A Note on First Order Unification

Rodrigo Readi Nasser*
Universität München

March 27, 2000

A connection between deduction in first order logic, term unification and completion algorithms for term reduction systems is given. Unification is expressed as solvability of systems of equations between applicative terms containing "unknowns" to be determined. The concept of term unification is extended for enabling the search for unknown functions, properties characteristic of term unification remain valid.

§1
Preliminary notions

We consider in this article a language L for the *predicate calculus* whose only predicate symbol $=$ is the 2-ary one for the **equality** and whose function symbols are a 2-ary one ap for **application** together with a set of 0-ary symbols divided into **ground symbols** and **unknowns**. The unknowns correspond to the *function letters* of [2], page 263, while the ground symbols to its *function symbols*. We **remark**: *unknowns are function symbols of the language L that should not be confused with variables of L.* Usually, we use capital letters X, Y, Z, perhaps indexed, for denoting unknowns; the letters u, v, w for variables; other small letters and numbers for ground symbols.

The terms in such an L are **applicative terms**. The atomic formulae in L, all of the form $s = t$, are called **equations** between applicative terms, their (free) variables are to be seen as universal quantified. As [3] in page 16, we adopt the *infix notation* for (applicative) terms and the notational simplification by *dropping parenthesis* under the *convention of association to the left*: *Instead of writing* $\mathrm{ap}(s,t)$, *we just write* (st); *we also write* (t_0) *for a term* t_0, *and recursively,* $(t_0 \cdots t_{k-1} t_k)$ *for* $((t_0 \cdots t_{k-1}) t_k)$. If the t_0 in $(t_0 \cdots t_{k-1} t_k)$ is not a function symbol, then it can be substituted by an expression (st), where s is a smaller term, and the parenthesis can be dropped; this can be done recursively, hence: *every applicative term t can be uniquely expressed in the form $(t_0 t_1 \cdots t_n)$, where t_0 is either a function symbol or a variable and $t_1, \ldots, t_n$ a possibly empty list of smaller terms.* The above t_0 is called here the **operator** of t and the $t_1, \ldots, t_n$ its **arguments**, the number n (perhaps 0) is the **number of arguments** of t. By allowing only an application operator and non 0-ary function symbols, we are not making an essential restriction: *an n-ary function symbol f can be expressed with a 0-ary symbol and n copies of the application operator.*

Now, we can classify equations:

Definition 1 (Classification of equations) *An equation $s = t$ is a* **reductor**, *if the operator of s is an unknown; it is a* **reversed reductor**, *if the operator of t is an unknown; it*

*Lehrstuhl für Mathematische Logik, Mathematisches Institut, Zimmer 417, Theresien-Str. 39, 80333-München. readi@rz.mathematik.uni-muenchen.de

is **bidirectional**, *if it is both, a reductor and a reversed reductor. An equation is* **directed**, *if it is either a reductor or a reversed reductor (or both, bidirectional). An equation is* **decomposable** *if it is not directed and if its both members have the same operator and the same number of arguments, namely, if it is of the form* $(ft_1 \cdots t_n) = (fs_1 \cdots s_n)$, *where* f *is either a ground symbol or a variable, but not an unknown. An equation is a* **contradiction** *if it is neither directed nor decomposable. An equation* $s = t$ *is an* **operators clash** *if it is not directed and the operators of* s *and* t *are different, it is an* **arguments clash** *if it is not directed and the numbers of arguments of* s *and* t *differ.*

We can easily see: *An equation is either directed or decomposable or a contradiction, it belongs to exactly one of these three classes. An equation is a contradiction if and only if it is either an operators clash or an arguments clash.*

Since we are not only interested in the operational aspect of unification, but also in the logical aspect, we need some axioms for equality.

