# Strongly Reduced Systems of Equations

Philippe DEVIENNE
ICOT – Lille University

## 1 Introduction

Generally, the idempotent most general unifier is used for expressing the constraints of unification [Eder 85]. A more general syntactic object is the *system of equations*, that is, a set of equations of the form $t = t'$ where t and t' are trees.

However, the advantages of the systems of equations are first the memory size, secondly the efficency of the unification algorithm, and also that within the context of termination and complexity, the behaviour of resolution can be understood better through the constraints between the variables than from the final susbtitution of these variables.

The reduced systems of equations introduced by [Colmerauer 84] express this feeling and this is strongly linked with the representative notion introduced by [Huet 1976] and the directed acyclic graph and the directed graph are based on the same idea: the common informations are shared [Fages 83]. Similarly, within backtracting intelligent, an important aspect is to express the dependancy graph of variables.

The goal of this paper is to show that the shortest expression of a reduced system can be used to express as clearly as possible the constraints generated by a system of equations, however also to understand the convergent, invariant or periodic phenomenons for some recursive Prolog programs.

## 2 Preliminaries

A system of equations is said to be solvable if there exists a grounding substitution, $\sigma$, which makes t and t' ground and equal for all equations, $t = t'$ of the system. Two systems, E1 and E2, are said to be equivalent, if they have the same grounding substitutions.

**Definition 1 ( Colmerauer 84 )** *An endless system is a system of equations in which every variable which occurs as a right hand side of an equation also occurs as the left hand side of an equation.*

**Definition 2 ( Colmerauer 84 )** *A system of equations is said to be reduced if the left hand sides of its equations are distinct variables,and it does not contain an endless subsystem.*

---

STRONGLY REDUCED SYSTEMS OF EQUATIONS.
Philippe Devienne - supported by INRIA grant
ICOT – Lille University, France.

Let t be a tree, E be a system of equations and rs be a reduced system, we will denote:

. $fsize(t \text{ or } E)$ = *Number of its function occurrences*
. $depth(x) = 1$ when x is a variable or a constant
$depth(f(t_1, \cdots, t_n)) = 1 + max(depth(t_i))$
. $domain(rs)$ is the set of variables occurring in the left hand side of rs.

### Reduction algorithm

The first stage :

| | |
|---|---|
| $f(t_1, \cdot\cdot, t_n) = f(t'_1, \cdot\cdot, t'_n)$ | replace by $t_1 = t'_1, \cdot\cdot, t_n = t'_n$ |
| $f(t_1, \cdot\cdot, t_n) = g(t'_1, \cdot\cdot, t'_n)$ | halt with failure |
| $U = U$ | delete the equation |
| $t = U \ (t \notin Var)$ | replace by $U = t$ |
| $U = V$ $(V \neq U)$ | replace U by V in every other equation. |
| $U = t\ ,\ U = t'$ $(depth(t) \leq depth(t'))$ | replace (U = t') by (t = t') |

The second stage is: there is no endless subsystem.

**Theorem 1** *The reduction algorithm applied to a set of equations E will return an equivalent set of equations in reduced form if and only if E is solvable. It will return failure otherwise.*

In the [Lassez, Maher, Marriott] algorithm which defines the most general unifier of a system, the two last transformations are replaced by the following one:

| | |
|---|---|
| $U = t$ $(t \neq U)$ | if $U \in t$ then halt with failure, otherwise replace U by t in every other equation. |

**Theorem 2** *Any reduced system of equations is solvable and its idempotent most general unifier can be defined through the function, representative:*

$$repr(\ f(t_1, \cdots, t_n)\ ) = f(repr(t_1), \cdots, repr(t_n))$$
$$repr(U) = repr(t) \ \ if\ \exists (U = t) \in rs$$
$$repr(U) = U \ \ otherwise.$$

*A most general unifier of the reduced system, rs, is:*

$$\{U \leftarrow repr(U) \ / \ \forall\, U \in domain(rs)\ \}.$$

## 3 Strongly reduced systems

In some cases, it is quite important to express the most clearly as possible the constraints between the variables.

**Definition 3** *A reduced system, rs, is said to be strongly reduced if:*
. $\forall\,(U = t),\ (V = t') \in rs$, *if* $t' \notin Var$ *then* $t'$ *is not a sub-tree of* $t$.
. $\forall\,(U = V) \in rs$, *the variable* $U$ *does not occur in the other equations.*

Strong reduction algorithm

| | |
|---|---|
| $U = V$ | replace U by V in the |
| $(U \neq V)$ | other equations, |
| $(U = t)\ ,\ (V = t')$ | if $t' \in t$, then replace t' by V |
| $(t, t' \notin Var)$ | V in t. |

**Theorem 3** *The strong reduction algorithm applied to a reduced system of equations, rs, returns an equivalent set of equations in strongly reduced form.*

Hence, any system of equations has a strongly reduced form iff it is solvable.

