



HAL
open science

Algorithms for equivalence and reduction to minimal form for a class of simple recursive equations

Bruno Courcelle, Gilles Kahn, Jean Vuillemin

► **To cite this version:**

Bruno Courcelle, Gilles Kahn, Jean Vuillemin. Algorithms for equivalence and reduction to minimal form for a class of simple recursive equations. Yves Bertot; Gérard Huet; Jean-Jacques Lévy; Gordon Plotkin. From Semantics to Computer Science Essays in Honour of Gilles Kahn, Cambridge University Press 1974, From Semantics to Computer Science Essays in Honour of Gilles Kahn, 9780521518253. 10.1017/CBO9780511770524.009 . hal-01239749

HAL Id: hal-01239749

<https://hal.science/hal-01239749>

Submitted on 15 Dec 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Algorithms for equivalence and reduction to minimal form for a class of simple recursive equations

Bruno Courcelle, Gilles Kahn, Jean Vuillemin
IRIA Laboria, Rocquencourt, France

Foreword

This document presents a translation for historical perspective of the paper: “B. Courcelle, G. Kahn, and J. Vuillemin. Algorithmes d’équivalence et de réduction à des expressions minimales dans une classe d’équations récursives simples, in, Jacques Loeckx, editor, *Automata, Languages and Programming*, volume 14 of *Lecture Notes in Computer Science*, pages 200–213. Springer Verlag, 1974”. This text is published with kind permission of Springer Science and Business Media. This text was translated from French to English by T. Veldhuizen of Waterloo University, Canada.

Abstract

In this paper, we describe an algorithm for deciding equivalence in a domain whose objects are defined by uninterpreted fixpoint equations. The algorithm is then applied to finding minimal representations of those objects.

8.1 Introduction

Many recent works, for example [4, 8, 9, 11] use the notion of fixpoint equation to express semantics of programming languages. We study here a “pure language of fixpoints” with uninterpreted function symbols, which omits in particular the conditional operator *if-then-else*.

In the study of fixpoint equations, of which a typical example is the equation $X = f(X, g(X))$, we ask certain questions, for example:

- is the equation $X = f(X, g(X))$ equivalent to the equation $Y = f(f(Y, g(Y)), g(Y))$?
- does there exist a simpler equation equivalent to

$$Z = g(g(Z))$$

or to the system

$$\begin{cases} X &= f(Y, g(X), Y) \\ Y &= g(X) \end{cases}$$

- can the variable X defined by the system

$$\begin{cases} X &= f(X, Y) \\ Y &= g(X, Y) \end{cases}$$

be defined by a single equation?

In the second section, we study *simple recursive equations*. We show the existence of *canonical forms* characterizing a class of equivalent equations. The canonical form minimizes the size of the equation in its equivalence class.

The third section, independent of the first (except where definitions of syntax and semantics are concerned), studies the same problems for *systems of recursive equations*, deriving a *notion of canonical form* that minimizes the *number of equations in the system*. This last problem is then addressed and resolved.

This work is motivated by a variety of questions such as the study of recursive datatype definitions in Algol 68 (C. Lewis and B. Rosen [6]), the formalization of equivalence proofs of parallel programs (G. Kahn [4]) and the study of decidable sub-theories of the theory of program schemas.

Other authors (J. Engelfriet [3], C. Pair [10], J. Kral [5]) have independently obtained related results in syntactically and semantically different frameworks.

8.2 Simple recursive equations

For clarity of exposition, we start by defining fixpoint equations in a single unknown.

8.2.1 Syntax

Terms are constructed from function symbols $\{F, G, H, \dots\}$ each having some arity, and from the variable symbol X by the rules.

- (i) Function symbols of arity 0 (or *constants*) and the symbol X are terms.
- (ii) If T_1, T_2, \dots, T_n are terms and F is a function symbol of arity n , then $F(T_1, T_2, \dots, T_n)$ is a term.

If T is a term, $X = T$ is a fixpoint equation.

In what follows, it is useful to define a partial order \leq on the set of terms by the following rules:

- (i) For all terms T , we have $X \leq T$.
- (ii) If $T_1 \leq T'_1, \dots, T_n \leq T'_n$ then $F(T_1, \dots, T_n) \leq F(T'_1, \dots, T'_n)$.

For other terms, $T_1 \leq T_2$ if and only if T_2 is the result of substituting in T_1 some terms for some occurrences of X .

Example 8.1

$$F(X, G(X, X)) \leq F(H(X), G(X, H(X)))$$

but $F(X, G(X, X)) \not\leq F(X, H(X))$.

