Normalization and linearity in unification logic

Philippe Le Chenadec

INRIA

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tel (1) 39 63 55 11

Rapports de Recherche

N°922

Programme 1

NORMALIZATION AND LINEARITY IN UNIFICATION LOGIC

Philippe LE CHENADEC

Octobre 1988

RR_0922

Normalization and Linearity in Unification Logic

Philippe LE CHENADEC
INRIA B.P. 105 - Rocquencourt
78153 Le Chesnay Cedex France

October 18, 1988

Abstract

We present a normalization process for natural deduction style proofs in an equational logic dedicated to unification. The reductions enjoy the classical strong normalization and Church-Rosser properties. The normal forms possess all the interesting symmetries of cut-free proofs in natural deduction or sequent calculi. Especially, the inference rules display a left-right symmetry and were chosen for their atomicity. The system presents a deep analogy with linear logic, by the presence of trips associated to proofs. This Hauptsatz for equational logic provides an essential tool for a proof-theoretic investigation of 1st and 2nd order unification.

Normalisation et Linéarité dans la Logique de l'Unification

Résumé

Nous présentons une procédure de Normalisation des preuves en Déduction Naturelle dans une logique équationnelle liée à l'unification du 1er ordre. Les réductions possèdent les propriétés classiques de Normalisation Forte et de Church-Rosser. Les formes normales vérifient les propriétés de symétrie des preuves sans coupures des calculs de séquent ou de déduction naturelle. Plus particulièrement, les règles d'inférence réflètent une symétrie gauche-droite et ont té choisies pour leur atomicité. Le système présente une analogie profonde avec la logique linéaire, notamment par la presence de circuits associés aux preuves. Ce Hauptsatz pour logique équationnelle donne un outil essentiel pour une étude style théorie de la démonstration de l'unification au 1er et au 2nd ordre.

Normalization and Linearity in Unification Logic

Philippe LE CHENADEC
INRIA BP 105 - Rocquencourt
78153 Le Chesnay Cedex

October 18, 1988

Abstract

We present a normalization process for natural deduction style proofs in an equational logic dedicated to unification. The reductions enjoy the classical strong normalization and Church-Rosser properties. The normal forms possess all the interesting symmetries of cut-free proofs in natural deduction or sequent calculi. Especially, the inference rules display a left-right symmetry and were chosen for their atomicity. The system presents a deep analogy with linear logic, by the presence of trips associated to proofs. This Hauptsatz for equational logic provides an essential tool for a proof-theoretic investigation of 1st and 2nd order unification.

1 Introduction

Since the original work of Gentzen [4] on sequent calculus, much work has been devoted to the normalization process of various logics [5]. Such an analysis was lacking in equational logic (the only exceptions we are aware of are [24,16]). There is a very simple explanation for this oversight: equational logic as traditionally presented lacks an elimination rule. However, this elimination rule is omnipresent in Computer Science, disguised under unification, e.g. in Resolution, Rewriting and Type Inference. Adding this rule to the traditional introduction rule (Leibniz's principle of substitution of equals for equals) gives us a logic with strong properties, in essence those of linear logic [7].

This logic was encountered while working on higher-order equations, in relation with the type inference problem for second-order lambda calculus [6,21], a major open problem in type theory. These equations roughly have the following syntax:

$$\lambda\overline{\alpha}.\Phi(\overline{\alpha},\overline{X}(\overline{\alpha})) = \lambda\overline{\alpha}.F((\Pi(\lambda\overline{\beta}.\Psi(\overline{\alpha},\overline{\beta},\overline{Y}(\overline{\alpha},\overline{\beta})))),(\Pi(\lambda\overline{\gamma}.\Theta(\overline{\alpha},\overline{\gamma},\overline{Z}(\overline{\alpha},\overline{\gamma})))))$$

where e.g. F is the function space type constructor of type $Type \rightarrow Type \rightarrow Type$, and Π is Church's quantifier of type $(Type \rightarrow Type) \rightarrow Type$. Overlines denote sequences of variables. The free functional variables split into two sets: on one hand we have Φ, Ψ, Θ (subterm types), on the other hand we have the sequences of X's, Y's and Z's (the type extractions). These two sets are *disjoint*. Also to these equations is naturally associated a set of first-order equations, obtained by stripping the higher-order structure, here we get $\phi = F(\Pi(\psi),\Pi(\theta))$. This is nothing but the familiar idea of first-order approximation of some higher-order complex objects.