**Definition 4** *Let* $\sigma$ *be a susbtitution and* $s$ *be a system of equations, then we write* $\sigma(s)$ *the system,* $s$*, where all the terms* $t$ *are been replaced by* $\sigma(t)$.

**Definition 5** *A susbtitution,* $\sigma$*, is said to be with respect to a congruence,* $\Re$*, if any binding of* $\sigma$ *is in the* $(U \leftarrow V)$ *form where* $U$ *and* $V$ *are equivalent in* $\Re$.

**Definition 6** $\Re_{srs}$ *is the congruence on the variables defined from the equalities of variables* $(U = V)$ *in the strongly reduced system, srs.*

**Theorem 4** *Let srs be a strongly reduced system and* $\sigma$ *be a most general unifier of* $E$*, then two variables,* $U$ *and* $V$*, are equivalent in* $\Re_{srs}$*, iff* $\sigma(U) = \sigma(V)$.

This means that all the constraints of equality between the variables are explicit in the strongly reduced system, and all the equivalent strongly reduced systems have the same congruence.

**Theorem 5** *Two strongly reduced systems,* $srs_1$ *and* $srs_2$*, are equivalent iff they have the same congruence,* $\Re$ *and they are equal with a renaming susbtitution with respect to* $\Re$.

**Corollary 1** *Let* $s$ *be a solvable system and srs be its strongly reduced form, then:* $fsize(srs) \leq fsize(s)$

Therefore, the strongly reduced form is the shortest expression of a solvable system of equations.

**Theorem 6** *An equivalent definition of the strongly reduced systems is a congruence,* $\Re$*, in Var and a mapping function from* $Var/\Re$ *to* $M(F,V)/R$*, that is, the set of finite trees modulo of* $\Re$.

Within this definition, there is one and only one strongly reduced form of any solvable system of equations.

To show some invariant phenomenons during the resolution, an interesting notion is the orthogonal system.

**Definition 7** *Two systems of equations,* $E_1$ *and* $E_2$*, are said to be orthogonal if there exist two reduced systems,* $rs_1$ *and* $rs_2$*, respectively equivalent to* $E_1$ *and* $E_2$ *such that the union of* $rs_1$ *and* $rs_2$ *is also reduced.*

Intuitively, two systems are orthogonal if their union does not produce others informations about the free variables, that is, all the informations were already explicitly written in one of these systems. However, generally, the reduced systems which verify the orthogonal condition are not strongly reduced.

**Theorem 7** *Two systems,* $E_1$ *and* $E_2$*, are orthogonal, iff there exist strongly reduced systems,* $srs_1$ *and* $srs_2$*, respectively equivalent to* $E_1$ *and* $E_2$ *such that* $E_1 \cup E_2$ *is equivalent to* $srs_1 \oplus srs_2$*:*

$$rs_1 \oplus rs_2 \; = rs_1 \cup \{\, (U = t) \in rs_2 \;/\; \nexists\,(U = t') \in rs_1 \,\}$$

**Theorem 8** *Let* $E_1$ *and* $E_2$ *be two non orthogonal systems, and srs be the strongly reduced form of* $E_1 \cup E_2$*, then*

$$fsize(srs) \; < \; fsize(E_1) + fsize(E_2)$$

If some new informations are produced from the union of $rsr_1$ and $srs_2$, then at least one redundancy appears in this union. In others words, the increase of domain is strongly linked with the decrease of the fsize of system.

Using the new notions and the weighted graph properties, it is possible to prove the convergence about some recursive Prolog programs.

**Theorem 9** *Let* $\pi_1$ *be a Prolog program composed of linear facts and one clause where the body has only one litteral, then for any linear goal, the halting problem and the existence of solutions is decidable and their complexities are linear functions from the goal depth.*

Moreover, let $\pi_2$ be a Prolog program composed of facts and one clause with one recursive litteral and one fact litteral in the body part, then for this slightly more complex structure, the existence of solutions is undecidable.

## References


A. Colmerauer, *Equations and Inequations on Finite and Infinite Trees*, FGCS'84 Proceedings, Nov. 1984., 85-99.

P. Devienne, P. Lebegue, *Weighted graphs, a tool for logic programming*, CAAP'86, pp 100-111.

E. Eder, *Properties of substitutions and unifications*, 1985, Journal of Symb. Comp., 1, 31-46.

F. Fages, *Notes sur l'unification des termes du premier ordre finis ou infinis*, INRIA, France

G. Huet, *Resolution d'equations dans les langages d'ordre 1, 2, ⋯ ,Ω*, 1976, These de doctorat d'etat, Universite Paris VI.

J.L. Lassez, M.J. Maher, K. Marriot, *Unification Revisited*, 1987, Workshop on Logic and Data Bases, J. Minker.