables (for \( S \), for short) if \( \alpha \) is of the form \( \{V_1/t_1, \ldots, V_n/t_n\} \) for \( \{V_1, \ldots, V_n\} = S \). Given two disjoint sets of variables \( S_1 \) and \( S_2 \), in the following we will denote by \( S_1 \prec S_2 \) any variable ordering of the form \( S_{11}, \ldots, S_{1h}, S_{21}, \ldots, S_{2k} \) such that \( S_1 = \{S_{11}, \ldots, S_{1h}\} \) and \( S_2 = \{S_{21}, \ldots, S_{2k}\} \).

Definition A.9. Let \( \alpha \) and \( \beta \) be two substitutions for variables of type \( \mathsf{rat} \), then \( \alpha \equiv \beta \) if for every variable \( V \) we have: (i) \( \mathcal{V}/t \in \alpha \iff \mathcal{V}/u \in \beta \) and (ii) \( \mathcal{Q} \models \forall (t = u) \).

In the following Definitions A.10 and A.11, and in Lemma A.12 we will denote by \( X, Y, \) and \( Z \) three disjoint sets of variables of type \( \mathsf{rat} \), by \( c \) a constant such that \( \mathcal{V}/c \subseteq X \cup Z \), by \( R \) a goal such that \( \mathcal{V}/R \cap Y = \emptyset \), and by \( H \) a formula such that: (i) \( \mathcal{V} \subseteq \mathcal{V}/R \), (ii) \( \mathcal{V}/R \cap \mathcal{V}/\mathcal{H} = \emptyset \), and (iii) \( \mathcal{V}/R \cap \mathcal{V}/\mathcal{H} = \emptyset \).

Definition A.10. A CM-redex is either fail or a triple \( (a \leftrightarrow b, S, \sigma) \) such that: (i) \( a \) is a constraint and \( \mathcal{V}/a \subseteq X \cup Z \), (ii) \( b \) is a conjunction and \( S \) is a finite set of formulas of the form \( pp0 \), where \( p \in \{\geq, >\} \) and \( p \) is a polynomial bilinear in the partition \( \mathcal{V}/S = (X \cup Y \cup Z) \), \( X \cup Y \cup Z \), (iii) for every \( pp0 \in S \), the polynomial \( p \) is in normal form w.r.t. the variable ordering \( X < Y < X \), (iv) for every monomial \( u \) occurring in \( b \) or in \( S \), either \( \mathcal{V}/u \cap Y = \emptyset \) or \( \mathcal{V}/u \cap (\mathcal{V}/S - (X \cup Y \cup Z)) = \emptyset \), (v) \( (\mathcal{V}/S - (X \cup Y \cup Z)) \cap \mathcal{V}/R = \emptyset \), and (vi) \( \sigma \) is a substitution for variables of type \( \mathsf{rat} \) such that \( \sigma \sigma = c, b \sigma = b, \) and \( \sigma \sigma = S \).

Definition A.11. Let \( D \) be a CM-redex of the form \( (a \leftrightarrow b, \{f_1, \ldots, f_n\}, \sigma) \) and \( \beta \) a substitution for variables of type \( \mathsf{rat} \) of the form \( \{Y_1/s_1, \ldots, Y_h/s_h\} \), where \( Y \subseteq \{Y_1, \ldots, Y_h\}, \{Y_1, \ldots, Y_h\} \cap (X \cup Z) = \emptyset \), and, for \( i = 1, \ldots, h \), \( \mathcal{V}/s_i \subseteq \mathcal{V}/R \) and \( \mathcal{V}/s_i \cap \mathcal{V}/R = \emptyset \). Then we say that \( D(\beta) \) holds if there exists a substitution \( \tau \) for variables of type \( \mathsf{rat} \) such that:

(a) \( \tau \) is of the form \( \{W_1/t_1, \ldots, W_k/t_k\} \), where \( W_1, \ldots, W_k \) is the set \( \mathcal{V}/\{f_1, \ldots, f_n\} = (X \cup Y \cup Z) \) and \( t_1, \ldots, t_k \in \mathcal{Q} \),

(b) \( \mathcal{Q} \models \forall X \forall Z (f_1 \tau \beta \wedge \ldots \wedge f_n \tau \beta) \),

(c) let a be of the form \( a_1 \wedge \ldots \wedge a_l \) and \( b \) of the form \( b_1 \wedge \ldots \wedge b_m \), where \( l \geq 0 \), \( m \geq 0 \), \( a_1, \ldots, a_l \) are atomic constraints, and \( b_1, \ldots, b_m \) are formulas of the form \( pp0 \), for some polynomial \( p \) and \( p \in \{\geq, >\} \), for all \( j \in \{1, \ldots, m\} \) either there exists \( i \in \{1, \ldots, l\} \) such that \( \mathcal{Q} \models \forall X \forall Z (a_i \leftrightarrow b_j \tau \beta) \) or \( \mathcal{Q} \models \forall X \forall Z (c \leftrightarrow b_j \tau \beta) \), and for all \( i \in \{1, \ldots, l\} \) there exists \( j \in \{1, \ldots, m\} \) such that \( \mathcal{Q} \models \forall X \forall Z (a_i \leftrightarrow b_j \tau \beta) \), and

(d) \( (\sigma \tau) |_Y \beta \equiv \beta \).
Lemma A.12. Let the relation \( \Rightarrow \) be defined as in the procedure CM and let \( D \) be a CM-redex. For every substitution \( \beta \) for variables of type \( \text{rat} \), \( D(\beta) \) holds if and only if (a.i) \( D \) is of the form \( \langle \text{true} \rightarrow \text{true}, S, \sigma \rangle \), (a.ii) \( \beta \) is a substitution of the form \( \{ Y_1/s_1, \ldots, Y_h/s_h \} \), where \( \{ Y_1, \ldots, Y_h \} \supseteq Y \) and \( \{ Y_1, \ldots, Y_h \} \cap (X \cup Z) = \emptyset \), and, for \( i = 1, \ldots, h \), \( \text{Vars}(s_i) \subseteq \text{Vars}(H) \) and \( \text{Vars}(s_i) \cap \text{Vars}(R) = \emptyset \), (a.iii) \( \text{Vars}(S) \cap (X \cup Y \cup Z) = \emptyset \) and \( \text{solve}(S) = \tau \), and (a.iv) \( (\sigma \tau)|_{Y \beta} \equiv \beta \), or there exists a CM-redex \( E \) such that: (b.i) \( D \Rightarrow E \) and (b.ii) \( E(\beta) \) holds.

Proof. (If part) Assume that \( D \) is a CM-redex and \( \beta \) is a substitution for variables of type \( \text{rat} \) such that they satisfy Conditions (a.i)–(a.iv). By Condition (a.i), \( D \) is of the form \( \langle \text{true} \rightarrow \text{true}, S, \sigma \rangle \), that is, it is different from fail. We want to show that \( D(\beta) \) holds, that is, Conditions (a)–(d) of Definition A.11 hold. Now, let \( S \) be the set \( \{ f_1, \ldots, f_n \} \). Then, by Condition (a.iii) and by the definition of the \( \text{solve} \) function, we have that the substitution \( \tau = \text{solve}(\{ f_1, \ldots, f_n \}) \) is of the form \( \{ W_1/t_1, \ldots, W_k/t_k \} \), where \( \{ W_1, \ldots, W_k \} \) is the set \( \text{Vars}(\{ f_1, \ldots, f_n \}) -(X \cup Y \cup Z) \) and \( t_1, \ldots, t_k \in \mathcal{Q} \) and, therefore, Condition (a) holds. Moreover, since \( \text{Vars}(S) \cap Y = \emptyset \) and the substitution \( \beta \) satisfies Condition (a.ii), we also have that \( \mathcal{Q} \models \forall X \forall Z (f_1 \tau \beta \wedge \ldots \wedge f_n \tau \beta) \) and Condition (b) holds. By Condition (a.i), the first element of the triple \( D \) is true \( \rightarrow \) true and, thus, we have that Condition (c) holds. Finally, by Condition (a.iv), we have \( (\sigma \tau)|_{Y \beta} \equiv \beta \), and we get that also Condition (d) holds and, therefore, \( D(\beta) \) holds.

Let us now assume that there exists a CM-redex \( E \) such that Conditions (b.i) and (b.ii) are satisfied. We want to show that \( D(\beta) \) holds. Since \( D \) is a CM-redex and, by using one of the rules (i)–(v) of \( D \), we can assume that \( D \) is different from fail, that is, \( D \) is of the form \( \langle a_1 \wedge \ldots \wedge a_i \rightarrow b_1 \wedge \ldots \wedge b_m, \{ f_1, \ldots, f_n \}, \sigma \rangle \), where \( a_1, \ldots, a_i \) are atomic constraints and \( b_1, \ldots, b_m \) are formulas of the form \( \rho \sigma \) for \( \rho \in \{ \geq, > \} \). In order to prove that \( D(\beta) \) holds, we proceed by cases considering the rule used for rewriting \( D \) into \( E \) and we show that Conditions (a)–(d) of Definition A.11 hold for \( D \) and \( \beta \). Suppose that we have obtained \( E \) from \( D \) by applying rule (i). Then, \( E \) is of the form \( \langle a_2 \wedge \ldots \wedge a_i \rightarrow b_1 \wedge \ldots \wedge b_m, \{ f_1, \ldots, f_n \}, \sigma \rangle \), where \( a_1 \) and \( b_i \) are of the form \( \rho \sigma \) for \( \rho \in \{ \geq, > \} \), and \( V \) is a new variable and, thus, it occurs neither in \( D \), nor in \( \beta \), nor in \( R \). Since \( E(\beta) \) holds, there exists a \( \tau' \) such that Conditions (a)–(d) hold for \( E \). Let \( \tau \) be defined as the substitution obtained from \( \tau' \) by removing the binding \( V/t \), where \( V \) is the new variable introduced by applying rule (i) to \( D \). Since \( D \) is a CM-redex, \( \text{Vars}(\{ p, q \}) \subseteq \text{Vars}(\{ f_1, \ldots, f_n \}) \cup X \cup Y \cup Z \) and, as a consequence, \( \tau \) is of the form \( \{ W_1/t_1, \ldots, W_k/t_k \} \), where \( \{ W_1, \ldots, W_k \} \) is the set \( \text{Vars}(\{ f_1, \ldots, f_n \}) -(X \cup Y \cup Z) \), and Condition (a) holds for \( D \). By hypothesis, we have that \( \mathcal{Q} \models \forall X \forall Z ((\rho \rho)(V \rho q) = 0 \wedge V > 0) \wedge f_1 \tau \beta \wedge \ldots \wedge f_m \tau \beta) \). Recalling that the variable \( V \) does not occur in \( f_1 \wedge \ldots \wedge f_n \), by the assumptions on \( E \), and by the definition of \( \tau \), we get that \( \mathcal{Q} \models \forall X \forall Z (f_1 \tau \beta \wedge \ldots \wedge f_n \tau \beta) \) and Condition (b) holds for \( D \). By the definition of \( \tau, \beta, \) and the function \( \rho \), we also have that \( \mathcal{Q} \models \exists V \forall X \forall Z ((\rho \rho)(V \rho p \beta \wedge q \rho \beta) = 0 \wedge V > 0) \). By the hypothesis that \( D \) is a CM-redex, we have that \( \text{Vars}(p) \subseteq X \cup Z \) and, thus, \( pr \beta = p \). Therefore, we get \( \mathcal{Q} \models \exists V \forall X \forall Z (V \rho q \beta = 0 \wedge V > 0) \), which entails, by Property P1, \( \mathcal{Q} \models \forall X \forall Z (a_1 \rightarrow b_1 \beta) \). Then, by the assumption that Condition (c) holds for \( E \), we get that Condition (c) holds for \( D \). Again, the variable \( V \) does not occur in \( \sigma \) and, thus, by the assumption that \( (\sigma \tau')|_{Y \beta} \equiv \beta \) and by the definition of \( \tau \), we get \( (\sigma \tau)|_{Y \beta} \equiv \beta \) and Condition (d) holds for \( D \). Therefore, if \( E \) has been obtained from \( D \) by applying rule (ii) and \( E(\beta) \) holds then \( D(\beta) \) holds.

