Joint NCC & IIS Bull., Comp. Science, 13 (2000), 1–11

A multi-branch narrowing: satisfiability and termination[*]

I. S. Anureev

A new notion of a multi-branch narrowing that allows case analysis to be built in is introduced. A narrowing strategy that preserves formula satisfiability is suggested. A formalism called formula rewriting systems specifying the strategy is defined. The termination of formula rewriting systems is considered.

Introduction

Automatization of formula proving have an important value for program verification and constraint satisfaction. Decision procedures for particular theories form the basis for automatization. Design of new decision procedures is often based on term rewriting systems. The formulas of base theories are considered as normal forms, and rewriting reduces wider class of formulas to normal forms.

Narrowing is a relation on terms that generalizes rewriting by using unification instead of matching. Therefore its usage in reducing formulas to their normal forms allows us to design more powerful decision procedures.

Narrowing was first introduced in [8, 9, 10] to perform unification in equational theories presented by a confluent and terminating rewriting system R. The narrowing process consists of building all possible narrowing derivations starting from the equation to be solved and computes in this way a complete set of unifiers modulo the equational theory defined by R.

Different strategies that restrict the size of the narrowing derivation tree have been proposed [5, 7, 9, 12, 14, 17, 18, 19, 20].

The problems of construction of a complete set of unifiers and proof of completeness of term rewriting systems are removed if narrowing is used as a method of reduction to normal forms. However, new problems of satisfiability preservation in each narrowing step and termination of the narrowing process need to be solved.

In this paper these problems are studied and some solutions are proposed.

To extend applicability of narrowing-based simplifications, a notion of a multi-branch narrowing is introduced in Section 2. It allows case analysis to be built in. In the same section the conditions that guarantee that narrowing preserves satisfiability are formulated. In Section 3 a formalism specifying the narrowing strategy that preserves satisfiability is considered. It is called formula rewriting systems. Sufficient conditions of satisfiability preservation for formula rewriting systems are stated. In Section 4 a special class of formula rewriting systems (constructor formula rewriting systems) is described. Termination of special classes of constructor formula rewriting systems w.r.t. innermost reduction strategy is considered in Sections 5, 6 and 7.

1. Preliminaries

The reader may refer to [6] for the concepts of terms, substitutions and rewriting systems. Notations used in this paper are listed below. Let Σ be the first-order signature $(\mathcal{F}, \mathcal{P}, \mathcal{X})$ composed of the set $\mathcal{F}$ of function symbols, the set $\mathcal{P}$ of predicate symbols and the set $\mathcal{X}$ of variables, $\mathcal{T}(\Sigma)$ denotes the set of terms over Σ, $\mathcal{F}(\Sigma)$ denotes the set of first-order formulas over Σ, $\mathcal{UF}(\Sigma)$ denotes the set of unquantified formulas over Σ, $\mathcal{E}(\Sigma)$ denotes the set $\mathcal{T}(\Sigma) \cup \mathcal{UF}(\Sigma)$ of expressions over Σ and $\mathcal{S}(\Sigma)$ denotes the set of substitutions over $\mathcal{T}(\Sigma)$, $\mathcal{A}r(f)$ is the arity of $f \in \mathcal{F} \cup \mathcal{P}$, $\mathcal{K}$ denotes a first-order

[*]Partially supported by RFBR under contract 00-01-00909 and youth grant 2000 of SB RAS.

algebraic Σ-structure. Whenever Σ contains the predicate symbol $=$, it will be interpreted as the equality relation in $\mathcal{K}$.

Given an expression $u \in \mathcal{E}(\Sigma)$, a set of expressions E, and a substitution $\sigma \in \mathcal{S}(\Sigma)$, $\mathcal{V}ar(u)$ denotes the set of variables of u, $\mathcal{MV}ar(u)$ denotes the multiset of variables of u, $\mathcal{MV}ar_E(u)$ denotes the multiset of variables of u except the variables occuring in subexpressions of u that belong to E, $root(u)$ denotes the root of u, $|u|$ denotes the number of occurences of functional and predicate symbols in u, $|u|_E$ denotes the number of occurences of functional and predicate symbols in u except the symbols occuring in subexpressions of u that belong to E, $\mathcal{P}(u)$ denotes the set of positions of u with Λ as the topmost position, $\mathcal{D}om(\sigma) = \{x | x \in \mathcal{X} \text{ and } x\sigma \neq x\}$ denotes the domain of σ and $\mathcal{VR}ange(\sigma) = \cup_{x \in \mathcal{D}om(\sigma)} \mathcal{V}ar(x\sigma)$ denotes the variable range of σ.

For distinct variables $x_1, \ldots, x_n$ and $t_1, \ldots, t_n \in \mathcal{T}(\Sigma)$, $(x_1 \to t_1, \ldots, x_n \to t_n)$ denotes the substitution σ such that $\mathcal{D}om(\sigma) \subseteq \{x_1, \ldots, x_n\}$ and $x_i\sigma = t_i$ for each $1 \leq i \leq n$. In particular, $(\ )$ is an identity substitution.

Given a set S and a multiset M, $\mathcal{FM}(S)$ denotes the set of all finite multisets of the elements of S and $\mathcal{O}(m, M)$ denotes the number of occurences of an element m in M.

Definition 1.1 A multiset $W \subseteq \mathcal{FM}(\mathcal{UF}(\Sigma))$ is satisfiable in $\mathcal{K}$ if the formula $\vee_{A \in W} A$ is satisfiable in $\mathcal{K}$. A binary relation $\to$ on the set $\mathcal{UF}(\Sigma) \cup \mathcal{FM}(\mathcal{UF}(\Sigma))$ is said to preserve satisfiability in $\mathcal{K}$ if W is satisfiable in $\mathcal{K}$ iff W' is satisfiable in $\mathcal{K}$ for all $W, W' \in \mathcal{UF}(\Sigma) \cup \mathcal{FM}(\mathcal{UF}(\Sigma))$ such that $W \to W'$.