If this new set of equations is solvable, say by usual unification, the higher-order equations are trivially solvable. Alternatively, when the associated set $\mathcal{E}$ of first-order equations is not solvable, we want some precise measure of the degree of failure. Ideally, this would allow us to check whether or not the lifting from 1st to 2nd order overcomes this failure. This is indeed so and will appear in print somewhere else [19]. Due to the origin of the problem, failures originate in the so-called occur-check. The desired measure is given by all the *minimal* deductions of the *minimal* fixpoint equations deducible from $\mathcal{E}$. The set $\mathcal{C}(\mathcal{E})$ of these deductions is finite provided $\mathcal{E}$ is and gives valuable informations about the higher-order equations.

Statements about $\mathcal{C}(\mathcal{E})$ required a normalization procedure for equational deduction, as well as a semantical analysis of redundancies in proofs. The first results of this study were presented in [18]. Namely we have a rewriting system complete in the Knuth-Bendix sense [12,17], for an equational logic LE_1 complete for a semantics of *unification graphs*. The normal forms enjoy the properties of cut-free or normal proofs in a Gentzen or natural deduction calculus: such a proof corresponds to sequences of eliminations followed by a sequence of introductions [5]. A set of *trips* is associated to each proof, à la logique linéaire, which characterizes the proof up to permutation. Finally, we have a (quite trivial) subformula property, and the occurrences of subterms in the proof can be partitionned according to trips into positive an negative occurrences, which is of the utmost importance for the higher-order equations that we have to solve in type inference. This defines an operational semantics of proofs, in terms of communication. Each inference rule, except the trivial symmetry rule, possesses a *proper term*, the inference being performed by identifying two occurrences, a positive one and a negative one, of this proper term in the premisses. We established in [18] that whenever these two occurrences were equal, the deduction was not (semantically) minimal, the proof of this fact being constructive. The aim of this paper is to give a cleaner explanation of these phenomenon than the sketches to be found in [18] and to generalize these results to full equational logic augmented with a unification inference LE_2. The reduction process was more delicate to handle than in the restricted case and required quite heavy computations. The short-circuit phenomenon of linear logic appears in *stupid* proofs that undo what they had previously built. The proofs in the restricted fragment outlines the sequential nature of unification in an intrinsic, qualitative way. This was already established in a complexity theory framework [3]. This sequentiality appears in the fact that the trips are *long-paths*, while in the general case, this is no longer true and the trips modelize the *data-flow* in the net of the proof.

Let us see an example. When we try to type the λ-term $M = (\lambda x.xx)\,(\lambda xy.xyx)$ in simple types [2], we get the following inference tree:

$$\dfrac{\dfrac{(1)\ \dfrac{x:\alpha \quad x:\alpha}{xx:\alpha'}}{\lambda x.xx:\alpha\to\alpha'} \quad \dfrac{\dfrac{(2)\ \dfrac{x:\beta \quad y:\gamma}{xy:\beta'} \quad x:\beta}{xyx:\beta''}}{\lambda xy.xyx:\beta\to\gamma\to\beta''}}{(4)\ \dfrac{}{M:\alpha'}}$$

We have to solve the equations:

$$\mathcal{E}_M \begin{cases} \alpha = \alpha\to\alpha' & (1)\\ \beta = \gamma\to\beta' & (2)\\ \beta' = \beta\to\beta'' & (3)\\ \alpha = \beta\to(\gamma\to\beta'') & (4)\end{cases}$$

Unification fails (cf. the unification graph in Fig. 1). The reader may check that, up to redundancy and permutation of subdeductions, the set $\mathcal{C}(\mathcal{E}_M)$ contains the three deductions (hint: this checking is difficult at this point of the paper):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha=\alpha\to\alpha'}\quad su\ \dfrac{\beta'=\beta\to\beta'' \quad \beta=\gamma\to\beta'}{\beta=\gamma\to(\beta\to\beta'')}}{\alpha=\beta\to(\gamma\to\beta'')}\ u \quad \dfrac{\alpha=\beta\to(\gamma\to\beta'') \quad \alpha=\alpha\to\alpha'}{\beta=\alpha}}{\beta=\beta\to(\gamma\to\beta'')}\ t \quad \beta=\gamma\to\beta'}{\beta'=\gamma\to\beta''}\ u \quad \beta'=\beta\to\beta''}{\gamma=\beta}\ u \quad \beta=\gamma\to\beta'}{\gamma=\gamma\to\beta'}\ t$$