Now suppose that we have obtained \( E \) from \( D \) by applying rule (ii). Then, \( E \) is of the form \( \langle \text{true} \rightarrow b_2 \wedge \ldots \wedge b_m, \{ (\rho \rho)(V \rho p_1 + \ldots + V \rho p_r + V \rho r+1 \rho q) = 0, V_1 \geq 0, \ldots, V_{r+1} \geq 0 \} \cup \{ f_1, \ldots, f_n \}, \sigma \rangle \), where \( p_1, \ldots, p_r \) are polynomials such that \( e \) is of the form \( p_1 \rho_1 \rho_1 0 \wedge \ldots \wedge p_r \rho_r 0, b_1 \) is of the form
By hypothesis, we have that 

\[ v_0 = 0 \]

where 

\[ \tau' = r_1 \otimes \ldots \otimes r_n \]

are in \( Q \). Since we have proved that \( \tau \) holds, we get \( \forall X \forall Z (f_1 \otimes \ldots \otimes f_n \rho \tau' = \tau) \). By the definition of \( \tau \), and the first component of the definition of the substitution \( \rho \), we have \( Q \models \forall X \forall Z (f_1 \otimes \ldots \otimes f_n \rho \tau' = \tau) \). Thus, by the assumption that Condition (c) holds for \( \tau' \) and \( \tau = \rho \tau' \), and Condition (d) holds for \( \tau' \). As a consequence, if \( E \) has been obtained from \( D \) by applying rule (iii) and \( E(\beta) \) holds then \( D(\beta) \) holds.

By using similar arguments, we can show that if \( E \) has been obtained from \( D \) by applying rule (iv). Then, \( E \) is of the form \( \langle a \leftarrow b, \{p = 0, q = 0\} \rangle \cup \{f_1, \ldots, f_n\}, \sigma \rangle \), where \( U \in X \cup Z \). By the hypothesis that \( E(\beta) \) holds, we have that there exists a substitution \( \tau' \) such that Conditions (a)–(d) hold for \( E \), which entails \( Q \models \forall X \forall Z ((p = 0) \tau' \land (q = 0) \tau' \land f_1 \tau' \land \ldots \land f_n \tau' = \tau) \). Now let \( \tau \) be the substitution \( \tau' \). Therefore, since \( U \subseteq X \cup Z \), we have \( U \tau' = U \) and \( Q \models \forall X \forall Z ((pU + q = 0) \tau' \land f_1 \tau' \land \ldots \land f_n \tau' = \tau) \). This observation and the definition of the substitution \( \tau \) entail that Conditions (a) and (b) hold for \( D \). Since \( \tau = \tau' \) and the first component of \( D \), that is, the formula \( a \leftarrow b \), is equal to the first component of \( E \), Condition (c) holds for \( D \). Finally, since the third component of \( D \), that is, the substitution \( \sigma \), is equal to the third component of \( E \), Condition (d) holds for \( D \). Hence, if \( E \) has been obtained from \( D \) by applying rule (iv) and \( E(\beta) \) holds then \( D(\beta) \) holds.

Finally, suppose that we have obtained \( E \) from \( D \) by applying rule (v). Hence, \( E \) is a CM-redux of the form \( \langle a \leftarrow b, \{b(U - \frac{2}{p})\} \rangle \cup \{\rho_1, \ldots, \rho_n\}, \sigma \rangle \), where \( U \in Y \), \( p \in (Q - \{0\}) \), \( \rho_1, \ldots, \rho_n \) are polynomials, and the predicate symbols \( \rho_1, \ldots, \rho_n \) are in \( \{\geq, >, =\} \). As a consequence, \( D \) is a CM-redux of the form \( \langle a \leftarrow b, \{pU + q = 0, p\rho_1 0, \ldots, p\rho_n 0\} \rangle \). Since \( E(\beta) \) holds, there exists a substitution \( \tau' \) such that Conditions (a)–(d) hold for \( D \). Let \( \tau \) be the substitution \( \tau' \). Since \( U \subseteq Y \), Condition (a) holds for \( D \). By hypothesis, \( D \) is a CM-redux, we have \( b = bU = bU + q = 0, p\rho_1 0, \ldots, p\rho_n 0 \) \( \Rightarrow \) \( \sigma = \{pU + q = 0, p\rho_1 0, \ldots, p\rho_n 0\} \), and thus, \( U \rho \{U - \frac{2}{p}\} = -\frac{q}{p} \). Moreover, by hypothesis we have also \( (\sigma) U = \{U - \frac{2}{p}\} \} \) \( \Rightarrow \) \( \rho \beta \equiv \beta \) and, thus, \( Q \models \forall (U = U) = \frac{2}{p} \). By the hypothesis that \( Q \models \forall X \forall Z ((n\rho_1 (U - \frac{2}{p})\rho_1 0) \tau' \land \ldots \land (n\rho_n (U - \frac{2}{p})\rho_n 0) \tau' = \tau \), our previous observations on \( U \rho \) and by the fact that \( \tau = \tau' \), we get \( Q \models \forall X \forall Z ((n\rho_1 (U - \frac{2}{p})\rho_1 0) \tau' \land \ldots \land (n\rho_n (U - \frac{2}{p})\rho_n 0) \tau' = \tau \), which entails \( Q \models \forall X \forall Z ((p\rho_1 0) \tau' \land \ldots \land (p\rho_n 0) \tau' \). Finally, we also have that \( Q \models \forall X \forall Z (pU + q = 0) \tau' \). As a consequence, Condition (b) holds for \( D \). Since we have proved that \( Q \models \forall (-\frac{2}{p}) = \frac{2}{p} \) and since Condition (c) holds for \( E \), then
in the form (Case a.i) Let us assume that $D$ is a CM-redex, or rule (iii), depending on the relation symbol of the leftmost atomic constraint in $D$. Now let the variable $V$ be the variable $U$ considered in the application of rule (v). We have that $U \sigma \{ U/ - \frac{q}{p} \} \tau = - \frac{q}{p}$ and, moreover, $Q \models \forall ( - \frac{q}{p} \beta = U \beta)$. Thus, we get that Condition (d) holds for $D$ and, hence, if $E$ has been obtained from $D$ by applying rule (v) and $E(\beta)$ holds then also $D(\beta)$ holds.

(Only If part) We prove that if $D(\beta)$ holds and it does not satisfy at least one among Conditions (a.i)–(a.iv) then there exists a CM-redex $E$ such that $D \Rightarrow E$ and $E(\beta)$ holds. Assume that $D(\beta)$ holds and, thus, it is of the form $\langle a \mapsto b, \{ f_1, \ldots, f_n \}, \sigma \rangle$. In what follows we will denote by $W_1$ the set $\text{Vars}(\{ f_1, \ldots, f_n \}) - (X \cup Y \cup Z)$.

(Case a.i) Let us assume that $D(\beta)$ does not satisfy Condition (a.i) of Lemma A.12. Then $D$ is not of the form $(true \mapsto true, \{ f_1, \ldots, f_n \}, \sigma)$. Since, by hypothesis, Condition (c) of Definition A.11 holds for $D$, we have that the number of literals in $a$ is not greater than the number of literals in $b$ and, thus, either (Case a.i.1) both $a$ and $b$ are different from true or (Case a.i.2) $a$ is true and $b$ is different from true.