Given a partial order $\succ$ on $\mathcal{T}(\Sigma)$, $\succ_m$ denotes the multiset extension of $\succ$. Let N be a set of nonnegative integers with a usual relation $>$.

Definition 1.2 Let $t_1, t_2 \in \mathcal{T}(\Sigma)$, $\theta, \phi, \tau, \sigma_1, \sigma_2 \in \mathcal{S}(\Sigma)$. The substitution θ is a unifier of t_1 and t_2 if $t_1\theta = t_2\theta$. A unifier θ is a most general unifier (MGU for short) of t_1 and t_2 if for each unifier ϕ of t_1 and t_2 there exists a substitution τ such that $\phi = \theta\tau$. The terms t_1 and t_2 are unifiable if there exists a unifier of t_1 and t_2. The substitution θ is a unifier of σ_1 and σ_2 if $x\sigma_1\theta = x\sigma_2\theta$ for each $x \in \mathcal{D}om(\sigma_1) \cup \mathcal{D}om(\sigma_2)$. A unifier θ of σ_1 and σ_2 is a most general unifier if for each unifier ϕ of σ_1 and σ_2 there exists a substitution τ such that $\phi = \theta\tau$. The substitutions σ_1 and σ_2 are unifiable if there exists a unifier of σ_1 and σ_2

Let us remind the unification algorithm.

Definition 1.3 Let U and V be sets of equalities, $x \in \mathcal{X}$, $t \in \mathcal{T}(\Sigma)$, and U^x_t denote the result of replacement of all occurences of the variable x in U by the term t. The unification algorithm consists in indeterministic application of the following rules:
- $(U \cup \{x = x\}, V) \to (U, V)$,
- $(U \cup \{x = t\}, V) \to (U^x_t, V^x_t \cup \{x = t\})$ if $x \notin \mathcal{V}ar(t)$,
- $(U \cup \{t = x\}, V) \to (U^x_t, V^x_t \cup \{x = t\})$ if $x \notin \mathcal{V}ar(t)$,
- $(U \cup \{f(t_1, \ldots, t_n) = f(t'_1, \ldots, t'_n)\}, V) \to (U \cup \{t_1 = t'_1, \ldots, t_n = t'_n\}, V)$.

Let W' be the result of application of the unification algorithm to the set

$$W = (\{x\sigma_1 = x\sigma_2 | x \in \mathcal{D}om(\sigma_1) \cup \mathcal{D}om(\sigma_2)\}, \emptyset).$$

If W' has the form $(\emptyset, \{x_1 = t_1, ..., x_n = t_n\})$ where $x_i \in \mathcal{X}$ and $t_i \in \mathcal{T}(\Sigma)$ for all $1 \leq i, j \leq n$, then the substitution $\theta = (x_1 \to t_1, ..., x_n \to t_n)$ is a unifier of the substitutions σ_1 and σ_2. If W' does not have the above form, then the substitutions σ_1 and σ_2 are not unifiable.

To guarantee preservation of satisfiability, we separately consider the case when an MGU is found without application of the decomposition rule. In this case the unification algoritm takes the form
- $(U \cup \{t = t\}, V) \to (U, V)$,
- $(U \cup \{x = t\}, V) \to (U^x_t, V^x_t \cup \{x = t\})$ if $x \notin \mathcal{V}ar(t)$,
- $(U \cup \{t = x\}, V) \to (U^x_t, V^x_t \cup \{x = t\})$ if $x \notin \mathcal{V}ar(t)$.

Definition 1.4 The terms t_1 and t_2 is said to be unifiable without decomposition if a unifier of t_1 and t_2 is found with the help of the above rules.

Example 1.5 The substitution $(z \rightarrow succ(x))$ is an MGU of the substitutions $(y \rightarrow z)$ and $(y \rightarrow succ(x))$. It is found by applying the rule $(\{z = succ(x)\}, \emptyset) \rightarrow (\emptyset, \{z = succ(x)\})$. Therefore the substitutions $(y \rightarrow z)$ and $(y \rightarrow succ(x))$ are unifiable without decomposition. □

The following example will illustrate the paper.

Example 1.6 Let $\mathcal{F}_{nat} = \{succ,\ pred,\ 0\}$ with arities 1, 1, and 0, respectively. Then Σ_{nat} denotes the signature $\{\mathcal{F}_{nat}, \{=\}, \mathcal{X}\}$ and the Σ_{nat}-structure K_{nat} specifies the natural numbers with a successor, predecessor ($pred(0) = 0$) and zero. □

2. Narrowing with satisfiability preservation

Our aim is to state the conditions guaranteeing that narrowing preserves satisfiability. But first we generalize narrowing to case analysis, the same conditions guaranteeing satisfiability preservation for the extension called multi-branch narrowing.

Let $\mathcal{B}$ be the conditional term rewriting system $\{p_i | l_i \rightarrow r_i \mid i \in I\}$ over Σ. Here p_i are arbitrary unquantified formulas.

Definition 2.1 A multi-branch $\mathcal{B}$-narrowing $\rightsquigarrow_{\mathcal{B}}$ is a set of pairs $(A, \{(p_i \wedge A[r_i]_q)\theta_i \mid i \in I\})$ such that $A \in \mathcal{UF}(\Sigma)$, $q \in \mathcal{P}(A)$, and θ_i is an MGU of the terms l_i and A_q.

Let R be a set of CTRSs.

Definition 2.2 A multi-branch R-narrowing $\rightsquigarrow_R$ is a set of pairs $(U \cup \{A\}, U \cup W)$ such that $U, W \in \mathcal{FM}(\mathcal{UF}(\Sigma))$, $A \in \mathcal{UF}(\Sigma)$, and $A \rightsquigarrow_{\mathcal{B}} W$ for some $\mathcal{B} \in R$.

Definition 2.3 A term t is called a $\mathcal{B}$-redex if the terms l_i and t are unifiable for each $i \in I$. A term t is called a redex of R if t is a redex of some $\mathcal{B} \in R$. The relation $\rightsquigarrow_{\mathcal{B},t}$ is defined by the set of pairs $(A, \{(p_i \wedge A[r_i]_q)\theta_i \mid i \in I\})$ such that $A \in \mathcal{UF}(\Sigma)$, $q \in \mathcal{P}(A)$, $t = A_q$, and θ_i is an MGU of the terms l_i and A_q for each $i \in I$.