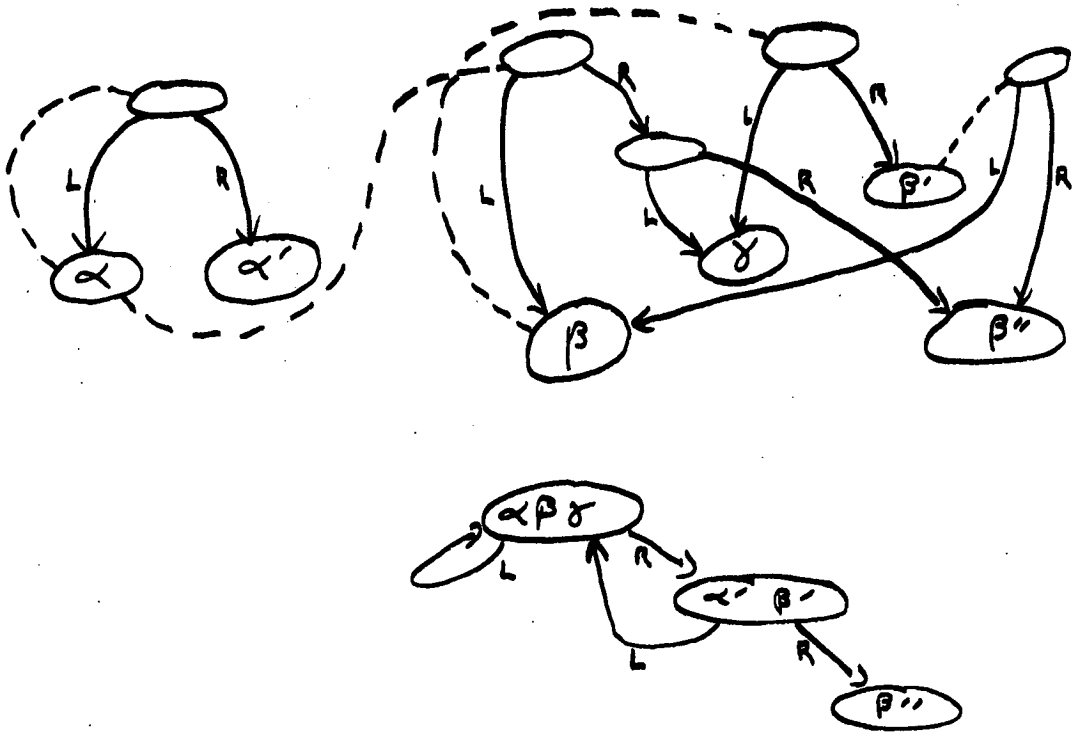

Figure 1: Term and Unification Graph for $\mathcal{E}_M$

The paper is organized as follows. Section 2 presents the results of [18], with an important improvement: a rewriting system approach is introduced where we used an ad-hoc normalization. This gives the correct intuitions for the extension of the results to a full equational logic and completes the proof-theoretic analysis of unification initiated in [18]. The definition of trips gives a semantics for section 3, which introduces a complete system RE_2 for our full equational logic LE_2 augmented with a unification closure rule playing the rôle of the elimination rule. The familiar substitution rule is split into two rules, a left one and a right one, that perform one substitution at a single occurrence of some variable or term. The confluence of the rewriting system forces us to break the symmetry L/R present in the inference rules. This system choose between *parallel* substitutions: if we substitute M_i for N_i, $i = 1,2$, in $C[N_1, N_2]$, we first perform the outermost-leftmost substitution. By taking into account this (arbitrary) ordering (sequentialization) of substitution, we obtain an infinite complete system, described by the first system plus some meta-rules. The system becomes infinite as we break the L/R symmetry: this denotes the imperfection of the syntax that fails to correctly express the parallelism present in deductions. Finally, we generalize the properties of the restricted system LE_1 to LE_2 and provide a semantic notion of minimality for deductions. The syntactic form and the semantic minimality allows the design of proof-search procedures. To conclude the paper, we sketch the use of these results in handling higher-order equations. We assume known the basic theory of typed λ-calculi, rewriting systems and unification [2,6,12,17].

2 A Proof-Theoretic Analysis of Unification

This section is a case analysis of the well-known Curry-Howard [11] isomorphism between Constructive Mathematics and Computer Science. We first precise some syntax. For convenience, we assume that terms are built up over a binary function symbol f. The other cases are a straightforward generalization. Variables are denoted by lower case latin letters, terms by upper case latin letters. The set of variables occurring in a set of equations $\mathcal{E}$ is noted $V(\mathcal{E})$. The set of occurrences of a term is noted $\mathcal{O}(M)$, ϵ is the empty occurrence. For $O \in \mathcal{O}(M)$, the subterm of M at occurrence O is noted M/O. We use the context notation

$C[\_]$ for terms. The trivial context is noted $\emptyset$. The occurrence of the hole is noted O_C. The contexts $C_1[\_]$ and $C_2[\_]$ are equivalent, noted $C_1[\_] \sim C_2[\_]$, iff $O_{C_1} = O_{C_2}$. Let $C_1[\_]$ and $C_2[\_]$ be two contexts, their composition $C_1[C_2[\_]]$ will be denoted by juxtaposition $C_1C_2[\_]$. We also need a multiple hole notation $C[\_,\_]$. By convention the occurrence of the first hole is of the form $O0O_1$ while the occurrence of the second one is $O1O_2$, i.e. the left one is... the leftmost one.