In Case (a.i.1), $D$ is of the form $\langle a_1 \wedge \ldots \wedge a_l \mapsto b_1 \wedge \ldots \wedge b_m, \{ f_1, \ldots, f_n \}, \sigma \rangle$, where $a_1, \ldots, a_l$ are atomic constraints and $b_1, \ldots, b_m$ are formulas of the form $tp0$, for $p \in \{ \geq, > \}$. Since Condition (c) holds for $D$, there exist $i \in \{ 1, \ldots, l \}$ and $j \in \{ 1, \ldots, m \}$ such that $Q \models \forall (a_i \mapsto b_j)$ and, thus, it is possible to apply rule (i) to $D$. We get that $E$ is of the form $\langle a_2 \wedge \ldots \wedge a_l \mapsto b_1 \wedge \ldots \wedge b_{i-1} \wedge b_{i+1} \wedge b_m, \{ nf(Vp + q) = 0, V > 0 \} \cup \{ f_1, \ldots, f_n \}, \sigma \rangle$, where $a_1$ is $pp0$, $b_i$ is $q \rho 0$, and $V$ is a new variable which occurs neither in $D$, nor in $\beta$, nor in $R$. Now we show that $E$ is a CM-redex, that is, Conditions (i)–(vi) of Definition A.10 hold. By hypothesis, $a_1 \wedge \ldots \wedge a_l$ is a constraint whose variables are in $X \cup Z$ and thus, Condition (i) holds for $E$. In the following we will denote by $W_2$ the set $\text{Vars}(\{ nf(Vp - q) = 0, V > 0, f_1, \ldots, f_n \}) - (X \cup Y \cup Z)$. The polynomial $nf(Vp - q)$ is bilinear in the partition $W_2, X \cup Y \cup Z$ because $\text{Vars}(p) \subseteq X \cup Z$, $V$ is a new variable, and $q$ is bilinear in the partition $(W_1, X \cup Y \cup Z)$ (note that the function $nf$ preserves bilinearity). Thus, by the assumption that $E$ is a CM-redex, we have that Condition (ii) holds for $E$. By definition of the function $nf$, the polynomial $nf(Vp - q)$ is in normal form w.r.t. the variable ordering $Z < Y < X$ and thus, Condition (iii) holds for $E$. Since $\text{Vars}(p) \subseteq X \cup Z$, there is no monomial $u$ in $nf(Vp - q) = 0$ such that $\text{Vars}(u) \cap Y \neq \emptyset$ and $\text{Vars}(u) \cap W_2 \neq \emptyset$ and Condition (iv) holds for $E$. Since $V$ is a new variable, we get that also Conditions (v) and (vi) hold for $E$. As a consequence, $E$ is a CM-redex. Now we want to show that $E(\beta)$ holds, that is, Conditions (a)–(d) of Definition A.11 hold for $E$. Since $D(\beta)$ holds, there exists a substitution $\tau$ such that Conditions (a)–(d) hold for $D$, $\beta$, and $\tau$. In particular, there exists $j \in \{ b_1, \ldots, b_m \}$ such that $Q \models \forall X \forall Z (a_i \mapsto b_j \beta \tau)$. Without loss of generality we can assume that $i = j$ and thus, $Q \models \forall X \forall Z (pp0 \mapsto (qp0) \tau \beta)$. By Property P1 we get that there exists a rational number $k \succ 0$ such that $Q \models \forall X \forall Z (kp - q \beta \tau = 0)$. Now let $\tau' = \tau \cup \{ V/k \}$. Then Condition (a) holds for $E$ and $\tau'$. Moreover, since $\text{Vars}(p) \subseteq X \cup Z$, by the definition of $\tau'$ and $\beta$, we get that $Q \models \forall X \forall Z (nf(Vp - q) \tau' \beta = 0 \wedge V > 0 \wedge f_1 \tau' \beta \wedge \ldots \wedge f_n \tau' \beta)$ and Condition (b) holds for $E$. Condition (c) follows easily from the hypotheses. Finally, Condition (d) holds for $E$ since $V$ is a new variable which does not occur in $D$ and $\beta$. Therefore, we get that $E(\beta)$ holds.

In Case (a.i.2), where $a$ is true and $b$ is not true, since $D$ is a CM-redex and, thus, for $i = 1, \ldots, m$, $b_i$ is of the form $qp0$, where $p \in \{ \geq, > \}$, it is possible to apply either rule (ii) or rule (iii), depending on the relation symbol of the leftmost atomic constraint in $b$. Let us
assume that $b_1$ is of the form $q \geq 0$ and we apply rule (ii). We obtain, from $D$, the triple $E$ of the form \( \langle \text{true} \rightarrow b_2 \land \ldots \land b_m, \{nf(V_1p_1 + \ldots + V_rp_r + V_{r+1} - q) = 0, V_1 \geq 0, \ldots, V_{r+1} \geq 0\} \cup \{f_1, \ldots, f_n\}, \sigma \rangle \), where the constraint $c$ is $p_1p_10 \land \ldots \land p_rp_r0$ and $V_1, \ldots, V_{r+1}$ are new variables. Now we want to show that $E$ is a CM-redex, that is, Conditions (i)–(vi) of Definition A.10 hold. In the following we will denote by $W_2$ the set $\text{Vars}(\{nf(V_1p_1 + \ldots + V_rp_r + V_{r+1} - q) = 0, V_1 \geq 0, \ldots, V_{r+1} \geq 0\} \cup \{f_1, \ldots, f_n\}, \sigma)$, where the constraint $c$ is $p_1p_10 \land \ldots \land p_rp_r0$ and $V_1, \ldots, V_{r+1}$ are new variables. Thus, we are left with the first case. Since

\[ \forall \emptyset = 0 \text{ the polynomials } \langle f_1, \ldots, f_n \rangle - (X \cup Y \cup Z) \text{. Condition (i) trivially holds for } E. \text{ Since, by hypothesis, } p_1, \ldots, p_r \text{ are linear polynomials in the variables } X \cup Z, V_1, \ldots, V_{r+1} \text{ are new variables, and the polynomial } nf(V_1p_1 + \ldots + V_rp_r + V_{r+1} - q) \text{ is bilinear in the partition } \langle W_1, X \cup Y \cup Z \rangle, \text{ we have that there is no monomial } u \text{ in } nf(V_1p_1 + \ldots + V_rp_r + V_{r+1} - q) \text{ such that } \text{Vars}(u) \cap Y \neq \emptyset \text{ and } \text{Vars}(u) \cap W \neq \emptyset \text{ and, thus, Condition (iv) holds for } E. \text{ Since } V_1, \ldots, V_{r+1} \text{ are new variables, we have that also Condition (v) holds for } E. \text{ Finally, we have } (b_2 \land \ldots \land b_m) = b_2 \land \ldots \land b_m. \text{ Since } V_1, \ldots, V_{r+1} \text{ are variables not occurring in } D \text{ and } \beta, \text{ and } q \text{ occurs in } b_1 \land \ldots \land b_m, \text{ we have that also Condition (vi) holds for } E \text{ and, thus, } E \text{ is a CM-redex. Now let us show that } E(\beta) \text{ holds, that is, Conditions (a)–(d) of Definition A.11 are satisfied. By the assumption that } D(\beta) \text{ holds, there exists a substitution } \tau \text{ such that } Q \models \forall X \forall Z (f_1\tau_1 \land \ldots \land f_n\tau_1) \text{ and } Q \models \forall X \forall Z (c \rightarrow b_1\tau_1). \text{ As a consequence, by the hypothesis that } c \text{ is a satisfiable constraint, Theorem 4.3 entails that } Q \models \exists V_1 \ldots \exists V_{r+1} \forall X \forall Z (nf(V_1p_1 + \ldots + V_rp_r + V_{r+1} - q\tau) = 0 \land V_1 \geq 0 \land \ldots \land V_{r+1} \geq 0). \text{ Recalling that } V_1, \ldots, V_{r+1} \text{ are new variables which occur neither in } D \text{ nor in } \beta \text{, we can extend the scope of the existential quantifier for these variables over the conjunction } f_1\tau_1 \land \ldots \land f_n\tau_1, \text{ and we get that there exists a substitution } \tau' \text{ such that } \tau' = \tau \cup \{V_1/t_1, \ldots, V_{r+1}/t_{r+1}\}, \text{ for some } t_1, \ldots, t_{r+1} \in Q, \text{ and Conditions (a) and (b) hold for } E. \text{ By the hypotheses and by definition of } \tau' \text{ we have that Condition (c) holds for } E. \text{ Finally, by hypothesis we have that } \{\sigma(\tau)\}_{\langle Y^1 \beta \equiv \beta \rangle \land \text{Restrict } \tau'} \text{ and, thus, by the definition of } \tau', \text{ since } V_1, \ldots, V_{r+1} \text{ do not occur in } \sigma, \text{ Condition (d) holds for } E. \text{ As a consequence, } E(\beta) \text{ holds. The case where we obtain } E \text{ from } D \text{ by applying rule (iii) can be addressed by similar arguments.}

(Case a.ii) By hypothesis, Condition (ii) of Definition A.11 holds for $\beta$ and, thus, also Condition (a.ii) of Lemma A.12 holds for $\beta$.

(Case a.iii) Let $D$ be such that Condition (a.iii) of Lemma A.12 is not satisfied. In this case we have that either $\text{Vars}(\{f_1, \ldots, f_n\}) \cap (X \cup Y \cup Z) \neq \emptyset$ or $\text{Vars}(\{f_1, \ldots, f_n\}) \cap (X \cup Y \cup Z) = \emptyset$ and $\{f_1, \ldots, f_n\}$ is not satisfiable. The second case is impossible because by hypothesis we have

\[ Q \models \forall X \forall Z (f_1\tau_1 \land \ldots \land f_n\tau_1). \text{ Thus, we are left with the first case. Since } D(\beta) \text{ holds, for } i = 1, \ldots, n, f_i \text{ is a formula of the form } pp0, \text{ where the polynomial } p \text{ is bilinear in the partition } \langle W_1, X \cup Y \cup Z \rangle \text{ and it is in normal form w.r.t. the variable ordering } Z < Y < X, \text{ and } \rho \in \{\geq, \succ, =\}. \text{ Since } \text{Vars}(\{f_1, \ldots, f_n\}) \cap (X \cup Y \cup Z) = \emptyset, \text{ we can assume without loss of generality that } f_1 \text{ is of the form } pp0 \text{ and the polynomial } p \text{ is of the form } q_1U + q_2, \text{ where } \text{Vars}(q_1) \subseteq W_1, U \in (X \cup Y \cup Z), \text{ and } q_2 \text{ is bilinear in the partition } \langle W_1, X \cup Y \cup Z \rangle. \text{ Let us assume that } U \text{ is a variable in } X \cup Z. \text{ Then we can rewrite } D \text{ into } E \text{ by using rule (iv). Then, } E \text{ is of the form } (a \leftrightarrow b, \{q_1 = 0, q_2 = 0, f_2, \ldots, f_n\}, \sigma). \text{ In the following we will denote by } W_2 \text{ the set } \text{Vars}(\{q_1 = 0, q_2 = 0, f_2, \ldots, f_n\}) - (X \cup Y \cup Z). \text{ We first show that } E \text{ is a CM-redex, that is, Conditions (i)–(vi) of Definition A.10 hold. The formulas } a \text{ and } b \text{ are not modified by rule (iv). Thus, by the hypotheses, Condition (i) holds for } E. \text{ By construction, in the formulas } q_1 = 0 \text{ and } q_2 = 0, \text{ the polynomials } p_1 \text{ and } p_2 \text{ are bilinear in the partition } \langle W_2, X \cup Y \cup Z \rangle \text{ and in normal form w.r.t. the variable ordering } Z < Y < X. \text{ Therefore, Conditions (ii) and (iii) hold for}
E. By construction and by the hypotheses, also Conditions (iv)–(vi) hold. As a consequence, $E$ is a CM-redex. Now, we show that $E(\beta)$ holds, that is, Conditions (a)–(d) of Definition A.11 hold. By hypothesis, there exists a substitution $\tau$ such that Conditions (a)–(d) hold for $D$. Let $\tau'$ be $\tau$. Since, by applying rule (iv), we eliminate from $\{f_1, \ldots, f_n\}$ one occurrence of a variable $U \in X \cup Z$, Condition (a) holds for $\tau'$. Moreover, $Q \models \forall X \forall Z ((q_1 U + q_2 = 0)\tau' \beta, U \tau' \beta = U$, and $\tau' = \tau$ entail $Q \models \forall X \forall Z ((q_1 = 0)\tau' \beta \land (q_2 = 0)\tau' \beta)$ and, thus, Condition (b) holds for $E$. Conditions (c) and (d) hold for $E$ by hypothesis. Thus, $E(\beta)$ holds.