The satisfiability preservation of a multi-branch narrowing imposes a limitation on redexes and conditional term rewriting systems (or CTRS for short). Let us describe the limitations.

Let $\mathcal{K}$ be a Σ-structure.

Definition 2.4 Let t be a redex of $\mathcal{B}$ and θ_i be an MGU of the terms l_i and t for each $i \in I$. The term t is said to have the completeness property in $\mathcal{K}$ if the formula $\forall \bar{x} \vee_{i \in I} (\exists \bar{y}_i (p_i \wedge \bar{x} = \bar{x}\theta_i))$ is valid in $\mathcal{K}$ where $\bar{x}$ is the set $\mathcal{V}ar(t)$ and $\bar{y}_i$ is the set $\mathcal{V}ar(p_i) \cup \mathcal{VR}ange(\theta_i)$ for each $i \in I$. A CTRS $\mathcal{B}$ is correct in $\mathcal{K}$ if $p_i \Rightarrow l_i = r_i$ is valid in $\mathcal{K}$ for each $i \in I$.

Theorem 2.5 If $\mathcal{B}$ is correct in $\mathcal{K}$ and a redex t of $\mathcal{B}$ has the completeness property in $\mathcal{K}$, then $\rightsquigarrow_{\mathcal{B},t}$ preserves satisfiability in $\mathcal{K}$.

Proof Let $W = \{(p_i \wedge A[r_i]_q)\theta_i \mid i \in I\}$, $A \in \mathcal{UF}(\Sigma)$, $q \in \mathcal{P}(A)$, $A_q = t$, and $A \rightsquigarrow_{\mathcal{B},t} W$.

Let A be satisfiable in $\mathcal{K}$. It follows from the completeness property that there exists $i \in I$ such that the formula $p_i\theta_i \wedge A\theta_i$ is satisfiable in $\mathcal{K}$.

Since $t\theta_i = l_i\theta_i$, $p_i\theta_i$ is satisfiable in $\mathcal{K}$, and the formula $p_i \Rightarrow l_i = r_i$ is valid in $\mathcal{K}$, the formula $p_i\theta_i \wedge A[r_i]_q\theta_i$ is satisfiable in $\mathcal{K}$. Then W is also satisfiable in $\mathcal{K}$.

Let W be satisfiable in $\mathcal{K}$. Then there exists $i \in I$ such that the formula $p_i\theta_i \wedge A\theta_i$ is satisfiable in $\mathcal{K}$.

Since $s\theta_i = l_i\theta_i$, $p_i\theta_i$ is satisfiable in $\mathcal{K}$, and the formula $p_i \Rightarrow l_i = r_i$ is valid in $\mathcal{K}$, the formula $A\theta_i$ is satisfiable in $\mathcal{K}$. Then A is also satisfiable in $\mathcal{K}$. □

Proposition 2.6 formulates the conditions when an instance of a redex is a redex.

Proposition 2.6 Let s be a redex of $\mathcal{B}$, θ_i be an MGU of the terms s and l_i for each $i \in I$, $\sigma \in \mathcal{S}(\Sigma)$, and the substitutions σ and θ_i are unifiable for each $i \in I$. Then $s\sigma$ is a redex of $\mathcal{B}$.

Proof Let ϕ_i be an MGU of θ_i and σ for each $i \in I$. Since

$$(s\sigma)\phi_i = (s\theta_i)\phi_i = (l_i\theta_i)\phi_i = (l_i\sigma)\phi_i = l_i\phi_i$$

for each $i \in I$, the substitution ϕ_i is a unifier of $s\sigma$ and l_i. Then $s\sigma$ is a redex of $\mathcal{B}$. □

Let us define the conditions when satisfiability is preserved for an instance of a redex that has the completeness property.

Theorem 2.7 Let s be a redex of $\mathcal{B}$, θ_i be an MGU of l_i and s for each $i \in I$, and $\sigma \in \mathcal{S}(\Sigma)$. If $\mathcal{B}$ is correct in $\mathcal{K}$, s has the completeness property, and the substitutions σ and θ_i are unifiable without decomposition for each $i \in I$, then $\rightsquigarrow_{\mathcal{B},s\sigma}$ preserves satisfiability in $\mathcal{K}$.

Proof Let $W = \{(p_i \wedge A[r_i]_q)\theta'_i \mid i \in I\}$ where θ'_i is an MGU of $s\sigma$ and l_i for each $i \in I$, $A \in \mathcal{UF}(\Sigma)$, $A_q = s\sigma$, and $A \rightsquigarrow_{\mathcal{B},s\sigma}$. The substitution θ'_i exists for each $i \in I$, since $s\sigma$ is a redex of $\mathcal{B}$ by Proposition 2.6.

Let $\bar{x}$ be the set $\mathcal{V}ar(s)$, $\bar{x}'$ be the set $\mathcal{V}ar(s\sigma)$, θ_i be an MGU of l_i and s for each $i \in I$, and $\bar{y}_i$ be the set $\mathcal{V}ar(p_i) \cup \mathcal{VR}ange(\theta_i)$ for each $i \in I$.

Let A be satisfiable in $\mathcal{K}$. It follows from the completeness property $\forall\bar{x} \vee_{i\in I} (\exists\bar{y}_i(p_i \wedge \bar{x} = \bar{x}\theta_i))$ that $\forall\bar{x}' \vee_{i\in I} (\exists\bar{y}_i(p_i \wedge \bar{x}\sigma = \bar{x}\theta_i))$.

Let ϕ_i be an MGU of σ and θ_i for each $i \in I$ and $\bar{y}'_i$ be the set $\mathcal{VR}ange(\theta_i) \cup \mathcal{VR}ange(\sigma)$ for each $i \in I$.