Definition 2 (Equality axioms) *In the above language* L *for equations of applicative terms, we consider theories containing the following set* A_L *of postulates, called* **equality axioms***:*

(1) *The* **axioms of equivalence**

$$\forall v(v = v),$$
$$\forall u \forall v(u = v \supset v = u),$$
$$\forall u \forall v \forall w((u = v \wedge v = w) \supset u = w);$$

(2) *The* **axiom of compatibility with application**

$$\forall u_1 \forall u_2 \forall v_1 \forall v_2 (u_1 = v_1 \supset (u_2 = v_2 \supset (u_1 u_2) = (v_1 v_2)));$$

(3) *The* **axioms of decomposition**

$$\forall u_1 \cdots \forall u_n \forall v_1 \cdots \forall v_n (fu_1 \cdots u_n) = (fv_1 \cdots v_n) \supset u_i = v_i$$

for every ground symbol f, $n \geq 0$ *and* $0 \leq i \leq n$*; as well as*

$$\forall v_1 \cdots \forall v_m ((\forall w(wr_1 \cdots r_n) = (ws_1 \cdots s_n)) \supset (\forall w r_i = s_i)),$$

where r_k *and* s_k *are terms whose variables are among* $v_1, \ldots, v_n, w$*;*

(4) *The* **contradiction axioms**

$$(\forall \bar{v} s = t) \supset \Box,$$

where $s = t$ *is a contradiction as in definition 1,* $\forall \bar{v}$ *a block of universal quantifiers binding all variables in* $s = t$ *and* $\Box$ *a logical symbol for contradiction (instead of negation symbol).*

The decomposition and contradiction axioms is what postulates the difference between unknowns and ground symbols. The idea is that the *universe* be terms built with ground symbols and that the unknowns represent functions on this universe, perhaps functions of functions, or more complicated domains. Of course, we need postulates determining the objects represented by unknowns, these are the **proper axioms of the theory**, they are a set S of additional equations.

Definition 3 (Equational derivability, consistency) *We say that* $S \models a$ *holds, where* a *is either an equation or* $\Box$*, if* a *is minimal-intuitionistic (i.e. without postulate* $8°$ *of page 82 of [2]) derivable in the predicate calculus with the above equality postulates* A_L *and with* S *as proper postulates of the theory. For another set of equations* T*, we write* $S \models T$ *if* $S \models a$ *for each* $a \in T$*, and* $S \equiv T$ *if both,* $S \models T$ *and* $T \models S$*. We say that* S *is* **consistent** *if* $S \models \Box$ *does not hold.*

The equality postulates A_L, as well as any equation, are *generalized horn formulae*, as introduced in [7], page 154, from there we have: *a is classical (intuitionistic) derivable from* S *if and only if either* $S \models a$ *or* $S \models \Box$.

The following definition, together with lemma 1, reformulates the concept of $S \models r$. This is the way we deal with derivability in the whole article.

Definition 4 (Calculus for equational derivability) *We define the following schemata for deriving equations from equations.*

(1) *An* **instance** *of an equation* $s = t$ *is an equation* $s^\alpha = t^\alpha$*, where* α *is a "v-substitution", namely, an assignation of terms to variables. The* **instantiation schema** *generates rules of the form*

$$\textbf{ins}: \frac{s = t}{s^\alpha = t^\alpha},$$

where α *is a v-substitution.*

(2) *We define, for each equality postulate in definition 2, a corresponding* **schema***:*

$$\textbf{ref}: \frac{}{s = s}, \quad \textbf{sym}: \frac{r = s}{s = r}, \qquad \textbf{trans}: \frac{r = s,\ s = t}{r = t},$$

$$\textbf{cons}: \frac{r_1 = s_1,\ r_2 = s_2}{(r_1 r_2) = (s_1 s_2)},$$

$$\textbf{dec}: \frac{(f r_1 \cdots r_n) = (f s_1 \cdots s_n)}{r_i = s_i}, \quad \textbf{sep}: \frac{(w r_1 \cdots r_n) = (w s_1 \cdots s_n)}{(r_i = s_i)^{w \leftarrow t}},$$

where r*,* s*,* t*,* r_i *and* s_i *represent terms,* f *in dec a ground symbol,* w *in sep a variable,* $(r_i = s_i)^{w \leftarrow t}$ *in sep the instance of* $r_i = s_i$ *obtained by substituting the variable* w *by the term* t*,* i *in dec and sep any subindex between* 1 *and* n*.*

We have:

Lemma 1 (Correctness of the calculus for equational derivability) *The relation* $S \models s = t$ *holds if and only if the equation* $s = t$ *is derivable from instances of the equations in* S *with the schemata ref, trans, cons, dec, sep corresponding to the equality postulates. The set* S *is contradictory if and only if* $S \models s = t$ *holds for a contradiction* $s = t$ *(cf. definition 1).*

The literature, for example [3], page 41, is full of calculi similar to the above one. Most of the readers will find the above lemma trivial, some as an easy consequence of the results in [5], and few scrupulous ones would begin to calculate with λ-terms. Well, all this is right. We **remark**: *Lemma 1 remains valid if an equality postulate in definition 2 and its corresponding schema in definition 4 are deleted.*

We can see a set of equations S, specially of reductors, as a term rewriting system (TRS) (see for example [3]). The following lemma can be seen as an alternative definition of term reduction.