Now let us assume that $U \in Y$. In order to apply rule (v) to $D$ we must have $\text{Vars}(q_2) \cap \text{Vars}(R) = \emptyset$ and $q_1 \in Q - \{0\}$. By the hypotheses, there is no monomial $u$ in $q_1 U + q_2$ such that $\text{Vars}(u) \cap Y \neq \emptyset$ and $\text{Vars}(u) \cap W_1 \neq \emptyset$. Thus, since $U \in Y$ and $q_1 U + q_2$ is bilinear in the partition $(W_1, X \cup Y \cup Z)$ and in normal form w.r.t. the variable ordering $Z < Y < X$, we have that $q_1 \in Q - \{0\}$ and $\text{Vars}(q_2) \cap Z = \emptyset$. By definition of $R$ we have that $\text{vars}(R) \cap Y = \emptyset$ and, since by the hypothesis that $D(\beta)$ holds, we have $W_1 \cap \text{Vars}(R) = \emptyset$, we have to prove that there is no variable $V \in (X \cap \text{Vars}(q_2))$ such that $V \in \text{Vars}(R)$. By the hypothesis that $D(\beta)$ holds, there exists a substitution $\lambda$ such that $Q \models \forall X \forall Z ((q_1 U + q_2)\lambda \tau' \beta = 0)$ and, thus, by our previous observations, $Q \models \forall X \forall Z (U \beta = -\frac{q_1}{q_1} \lambda \tau' \beta)$. Recalling that $\text{Vars}(q_2) \subseteq \text{Vars}(Y) \cup \text{Vars}(W_1)$, $q_1$ is a constant of type $\text{rat}$, and $q_2$ is in normal form (in particular, there is at most one monomial for each variable), and by the definition of the substitution $\beta$, we have that $\text{Vars}(q_2) \subseteq \text{Vars}(Y) \cup \text{Vars}(W_1)$.

By hypothesis, we have that $\text{Vars}(Y \beta) \cap \text{Vars}(R) = \emptyset$. Thus, $\text{Vars}(q_2) \cap \text{Vars}(R) = \emptyset$ and we can apply rule (v) to $D$. Therefore, $E$ is of the form $(a \leftrightarrow (b U - \frac{q_2}{q_1})), \{nf(p_2 (U - \frac{q_2}{q_1}))p_0, \ldots, nf(p_n (U - \frac{q_2}{q_1}))p_0, \sigma(U - \frac{q_2}{q_1})\}$, where $U \in Y$, $q_1 \in (Q - \{0\}), q_2, p_1, \ldots, p_n$ are polynomials, and the predicates symbols $p_2, \ldots, p_n$ are in $\{\geq, >, =\}$. We first show that $E$ is a CM-redex, that is it satisfies Conditions (i)–(vi) of Definition A.10. In the following we will denote by $W_2$ the set $\{nf(p_2 (U - \frac{q_2}{q_1}))p_0, \ldots, nf(p_n (U - \frac{q_2}{q_1}))p_0\} - (X \cup Y \cup Z)$. By hypothesis, Condition (i) holds for $E$ and the formula $b$ is a conjunction of formulas of the form $pp0$, where the polynomial $p$ is bilinear in the partition $(W_1, X \cup Y \cup Z)$ and it is in normal form w.r.t. the variable ordering $Z < Y < X$. By the hypothesis that there is no monomial $u$ in $b$ such that $\text{Vars}(u) \cap Y \neq \emptyset$ and $\text{Vars}(u) \cap W_1 \neq \emptyset$, we get that the variable $U$ occurs in $b$ in monomials of the form $aU$ where $a$ is a constant in $Q - \{0\}$. Note that, since $U \in Y$, we have $W_1 = W_2$. Therefore, since $q_1$ is a constant in $Q - \{0\}$ and $q_2$ is bilinear in the partition $(W_1, X \cup Y \cup Z)$, we get that the polynomials in $b(U - \frac{q_2}{q_1})$ are bilinear in this partition. By these observations we get also that there is no monomial $u$ in $b(U - \frac{q_2}{q_1})$ such that $\text{Vars}(u) \cap Y \neq \emptyset$ and $\text{Vars}(u) \cap W_1 \neq \emptyset$. By similar observations we can prove that the polynomials $nf(p_2 (U - \frac{q_2}{q_1}))$, $\ldots$ and $nf(p_n (U - \frac{q_2}{q_1}))$ are bilinear in the same partition, they are in normal form w.r.t. the variable ordering $Z < Y < X$, and there is no monomial $u$ in $nf(p_2 (U - \frac{q_2}{q_1}))$, $\ldots$, $nf(p_n (U - \frac{q_2}{q_1}))$ such that $\text{Vars}(u) \cap Y \neq \emptyset$ and $\text{Vars}(u) \cap W_1 \neq \emptyset$. As a consequence, Conditions (ii)–(iv) hold for $E$. Let us denote the set $\{nf(p_2 (U - \frac{q_2}{q_1}))p_0, \ldots, nf(p_n (U - \frac{q_2}{q_1}))p_0\}$ by $S'$. Since $U$ occurs neither in $S'$ nor in $\sigma$, we get $S' \sigma(U - \frac{q_2}{q_1}) = S'$, Condition (vi) holds for $E$, and $E$ is a CM-redex. Now let us prove that $E(\beta)$ holds, that is, Conditions (a)–(d) of Definition A.11 hold. By the hypotheses, there exists a substitution $\tau$ such that Conditions (a)–(d) hold for $D$. Now let us define the substitution $\tau'$ to be $\tau$. We have that Condition (a) holds for $E$. We have also that $Q \models \forall X \forall Z ((q_1 U + q_2 = 0)\tau' \beta \land (p_2 \tau' \beta) \land \ldots \land (p_n \tau' \beta) \land \tau' \beta)$. Since, by hypothesis, $D$ is a CM-redex, we have $b \sigma = b, \{q_1 U + q_2 = 0, p_2 \rho_2, \ldots, p_n \rho_n\} \sigma = \{q_1 U + q_2 = 0, p_2 \rho_2, \ldots, p_n \rho_n\}$, and, thus, $U \sigma(U - \frac{q_2}{q_1}) = -\frac{q_2}{q_1}$. Moreover, since by hypothesis $\sigma(U - \frac{q_2}{q_1})\tau' \beta \equiv \beta$, we have that $Q \models \forall (U \beta = -\frac{q_2}{q_1} \beta)$. Therefore, we have $Q \models \forall X \forall Z ((\,nf(p_2 (U - \frac{q_2}{q_1}))p_0, \ldots, nf(p_n (U - \frac{q_2}{q_1}))p_0)\tau' \beta \land \ldots \land (\,nf(p_n (U - \frac{q_2}{q_1}))p_0)\tau' \beta)$. Finally, we have
also that \( \forall X \forall Z ((q_1 U + q_2 = 0) \tau \beta) \). As a consequence, Condition (b) holds for \( E \). Since we have proved that \( \forall (-\frac{p}{\beta} \beta = U \beta) \), and since Condition (c) holds for \( D \), then Condition (c) holds also for \( E \). Now, we only need to show that \( (\sigma \tau)|_{Y \beta} \equiv \beta \) entails \( (\sigma (U - \frac{q}{p}) \tau')|_{Y \beta} \equiv \beta \).

Let us consider a variable \( V \in Y \). If \( V \) is not the variable \( U \) considered in the application of rule (v), then, by the definition of \( \tau \) and by the hypothesis that \( D \) is a CM-redex, we have \( V(\sigma (U - \frac{q}{p}) \tau')|_{Y \beta} = V(\sigma \tau)|_{Y \beta} \). Now let the variable \( V \) be the variable \( U \) considered in the application of rule (v). We have that \( U \sigma (U - \frac{q}{p}) \tau = -\frac{p}{\beta} \) and, moreover, \( \forall (-\frac{p}{\beta} \beta = U \beta) \). Thus, we get that Condition (d) holds for \( E \). Hence, \( E(\beta) \) holds.

(Case a.iv) Finally, Condition (a.iv) of Lemma A.12 holds for \( D \) because of the hypothesis that Condition (d) of Definition A.11 holds for \( D \).

Thus, we have proved that if \( D \) does not satisfy one of the Conditions (a.i)–(a.iv) then it can be rewritten into a CM-redex \( E \) such that \( E(\beta) \) holds.

**Theorem A.13 (Termination, Soundness, and Completeness of CM)** Let \( \gamma \colon H \leftarrow c \land G \) and \( \delta \colon K \leftarrow d \land B \) be clauses in normal form and without variables in common. Suppose that \( \gamma \) and \( \delta \) are the input to the procedure \( \text{GM} \) and let the substitution \( \alpha \) and the goal \( R \) be an output of \( \text{GM} \). Let clauses \( \gamma' \colon H \leftarrow c \land B \alpha \land R \) and \( \delta' \colon K \alpha \leftarrow d \alpha \land B \alpha \) in normal form be the output to the constraint matching procedure \( \text{CM} \). Then the following properties hold:

(a) \( \text{CM} \) terminates, that is: (1) given a CM-redex \( D_0 \) and the rewriting relation \( \Longrightarrow \) defined in the procedure \( \text{CM} \), every sequence \( D_0 \Longrightarrow D_1 \Longrightarrow \ldots \) is finite and (2) for every CM-redex \( D \), there are finitely many CM-redexes \( E_1, \ldots, E_n \) such that, for \( i = 1, \ldots, n \), \( D \Longrightarrow E_i \);

(b) For all constraints \( e \) and substitutions \( \beta \), variables of type \( \text{rat} \), if \( e \) and \( \beta \) are an output of \( \text{CM} \), then:

1. \( \gamma' \equiv H \leftarrow e \land d \alpha \beta \land B \alpha \land R \),
2. \( B \alpha \beta = B \alpha \),
3. \( \text{Vars}(K \alpha \beta) \subseteq \text{Vars}(H) \), and
4. \( \text{Vars}(e) \subseteq \text{Vars}\{H, R\} \);

(c) For all constraints \( e \) and substitutions \( \beta \), variables of type \( \text{rat} \), if \( c \) is either unsatisfiable or admissible, and the following conditions hold:

1. \( \gamma' \equiv H \leftarrow e \land d \alpha \beta \land B \alpha \land R \),
2. \( B \alpha \beta = B \alpha \),
3. \( \text{Vars}(K \alpha \beta) \subseteq \text{Vars}(H) \), and
4. \( \text{Vars}(e) \subseteq \text{Vars}\{H, R\} \),

then an output of \( \text{CM} \) is a constraint \( e' \) and a substitution \( \beta' \) such that \( \forall (e' \land d \alpha \beta' \leftarrow e \land d \alpha \beta) \) and \( \beta' \equiv \beta|_{\text{Vars}_{\text{rat}}(K \alpha)} \).