When given two terms T and T' , we can define a lower bound $\Sigma(T, T')$ such that $\Sigma(T, T') \leq T$ and $\Sigma(T, T') \leq T'$ in the following manner:

$$\begin{aligned} \Sigma(T, X) &= \Sigma(X, T) = X \\ \Sigma(F(T_1, T_2, \dots, T_n), G(T'_1, T'_2, \dots, T'_m)) &= X \quad \text{if } F \neq G \\ \Sigma(F(T_1, T_2, \dots, T_n), F(T'_1, T'_2, \dots, T'_n)) &= F(\Sigma(T_1, T'_1), \dots, \Sigma(T_n, T'_n)) \end{aligned}$$

Example 8.2 $\Sigma(F(X, G(X)), F(G(X), H(X))) = F(X, X)$

Notation For a term T in which the letter X occurs m times, we denote by $T\{T_1, \dots, T_m\}$ the result of substituting the term T_i for the i^{th} occurrence of X in T , for each i in $[1, m]$.

Lemma 8.3 *For all terms T and T' there exist terms T_1, \dots, T_m and T'_1, \dots, T'_m such that:*

$$\begin{aligned} T &= \Sigma(T, T')\{T_1, \dots, T_m\} \\ T' &= \Sigma(T, T')\{T'_1, \dots, T'_m\} \end{aligned}$$

8.2.2 Semantics

(a) We interpret fixpoint equations in a *domain* \mathcal{D} which must satisfy the following requirements:

- (i) The set \mathcal{D} is provided with a partial order relation \subseteq . We write \equiv for the induced equivalence relation.
- (ii) There exists in \mathcal{D} a least element \perp .
- (iii) Every denumerable ascending chain has a least upper bound.

This structure, which is slightly less restrictive than that of complete lattice, used by D. Scott [11], and was also used in [7], [8] and [12].

(b)

- (i) With each constant symbol C is associated an element c in \mathcal{D} .
- (ii) With each function symbol F of arity n is associated a map f from \mathcal{D}^n to \mathcal{D} which is *monotone* and *continuous* in each of its arguments. (See R. Milner [7] for a definition of these notions).
- (iii) With each term T is associated in a natural way a map t from \mathcal{D} to \mathcal{D} , and with the fixpoint equation $X = T$ we associate the least fixpoint of $t: \mathcal{D} \rightarrow \mathcal{D}$ which we again call the least fixpoint of the equation. It is defined as the upper bound in \mathcal{D} of the set $\{t^n(\perp) \mid n \geq 0\}$. (See again [7]).

Convention We systematically use upper-case letters to designate syntactic objects, and lower-case letters to designate the associated semantic objects.

(c) We are now going to construct a *canonical interpretation* of our language, which plays the role of the Herbrand universe for first-order theories. The domain \mathcal{D} of the canonical interpretation consists of the set of infinite sequences of terms $\{T_i \mid i \in \mathbb{N}\}$ constructed with the variable X such that $T_i \leq T_{i+1}$ for all $i \geq 0$.

We now define an order relation \subseteq on \mathcal{D} : if $\tau = \{T_i \mid i \in \mathbb{N}\}$ and $\tau' = \{T'_i \mid i \in \mathbb{N}\}$ are two elements of \mathcal{D} , then $\tau \subseteq \tau'$ if and only if $\forall i \exists j T_i \leq T'_j$.

The minimal element of \mathcal{D} is $\perp = \{T_i \mid \forall i \in \mathbb{N}, T_i = X\}$. Every chain $\tau_1 \subseteq \tau_2 \subseteq \dots \subseteq \tau_i \subseteq \dots$ in which $\tau_i = \{T'_i \mid j \in \mathbb{N}\}$ admits an upper bound $\tau = \bigcup_{i \in \mathbb{N}} \tau_i = \{T'_i \mid i \in \mathbb{N}\}$ where for all i , $T'_i = T_{n_i}^i$, and the sequence n_i chosen so that $\forall i (n_i \geq i \text{ and } T_{n_i}^i \leq T_{n_{i+1}}^{i+1})$. (There exists such a sequence since the τ_i 's form an ascending chain.)

A constant C is therefore interpreted as the sequence:

$$c = \{T_i \mid T_i = C, i \in \mathbb{N}\}$$

The interpretation f of an n -ary symbol F maps n sequences $\{T_k^i \mid k \in \mathbb{N}\}$ for i in $[1, n]$ to the sequence $\{F(T_k^1, T_k^2, \dots, T_k^n) \mid k \in \mathbb{N}\}$. It is easy to verify that we have a legitimate interpretation. (More rigorously, the interpretation domain that we have considered is \mathcal{D}/\equiv).