From the form of the rules of unification without decomposition it follows that formulas $\bar{x}\sigma = \bar{x}\theta_i$ and $\bar{x}' = \bar{x}'\phi_i$ are equivalent. Then $\forall\bar{x}' \vee_{i\in I} (\exists\bar{y}'_i(p_i\phi_i \wedge \bar{x}' = \bar{x}'\phi_i))$.

Since $s\theta_i = l_i\theta_i$, $p_i\theta_i$ is satisfiable in $\mathcal{K}$, and the formula $p_i \Rightarrow l_i = r_i$ is valid in $\mathcal{K}$, there exists $i \in I$ such that the formula $p_i\phi_i \wedge A[r_i]_q\phi_i$ is satisfiable in $\mathcal{K}$.

By the proof of Proposition 2.6, the substitution ϕ_i is a unifier of $s\sigma$ and l_i for each $i \in I$. Since $\phi_i = \theta'_i\tau$ for some $\tau \in \mathcal{S}(\Sigma)$, $p_i\theta'_i \wedge A[r_i]_q\theta'_i$ is satisfiable in $\mathcal{K}$. Then W is also satisfiable in $\mathcal{K}$.

Let W be satisfiable in $\mathcal{K}$. The proof that A is satisfiable in $\mathcal{K}$ is analogous to that of Theorem 2.5. □

3. Formula rewriting systems

Consider a formalism specifying the narrowing strategy that preserves satisfiability.

Let $\mathcal{B} = \{p_i | l_i \to r_i \mid i \in I\}$ be a CTRS over Σ.

Definition 3.1 Let $s \in \mathcal{T}(\Sigma) \setminus X$. A pair $\rho = (\mathcal{B}, s)$ is called a formula rewrite rule over Σ if the terms l_i and s are unifiable for each $i \in I$. The term s is called a sample of ρ. A finite set of formula rewrite rules over Σ is called a formula rewriting system (or FRS for short) over Σ.

Note that a term rewrite rule $l \to r$ can be treated as the formula rewrite rule $(\{l \to r\}, l)$. Therefore, term rewriting systems are a special case of FRSs.

Example 3.2 The pair $\rho_{nat} = (\{pred(succ(x)) \to x,\ pred(0) \to 0\},\ pred(y))$ is a formula rewrite rule over the signature Σ_{nat}, since the terms $pred(succ(x))$ and $pred(0)$ are unifiable with $pred(y)$. The substitution $\theta_1^{nat} = (y \to succ(x))$ is an MGU of $pred(succ(x))$ and $pred(y)$, and the substitution $\theta_2^{nat} = (y \to 0)$ is an MGU of $pred(0)$ and $pred(y)$. □

Let $\rho = (\mathcal{B}, s)$ be a formula rewrite rule over Σ, θ_i be an MGU of the terms l_i and s for each $i \in I$.

Definition 3.3 A term t is called a redex of the rule ρ if there exists a substitution σ such that $t = s\sigma$ and substitutions σ and θ_i are unifiable without decomposition for each $i \in I$. A term t is called a redex of an FRS R if t is a redex of some rule of R.

Let ϕ_i be an MGU of the substitutions σ and θ_i for each $i \in I$.

Example 3.4 The term $pred(z)$ is a redex of ρ_{nat}, since $pred(z) = pred(y)\sigma_{nat}$, where $\sigma_{nat} = (y \to z)$, the substitution $\phi_1^{nat} = (z \to succ(x))$ is an MGU of the substitutions σ_{nat} and θ_1^{nat} and the substitution $\phi_2^{nat} = (z \to 0)$ is an MGU of the substitutions σ_{nat} and θ_2^{nat}. These MGUs are found without application of the decomposition rule of the unification algorithm. □

Definition 3.5 An ρ-reduction relation $\to_\rho$ is a set of pairs $(A, \{(p_i \wedge A[r_i]_q)\phi_i \mid i \in I\}$ such that $A \in \mathcal{UF}(\Sigma)$, $q \in \mathcal{P}(A)$, and A_q is a redex of ρ. Let R be an FRS. An R-reduction relation $\to_R$ is a set of pairs $(U \cup \{A\}, U \cup W)$ such that $\rho \in R$, $U, W \in \mathcal{FM}(\mathcal{UF}(\Sigma))$, $A \in \mathcal{UF}(\Sigma)$, and $A \to_\rho W$.

Term rewriting systems can be used for rewriting both formulas and terms. From the definition of $\to_\rho$ we see that FRSs can be used only for rewriting formulas. That is why they are called formula rewriting systems.

Example 3.6 $pred(z) = z \to_{\rho_{nat}} \{x = succ(x),\ 0 = 0\}$. □

For each FRS R over Σ there is a corresponding abstract reduction system [13], namely

$$(\mathcal{FM}(\mathcal{UF}(\Sigma)), \to_R).$$

Therefore, all concepts defined for abstract reduction systems (termination, normal form and so on) are inherited by R.

The termination property is undecidable for FRSs [1]. The conditions of satisfiability preservation is given by the following theorem.

Theorem 3.7 Let K be a Σ-structure and $\rho = (\mathcal{B}, s)$ be a formula rewrite rule such that $\mathcal{B}$ is correct in $\mathcal{K}$ and the redex s of $\mathcal{B}$ has the completeness property in $\mathcal{K}$. Then $\to_\rho$ preserves satisfiability in $\mathcal{K}$.

Proof Let t be a redex of ρ. Then there exists a substitution σ such that $t = s\sigma$ and substitutions σ and θ_i are unifiable without decomposition for each $i \in I$.

Since $\mathcal{B}$ is correct in $\mathcal{K}$ and s has the completeness property in $\mathcal{K}$, from Theorem 2.7 it follows that $\rightsquigarrow_{\mathcal{B},t}$ preserves satisfiability in $\mathcal{K}$.

By definition of $\to_\rho$, if $A[t]_q \to_\rho W$ then $A[t]_q \rightsquigarrow_{\mathcal{B},t} W$ for all $A \in \mathcal{UF}(\Sigma)$, $q \in \mathcal{P}(A)$, and $W \subseteq \mathcal{UF}(\Sigma)$.