**Proof.** (a) We first prove that, given a CM-redex \( D_0 \) and the rewriting relation \( \Longrightarrow \) defined in the procedure \( \text{CM} \), every sequence \( D_0 \Longrightarrow D_1 \Longrightarrow \ldots \) is finite. We will use a well-founded lexicographical ordering on \( \mathbb{N} \times \mathbb{N} \times \mathbb{N} \) defined as follows. Given \( (l_1, m_1, n_1) \) and \( (l_2, m_2, n_2) \) in \( \mathbb{N} \times \mathbb{N} \times \mathbb{N}, (l_1, m_1, n_1) >_{\text{lex}} (l_2, m_2, n_2) \) iff either \( l_1 > l_2 \), or \( l_1 = l_2 \) and \( m_1 > m_2 \), or \( l_1 = l_2 \) and \( m_1 = m_2 \) and \( n_1 > n_2 \). The relation \( >_{\text{lex}} \) is a well-founded partial order on the set \( \mathbb{N} \times \mathbb{N} \times \mathbb{N} \). Let us
now introduce the termination function $\xi$ that maps CM-redexes to elements of $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$ and is defined as follows: $\xi(D) = (0, 0, 0)$ if $D$ is the CM-redex fail and $\xi(D) = (l, m, n)$ if $D$ is of the form $(a \leftarrow b, S, \sigma)$, $l$ is the number of occurrences of formulas of the form $p \rho 0$ in $b$, for some polynomial $p$ and relation symbol $\rho$, $m$ is the cardinality of the set $\text{Vars}(S) \cap Y$, and $n$ is the number of occurrences in $S$ of the variables in $X \cup Z$. We will show that, for any two CM-redexes $D$ and $E$, if $D \Rightarrow E$ then $\xi(D) >_{lex} \xi(E)$. We proceed by cases: let us first consider the case where $D \Rightarrow E$ by using rule (i). Let $D$ be the CM-redex $(p \rho 0 \land f \leftarrow g_1 \land q \rho 0 \land g_2, S, \sigma)$. Then $E$ is the CM-redex $(f \leftarrow g, \{pU + q = 0\} \cup S, \sigma)$, where $V$ is a new variable and $\rho \in \{\geq, >\}$. If $\xi(D) = (l, m, n)$, then $\xi(E)$ is $(l - 1, m', n')$, for some $m'$ and $n'$, and, thus $\xi(D) >_{lex} \xi(E)$. Similarly, if $D \Rightarrow E$ by using rule (ii) or rule (iii) and $\xi(D) = (l, m, n)$, then $\xi(E) = (l - 1, m', n')$ and, thus, $\xi(D) >_{lex} \xi(E)$. Now let us consider the case where $D \Rightarrow E$ by using rule (iv). Let $D$ be the CM-redex $(f \leftarrow g, \{pU + q = 0\} \cup S, \sigma)$, then $E$ is the CM-redex $(f \leftarrow g, \{p = 0, q = 0\} \cup S, \sigma)$, where $U$ is a variable in $X \cup Z$. If $\xi(D) = (l, m, n)$, then $\xi(E)$ is $(l, m, n - 1)$ and, thus, $\xi(D) >_{lex} \xi(E)$. Finally, we consider the case where $D \Rightarrow E$ by using rule (v). Let $D$ be the CM-redex $(f \leftarrow g, \{aU + q = 0\} \cup S, \sigma)$, then $E$ is the CM-redex $(f \leftarrow \langle g\{U/ - \frac{q}{a}\}\rangle, \{nf\{U/ - \frac{q}{a}\}\rho 0 \mid p \rho 0 \in S\}, \sigma\{U/ - \frac{q}{a}\})$, where $U$ is a variable in $Y$. Let $\xi(D)$ be $(l, m, n)$ and let $\xi(E)$ be $(l', m', n')$. The number of occurrences of formulas of the form $p \rho 0$ in $g$ is equal to that in $g\{U/ - \frac{q}{a}\}$ and, therefore, $l = l'$. Since $D$ is a CM-redex, the polynomial $aU + q$ is in normal form w.r.t. the variable ordering $Z \prec Y \prec X$ and, thus, $U \notin \text{Vars}(q)$. As a consequence, $m' = m - 1$ and, hence, $\xi(D) >_{lex} \xi(E)$. Since $>_{lex}$ is a well-founded order, we have that, given a GM-redex $D_0$, every sequence $D_0 \Rightarrow D_1 \Rightarrow \ldots$ is finite.

Now we prove that, for every CM-redex $D$, there are finitely many CM-redexes $E_1, \ldots, E_n$, such that, for $i = 1, \ldots, n$, $D \Rightarrow E_i$. Let $D$ be of the form $(a \leftarrow b, S, \sigma)$. Since $b$ is a finite conjunction of literals, there are finitely many GM-redexes $E_1, \ldots, E_n$ such that, for $i = 1, \ldots, n$, $D \Rightarrow E_i$, by using rule (i), or rule (ii), or rule (iii). In the case where $D$ is rewritten by using rule (iv) or rule (v), we can use arguments similar to the ones for the case of rules (i)–(iii) because, by definition of CM-redex, $S$ is a finite set. Thus, we get the thesis.

(b) We assume that, given the input clauses $\gamma'$ and $\delta'$, the output of the procedure CM is the constraint $e$ and the substitution $\beta$. In the following by $\gamma''$ we will denote the clause $H \leftarrow e \land da \beta \land Ba \land R$. Assume that the constraint $e$ is unsatisfiable. Then $e$ is an unsatisfiable constraint such that $\text{Vars}(e) \subseteq \text{Vars}\{H, R\}$ and $\beta$ is a substitution of the form $\{U_1/a_1, \ldots, U_s/a_s\}$, where $\{U_1, \ldots, U_s\} = \text{Vars}_{\text{rat}}(Ka)$ and $a_1, \ldots, a_s$ are arbitrary terms of type $\text{rat}$ such that, for $i = 1, \ldots, s$, $\text{Vars}(a_i) \subseteq \text{Vars}(H)$. Now we will show that Conditions (b.1)–(b.4) hold. By Theorem A.4 we have that $\gamma'$ and $\delta'$ are in normal form. We have also that clause $\gamma''$ is in normal form. Indeed, the following properties hold: (i) the terms of type $\text{rat}$ occurring in $Ba \land R$ are distinct variables that do not occur in $H$ and (ii) $\gamma''$ has no constraint-local variables because: (ii.1) $\delta'$ is in normal form and, thus, $\text{Vars}(da) \subseteq \text{Vars}\{\{Ka, Ba\}\}$, (ii.2) $\beta$ is a substitution such that $\text{Vars}_{\text{rat}}(Ka, \beta) \subseteq \text{Vars}(H)$, and (ii.3) $e$ is a constraint such that $\text{Vars}(e) \subseteq \text{Vars}\{\{H, R\}\}$. By assumption, the constraint $e$ is unsatisfiable and, by construction, also the constraint $e$ is unsatisfiable, which entails $Q = \forall(c \leftarrow e \land da \beta)$. Therefore, by Lemma A.5, $\gamma' \equiv \gamma''$. Thus Condition (b.1) is satisfied. Since $\delta'$ is in normal form, the variables of type $\text{rat}$ in $Ba$ do not occur in $Ka$. Therefore, by definition of $\beta$, we get $Ba \beta = Ba$ and Condition (b.2) is satisfied. By Theorem A.4 we have that $\text{Vars}_{\text{tree}}(Ka) \subseteq \text{Vars}(H)$. Moreover, by the definition of $\beta$, we have that $\text{Vars}_{\text{rat}}(Ka, \beta) \subseteq \text{Vars}(H)$. Since $\text{Vars}_{\text{tree}}(Ka, \beta) = \text{Vars}_{\text{tree}}(Ka)$, Condition (b.3) is satisfied. Finally, the definition of $e$ entails that also Condition (b.4) is satisfied.