We write $P\{A/X\}$ for the result of substituting A for all occurrences of X in a term P . We can verify that in our canonical interpretation $Y(t) = \{T^i \mid i \in \mathbb{N}\}$ with

$$T^1 = T \text{ and } T^{i+1} = T\{T^i/X\}$$

The interest in the canonical interpretation arises from the following lemma:

Lemma 8.4 *Two fixpoint equations are equivalent if and only if they are equivalent in the canonical interpretation.*

Proof It suffices to demonstrate that if $X = T$ and $X = T'$ are equivalent in the canonical interpretation c , they must be equivalent in all other interpretations I .

Let the fixpoints of these equations be $Y(t) = \{T^i \mid i \in \mathbb{N}\}$ and $Y(t') = \{T'^j \mid j \in \mathbb{N}\}$ in the canonical interpretation. It is easy to verify that $T_1 \leq T_2$ implies $t_1(\perp) \subseteq_I t_2(\perp)$ for all interpretations I . From $\forall i \exists j T^i \leq T'^j$ we deduce

$$\forall i \exists j t^i(\perp) \subseteq_I t'^j(\perp)$$

and by symmetry

$$\forall k \exists l t'^k(\perp) \subseteq_I t^l(\perp).$$

Consequently, in I , $\bigcup_{i \in \mathbb{N}} t^i(\perp) \equiv_I \bigcup_{i \in \mathbb{N}} t'^i(\perp)$. □

We can now state two technical lemmas that will be useful later.

Lemma 8.5 *If two terms $F(T_1, \dots, T_n)$ and $G(T'_1, \dots, T'_m)$ are equal in the canonical interpretation, then $F = G$, $n = m$, and for all i , $T_i = T'_i$ in the canonical interpretation.*

Proof Let $\{T_k^i \mid i \in \mathbb{N}\}$ and $\{T'_k{}^i \mid i \in \mathbb{N}\}$ be the interpretations of T_k and T'_k . Since:

$$\forall i \exists j F(T_1^i, \dots, T_n^i) \leq G(T'_1{}^j, \dots, T'_m{}^j)$$

from which we deduce $F = G$, $n = m$ and

$$\forall i \exists j T_k^i \leq T_k^{j'} \quad (k \in [1, n])$$

and the opposite inequality also holds. □

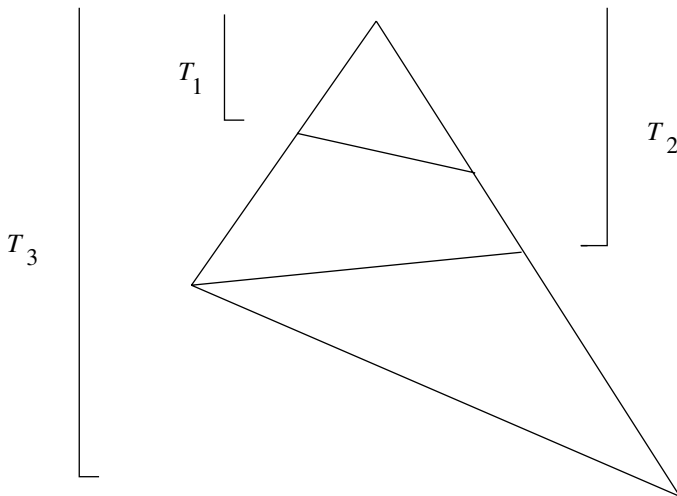
Lemma 8.6 *For all terms T and T' , if $Y(t) \subseteq Y(t')$ in the canonical interpretation, then either $T = X$ or $Y(t) \equiv Y(t')$.*

Proof (a) First we define two notions of “depth” of a term T , $\text{pmax}(T)$ and $\text{pmin}(T)$:

$$\begin{cases} \text{pmax}(X) = \text{pmax}(C) = 1 \\ \quad \text{(if } C \text{ is a constant)} \\ \text{pmax}(F(T_1, \dots, T_n)) = 1 + \max_{1 \leq i \leq n} \{\text{pmax}(T_i)\} \end{cases}$$

$$\begin{cases} \text{pmin}(X) = 1 ; \text{pmin}(C) = +\infty \\ \quad \text{(if } C \text{ is a constant)} \\ \text{pmin}(F(T_1, \dots, T_n)) = 1 + \min_{1 \leq i \leq n} \{\text{pmin}(T_i)\} \end{cases}$$