According to the Curry-Howard isomorphism, a unification algorithm, given a set of equations $\mathcal{E}$, returns either a proof $\mathcal{D} \vdash x = C[x]$ where $C[x]$ is a non-variable term with a distinguished occurrence of x, or a sequence of proofs $\mathcal{D}_i \vdash x_i = t_i$, $i = 1, \ldots, n$, defining a most general unifier for $\mathcal{E}$ by equations in solved form. In both cases the proofs proceed from the axioms $\mathcal{E}$, according to inference rules encoded in the unification algorithm. We analyse the structure of such proofs according to a specific proof system, well-suited for unification. A remarkable fact here is the disclosure of a (proof-system independant) invariant of unification problems: the trips.

The inference rules of our first system LE_1 have the following salient features: they are dedicated to proofs of equations of the form $x = M$, x a variable. They present a strong atomicity: an inference step performs an elementary operation by identifying two occurrences of some variable. Especially it is difficult to imagine another proof system that will break down these inferences into more elementary parts [15]. The three rules (s), (t) and (su), of symmetry, transitivity and substitution, are standard in equational reasoning, despite their peculiar form; the rule (u) is specific to the unification problem and encodes the well-known unification (downward) closure.

$$LE_1 \left\{ \begin{array}{ll} s\,\dfrac{x = y}{y = x} & t\,\dfrac{x = y \quad y = M}{x = M} \\[2ex] u\,\dfrac{x = C[y] \quad x = D[M] \quad C \sim D}{y = M} & su\,\dfrac{x = M \quad y = C[x]}{y = C[M]} \end{array} \right.$$

Provability in LE_1 from hypotheses $\mathcal{E}$ is denoted as usual by $\vdash^{\mathcal{E}}_{LE_1}$. The set of hypotheses of a deduction $\mathcal{D}$ is noted $\mathcal{A}(\mathcal{D})$. We have soundness and completeness for LE_1, with respect to the semantics of *unification graphs*. These graphs are defined as follows. To a set $\mathcal{E}$ of equations we associate its dag representation $\mathcal{D}(\mathcal{E})$. This dag has one vertex per variable in $V(\mathcal{E})$, each left- and right-hand side of equations from $\mathcal{E}$ defines in the standard way a tree, whose leaves are identified according to their variables giving a so-called *term dag* (cf. Fig. 1), members of distinct equations are associated to distinct term dags. The unification graph $\mathcal{G}(\mathcal{E})$ is the quotient of $\mathcal{D}(\mathcal{E})$ by the smallest downward closed congruence on vertices generated by the equations in $\mathcal{E}$. A set of equations $\mathcal{E}$ is unifiable iff $\mathcal{G}(\mathcal{E})$ does not include (directed) cycles. A path may be represented by a pair (v, O), v a vertex, O an occurrence. These graphs encode in the standard way a set of terms, usually a proper subset of the set of all terms on $V(\mathcal{E})$ (cf. the example $\mathcal{E}_M$). Two such terms t_1, t_2 are equal modulo $\mathcal{E}$, noted $\models^{\mathcal{E}} t_1 = t_2$, iff their roots are equal vertices of $\mathcal{G}(\mathcal{E})$. The vertex of a term t is noted $V(t)$ (formally this is the vertex of an *occurrence* of t in $\mathcal{E}$, however the notation will be unambiguous). Naturally, we are interested in fundamental cycles, in the graph-theoretic sense [1], and associated deductions: the fundamental cycle $c = (V(x), O)$ is equationnally generated by a proof $\mathcal{D} \vdash^{\mathcal{E}}_{LE_1} x = C[x]$ when $O = O_C$.

Theorem 2.1 $\vdash^{\mathcal{E}}_{LE_1} x = M$ *iff* $\models^{\mathcal{E}} x = M$.