Now let us assume that $c$ is satisfiable. In the procedure CM, the set $X$ is defined to be
\( Vars(c) - Vars(Ba) \), the set \( Y \) is defined to be \( Vars(da) - Vars(Ba) \), and the set \( Z \) is defined to be \( Vars_{rat}(Ba) \). First, we show that the constraint \( c \), the sets \( X, Y, \) and \( Z, \) the goal \( R, \) and the atom \( H, \) satisfy the assumptions made in Definitions A.10 and A.11, and in Lemma A.12. By definition, \( X, Y, \) and \( Z \) are sets of variables of type \( rat \). By construction, the substitution \( \alpha \) is of the form \( \{ T_1/s_1, \ldots, T_h/s_h \} \), where \( \{ T_1, \ldots, T_h \} \subseteq Vars(B) \). Since \( \delta \) is in normal form, we have that \( Vars_{rat}(K) \alpha = Vars_{rat}(K) \) and, thus, by the hypothesis that \( \gamma \) and \( \delta \) have no variables in common and by the definition of \( X \) and \( Y \), we have that \( X \) and \( Y \) are disjoint sets. By definition, \( Z \) is disjoint from \( X \) and \( Y \). Moreover, by definition of \( X \) and \( Z \), we have \( Vars(c) \subseteq X \cup Z \) and by assumption \( c \) is satisfiable. Since \( \gamma \) and \( \delta \) have no variables in common, by construction of \( R \) and by definition of \( Y \), we have \( Vars(R) \cap Y = \emptyset \). Finally, since by Theorem A.4 the clause \( \gamma' \) is in normal form, \( X \subseteq Vars_{rat}(H) \), \( Vars_{rat}(H) \cap Vars_{rat}(R) = \emptyset \), and \( Vars_{rat}(H) \cap Vars_{rat}(Ba) = \emptyset \). Since in the procedure \( CM \) the constraint \( e \) is defined to be \( project(x, C) \), by Lemma 4.1 we have that Conditions (b.1)–(b.4) hold only if \( Q = \forall (c \leftrightarrow (e \land da \beta)) \), and Conditions (b.2) and (b.3) hold. The rewriting process of the procedure \( CM \) starts from the initial triple \( \{ c \leftrightarrow e \land da, \emptyset, \emptyset \} \). Since the output of \( CM \) is not \textit{fail}, at the end of the rewriting process we obtain a triple \( \{ true \leftrightarrow true, C, \sigma_Y \} \) such that: (1) for all \( f \in C \), \( f \) is a formula of the form \( pp \rho \), where \( \rho \in \{ \geq, >, = \} \), and \( Vars(p) \subseteq W \), where \( W \) is the set of new variables introduced during the rewriting process, and (2) \( C \) is a satisfiable set of atomic constraints and \( solve(C) = \sigma_W \). Thus, the output of \( CM \) is the substitution \( \beta = (\sigma_Y \sigma_W) \sigma_G \), where \( \sigma_G = \{ V_1/u_1, \ldots, V_l/u_l \} \), \( \{ V_1, \ldots, V_l \} = Vars_{rat}(K \sigma_Y \sigma_W) - Vars(H) \), and, for \( i = 1, \ldots, l \), \( Vars(u_i) \subseteq Vars(H) \). Now we show that the triple \( \{ true \leftrightarrow true, C, \sigma_Y \} \) is a \( CM \)-redex, that is, Conditions (i)–(vi) of Definition A.10 hold. Condition (i) trivially holds. Since, by hypothesis, \( C \) is a set of atomic constraints and \( Vars(C) \cap (X \cup Y \cup Z) = \emptyset \), Conditions (ii)–(iv) hold. By hypothesis, \( Vars(C) \) is a set of new variables, which, therefore, do not occur in \( R \) and, thus, Condition (v) holds. By construction, \( \sigma_Y \) is a substitution of the form \( \{ U_1/t_1, \ldots, U_k/t_k \} \) where \( \{ U_1, \ldots, U_k \} \subseteq Y \). Therefore, we have that \( \sigma_Y \) is a substitution for variables of type \( rat \) and, by the hypotheses on \( c, c \sigma_Y = c \). Moreover, since \( Vars(C) \) is a set of new variables, we also have that \( \delta \sigma_Y = S \). Condition (vi) holds, and, thus, \( \{ true \leftrightarrow true, C, \sigma_Y \} \) is a \( CM \)-redex. Now we show that \( \{ true \leftrightarrow true, C, \sigma_Y \}(\beta) \) holds (in the sense of Definition A.11) by proving that Conditions (a.i)–(a.iv) of Lemma A.12 hold. Condition (a.i) holds by hypothesis. By hypothesis, \( \beta \) is the substitution \( (\sigma_Y \sigma_W) \sigma_G \). Since the substitution \( \sigma_Y \) is constructed only by rule (v) of the procedure \( CM \), we have that for every binding \( V/t \in \sigma_Y \) the variable \( V \) does not occur in the term \( t \) and in the rest of the \( CM \)-redex, and by the definition of \( \sigma_G \), we have that \( \{ U_1, \ldots, U_k \} \cap \{ V_1, \ldots, V_l \} = \emptyset \). As a consequence, by the definitions of \( \beta \) and \( \sigma_W \), we have that \( \beta \) is of the form \( \{ Y_1/s_1, \ldots, Y_h/s_h \} \) and \( \{ Y_1, \ldots, Y_h \} = \{ U_1, \ldots, U_k \} \cup \{ V_1, \ldots, V_l \} \). We want to show that \( Y \subseteq \{ U_1, \ldots, U_k \} \cup \{ V_1, \ldots, V_l \} \). By construction, we have that \( \{ U_1, \ldots, U_k \} \subseteq Y \) and, by the hypothesis that \( \delta' \) is in normal form, \( Y \subseteq Vars_{rat}(Ka) \). Since, by the definition of \( \sigma_G \), \( Vars_{rat}(Ka) \subseteq \{ U_1, \ldots, U_k \} \cup \{ V_1, \ldots, V_l \} \), we get \( Y \subseteq \{ Y_1, \ldots, Y_h \} \). By the definition of the set \( Z \), by the fact that the clauses \( \gamma \) and \( \delta \) have no variables in common, and by the definition of \( \beta \), we have that \( \{ Y_1, \ldots, Y_h \} \cap (X \cup Z) = \emptyset \). Finally, since \( \sigma_Y \) is constructed by rule (v) and due to the ordering \( Z < Y < X \) on the variables, we have that \( Vars(Y \sigma_Y) \cap Z = \emptyset \). Therefore, by the definition of \( \sigma_W \) and \( \sigma_G \), we get that, for \( i = 1, \ldots, s \), \( Vars(s_i) \subseteq X \), \( Vars(s_i) \cap Vars(R) = \emptyset \), and, thus, Condition (a.ii) holds. Since, by hypothesis, \( C \) is a set of atomic constraints and \( Vars(C) \subseteq W \) and by the definition of the function \( solve \), we get that Condition (a.iii) holds. Finally, since \( \beta \) is \( (\sigma_Y \sigma_W) \sigma_G \) and since the terms \( u_1, \ldots, u_l \) in the definition of \( \sigma_G \) are arbitrary terms, we get that also Condition (iv) holds. Therefore, \( \{ true \leftrightarrow true, C, \sigma_Y \}(\beta) \) holds and, since \( \{ c \leftrightarrow e \land da, \emptyset, \emptyset \} \implies^* \{ true \leftrightarrow true, C, \sigma_Y \} \), by Lemma A.12, we get that \( \{ c \leftrightarrow e \land
such that Conditions (a)–(d) of Definition A.11 hold for \( \langle c \leftrightarrow e \land da, \emptyset, \emptyset \rangle \) and \( \beta \). Since, in this case, the set \( \{W_1, \ldots, W_k\} \), as defined in Condition (a) of Definition A.11, is empty, we get that \( \tau \) is the identity substitution.

Let the constraint \( c \) be of the form \( a_1 \land \ldots \land a_l \) and the constraint \( e \land da \) be of the form \( b_1 \land \ldots \land b_m \), where \( a_1, \ldots, a_l \) and \( b_1, \ldots, b_m \) are atomic constraints. Since, by hypothesis, \( \text{Vars}(c) \subseteq X \cup Z \) and \( \text{Vars}(e \land da) \subseteq X \cup Y \cup Z \), we get that for all \( j \in \{1, \ldots, m\} \) either there exists \( i \in \{1, \ldots, l\} \) such that \( Q \models \forall X \forall Y (a_i \leftrightarrow b_j) \) or \( Q \models \forall X \forall Z (e \rightarrow b_j) \), and for all \( i \in \{1, \ldots, l\} \) there exists \( j \in \{1, \ldots, m\} \) such that \( Q \models \forall X \forall Y (a_i \leftrightarrow b_j) \), which entails that \( Q \models \forall (e \leftrightarrow (e \land da)) \). By definition, \( Z = \text{Vars}(\text{Bo}) \) and, by the fact that \( \langle c \leftrightarrow e \land da, \emptyset, \emptyset \rangle(\beta) \) holds, \( Z \beta = Z \). Therefore, Condition (b.2) holds. Finally, we have that \( \text{Vars}(K\alpha\beta \text{rat}) \cap \text{Vars}(R) = \emptyset \). By Theorem A.4, we have that \( \text{Vars}_{\text{tree}}(K\alpha) \subseteq \text{Vars}(H) \) and, by the definition of \( \beta \), \( \text{Vars}_{\text{tree}}(K\alpha\beta \text{rat}) \subseteq \text{Vars}(H) \).

Since \( \delta' \) is in normal form, \( \text{Vars}_{\text{rat}}(K\alpha) \cap \text{Vars}_{\text{rat}}(\text{Bo}) = \emptyset \). Since Condition (b.2) holds, we have that \( \text{Vars}_{\text{rat}}(K\alpha\beta \text{rat}) \cap \text{Vars}_{\text{rat}}(\text{Bo}) = \emptyset \). Finally, since \( \gamma' \) is in normal form, \( \text{Vars}(c) \subseteq \text{Vars}(\{H, \text{Bo} \cup R\}) \), \( \text{Vars}_{\text{rat}}(K\alpha\beta) \subseteq \text{Vars}(H) \), and, thus, Condition (b.3) holds. Therefore, we get the thesis.

(c) Let us consider a constraint \( e \) and a substitution \( \beta \) such that Conditions (c.1)–(c.4) hold.

In the following by \( \gamma'' \) we will denote the clause \( H \leftrightarrow e \land da \beta \land \text{Bo} \land R \).

Let us assume that \( c \) is an unsatisfiable constraint. Since the clause \( \delta' \) is in normal form, by Condition (c.3) we have that \( \text{Vars}(da) \subseteq \text{Vars}(K\alpha) \cup \text{Vars}(\text{Bo}) \) and by Condition (c.4) we have that clause \( \gamma'' \) has no constraint-local variables. Since the clause \( \gamma' \) is in normal form, we have also that clause \( \gamma'' \) is in normal form. Thus, by Lemma A.5, we have that Condition (c.1) entails \( Q \models \forall (c \leftrightarrow e \land da \beta) \). Since \( c \) is unsatisfiable, the output of \( \text{CM} \) is an unsatisfiable constraint \( e' \) such that \( \text{Vars}(e') \subseteq \text{Vars}(\{H, R\}) \) and a substitution \( \beta' \) of the form \( \{U_1/a_1, \ldots, U_s/a_s\} \), where \( \{U_1, \ldots, U_s\} = \text{Vars}_{\text{rat}}(K\alpha) \) and \( a_1, \ldots, a_s \) are arbitrary terms of type \( \text{rat} \) such that, for \( i = 1, \ldots, s \), \( \text{Vars}(a_i) \subseteq \text{Vars}(H) \). As a consequence, we have that \( Q \models \forall (e \land da \beta \leftrightarrow e' \land da \beta') \). In order to show that \( \beta' \equiv \beta|_{\text{Vars}_{\text{rat}}(K\alpha)} \), we will show that Conditions (i)–(iii) of Definition A.9 hold. Condition (i) holds because of the definition of \( \beta' \). Finally, Conditions (ii) and (iii) hold because, by Condition (c.4), \( \text{Vars}(K\alpha\beta) \subseteq \text{Vars}(H) \) and \( \beta' \) is any substitution such that \( \text{Vars}(K\alpha \beta') \subseteq \text{Vars}(H) \). Thus, we get the thesis.