These notions allow us to state the following “alignment” property: if T_1, T_2, T_3 satisfy $T_1 \leq T_3$ and $T_2 \leq T_3$ and $\text{pmax}(T_1) \leq \text{pmin}(T_2)$ then $T_1 \leq T_2$. The proof is done easily by structural induction. We can represent this situation by the following figure:



(b) Now, given two fixpoints $Y(t) = \{T^i \mid i \in \mathbb{N}\}$ and $Y(t') = \{T'^i \mid i \in \mathbb{N}\}$, if they satisfy $Y(t) \subseteq Y(t')$ then $\forall i \exists j T^i \leq T'^j$.

Let us also show that if we have $\forall j \exists i T'^j \leq T^i$, if $T \neq X$. Given T'^j , we can always find an i such that

$$\text{pmin}(T^i) \geq \text{pmax}(T'^j)$$

It is therefore possible to choose k such that $k \geq j$ and $T^i \leq T'^k$. Then $T'^j \leq T'^k$ and by the alignment property $T'^j \leq T^i$. Then $Y(t') \subseteq Y(t)$. \square

8.2.3 Normal form and equivalence algorithm

We are going to show that the set of terms leading to equivalent fixpoint equations is closed under the operation Σ .

Notation If two terms T and T' have interpretations t and t' in the canonical interpretation, we write $\sigma(t, t')$ for the interpretation of $\Sigma(T, T')$.

Lemma 8.7 *If $t(a) \equiv t'(a)$ for some a , then $\sigma(t, t')(a) \equiv t(a) \equiv t'(a)$.*

Proof By structural induction on T :

- (1) If $T = X$ or $T = C$ the property is obvious.
- (2) If $T = F(T_1, \dots, T_n)$ two cases arise:
 - $T' = X$, then $\Sigma(T, T') = X$ and the property holds;
 - $T' = G(T'_1, \dots, T'_m)$ and so necessarily $F = G$, $m = n$ and $t'_i(a) = t_i(a)$ for all i in $[1, n]$. By the induction hypothesis, $\sigma(t_i, t'_i)(a) = t_i(a)$ and therefore

$$t(a) = f(\sigma(t_1, t'_1)(a), \dots, \sigma(t_n, t'_n)(a)) = \sigma(t, t')(a).$$

\square

Lemma 8.8 *If two terms T and T' have the same fixpoint, then $\Sigma(T, T')$ also has the same fixpoint. In other words $Y(t) \equiv Y(t')$ implies $Y(t) = Y(\sigma(t, t'))$.*

Proof We have $Y(t) \equiv t(Y(t)) \equiv Y(t') \equiv t'(Y(t')) \equiv t'(Y(t))$. By Lemma 8.7 we obtain $\sigma(t, t')(Y(t)) = Y(t)$ and by minimality $Y(\sigma(t, t')) \subseteq Y(t)$. Lemma 8.6 therefore implies:

- either $Y(\sigma(t, t')) \equiv Y(t)$ and the proof is finished;
- or $\Sigma(T, T') \equiv X$, but this is compatible with $Y(t) = Y(t')$ only if $T = T' = X$ in which case we again have $Y(\sigma(t, t')) = Y(t)$.

□

We are now ready to show (non-constructively) the existence of a minimal form for the set of terms leading to equivalent fixpoint equations.

We write $\|T\|$ for the size of a term T , defined recursively by:

- (1) $\|X\| = 0$
- (2) $\|F(T_1, \dots, T_n)\| = 1 + \sum_{i=1}^n \|T_i\|$.

Lemma 8.9 *In the set $E(T) = \{T' \mid Y(t') \equiv Y(t)\}$ of terms having the same least fixpoint as T , there exists an element T^* of minimal size.*

Proof Let T_1 and T_2 be two different terms of $E(T)$ of minimal size. Then T_1 and T_2 are inevitably incomparable, otherwise, from Lemma 8.3, one of the two would be of size strictly less than the other. But then, again $\|\Sigma(T_1, T_2)\| < \|T_1\|$ from the lemma. Since $E(T)$ is closed under Σ by Lemma 8.8, T_1 and T_2 cannot be minimal. □

For the moment, Lemma 8.9 does not allow the construction of T^* , but we will present a syntactic relation between all the terms having the same fixpoint as T^* . We write $T' \rightarrow T''$ for the relation defined by these axioms:

- (i) $\vdash T \rightarrow T$
- (ii) $T \rightarrow U \vdash T \rightarrow U\{\{X/T\}\}$

where the notation $U\{\{X/T\}\}$ indicates that some occurrences of X in U have been replaced by T . We write $D(T) = \{U \mid \vdash T \rightarrow U\}$.