Proof. Cf. [18]. The proof of soundness is immediate by structural induction on proofs. Completeness follows from a detailed analysis of *sharing* in unification graphs. □

Simpler proofs of completeness exist (e.g. by contraposition). However, the constructive proof of completeness for LE_1 suggests a normal form result for deductions. A first proof of such a result can be found in the expanded version of [18]. We clean up this result by giving a *rewriting* proof. Let RE_1 be the following rewrite system :

$$s\,\frac{s\,\dfrac{x = y}{y = x}}{x = y} \quad \Longrightarrow \quad x = y \tag{1}$$

$$s\,\dfrac{u\,\dfrac{x=C[y]\quad x=D[z]}{y=z}}{z=y} \;\Longrightarrow\; u\,\dfrac{x=D[z]\quad x=C[y]}{z=y} \tag{2}$$

$$s\,\dfrac{t\,\dfrac{x=y\quad y=z}{x=z}}{z=x} \;\Longrightarrow\; t\,\dfrac{s\,\dfrac{y=z}{z=y}\quad s\,\dfrac{x=y}{y=x}}{z=y} \tag{3}$$

$$t\,\dfrac{x=y\quad t\,\dfrac{y=z\quad z=M}{y=M}}{x=M} \;\Longrightarrow\; t\,\dfrac{t\,\dfrac{x=y\quad y=z}{x=z}\quad z=M}{x=M} \tag{4}$$

$$u\,\dfrac{x=C_1[z]\quad t\,\dfrac{x=y\quad y=C_2[M]}{x=C_2[M]}}{z=M} \;\Longrightarrow\; u\,\dfrac{t\,\dfrac{s\,\dfrac{x=y}{y=x}\quad x=C_1[z]}{y=C_1[z]}\quad y=C_2[M]}{z=M} \tag{5}$$

$$su\,\dfrac{y=M\quad x=y}{x=M} \;\Longrightarrow\; t\,\dfrac{x=y\quad y=M}{x=M} \tag{6}$$

$$u\,\dfrac{x=y\quad x=M}{y=M} \;\Longrightarrow\; t\,\dfrac{s\,\dfrac{x=y}{y=x}\quad x=M}{y=M} \tag{7}$$

$$su\,\dfrac{y=M\quad su\,\dfrac{x=y\quad z=C[x]}{z=C[y]}}{z=C[M]} \;\Longrightarrow\; su\,\dfrac{t\,\dfrac{x=y\quad y=M}{x=M}\quad z=C[x]}{z=C[M]} \tag{8}$$

$$su\,\dfrac{su\,\dfrac{x=M\quad z=C_1[x]}{z=C_1[M]}\quad t=C_2[z]}{t=C_2C_1[M]} \;\Longrightarrow\; su\,\dfrac{x=M\quad su\,\dfrac{z=C_1[x]\quad t=C_2[z]}{t=C_2C_1[x]}}{t=C_2C_1[M]} \tag{9}$$

$$t\,\dfrac{z=y\quad su\,\dfrac{x=M\quad y=C[x]}{y=C[M]}}{z=C[M]} \;\Longrightarrow\; su\,\dfrac{x=M\quad t\,\dfrac{z=y\quad y=C[x]}{z=C[x]}}{z=C[M]} \tag{10}$$

$$u\,\dfrac{x=C_1[y]\quad su\,\dfrac{z=M\quad x=C_2C_3[z]}{x=C_2C_3[M]}}{y=C_3[M]} \;\Longrightarrow\; su\,\dfrac{z=M\quad u\,\dfrac{x=C_1[y]\quad x=C_2C_3[z]}{y=C_3[z]}}{y=C_3[M]} \tag{11}$$

$$u\,\dfrac{x=C_1C_2[y]\quad su\,\dfrac{z=C_3[M]\quad x=C_4[z]}{x=C_4C_3[M]}}{y=M} \;\Longrightarrow\; u\,\dfrac{u\,\dfrac{x=C_4[z]\quad x=C_1C_2[y]}{z=C_2[y]}\quad z=C_3[M]}{y=M} \tag{12}$$

$$u\,\dfrac{su\,\dfrac{x=C_1[y]\quad z=C_2[x]}{z=C_2C_1[y]}\quad z=C_3C_4[M]}{y=M} \;\Longrightarrow\; u\,\dfrac{x=C_1[y]\quad u\,\dfrac{z=C_2[x]\quad z=C_3C_4[M]}{x=C_4[M]}}{y=M} \tag{13}$$

Some comments are in order: the firsts three rules lift up the symmetry rule to axioms, this is similar to the fact that in sequent calculus, the axiom $\Phi \vdash \Phi$ can be restricted to atomic Φ. Rules (4) and (5) reorders proofs restricted to (s), (t) and (u)-inferences. Their meaning will be clarified in the next section. Rules (6)–(8) perform some unessential cleaning-up. Finally, the most significant rules, (9) to (13), regulates the interaction between (t), (u) and (su). We have cut-elimination rules: interpreting the rule (su) as

an introduction (it *builts* terms, while the rule (u) *destroys* them), rules (10)–(13) removes the cuts (=an introduction followed by the corresponding elimination). The rule (su) clearly possesses a *principal* formula: its right premiss. Also the deduction leading to the left premiss of a (su)-rule can be named the *auxiliary* deduction. Therefore rule (9) decreases the complexity of the auxiliary deductions. This explains the origin of each rule. But notice that we do not have yet any semantics for these reductions.