Now let us assume that \( c \) is a satisfiable, admissible constraint. We want to show that there exists a substitution \( \beta' \) and a constraint \( e' \) that are the output of \( \text{CM} \) such that \( Q \models \forall (e' \land da \beta' \leftrightarrow e \land da \beta) \) and \( \beta' \equiv \beta|_{\text{Vars}_{\text{rat}}(K\alpha)} \). Since \( c \) is satisfiable, the procedure \( \text{CM} \) begins by defining the set \( X \) as \( \text{Vars}(c) - \text{Vars}(\text{Bo}) \), the set \( Y \) as \( \text{Vars}(da) - \text{Vars}(\text{Bo}) \), the set \( Z \), and \( \text{Vars}_{\text{rat}}(\text{Bo}) \) as the constraint \text{project}(c, X). By following the same considerations given in Part (b) of this proof, we have that the constraint \( c \), the sets \( X, Y, \) and \( Z \), the goal \( R \), and the atom \( H \) satisfy the assumptions made in Definitions A.10 and A.11, and in Lemma A.12. After defining the sets \( X, Y, \) and \( Z \), and the constraint \( e' \) the triple \( \langle c \leftrightarrow e' \land da, \emptyset, \emptyset \rangle \) is rewritten by using the rewriting relation \( \equiv \) defined in the procedure \( \text{CM} \). By Part (a) of this proof, we know that the procedure \( \text{CM} \) terminates. First, we prove that \( \langle c \leftrightarrow e' \land da, \emptyset, \emptyset \rangle \) is a CM-redex, that is, it satisfies Conditions (i)–(vi) of Definition A.10. By the hypothesis that \( c \) is a constraint and by definition of \( X \) and \( Z \), we have that Condition (i) holds. By construction, \( e' \) is a constraint such that \( \text{Vars}(e') \subseteq X \). By hypothesis, \( da \) is a constraint and, by definition of \( Y \) and \( Z \), \( \text{Vars}(da) \subseteq Y \cup Z \). Therefore, Condition (ii) holds. Conditions (iii)–(vi) hold trivially. Let the substitution \( \beta|_{\text{Vars}_{\text{rat}}(K\alpha)} \) be of the form \( \{Y_1/s_1, \ldots, Y_h/s_h\} \). By assumption, the clauses \( \gamma \) and \( \delta \) have no common variables. Therefore, \( \text{Vars}(\gamma') \cap \text{Vars}(\delta') = \text{Vars}(\text{Bo}) \) and, since \( \delta' \) is in normal form, \( \text{Vars}_{\text{rat}}(\text{Bo}) \cap \text{Vars}_{\text{rat}}(K\alpha) = \emptyset \), which entails \( \text{Vars}_{\text{rat}}(K\alpha) \cap \text{Vars}_{\text{rat}}(H) = \emptyset \).
By Condition (c.3), \( \text{Vars}(K\alpha\beta) \subseteq \text{Vars}(H) \) and, thus, \( \text{Vars}_{\text{rat}}(K\alpha) = \{Y_1, \ldots, Y_h\} \). Since \( \delta' \) is in normal form, \( Y \subseteq \text{Vars}_{\text{rat}}(K\alpha) \) and, therefore, \( Y \subseteq \{Y_1, \ldots, Y_h\} \). By Condition (c.2), \( B\alpha\beta = B\alpha \), which entails \( \{Y_1, \ldots, Y_h\} \cap Z = \emptyset \). By our previous observations and by the definition of the set \( X \), \( \text{Vars}_{\text{rat}}(K\alpha) \cap X = \emptyset \) and, thus, \( \{Y_1, \ldots, Y_h\} \cap X = \emptyset \). Moreover, since, by Condition (c.3), \( \text{Vars}(K\alpha\beta) \subseteq \text{Vars}(H) \) and we have proved \( \{Y_1, \ldots, Y_h\} = \text{Vars}_{\text{rat}}(K\alpha) \), we get that, for \( i = 1, \ldots, h \), \( \text{Vars}(s_i) \subseteq \text{Vars}(H) \). Finally, since \( \gamma' \) is in normal form, we have that \( \text{Vars}_{\text{rat}}(H) \cap \text{Vars}_{\text{rat}}(R) = \emptyset \) and, thus, for \( i = 1, \ldots, h \), \( \text{Vars}(s_i) \cap \text{Vars}(R) = \emptyset \). Now we prove that \( \langle c \leftrightarrow e' \land \alpha, \emptyset, \emptyset \rangle (\beta|_{\text{Vars}_{\text{rat}}(K\alpha)}) \) holds, that is, there exists a substitution \( \tau \) such that Conditions (a)–(d) of Definition A.11 hold. Let \( \tau \) be the identity substitution. Since the second element of the CM-redex \( \langle c \leftrightarrow e' \land \alpha, \emptyset, \emptyset \rangle \) is the empty set, Conditions (a) and (b) hold. By using arguments similar to the ones for the case where \( c \) is unsatisfiable, at Point (c) of this proof, we have that \( \gamma'' \) is in normal form. Therefore, by Condition (c.1), \( Q \models \forall (c \leftrightarrow e' \land \alpha) \). Let the constraint \( c \) be the conjunction \( a_1 \land \ldots \land a_m \) and the constraint \( e' \land \alpha \beta \) be the conjunction \( b_1 \land \ldots \land b_n \), where \( a_1, \ldots, a_m \) and \( b_1, \ldots, b_n \) are atomic constraints. Since \( c \) is admissible, by Lemma 4.2, there exists an injection \( \mu : \{1, \ldots, m\} \rightarrow \{1, \ldots, n\} \) such that for \( i = 1, \ldots, m \), \( Q \models \forall (a_i \leftrightarrow b_{\mu(i)}) \) and for \( j = 1, \ldots, n \), if \( j \notin \{\mu(i) \mid 1 \leq i \leq m\} \), then \( Q \models \forall (a \rightarrow b_j) \). Since \( \mu \) is an injective function, we get that Condition (c) holds. Finally, since the third component of the CM-redex \( \langle c \leftrightarrow e' \land \alpha, \emptyset, \emptyset \rangle \) is the empty set, and \( \tau \) is the identity substitution, we get also that Condition (d) holds. Therefore, \( \langle c \leftrightarrow e' \land \alpha, \emptyset, \emptyset \rangle (\beta|_{\text{Vars}_{\text{rat}}(K\alpha)}) \) holds. By Lemma A.12 and by the termination of \( \text{CM} \), we have that \( \langle c \leftrightarrow e' \land \alpha, \emptyset, \emptyset \rangle \rightarrow^{*} \langle \text{true} \leftrightarrow \text{true}, C, \sigma_Y \rangle \) and Conditions (a.i)–(a.iv) of Lemma A.12 hold for the CM-redex \( \langle \text{true} \leftrightarrow \text{true}, C, \sigma_Y \rangle \) and the substitution \( \beta|_{\text{Vars}_{\text{rat}}(K\alpha)} \). Now we show that the procedure \( \text{CM} \) does not return \text{fail}, that is, Conditions (e) and (f) of the procedure \( \text{CM} \) hold for the CM-redex \( \langle \text{true} \leftrightarrow \text{true}, C, \sigma_Y \rangle \). In particular, Conditions (e) holds because, by Condition (a.iii) of Lemma A.12, \( \text{Vars}(C) \cap (X \cup Y \cup Z) = \emptyset \) and, since \( \langle \text{true} \leftrightarrow \text{true}, C, \sigma_Y \rangle \) is a CM-redex, the elements of the set \( C \) are atomic constraints. Moreover, Condition (e) holds because, by Condition (a.iii) of Lemma A.12, the set \( C \) of atomic constraints is satisfiable. Let \( \text{solve}(S) \) be \( \sigma_W \). Then the output of the procedure \( \text{CM} \) is a substitution \( \beta' = (\sigma_G|_{\text{rat}}) \). Let \( \alpha_1, \ldots, \alpha_t \) be an arbitrary term of type \( \text{rat} \) such that, for \( i = 1, \ldots, s \), \( \text{Vars}(s_i) \subseteq \text{Vars}(H) \). By Point (b.1) of Theorem A.13, which we proved above, since \( e' \) and \( \beta' \) are an output of \( \text{CM} \), we have that \( \text{true} \equiv H \leftrightarrow e' \land \alpha \beta' \land B\alpha \land R \). Clause \( H \leftrightarrow e' \land \alpha \beta' \land B\alpha \land R \) has no constraint-local variables because \( \text{Vars}(e' \land \alpha \beta') = \emptyset \) and, thus, it is in normal form. As a consequence, by Lemma A.5, \( Q \models \forall (c \leftrightarrow e' \land \alpha \beta') \). By Conditions (c.1)–(c.4) and recalling that also \( \gamma'' \) is in normal form, we also have that \( Q \models \forall (c \leftrightarrow e' \land \alpha \beta') \). Therefore, by transitivity, \( Q \models \forall (e' \land \alpha \beta' \leftrightarrow e' \land \alpha \beta) \). By Lemma A.12, we have that \( (\sigma_G|_{\text{rat}}) \beta \equiv \beta \). As a consequence, since by definition of \( \sigma_G \), for every variable \( V \) in \( \{U_1, \ldots, U_s\} \), \( \text{Vars}(K\sigma_G|_{\text{rat}}) \rightarrow^{*} \text{Vars}(H) \) the corresponding term \( V \sigma_G \) is any term of type \( \text{rat} \) such that \( \text{Vars}(V \sigma_G) \subseteq \text{Vars}(H) \), and since \( \text{Vars}(K\alpha\beta) \subseteq \text{Vars}(H) \), we have also \( \beta \equiv \beta|_{\text{Vars}_{\text{rat}}(K\alpha)} \) and we get the thesis.

**A.3. Termination, Soundness and Completeness of the Folding Algorithm**

At this point we are ready to show the termination, the soundness, and the completeness of the algorithm \( \text{FA} \).