Theorem 8.10 *The set $E(T)$ of terms having the same least fixpoint as T is identical to the set $D(T^*) = \{U \mid \vdash T^* \rightarrow U\}$ of terms deriving from the minimal element T^* of $E(T)$.*

Proof Of course $E(T) = E(T^*)$. The fact that $D(T^*) \subseteq E(T^*)$ is already known (for example cf. [10]). We now show that $E(T^*) \subseteq D(T^*)$: let T' be a term of minimal size belonging to $E(T^*)$ and not to $D(T^*)$. We necessarily have $T^* \leq T'$ because T^* is a normal form. So, $T' = T^*\{T_1, \dots, T_k\}$. Since T' is in $E(T^*)$, we have

$$Y(t^*) \equiv Y(t') \equiv t'(Y(t')) \equiv t'(Y(t^*)).$$

Therefore $t^*(Y(t^*)) = t'(Y(t^*))$.

From Lemma 8.5, we deduce $Y(t^*) \equiv t_i(Y(t^*))$ for $i \in [1, k]$. By minimality and with Lemma 8.6, we obtain $Y(t_i) = Y(t^*)$. But since the T_i 's are smaller than T' they are by hypothesis in $D(T^*)$. Consequently T' is in $D(T^*)$. \square

Theorem 8.10 implies that the normal form T^* of T is a subterm of T , namely the smallest subterm T^* such that $T^* \rightarrow T$. This gives us an algorithm for computing this normal form and a decision procedure for the equivalence of two fixpoint equations. Rather than presenting these two algorithms in greater detail here, we now proceed to the general case, that of systems of fixpoint equations.

8.3 Systems of recursive equations

The results obtained in the second section extend to the case of systems of fixpoint equations: with each system, one can associate a canonical system of minimal size (the "size" of a system is the sum of the sizes of its constituting equations). Thus, one obtains an algorithm for deciding the equivalence of two systems and one can also compute an equivalent system with a minimal number of equations. However, this last system is not necessarily unique.

8.3.1 Syntax

We use a set $\Xi = \{X_1, \dots, X_n\}$ of variables.

A *system* is:

- (i) a family of fixpoint equations $X_i = T_i$, for $i = 1, 2, \dots, n$. The terms T_i are constructed over the variables X_i , $i \in [1, n]$.
- (ii) A *main variable* X_1 .

A system is said to be *connected* if all its equations are needed to compute X_1 , with the following definition: the equation $X_i = T_i$ and all the equations necessary for variables of T_i are needed to compute X_i . The *size* of a system $S = \{X_i = T_i \mid i \in [1, n]\}$ is defined by $\|S\| = \sum_1^n \|T_i\|$ and a system is said to be *uniform* if for all i , $\|T_i\| = 1$ i.e., if T_i contains a single function symbol.

8.3.2 Semantics

If we interpret terms as in Section 8.2.2 in a domain \mathcal{D} , then for the set $\{T_i \mid i \in [1, n]\}$ one gets an obvious map from \mathcal{D}^n to \mathcal{D}^n and thus a fixpoint in \mathcal{D}^n . We write $Y(x_i)$ for the component of this vector corresponding to the variable X_i .

Two systems S and S' are *equivalent* if, for all interpretations we have $Y(x_1) \equiv Y(x'_1)$. The notion of canonical interpretation extends trivially and the reader may verify that Lemma 8.4 remains valid.

A system $\{X_i = T_i \mid i \in [1, n]\}$ is said to be *normal* if it is not equivalent to a system $\{X_j = T_j \mid j \in [1, n]\}$ in which the T_j 's are subterms of the T_i and at least one of them is a proper subterm or a variable.

A proper subterm of T is a subterm of T different from T .

The case of a system containing a single equation, the system is normal if the equation is in normal form.