Lemma 2 (Term reduction as derivability) *The term s is reducible to t ($s \to t$) with S if and only if $s = t$ is derivable from instances of equations in S with ref, cons and trans.*

By structural induction on such derivations with ref, cons and trans, we can easily prove the following intuitive result.

Lemma 3 (Form preservation) *If S is a set of reductors, f either a ground symbol or a variable, and $s_1, \ldots, s_m$, t terms such that $(fs_1 \cdots s_m) \to t$ with S holds, then t is of the form $(fs'_1 \cdots s'_m)$, so that $s_i \to s'_i$ also holds for each i.*

§2

Reformulation and extension of the concept of first order unification

Unknowns represent objects to be determined. A reductor $(Xt_1 \cdots t_n) = t$, where X is an unknown and t, $t_1, \ldots, t_n$ terms, perhaps containing unknowns and variables, determine the unknown X applied to the arguments $t_1, \ldots, t_n$.

Definition 5 (Equations systems, substitutions, solution)

(1) *An (equations)* **system** *is a set of equations. A* **substitution** *S is a confluent set of reductors.*

(2) *A substitution S is a* **solution** *of an equation $s = t$ (or* **unifies** *s and t) if s and t can be reduced to a common r with S as TRS. A substitution S is a* **solution** *of a system T if it is a solution of all its equations.*

Now, we can postulate our first theorem connecting solvability with derivability:

Theorem 1 (Solvability as derivability) *The substitution S is a solution of $s = t$ if and only if $S \models s = t$, it is a solution of a system T if and only if $S \models T$.*

Proof. If S is a solution of $s = t$, then, using lemma 2, $S \models s = r$ and $S \models t = r$ hold, where r is the common reduct of s and t, and hence also $S \models s = t$ holds. For proving the other direction, we consider the set of all equations for which S is a solution. This set contains S, is trivially closed under ins, ref, sym and cons, is closed under trans because of the confluence of S, is closed under dec and sep because of lemma 3: it contains all $s = t$ for which $S \models s = t$. **QED**.

In the context of conventional term unification [4], we say that a substitution α is more general than a substitution β if there is a γ such that $\beta \equiv \alpha\gamma$. If α and β are idempotent substitutions, as unifiers normally are, and if they are expressed as sets of equations of the form $X = t$, then the existence of such γ is equivalent to $\beta \models \alpha$. For our extended concept of substitution, we cannot define the composition of two substitutions, because TRS are not necessarily terminating, but the concept of "more general" can be paraphrased as:

Definition 6 (Generality as derivability)

(1) *A substitution S_1 is* **more general** *than a substitution S_2 if $S_2 \models S_1$.*

(2) *A substitution S is a* **most general** *solution of a system T if $S \models T$ (it is a solution) and $T \models S$ (namely $S \equiv T$).*

A trivial consequence of the above definition, of the above theorem and of derivability properties is what we expect for most general unifiers:

Observation 1 *If S is a most general solution of T, then S' is a solution of T if and only if S is more general than S'. Two most general solutions S and S' satisfy $S \equiv S'$.*

Definition 7 (Decomposition and orientation)

(1) *A set S is* **semi-directed** *if each equation in it is either a reductor or a contradiction.*

(2) *If T is a set of equations, then* **dec**(T) *is the set of non decomposable equations obtained by recursively substituting each equation of the form $(zr_1 \cdots r_n) = (zs_1 \cdots s_n)$, where z is either a ground symbol or a variable, by the smaller equations $r_i = s_i$. When $n \equiv 0$, this means deleting the equation $z = z$.*

(3) *If T consist of non decomposable equations, namely, of directed equations and contradictions, then an* **orientation** *of T is a semi-directed set obtained by substituting some (perhaps none) equations of the form $s = t$ by their symmetries $t = s$.*