Then $A[t]_q$ is satisfiable in $\mathcal{K}$ iff W is satisfiable in $\mathcal{K}$ for all $A \in \mathcal{UF}(\Sigma)$, $q \in \mathcal{P}(A)$, $W \subseteq \mathcal{UF}(\Sigma)$, and $t \in \mathcal{T}(\Sigma)$ such that $A[t]_q \to_\rho W$. Hence $\to_\rho$ preserves satisfiability in $\mathcal{K}$. □

Example 3.8 The sufficient conditions of satisfiability preservation for the rule ρ_{nat} are $pred(succ(x)) = x$, $pred(0) = 0$ and $\forall y(\exists x(y = succ(x)) \vee y = 0)$. It is obvious that the conditions are valid in K_{nat}.□

4. Constructor formula rewriting systems

Many FRSs arising in practice are constructor FRSs. A constructor FRS R is an FRS in which the set of functional symbols can be partitioned into a set $\mathcal{A}$ of defined functional symbols (or analyzers) and a set $\mathcal{C}$ of constructors, such that for every $\rho \in R$ its sample has the form $f(t_1, ..., t_n)$ with $f \in \mathcal{A}$ and $t_1, ..., t_n \in \mathcal{T}((\mathcal{C}, \mathcal{P}, \mathcal{X}))$. The related concept for term rewriting systems has been considered, for instance, in [13].

Let R be a constructor FRS over Σ with a set of analyzers $\mathcal{A}$ and a set of constructors $\mathcal{C}$. Let us introduce some concepts that allow us to analyze a structure of expressions w.r.t. R.

Definition 4.1 An expression u is called constructive if $u \in \mathcal{E}((\mathcal{C}, \mathcal{P}, \mathcal{X}))$. A substitution σ is constructive if $x\sigma \in \mathcal{T}((\mathcal{C}, \mathcal{P}, \mathcal{X}))$ for each variable $x \in \mathcal{X}$. A term in which no variable occurs twice or more is called linear. A substitution σ is linear on $X \subseteq \mathcal{X}$ if for all variables $x, y \in X$ the term $x\sigma$ is linear and $\mathcal{V}ar(x\sigma) \cap \mathcal{V}ar(y\sigma) = \emptyset$ if $x \neq y$. An expression u is nested if u has nested occurences of analyzers. An expression u is called simple if u is not nested. A term t is a call if $root(t) \in \mathcal{A}$.

$C_m(u)$ denotes the multiset of all simple calls that occur in $u \in \mathcal{E}$.

Definition 4.2 A map $\mu_c : \mathcal{E}(\Sigma) \to \mathcal{FM}(N)$ is a constructor measure if $\mu_c(u) = \{|t| \mid t \in C_m(u)\}$. A map $Dec_v : \mathcal{E}(\Sigma) \to \mathcal{FM}(\mathcal{X})$ is a variable decomposition if $Dec_v(u) = \cup_{t \in C_m(u)} \mathcal{MV}ar(t)$.

Let $C_e(u)$ denote the multiset of all nested calls of $u \in \mathcal{E}(\Sigma)$, μ_a and μ_v denote the maps such that $\mu_a(u) = |C_e(u)|$ and $\mu_v(u) = \{\mathcal{O}(x, Dec_v(u)) \mid \mathcal{O}(x, Dec_v(u)) \neq 0\}$, respectively.

Finding of classes of terminating FRSs is a very difficult problem. In the following sections three special classes of constructor FRSs are considered. They all are terminating w.r.t. a special strategy of application of rules.

Definition 4.3 Let $\mathcal{R}$ be the set of all FRSs over Σ. A function

$$s : \mathcal{R} \to \mathcal{FM}(\mathcal{UF}(\Sigma)) \times \mathcal{FM}(\mathcal{UF}(\Sigma))$$

is called a reduction strategy for FRSs if $s(R) \subseteq \to_R$ for all $R \in \mathcal{R}$. An FRS R is terminating w.r.t. s if $s(R)$ is terminating.

The following reduction strategy guarantees the termination of classes of FRSs mentioned above.

Definition 4.4 A reduction strategy for AESs is an innermost reduction strategy if the rules of AESs are applied to the redexes that do not contain redexes as their proper subterms.

5. Analyzer elimination systems

The following class of constructor FRSs (analyzer elimination systems or AESs for short) allows us to design simplifiers that eliminate analyzers. The idea of these systems consists in percolating analyzers through constructors to variables followed by their elimination by variable replacement. Unfortunately, even very strong restrictions imposed on AESs guarantee termination only w.r.t. a certain reduction strategy.

Definition 5.1 A constructor FRS R is an analyzer elimination system if any simple call is a redex of R and for each rule $(\{p_i | l_i \to r_i \mid i \in I\}, s) \in R$ and for each $i \in I$ the following properties hold:
- θ_i is linear on $\mathcal{V}ar(s)$ and constructive,
- p_i and r_i are simple,
- p_i and r_i are constructive if $l_i \neq s$,
- $Dec_v(s) \supseteq Dec_v(p_i) \cup Dec_v(r_i)$,
- $\mu_c(s) >_m \mu_c(p_i)$ and $\mu_c(s) >_m \mu_c(r_i)$.

Example 5.2 Let R be an FRS that consists of the following rules:
- $\rho_1 : pred(succ(x)) \to x$,
- $\rho_2 : pred(0) \to 0$,
- $\rho_3 : (\{pred(succ(x)) \to x,\ pred(0) \to 0\}, pred(y)\ )$.

Show that R is an AES with the analyzer *pred*.

Let t be a simple call. By the definiton of a simple call, t takes one of the following forms: $pred(succ(t'))$, $pred(0)$ or $pred(z)$, where t' is a constructive term and $z \in \mathcal{X}$.

Considering these cases, we see that t is a redex of the rules ρ_1, ρ_2, and ρ_3, respectively.