8.3.3 Canonical systems

Given a system $S = \{X_i = T_i \mid i \in [1, n]\}$ we start by constructing a uniform system \bar{S} equivalent to it. For example:

$$S = \begin{cases} X_1 & = & F(X_1, G(X_1, X_2)) \\ X_2 & = & H(F(X_1, X_2)) \end{cases}$$

(main variable X_1)

$$\bar{S} = \begin{cases} Y_1 & = & F(Y_1, Y_3) \\ Y_2 & = & H(Y_4) \\ Y_3 & = & G(Y_1, Y_2) \\ Y_4 & = & F(Y_1, Y_2) \end{cases}$$

(main variable Y_1)

Formally, \bar{S} is constructed as follows: let us call $\mathcal{J} = \{\tau_i \mid i \in [1, m]\}$ the set of proper subterms of S (in the case of our example: $\mathcal{J} = \{X_1, X_2, F(X_1, X_2), G(X_1, X_2)\}$.) The system \bar{S} is constructed over new variables $Y_i, i \in [1, m]$ associated as follows with elements of \mathcal{J} :

- (1) If $\tau_i = X_i$ and if $X_i = F(\tau_{i_1}, \dots, \tau_{i_k})$ is an equation of S , then $Y_i = F(Y_{i_1}, \dots, Y_{i_k})$ is an equation of \bar{S} .
- (2) If $\tau_i = F(\tau_{i_1}, \dots, \tau_{i_k})$ then $Y_i = F(Y_{i_1}, \dots, Y_{i_k})$ is an equation of \bar{S} .
- (3) The main variable of \bar{S} is the new variable Y_1 associated with X_1 .

Lemma 8.11 $S \equiv \bar{S}$

Lemma 8.12 *If S is connected, then \bar{S} is connected. If S is normal, then \bar{S} is normal.*

For a uniform system $S = \{X_i = F_i(X_{i_1}, \dots, X_{i_{k(i)}}) \mid i \in I\}$ we compute the equivalence relation over variables defined by $Y(x_i) \equiv Y(x_j)$ for all interpretations, which we write $X_i \equiv X_j$. We inductively define an increasing sequence of subsets of $\Xi \times \Xi$, if $\Xi = \{X_i \mid i \in I\}$:

- (1) $D_0 = \{(X_i, X_j) \in \Xi \times \Xi \mid F_i \neq F_j\}$
- (2) $D_{n+1} = D_n \cup \{(X_i, X_j) \in \Xi \times \Xi \mid F_i = F_j \wedge \exists m \in [1, k(i)] \text{ such that } (X_{i_m}, X_{j_m}) \in D_n\}$

Lemma 8.13

- (i) *There exists an integer l such that $D_l = \bigcup_{n=0}^{\infty} D_n$.*
- (ii) *$X_i \equiv X_j$ if and only if the pair (X_i, X_j) does not appear in D_l .*

Proof (i) The existence of l comes simply from the fact that Ξ is finite. Let us prove (ii) by Scott induction (cf. [5]) on the formula Φ :

$$\bigwedge \{X_i \equiv X_j \mid (X_i, X_j) \notin D_l\}$$

First of all, $\bigwedge \Omega \equiv \Omega$. In addition, Φ implies:

$$\Phi\{T_1/X_1, \dots, T_i/X_i, \dots, T_n/X_n\}$$

since $(X_i, X_j) \notin D_l$ leads to $T_i \equiv T_j$. □

Theorem 8.14 *The equivalence of two systems of fixpoint equations is decidable.*

Proof Lemma 8.11 makes it possible to reduce to the case of uniform systems. If $S = \{X_i = T_i \mid i \in I\}$ and $S' = \{X'_i = T'_i \mid i \in I'\}$, it is always possible to ensure that $\Xi = \{X_i \mid i \in I\}$ and $\Xi' = \{X'_i \mid i \in I'\}$ are *disjoint* and to consider the system

$$S'' = \{X_i = T_i, X'_j = T'_j \mid i \in I, j \in I'\}.$$

The algorithm of Lemma 8.11 tells us whether $X_1 \equiv X'_1$. □

Corollary 8.15 *The equivalence $T \equiv T'$, where T and T' are any terms on the variables of the two systems S and S' is decidable.*

Proof One adds the equations $Z = T$ and $Z' = T'$ and verifies whether $Z \equiv Z'$. \square

Theorem 8.16 *For any system S one can construct a corresponding equivalent system S' which is normal, whose size is at most the size of S and which has no more variables.*

Proof Let T be a proper subterm of S , $T \neq X_i$, such that $X_i \equiv T$. This equality permits to “reduce” the system S as follows:

- If T is a variable X_j , one replaces everywhere X_j by X_i and one removes from S the equation defining X_j , $X_j = T_j$. This transformation eliminates one variable from the system without increasing its size. One could also have eliminated X_i instead of X_j ; this degree of freedom will be exploited later.
- If T is a subterm of size greater or equal to 1, we replace all its occurrences in S with X_i . The size of the system can only decrease and the system obtained is equivalent to the initial system.