Observation 2

(1) *$T \equiv dec(T)$ holds: Each equation in $dec(T)$ is derivable from the ones in T with the schemata dec and sep. Each equation of T is derivable from the ones in $dec(T)$ with the schemata ref and cons.*

(2) *There is not necessarily a unique orientation, because bidirectional equations and contradictions can be arbitrarily inverted. A set is equivalent to an orientation of it, because of the schema sym.*

In the so called "higher order unification", we can find unifiers, but normally we miss the existence of most general ones. This does not happen in our context. Our second main theorem connects solvability with consistence:

Theorem 2 (Solvability as consistence) *For a system T, following statements are equivalent: (1) T is consistent, (2) T has a most general solution, (3) T has a solution.*

Proof. Let T be consistent and S the set of reductors derivable from it. We prove that if $T \models s = t$, then s and t has a common reduct with S. This implies that S is confluent and equivalent to T, and hence a most general solution of T. By (1) of observation 2, $T \models \text{dec}(\{s = t\})$ holds, and hence, $\text{dec}(\{s = t\})$ contains no contradiction: S contains an (every) orientation of $\text{dec}(\{s = t\})$. The set of equations whose members has a common reduct with S contains S, and hence an orientation of $\text{dec}(\{s = t\})$, contains $\text{dec}(\{s = t\})$ because it is closed by sym, contains $s = t$ because it is closed by ref and cons. Obviously item (2) implies item (3). If T has a solution S, then $S \models T$ and $S \models s = t$ for each $s = t$ derivable from T; hence $s = t$ cannot be a contradiction because of lemma 3. **QED.**— Since substitutions are solutions of themselves:

Corollary 1 *A substitution is consistent.*

Our third main theorem connects the unification algorithm with completion, it gives an idea of how to find a most general solution using results of the theory of TRS:

Theorem 3 (Unification as completion)

(1) A confluent semi-directed set is a substitution if it is consistent, contains a contradiction if it is inconsistent.

(2) Enumerating a confluent semi-directed system equivalent to a system T leads to a contradiction if T has no solution, or enumerates a most general solution of T if T has a solution.

Item (1) follows from the above corollary by reductio ad absurdum. A revision of the indirect proof leads to a constructive one. Item (2) follows from item (1).

§3

Epilogue

We extended the concept of term (or syntactic) unification for allowing the search of unknown functions, remaining in the framework of first order logic. Unifiability is now equivalent to consistence. The unification algorithm should be a modification of the completion algorithm for enumerating a confluent system that allows cancellation of ground symbols. A "confluence criterium", like the one with critical pairs, should indicate what equations are to be added to warranty confluence and could help to eventually stop enumeration. For example, orientations of $\mathrm{dec}(\{s = t\})$ for critical pairs (s, t) of all possible instances of equations could be added, but this is not practicable. The restriction to critical pairs taking most general unifiers of non-variable subterms of left sides of equations leads to a confluent system if the result is a terminating TRS, but in the general case it is not necessarily confluent. Unfortunately, known confluence criteria without demanding termination are complicated and demand strong restrictions to the TRS. The practical use of this extended concept of unification depends on future research for finding relevant families of equation systems having a practicable algorithm.

References

[1] Stephen Cole Kleene, Recursive Predicates and Quantifiers, Transactions of the American Mathematical Society, V 53, 1943.

[2] Stephen Cole Kleene, "Introduction to Metamathematics", Wolters-Noordhoff Publishing - Groningen, North-Holland Publishing Company - Amsterdam, First Published 1952, Sixth Reprint 1971.

[3] Jan Willem Klop, Term Rewriting Systems, In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors: Handbook of Logik in Computer Science, volume 2, chapter 1, pages 1–116. Oxford University Press, 1992.

[4] A. Martelli und U. Montanari, "An Efficient Unification Algorithm", ACM Transactions on Programming Languages and Systems 4, 2 (April 1982), pages 258-282.

[5] Dag Prawitz, "Natural Deduction", Almqvist & Wiksells, Uppsala, 1965.

[6] Readi Nasser, R. mj-Prolog, 1. Proof Theoretical Foundations, Internal Report, CIS, Uni-M"unchen, ftp://ftp.cis.uni-muenchen.de/pub/cis-berichte/CIS-Bericht-99-124.ps (Note lapse in lines 12–13 of §32).

[7] A. S. Troelstra, H. Schwichtenberg, Basic Proof Theory, Cambridge University Press, 1996.