Proposition 2.2 *The rewriting system RE_1 is strongly normalizing and Church-Rosser.*

Proof. The proof of local confluence is a tedious critical pair checking. The proof of well-foudedness uses a lexicographic ordering. Firstly the number of rules (su) decreases under reduction. Hence in an infinite sequence of reductions, their number is eventually constant and rules (6), (8), (12) and (13) can no longer be applied. The same argument applies to the number of (u)-inferences and rule (7) can no longer be applied. Then the sum over rules (su) of the number of rules (t) or (u) below this (su)-rule decreases, hence is eventually constant and rules (10), (11) can no longer be applied. Call a substitution *auxiliary* iff its conclusion is the left premiss of a substitution. Then the sum over auxiliary substitutions of the maximum of the number of rules (su) and (u) above them along some branch decreases under reductions, is eventually constant, afterwards rule (9) can no longer be applied. Next, we count the number of (t)-rules whose conclusion is the right premiss of a (t)- or (u)-rule. This number decreases and this eliminates rules (4) and (5). The remaining rules are commuting rules, which are easily seen to be terminating. This contradicts the existence of an infinite sequence of reductions. □

Naturally, this termination proof gives some hint on the structure of normal proofs. The structure of their inference tree is given by the linear grammar:

$$\begin{aligned} \mathcal{P} &::= \mathcal{A} | SU(\mathcal{A}, \mathcal{P}) \\ \mathcal{A} &::= \mathcal{D} | T(\mathcal{A}, \mathcal{D}) \\ \mathcal{D} &::= \mathcal{E} | U(\mathcal{A}, \mathcal{D}) \end{aligned}$$

where the non-terminals $\mathcal{P}$, $\mathcal{A}$ and $\mathcal{E}$ represent respectively the deductions, the auxiliary deductions, and the set of axioms (up to symmetry for strict axioms). The terminals SU, T and U represent the inference rules, their argument being their premisses. Each inference rule (except the symmetry rule) eliminates a variable. Call this variable the proper variable of the inference step. If one of the two occurrences of the proper variable is extracted from a non-strict right-hand side in the axioms, then this extraction is performed by a sequence (right branch) of consecutive instances of (u)-rules. For (su)-free deductions, we have a subformula property [4]: both sides of the conclusion are subterms of the axioms. The analogy with cut-free proofs in first-order sequent calculus [4] of normal deductions is stronger: a cut-free proofs splits the axioms by elimination rules, then recombines the pieces together by introduction rules. Here, the unification rule corresponds to elimination, the substitution rule to introductions.

We now define the trips associated to proofs. This brings us closer to the linear logic of Girard [7]. These are sets of paths in the term dag $\mathcal{D}(\mathcal{E})$ augmented with one edge per equation in $\mathcal{E}$, between the vertices associated to the members of equations (cf. Fig. 1). Call this graph $\mathcal{D}^+(\mathcal{E})$. It will be convenient to assume for a while that $\mathcal{D}^+(\mathcal{E})$ is unoriented. Formally a path is a sequence of edges. We will define them unambiguously by sequences of vertices. Trips are inductively defined on proofs. If $\mathcal{D} \vdash x = M$, there will be two trips by occurrence in $\mathcal{O}(M)$, one being the inverse of the other, plus some "internal" trips. Notice first on the following example that the hole occurrences in rules (su) and (u) are necessarily part of the inference:

$$u\,\dfrac{su\,\dfrac{z=z \quad x=f(z,z)}{x=f(z,z)} \qquad x=f(f(u,v),f(u,v))}{z=f(u,v)} \qquad u\,\dfrac{su\,\dfrac{z=z \quad x=f(z,z)}{x=f(z,z)} \qquad x=f(f(u,v),f(u,v))}{z=f(u,v)} \qquad (\mathcal{D}_0)$$

It follows that, for any rule and any occurrence in its conclusion, there exists a unique associated occurrence in the premisses. Let $\mathcal{D}$ be some deduction, its trips are:

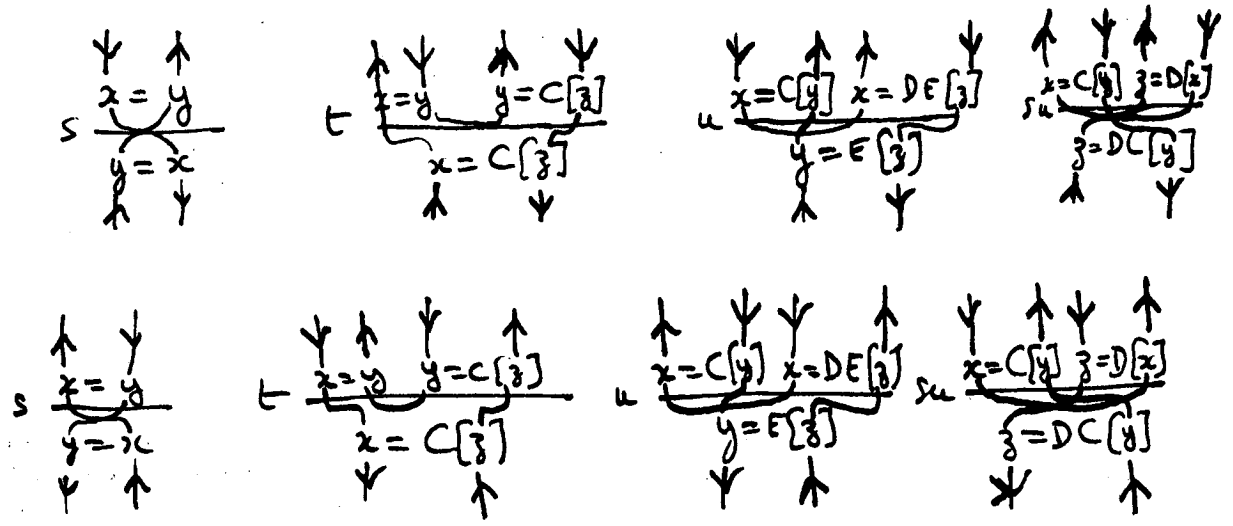

Figure 2: Traveling an inference rule

1. $\mathcal{D}$ is an axiom instance $x = M$, the path p_O, $O \in \mathcal{O}(M)$, starts with the edge between $V(x)$ and $V(M)$, then down in the term dag of M to the subterm M/O. We put $\mathcal{T}(\mathcal{D}) = \{p_O, \overline{p_O} \mid O \in \mathcal{O}(M)\}$, where $\overline{p}$ denotes the inverse path of p.

2. $\mathcal{D}$ ends with a symmetry rule, define $\mathcal{T}(\mathcal{D}) = \mathcal{T}(\mathcal{D}')$, with $\mathcal{D}$:

$$s\,\frac{\begin{matrix}\mathcal{D}'\\ x = y\end{matrix}}{y = x}$$

3. $\mathcal{D}$ ends with a transitivity rule:

$$t\,\frac{\begin{matrix}\mathcal{D}' & \mathcal{D}''\\ x = y & y = M\end{matrix}}{x = M}$$

Let p be the path of $\mathcal{T}(\mathcal{D}')$ associated to the unique occurrence ϵ in $\mathcal{O}(y)$ from $V(x)$ to $V(y)$ and, for an occurrence O in $\mathcal{O}(M)$, let p_O be the path of $\mathcal{T}(\mathcal{D}'')$ from $V(y)$ to $V(M/O)$, associated to O, we define $q_O = p; p_O$ (path concatenation) and put

$$\mathcal{T}(\mathcal{D}) = \mathcal{T}(D')) \cup \mathcal{T}(\mathcal{D}'') \cup \{q_O, \overline{q_O} \mid O \in \mathcal{O}(M)\} - \{p, \overline{p}, p_O, \overline{p_O} \mid O \in \mathcal{O}(M)\}.$$

4. $\mathcal{D}$ ends with a unification rule:

$$u\,\frac{\begin{matrix}\mathcal{D}' & \mathcal{D}''\\ x = C[y] & x = D[M]\end{matrix}}{y = M}$$

Let $O \in \mathcal{O}(M)$. This occurrence defines a unique path p_O in $\mathcal{D}''$ from $V(x)$ to $V(M/O)$. Let p be the path of $\mathcal{T}(\mathcal{D}')$ associated to the occurrence O_C from $V(y)$ to $V(x)$. We define

$$\mathcal{T}(\mathcal{D}) = \mathcal{T}(D') \cup \mathcal{T}(\mathcal{D}'') \cup \{p; p_O, \overline{p; p_O} \mid O \in \mathcal{O}(M)\} - \{p, \overline{p}, p_O, \overline{p_O} \mid O \in \mathcal{O}(M)\}$$