Conditions from the definition of AES for the rule ρ_1 with the sample $s = pred(succ(x))$ take the form:
– the identical substitution $\theta_1 = ()$ is linear on $\{x\}$,
– the right-hand side x is simple,
– the right-hand side x is constructive if $s = pred(succ(x))$,
– $Dec_v(s) = \{x\} \supseteq \emptyset = Dec_v(x)$,
– $\mu_c(s) = \{1\} >_m \emptyset = \mu_c(x)$.

Conditions from the definition of AES for the rule ρ_3 with the sample $s = pred(y)$ take the form:
– the substitutions $\theta_1 = (y \to succ(x))$ and $\theta_2 = (y \to 0)$ are linear on $\{y\}$,
– the right-hand sides x and 0 are simple,
– the right-hand sides x and 0 are constructive if $s \neq pred(succ(x))$ and $s \neq pred(0)$, respectively,
– $Dec_v(s) = \{y\} \supseteq \emptyset = Dec_v(x)$ and $Dec_v(s) = \{y\} \supseteq \emptyset = Dec_v(0)$,
– $\mu_c(s) = \{0\} >_m \emptyset = \mu_c(x)$ and $\mu_c(s) = \{0\} >_m \emptyset = \mu_c(0)$.

Verification of conditions for the rules ρ_1 and ρ_3 is straightforward. The rule ρ_2 is considered in similar way. □

The termination property of AESs is given by the following theorem.

Theorem 5.3 AESs are terminating w.r.t. the innermost reduction strategy.

Proof Let s_i be the innermost reduction strategy. To prove the termination of AESs w.r.t. s_i, it is sufficient to build a well-founded partial order $\succ$ such that $s_i(R) \subseteq \succ$ for any AES R.

The required partial order $\succ$ is a multiset extension of $\succ'$ such that $u \succ' v$ iff $(\mu_a(u), \mu_v(u), \mu_c(u))$ is lexicographically bigger than $(\mu_a(v), \mu_v(v), \mu_c(v))$ with orders $>$, $>_m$, and $>_m$ on the first, the second, and the third elements of the tuple, respectively.

A multiset extension of a well-founded order and a lexicographical order on tuples of the same number of elements with well-founded orders on the elements are well-founded. Therefore $\succ$ is well-founded. The check of $s_i \subseteq \succ$ is reduced to the routine case analysis and, therefore, is dropped.

□

Let us show that the use of the reduction strategy for AESs is a necessary condition of termination of AESs.

Example 5.4 Let R be the FRS that consists of the following rules:
$\rho_1 : (\{f(c(z)) \to z\}, f(x))$
$\rho_2 : (\{f(c(z)) \to z\}, f(c(z)))$
$\rho_3 : (\{f(d(x,y)) \to d(f(y), f(x))\}, f(d(x,y)))$
$\rho_4 : (\{h(c(z)) \to d(h(z), z)\}, h(c(z)))$
$\rho_5 : (\{h(d(x,y)) \to x\}, h(d(x,y)))$
$\rho_6 : (\{h(d(x,y)) \to x\}, h(z))$

The FRS R is an AES with analyzers f and h. However the following chain of reductions

$$\begin{array}{rcrl} d(f(x), f(h(x))) & \to_{\rho_1} & d(z, f(h(c(z)))) & \to_{\rho_4} \\ d(z, f(d(h(z), z))) & \to_{\rho_3} & d(z, d(f(z), f(h(z)))) & \to_{\rho_1} \ \ldots \end{array}$$

is infinite. In the chain the rule ρ_3 is applied to the redex $f(d(h(z), z))$ of R that contains the redex $h(z)$ as its proper subterm. □

6. Analyzer elimination systems with argument status

Consider the generalization of AESs (AESs with argument status, or SAES, for short) that also has the termination property w.r.t. the innermost reduction strategy.

Example 6.1 Let R be the constructor system $(\{f(g(g(x)),y) \to f(x,f(a,y))\}, f(g(g(x)),y))$ with the analyzer f and constructors g and a. It is obvious that R is terminating w.r.t. the innermost reduction strategy. However, R is not an AES, since the term $f(x,f(a,y))$ is nested. □

This problem can be decided if we take into account only certain arguments of function symbols. In the example the term $f(x,f(a,y))$ is simple if we take into account only the first argument of f. Let us introduce the corresponding definition.

Definition 6.2 A map $arg : \mathcal{A} \to \mathcal{P}(N)$ is called an argument status if $arg(f) \subseteq \{1,...,\mathcal{A}r(f)\}$.

The definitions of a nested expression, a call, a constructor measure and a variable measure are modified so as to take into account an argument status.

Definition 6.3 An expression u is nested w.r.t. arg if there is a position $q \in \mathcal{P}(u)$ such that $u_q = f(t_1,...,t_n)$, $f \in \mathcal{D}$ and the term t_i is not constructive for some $i \in arg(f)$. An expression u is called simple if u is not nested w.r.t. arg.

$C_m(u)$ denotes the multiset of all calls which are simple w.r.t. arg and occur in expression u.

Definition 6.4 A map $\mu_c^{arg} : \mathcal{E}(\Sigma) \to \mathcal{FM}(N)$ is a constructor measure w.r.t. arg if

$$\mu_c^{arg}(u) = \{\sum_{i \in arg(f)} |t_i| \mid f(t_1,...,t_n) \in C_m(u)\}.$$

Definition 6.5 A map $Dec_v^{arg} : \mathcal{E}(\Sigma) \to \mathcal{FM}(X)$ is a variable decomposition w.r.t. arg if

$$Dec_v^{arg}(u) = \cup_{f(t_1,...,t_n) \in C_m(u)} \cup_{i \in arg(f)} \mathcal{MV}ar(t_i).$$

Definition 6.6 A constructor FRS R is an analyzer elimination system with an argument status arg, if any simple call is a redex of R and for each rule $(\{p_i | l_i \to r_i \mid i \in I\}, s) \in R$ and for each $i \in I$ the following properties hold:
– θ_i is linear on $\mathcal{V}ar(s)$ and constructive,
– p_i and r_i are simple w.r.t. arg,
– p_i and r_i are constructive if $l_i \neq s$,
– $Dec_v^{arg}(s) \supseteq Dec_v^{arg}(p_i) \cup Dec_v^{arg}(r_i)$,
– $\mu_c^{arg}(s) >_m \mu_c^{arg}(p_i)$ and $\mu_c^{arg}(s) >_m \mu_c^{arg}(r_i)$,
– $t_{ij} = t_j$ for each $j \in arg(f)$ where $l_i = f(t_{i1}, \ldots, t_{in})$ and $s = f(t_1, \ldots, t_n)$.