**Proof of Theorem 4.4 (Termination, Soundness, and Completeness of \( \text{FA} \)).**

Let us assume that \( \gamma : H \leftarrow c \land G \) and \( \delta : K \leftarrow d \land B \) are clauses in normal form, without variables in common, and that they are the input of the algorithm \( \text{FA} \).
(1) By Point (a) of Theorem A.4, given a GM-redex $D_0$ and the rewriting relation $\Longrightarrow$ defined in the procedure $\text{GM}$, every sequence $D_0 \Longrightarrow D_1 \Longrightarrow \ldots$ is finite and, for every GM-redex $D$, there are finitely many GM-redexes $E_1, \ldots, E_n$ such that, for $i = 1, \ldots, n$, $D \Longrightarrow E_i$. Therefore, the number of possible outputs of $\text{GM}$ that are different from $\text{fail}$ is finite. Now let $\alpha$ and $R$ be an output of $\text{GM}$. By Point (b) of Theorem A.4, the clauses $\gamma'$: $H \leftarrow e \land B\alpha \land R$ and $\delta'$: $K\alpha \leftarrow d\alpha \land B\alpha$ are in normal form. Therefore, by Point (a) of Theorem A.13, given a CM-redex $D_0$ and the rewriting relation $\Longrightarrow$ defined in the procedure $\text{CM}$, every sequence $D_0 \Longrightarrow D_1 \Longrightarrow \ldots$ is finite and, for every CM-redex $D$, there are finitely many CM-redexes $E_1, \ldots, E_n$ such that, for $i = 1, \ldots, n$, $D \Longrightarrow E_i$. Now, assume that $\text{CM}$ returns $\text{fail}$. In this case the algorithm $\text{FA}$ takes a different output $\alpha$ and $R$ of $\text{GM}$ and executes the procedure $\text{CM}$ with the corresponding new input $\gamma'$ and $\delta'$. Since the number of possible outputs of $\text{GM}$ that are different from $\text{fail}$ is finite, we get that the algorithm $\text{FA}$ terminates.

(2) Let the clause $\eta$: $H \leftarrow e \land K\alpha\beta \land R$ be the output of the algorithm $\text{FA}$, where the substitution $\alpha$ and the goal $R$ are computed by $\text{GM}$, and the constraint $e$ and substitution $\beta$ are computed by $\text{CM}$. We want to show that clause $\eta$ can be derived by folding $\gamma$ using $\delta$ according to Definition 3.1. In order to do so, we need to show that Conditions (1)–(3) of Definition 3.1 hold for the constraint $e$, the substitution $\vartheta = \alpha\beta$, and the goal $R$. By Theorem A.4, we have $G \Rightarrow B\alpha \land R$ which, by the definition of $\text{\Rightarrow}$, implies that $H \leftarrow e \land G \equiv H \leftarrow e \land B\alpha \land R$. By Theorem A.4, we have also that $H \leftarrow e \land B\alpha \land R$ and $\delta\alpha$ are clauses in normal form. Moreover, by Theorem A.13, we have $H \leftarrow e \land B\alpha \land R \equiv H \leftarrow e \land d\alpha \land B\alpha \land R$. Since by Theorem A.13 we also have that $B\alpha\beta = B\alpha$, by transitivity of the equivalence relation $\equiv$, we conclude that $\gamma \equiv H \leftarrow e \land d\alpha \land B\alpha \land R$. As a consequence, Condition (1) holds, with $\vartheta = \alpha\beta$. Let us now consider a variable $X \in \text{EVars}(\delta)$. The substitution $\alpha$ satisfies Point (b.2) of Theorem A.4. Since $B\alpha\beta = B\alpha$, we have that $X\alpha\beta = X\alpha$, that is, $X\alpha\beta$ is a variable. Moreover, since $X\alpha \notin \text{V}(\{H,R\})$ and, by Theorem A.13, $\text{V}(e) \subseteq \text{V}(\{H,R\})$, we have that $X\alpha\beta \notin \text{V}(\{H,e,R\})$. Therefore Condition (2.1) of Definition 3.1 holds. Recall that if $X \in \text{EVars}(\delta)$ then Condition (b.2.2) of Theorem A.4 holds for the variable $X\alpha$. Let $Y$ be a variable in $\text{V}(d \land B)$ different from $X$. If $Y \in \text{EVars}(\delta)$ then $Y\alpha\beta = Y\alpha$ and, by Theorem A.4, $Y\alpha\beta$ does not occur in $Y\alpha\beta$. If $Y \notin \text{EVars}(\delta)$ then $Y \in \text{V}(K)$ and, by Theorem A.13, $\text{V}(K\alpha\beta) \subseteq \text{V}(H)$. Since $X\alpha\beta$ is a variable that does not occur in $\text{V}(\{H,e,R\})$, we have that it does not occur in $Y\alpha\beta$. As a consequence, Condition (2.2) of Definition 3.1 holds. Finally, since $\text{V}(K\alpha\beta) \subseteq \text{V}(H)$, also Condition (3) of Definition 3.1 holds.

(3) Let us assume that it is possible to fold the clause $\gamma$ using the clause $\delta$ according to Definition 3.1. That is, there exist a constraint $e$, a substitution $\vartheta$, and a goal $R$ such that Conditions (1)–(3) of Definition 3.1 are satisfied. Without loss of generality, we may assume that $\text{V}(e) \subseteq \text{V}(\{H,d\theta \land B\theta \land R\})$, because, in the case where $e$ has some variables not in $\text{V}(\{H,d\theta \land B\theta \land R\})$, we can obtain a clause equivalent to $\gamma$ by eliminating the extra variables using the function $\text{project}$. Now we want to show that, since Conditions (1)–(3) of Definition 3.1 are satisfied, the clause $\gamma'': H \leftarrow e \land d\theta \land B\theta \land R$ is in normal form. First, we show that $\text{V}(e) \cap \text{V}(B\theta) = \emptyset$. This is entailed by the following facts: $\text{V}_\text{rat}(B) \subseteq \text{EVars}(\delta)$, by the hypothesis that $\delta$ is in normal form, and, by Condition (2) of Definition 3.1, if $X \in \text{V}_\text{rat}(B)$ then $X\theta$ is a variable and it does not occur in $e$. Next, since $\delta$ is in normal form, we have $\text{V}(d\theta) \subseteq \text{V}(K\theta) \cup \text{V}(B\theta)$ and, thus, $\text{V}(d\theta) \subseteq \text{V}(H) \cup \text{V}(B\theta)$. By these observations, $H \leftarrow e \land d\theta \land B\theta \land R$ has no constraint-local variables. By Condition (2) of Definition 3.1 we also have that $\text{V}_\text{rat}(H) \cap \text{V}_\text{rat}(B\theta \land R) = \emptyset$, every term of type $\text{rat}$ in $B\theta \land R$ is a variable, and each variable of type $\text{rat}$ occurs at most once in $B\theta \land R$. 40.
Therefore, $\gamma''\text{ is in normal form. In the following we will denote the set } \text{Vars}(B) \cup \text{Vars}_{\text{tree}}(K)\text{ of variables by } V. \text{ Let us define the substitution } \alpha \text{ as } \vartheta|_V, \text{ then we have } G =_{AC} BA \land R. \text{ That is, Condition (c.1) of Theorem A.4 holds. Let us consider a variable } X \in \text{EVars}(\delta). \text{ By definition of } \alpha, \text{ Condition (c.2.1) of Theorem A.4 holds. Consider, now, a variable } Y \in \text{Vars}(d \land B) \text{ such that } Y \text{ is different from } X. \text{ If } Y \in V \text{ then by Condition (1) of Definition 3.1 we have that } X\alpha \text{ does not occur in } Y\alpha. \text{ If } Y \notin V \text{ then } Y\alpha = Y \text{ and, since } X\alpha \in \text{Vars}(G) \text{ and } \gamma \text{ and } \delta \text{ have no variables in common, } X\alpha \text{ does not occur in } Y\alpha. \text{ Therefore, Condition (c.2.2) of Theorem A.4 holds. Finally, by Condition (3) of Definition 3.1, we have } \text{Vars}(K\vartheta) \subseteq \text{Vars}(H) \text{ and thus, } \text{Vars}_{\text{tree}}(K\alpha) \subseteq \text{Vars}(H). \text{ Therefore, Condition (c.3) of Theorem A.4 holds. As a consequence, by Theorem A.4, the output of } \text{GM} \text{ is a substitution } \alpha' \text{ such that } \alpha'|_V, \text{ and the goal } R. \text{ By Theorem A.4, we have also that the clauses } \gamma'': H \leftarrow c \land Ba' \land R \text{ and } \delta': Ka' \leftarrow d\alpha' \land Bo' \text{ are in normal form.}

Now let $\gamma'$ and $\delta'$ be the input clauses of $\text{CM}. \text{ Since } G =_{AC} Ba' \land R, \text{ we have that } H \leftarrow c \land G \cong H \leftarrow c \land Ba' \land R. \text{ Therefore, by Condition (1) of Definition 3.1 and by transitivity of } \cong, \text{ we have } H \leftarrow c \land Ba' \land R \cong H \leftarrow e \land d\vartheta \land B\vartheta \land R. \text{ Let us define } \beta \text{ to be the substitution } \{X/s \mid X/s \in \vartheta, X/s \notin \alpha\}, \text{ where } \alpha \text{ is the substitution introduced above in this proof. Clearly, } \vartheta = \alpha \cup \beta \text{ and, by definition of } \alpha \text{ and by Condition (2) of Definition 3.1, we have also } \vartheta = \alpha\beta. \text{ Since } \alpha \text{ and } \alpha' \text{ differ only for the variables in } \text{Vars}_{\text{tree}}(K), \text{ we have } H \leftarrow c \land Ba' \land R \cong H \leftarrow e \land d\alpha' \land B\alpha' \land R. \text{ As a consequence, Condition (c.1) of Theorem A.13 holds. By definition of } \cong \text{ and } \beta, \text{ we have } Ba' \beta = Ba', \text{ and Condition (c.2) of Theorem A.13 holds. Condition (3) of Definition 3.1, the properties of } \alpha', \text{ and the definition of } \beta \text{ entail that Condition (c.3) of Theorem A.13 holds. By hypothesis, we have that } \text{Vars}(e) \subseteq \text{Vars}(\{H, d\vartheta \land B\vartheta \land R\}). \text{ Recalling that } \delta \text{ is in normal form, we have that } \text{Vars}(d) \subseteq \text{Vars}(\{K, B\}) \text{ and, thus, } \text{Vars}(d\vartheta) \subseteq \text{Vars}(\{K\vartheta, B\vartheta\}). \text{ Hence, we also get } \text{Vars}(e) \subseteq \text{Vars}(\{H, K\vartheta, R\}). \text{ Since, by Condition (3) of Definition 3.1, we have } \text{Vars}(K\vartheta) \subseteq \text{Vars}(H), \text{ we get that Condition (c.4) of Theorem A.13 holds. Therefore, by Theorem A.13, } \text{CM} \text{ does not return } \text{fail} \text{ and we get the thesis.} \quad \Box
References