This construction can only be iterated a finite number of times because there may be only a finite number of equivalences $X_i \equiv T$. By definition, the system which one then reaches is normal. \square

Example 8.17

$$S_0 \begin{cases} X = F(F(X, Y), Z) \\ Y = G(X, Z) \\ Z = G(X, G(X, Y)). \end{cases}$$

One finds that $Y \equiv Z$. But S_0 is equivalent to S_1 :

$$S_1 \begin{cases} X = F(F(X, Y), Y) \\ Y = G(X, Y). \end{cases}$$

But $X \equiv F(X, Y)$. Hence S_1 is therefore equivalent to S_2 :

$$S_2 \begin{cases} X = F(X, Y) \\ Y = G(X, Y). \end{cases}$$

So S_2 is normal.

The construction of Theorem 8.16 can be more easily carried out on uniform systems where the only proper subterms are variables.

Lemma 8.13 defines an equivalence relation between these variables and the normal system associated with a uniform system connects to the equivalence classes.

Notation If S is uniform, we write $n(S)$ for the normal system to which it corresponds. With each system S , one associates an equivalent uniform system \bar{S} , and $\hat{S} = n(\bar{S})$ that is normal and uniform.

Lemma 8.18 *If R and S are two equivalent, connected systems, then $\hat{R} = \hat{S}$.*

Proof Of course, the equality between \hat{R} and \hat{S} is understood to be up to variable renaming. We first show that for each variable X_i of \bar{R} there exists a variable X'_j of \bar{S} such that $X_i \equiv X'_j$, by recurrence. Since $R \equiv S$ and $\bar{R} \equiv \bar{S}$, we have $X_1 \equiv X'_1$.

If $X_i \equiv X'_j$ and $X_i = F(X_{i_1}, \dots, X_{i_n})$, $X'_j = F(X'_{j_1}, \dots, X'_{j_n})$, then $X_{i_1} \equiv X'_{j_1}, \dots, X_{i_n} \equiv X'_{j_n}$. Consequently, all the variables necessary to X_1 have a corresponding variable in \bar{S} . Since \bar{R} is connected, these are all the variables of \bar{R} . Of course, the symmetric property is true:

$$\forall j \exists i X'_j \equiv X_i$$

Therefore there exists a bijection between the equivalence classes of \bar{R} and \bar{S} . Since by construction the variables of (respectively) \hat{R} and \hat{S} are independent, we have $\hat{R} = \hat{S}$ modulo a renaming. \square

Lemma 8.18 justifies calling the system \hat{S} a canonical system.

Theorem 8.19 *In the set of systems equivalent to a given system S , the canonical system \hat{S} is of minimal size.*

Proof From the construction of \bar{S} , it is clear that $\|S\| \geq \|\bar{S}\|$. Since $\|\bar{S}\| \geq \|n(\bar{S})\| = \|\hat{S}\|$ and $\hat{R} = \hat{S}$ for all R equivalent to S , \hat{S} is of minimal size. \square

Remark There may exist multiple equivalent systems of minimal size, but only one may be canonical, as the following example illustrates:

$$S_1 = \begin{cases} X = F(X, Y) \\ Y = G(Z) \\ Z = H(X, Y) \end{cases} ; \quad S_2 = \begin{cases} X = F(X, Y) \\ Y = G(H(X, Y)) \end{cases} .$$

Here, $\|S_1\| = \|S_2\|$, $S_1 \equiv S_2$ and S_1 is canonical.

8.3.4 Minimizing the number of equations

One might be interested in a representation of a system that minimizes not the size but the number of equations. The following example illustrates that there is not always a unique system that is normal and minimal (in this sense):

$$S_1 = \begin{cases} X & = F(X, Y) \\ Y & = G(H(X, Y)) \end{cases} ; \quad S_2 = \begin{cases} X' & = F(X', G(Y')) \\ Y' & = H(X', G(Y')) \end{cases} .$$

Thus S_1 and S_2 are equivalent ($X \equiv X'$), normal and each has a minimum number of equations. But S_1 and S_2 are not identical up to renaming of variables.

We now show how to construct all normal systems and all minimal normal systems.

Definition 8.20 Let $S = \{X_i = T_i \mid i \in I\}$ and C be a set of variables of S , $C = \{X_k \mid k \in K\}$, containing X_1 and having the following property:

$$\forall l \in I \setminus K, \exists T(l, C),$$

a term using only variables of C such that $X_l \equiv T(l, C)$.¹

We call an S -cut, denoted by $C(S)$, the system associated with such a set C :

$$C(S) = \{X_i = T'_i \mid X_i \in C\}$$

in which T'_i is obtained from T_i by replacing for all l in $I \setminus K$ the variable X_l by $T(l, C)$. To a set C containing a minimal number of variables corresponds a minimal cut.