AESs are a special case of SAESs. It is obtained by taking $arg(f) = \{1, \ldots, \mathcal{A}r(f)\}$ for each $f \in \mathcal{A}$.

The termination property of SAESs is given by the following theorem.

Theorem 6.7 SAESs are terminating w.r.t. the innermost reduction strategy.

Proof Let s_i be the innermost reduction strategy. To prove the termination of AESs w.r.t. s_i it is sufficient to build a well-founded partial order $\succ$ such that $s_i(R) \subseteq \succ$ for any AES R.

Let $C_e^{arg}(u)$ denote the multiset of all nested calls of $u \in \mathcal{E}(\Sigma)$ w.r.t. arg, μ_a^{arg} and μ_v^{arg} denote the maps such that $\mu_a^{arg}(u) = |C_e(u)^{arg}|$ and $\mu_v^{arg}(u) = \{\mathcal{O}(x, Dec_v^{arg}(u)) \mid \mathcal{O}(x, Dec_v^{arg}(u)) \neq 0\}$, respectively.

The required partial order $\succ$ is a multiset extension of $\succ'$ such that $u \succ' v$ iff

$$(\mu_a^{arg}(u), \mu_v^{arg}(u), \mu_c^{arg}(u))$$

is lexicographically bigger than

$$(\mu_a^{arg}(v), \mu_v^{arg}(v), \mu_c^{arg}(v))$$

with orders $>$, $>_m$, and $>_m$ on the first, the second, and the third elements of the tuple, respectively.

A multiset extension of a well-founded order and a lexicographical order on tuples of the same number of elements with well-founded orders on the elements are well-founded. Therefore $\succ$ is well-founded. The check of $s_i \subseteq \succ$ is reduced to the routine case analysis and, therefore, is dropped.

□

7. Analyzer elimination systems with a substitution base

The third class of constructive systems that are terminating w.r.t. the innermost reduction strategy are analyzer elimination systems with a substitution base. These systems are built in the following way:

1. A class of expressions called a substitution base is chosen. The expression analysis does not take into account expressions of the substitution base. An example of such analysis of the expression structure is application of functions $\mathcal{V}ar_E$ $||_E$. These functions do not take into account subexpressions that occur in E.

2. Narrowing is restricted by MGUs that replace the variables by expressions of the substitution base.

Definition 7.1 Let $E \subseteq \mathcal{E}$, $S \subseteq \mathcal{S}$. E is called closed w.r.t. S, if $u\sigma \in E$ for every $u \in E$ and $\sigma \in S$. E is complete w.r.t. S, if the sets E, $\mathcal{E} \setminus (E \cup \mathcal{V})$ are closed w.r.t. S. E is closed (complete) w.r.t. substitutions, if E is closed (complete) w.r.t. $\mathcal{S}$. A map μ_c^E is a constructor measure w.r.t. E if $\mu_c^E(u) = \{|t|_E | t \in C_m(u)\}$. A map Dec_v^E is a variable decomposition w.r.t. E, if $Dec_v^E(u) = \{\mathcal{MV}ar_E(t) | t \in C_m(u)\}$.

Example 7.2 Let $E = \{h(x)\sigma | \sigma \in \mathcal{S}\}$, $E' = \{h(x)\sigma | \sigma$ is a constructive substitution$\}$. Then E is complete w.r.t. substitutions, E' is complete w.r.t. constructive substitutions. □

Definition 7.3 Let E be a set of expressions complete w.r.t. constructive substitutions. A constructor FRS R is an analyzer elimination system with the substitution base E, if any simple call is a redex of R and for each rule $(\{p_i | l_i \to r_i \mid i \in I\}, s) \in R$ and for each $i \in I$ the following properties hold:
– $\{x\theta_i \mid x \in \mathcal{V}ar(s)\} \subseteq E$,
– p_i and r_i are simple,
– p_i and r_i are constructive if $l_i \neq s$,
– $Dec_v^E(s) \supseteq Dec_v^E(p_i) \cup Dec_v^E(r_i)$,
– $\mu_c^E(s) >_m \mu_c^E(p_i)$ and $\mu_c^E(s) >_m \mu_c^E(r_i)$.

Example 7.4 Let R be a constructive FRS with the analyzer f that consists of the following rules:
– $\rho_1 : (\{f(h(x)) \to x\}, f(y))$,
– $\rho_2 : f(h(x)) \to x$,
– $\rho_3 : f(g(x)) \to g(f(x))$.

Let $\mathcal{S}_c$ be a set of all constructive substitutions. Show that R is an AES with the substitution base $E = \{h(x)\sigma | \sigma \in \mathcal{S}_c\}$.

Conditions from the definition of AES with the substitution base for the rule ρ_1 with the sample $s = f(y)$ take the form:
– $true$ and x are simple,
– $\{y(x \to z, y \to h(z))\} = \{h(z)\} \subseteq E$,
– $true$ and x are constructive if $f(h(x)) \neq s$,
– $Dec_v^E(s) \supseteq Dec_v^E(p_i) \cup Dec_v^E(r_i)$,
– $\mu_c^E(s) >_m \mu_c^E(p_i)$ and $\mu_c^E(s) >_m \mu_c^E(r_i)$.

Conditions from the definition of AES with the substitution base for the rule ρ_2 with the sample $s = f(h(x))$ take the form:
– $true$ and x are simple,
– $\{x(x \to z)\} = \{z\} \subseteq E$,
– $true$ and x are constructive if $f(h(x)) \neq s$,
– $Dec_v^E(s) = \emptyset \supseteq \emptyset \cup \emptyset$,
– $\{1\} >_m \emptyset$ and $\{1\} >_m \emptyset$.