Theorem 8.21 The normal systems equivalent to S are the cuts of \hat{S} .

Proof If R is a cut of \hat{S} , then $R \equiv S$ and R is normal otherwise \hat{S} would not be.

If R is normal and $R \equiv S$, then \bar{R} is normal since $\bar{R} = \hat{R} = \hat{S}$.

But R is a cut of \hat{S} if and only if $\bar{R} = \hat{S}$ and R is normal. \square

Corollary 8.22 The minimal normal systems equivalent to S are the minimal cuts of \hat{S} . Since there are only a finite number of cuts of \hat{S} , one can effectively construct these minimal systems.

¹ And by $T(l, C)$ we designate in what follows the least term allowing the definition of X_l .

Example 8.23

$$S = \begin{cases} X = F(F(X, Y), Y) \\ Y = G(H(X, G(Z))) \\ Z = H(X, G(Z)) \end{cases} \quad \hat{S} = \begin{cases} X = F(X, Y) \\ Y = G(Z) \\ Z = H(X, Y) \end{cases}$$

$$C_1 = \{X, Y, Z\}, \quad C_1(\hat{S}) = \hat{S}$$

$$C_2 = \{X, Y\}, \quad C_2(\hat{S}) = \begin{cases} X = F(X, Y) \\ Y = G(H(X, Y)) \end{cases}$$

$$C_3 = \{X, Z\}, \quad C_3(\hat{S}) = \begin{cases} X = F(X, G(Z)) \\ Z = H(X, G(Z)) \end{cases}$$

So, we find the two minimal normal forms of S .

Remark If one starts from *a single* recursive equation as in the second section, one always finds a single minimal normal form, the normal form of this equation.

Example 8.24

$$\begin{aligned} S &= \{X = F(X, G(F(X, G(X, X)), X))\} \\ \hat{S} &= \begin{cases} X = F(X, X') \\ X' = G(X, X) \end{cases} \\ C &= \{X\}, \quad C(\hat{S}) = \{X = F(X, G(X, X))\}. \end{aligned}$$

8.4 Conclusion

In the opinion of the authors, the interest of this work lies not in the decidability results obtained (which can be with less effort) but in the methods used, in particular the construction of a canonical domain.

This method was used by B. Courcelle and J. Vuillemin [1] for functional systems.

Finally, it is possible to consider the results obtained as completeness results of subtheories of the logic LCF (Logic for Computable Functions) studied by R. Milner [7].

Acknowledgement

Some of the results obtained here were already known by R. Milner, who collaborated in the early phase of this work. B. Courcelle and J. Vuillemin thank T. Veldhuizen for the translation.

Bibliography

- [1] B. Courcelle and J. Vuillemin, Completeness results for the equivalence of recursive schemes, *Journal of Computer System Science* **12**:179–197, 1976.
- [2] W. P. De Roeper, Operational and mathematical semantics for first-order recursive program schemas, (private communication).
- [3] J. Engelfriet, A note on infinite trees, *Information Processing Letters* **1**:229–232, 1972.
- [4] G. Kahn, A preliminary theory for parallel programs. (Rapport Laboria no. 6, January 1973).
- [5] J. Kral, Equivalence of modes and the equivalence of finite automata, *Algol Bulletin* **35**:34–35, 1973.
- [6] C. H. Lewis and B. K. Rosen, Recursively defined data types: part 1. *Proceedings of the 1st annual ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, pp. 125–138. ACM, New York, 1973.
- [7] R. Milner, Models of LCF. Stanford Computer Science Department Report. CS-332, 1973.
- [8] R. Milner and R. Weyrauch, Proving compiler correctness in a mechanized logic. In B. Meltzer and D. Michie (eds), *Machine Intelligence 7*, pp. 51–72. Edinburgh University Press, 1972.
- [9] M. Nivat, Sur l'interprétation des schémas de programmes monadiques. Rapport Laboria No. 1, 1972.
- [10] C. Pair, Concerning the syntax of Algol 68, *Algol Bulletin* **31**:16–27, 1970.
- [11] D. Scott, *Outline of a Mathematical Theory of Computation*. Oxford Monograph PRG-2. Oxford University, 1970.
- [12] J. Vuillemin, *Proof Techniques for Recursive Programs*. PhD thesis, Stanford Computer Science Department. 1973.