Conditions from the definition of AES with the substitution base for the rule ρ_3 with the sample $s = f(g(x))$ take the form:
– $true$ and $g(f(x))$ are simple,
– $\{x(x \to z)\} = \{z\} \subseteq E$,
– $true$ and $g(f(x))$ are constructive if $f(g(x)) \neq s$,
– $\{x\} =\supseteq_m \emptyset \cup \{x\}$,
– $\{3\} >_m \emptyset$ and $\{3\} >_m \{2\}$.

Verification of conditions for the rules ρ_1, ρ_2, and ρ_3 is straightforward. The form of the samples of the rules of R guarantees that any simple call is a redex of R. So R is an AES with a substitution base. □

AESs is a special case of AESs with a substitution base. It is obtained by taking $E = \emptyset$.

The termination property of AESs with a substitution base is given by the following theorem.

Theorem 7.5 Analyzer elimination systems with a substitution base is terminating w.r.t. the innermost reduction strategy.

Proof Let s_i be the innermost reduction strategy. To prove the termination of AESs with a substitution base w.r.t. s_i, it is sufficient to build a well-founded partial order $\succ$ such that $s_i(R) \subseteq \succ$ for any AES R.

Let R be an AES with the substitution base E. The required partial order $\succ$ is a multiset extension of $\succ'$ such that $u \succ' v$ iff $(\mu_a(u), \mu_c^E(u))$ is lexicographically bigger than $(\mu_a(v), \mu_c^E(v))$ with orders $>$ and $>_m$ on the first and the second elements of the tuple, respectively.

A multiset extension of a well-founded order and a lexicographical order on tuples of the same number of elements with well-founded orders on the elements are well-founded. Therefore $\succ$ is well-founded. The check of $s_i \subseteq \succ$ is reduced to the routine case analysis and, therefore, is dropped.

□

Conclusion

The paper presents a method of application of narrowing to formula simplification. The method includes the following new features:
– the concept of a multi-branch narrowing that allows case analysis to be built;
– formula rewriting systems formalizing narrowing strategy that preserves satisfiability;
– the method for proving termination of formula rewriting systems.

Proving correctness conditions that appear in program verification is an important area of application of the method. Proving correctness conditions is performed in the interactive mode in most verification systems. Our method allows the proof of correctness conditions to be done automatically. The program verification system SPECTRUM [15, 16] has been developed for which a new prover based on the method is being designed. In particular, experiments on automatic verification of programs of array sorting and file sorting have been performed in the framework of the project SPECTRUM by this method.

Some details of the theory of formula rewriting systems and their application to problem-oriented verification has been considered in [1, 2, 3, 4]. The methods of proving termination of formula rewriting systems have been studied in [2]. In particular, other classes of terminating formula rewriting systems have been proposed [1, 2].

References

[1] I. Anureev, *A method for simplification procedure design based on formula rewriting systems*, Joint Bulletin of NCC & IIS, Ser.: Comput. Sci., **8**, 1998, 1–18.

[2] I. Anureev, *Formula rewriting systems and their application to automated program verification*, PhD thesis, A.P. Ershov Institute of Informatics Systems, 1998 (in Russian).

[3] I. Anureev, *Formula rewriting systems and their application to automated program verification*, Joint Bulletin of NCC & IIS, Ser.: Comput. Sci., **10**, 1999, 1–6.

[4] I. Anureev, *A data structure elimination method*, Programming, **4**, 1999, 5–15 (in Russian).

[5] J. Christian, *Some termination criteria for narrowing and E-narrowing*, Lect. Notes Comput. Sci., **607**, 1992, 582–588.

[6] N. Dershowitz, J.-P. Jouannaud, *Rewrite systems*, Handbook of Theoretical Computer Science, **B(6)**, 1990, 243–320.

[7] N. Dershowitz, G. Sivakumar, *Goal-directed equation solving*, Proc. 7th National Conf. on Artificial Intelligence, St. Paul, Minn. (USA), 1988, 166–170.

[8] M. Fay, *First order unification in equational theories*, Proc. 4th Workshop on Automated Deduction, Austin, Tex. (USA), 1979, 161–167.

[9] J.-M. Hullot, *Canonical forms and unification*, Lect. Notes Comput. Sci., **87**, 1980, 318–334.

[10] J.-P. Jouannaud, C. Kirchner, H. Kirchner, *Incremental construction of unification algorithms in equational theories*, Lect. Notes Comput. Sci., **154**, 1983, 361–373.

[11] H. Kirchner, *Term rewriting*, Algebraic foundations of systems specification. IFIP state-of-the-art reports, 1999, 273–320.

[12] S. Krischer, A. Bockmayr, *Detecting redundant narrowing derivations by the lse-sl reducibility test*, Lect. Notes Comput. Sci., **488**, 1991, 74–85.

[13] J.-W. Klop, *Term rewriting systems*, Handbook of Logic in Comput. Sci., **2**, 1993, 1–116.

[14] S. Limet, P. Rety, *Conditional directed narrowing*, Lect. Notes Comput. Sci., **1101**, 1996, 637–640.

[15] V.A. Nepomniaschy, A.A. Sulimov, *Problem-oriented means of program specification and verification in project SPECTRUM*, Lect. Notes Comput. Sci., **722**, 1993, 374–378.

[16] V.A. Nepomniaschy, A.A. Sulimov, *Problem-oriented verification system and its application to linear algebra programs*, Theor. Comput. Sci., **119**, 1993, 173–185.

[17] W. Nutt, P. Rety, G. Smolka, *Basic narrowing revisited* J. of Symbolic Computation, **7(3-4)**, 1989, 295–318.

[18] P. Rety, *Improving basic narrowing*, Lect. Notes Comput. Sci., **256**, 1987, 228–241.

[19] P. Rety, C. Kirchner, P. Lescanne, Lect. Notes Comput. Sci., **202**, 1985, 141–157.

[20] J.-H. You, *Enumerating outer narrowing derivations for constructor-based term rewriting systems*, J. of Symbolic Computation, **7(3-4)**, 1989, 319–342.