5. $\mathcal{D}$ ends with a substitution rule:

$$su\ \frac{\begin{array}{cc}\mathcal{D}' & \mathcal{D}'' \\ x = M & y = C[x]\end{array}}{y = C[M]}$$

We have two cases according to $O \in \mathcal{O}(C[M])$. First O_C is prefix of O, say $O = O_C O'$. Let $\mathcal{O}_1$ be the set of such occurrences. Let p_O be the path of $\mathcal{T}(\mathcal{D}')$ associated to the occurrence O' of M from $V(x)$ to $V(M/O')$, and p'_O be the path of $\mathcal{T}(\mathcal{D}'')$ associated to O_C from $V(y)$ to $V(C[x]/O_C)$ We define $q_O = p'_O; p_O$. Second O_C is not a prefix of O, also $O \in \mathcal{O}(C[x])$ and $O \neq O_C$. Let $\mathcal{O}_2$ be the set of such occurrences. Let p_O to be the path of $\mathcal{T}(\mathcal{D}'')$ associated to O from $V(y)$ to $V(C[x]/O)$. In this case we also denote p_O by q_O. Then

$$\mathcal{T}(\mathcal{D}) = \mathcal{T}(D') \cup \mathcal{T}(\mathcal{D}'') \cup \{q_O, \overline{q_O} \mid O \in \mathcal{O}(M)\} - \left((\bigcup_{O \in O_1} \{p_O, \overline{p_O}, p'_O, \overline{p'_O}\}) \cup (\bigcup_{O \in O_2} \{p_O, \overline{p_O}\}) \right)$$

The meaning of these trips will be clarified in the next section. Trips should be viewed geometrically as shown on Fig. 2. The reader may find the trips of the deductions in the Introduction, of $\mathcal{D}_0$ above and $\mathcal{D}_1$ below; and experiments the existence of long-trips that visit once each member of each equation in a deduction, of short-circuits that fail to reach the conclusion of the deduction, and of partial trips.

$$su\ \frac{v = f(c,d) \quad su\ \dfrac{u = f(a,b) \quad su\ \dfrac{x = f(u,v) \quad y = f(x,z)}{y = f(f(u,v),z)}}{y = f(f(a,b),v),z)}}{y = f(f(f(a,b),f(c,d)),z)} \qquad (\mathcal{D}_1)$$

Here are some immediate properties of trips:

- To each path in $\mathcal{T}(\mathcal{D})$ is associated at least one subdeduction of $\mathcal{D}$ defined to be (one of) the largest subdeduction containing it.
- A member of some conclusion of an inference is visited at most once by a path.
- If a path visits some member of the conclusion of some inference, it also visits the other member.
- Every path visits at least one axiom.
- If a path visits all axioms of $\mathcal{D}$, it visits all members of conclusions of inferences in $\mathcal{D}$.

Definition 2.1 *Let $\mathcal{D}$ be some deduction.*
A path in $\mathcal{T}(\mathcal{D})$ is a short-circuit iff it is a path of some proper subdeduction of $\mathcal{D}$.
A path in $\mathcal{T}(\mathcal{D})$ is a long-path iff each left-hand side of each axiom is visited.
A path in $\mathcal{T}(\mathcal{D})$ is trivial iff it visits a single axiom.

Examples of short-circuits can be found in $\mathcal{D}_0$, of trips that are neither short-circuits nor long-paths in $\mathcal{D}_1$.

Lemma 2.3 *Let $\mathcal{D}$ be some deduction. If $\mathcal{D}$ possesses a long-trip, then all other paths in $\mathcal{T}(\mathcal{D})$ are trivial.*

Proof. Let $\mathcal{D} \vdash x = M$. If the occurrence $O \in \mathcal{O}(M)$ defines a long-trip, any occurrence having O as prefix also defines a long-trip. Hence we can define minimal long-trip occurrences. By induction on $\mathcal{D}$ it is easily established that there exists a *minimum* long-trip occurrence O_m, the only slight difficulty being with the rule (su). Finally, an induction establishes that if $O' \in \mathcal{O}(M)$ possesses O_m as prefix, then it defines a long-trip, otherwise it defines a trivial path. Naturally, O_m is ϵ when $\mathcal{D}$ ends with a (u)-rule, and O_C if it ends with a substitution whose right premiss context is $C[\_]$. □

Definition 2.2 *A deduction $\mathcal{D}$ is sequential iff $\mathcal{T}(\mathcal{D})$ possesses a long-trip.*

Trips modelizes data-flow in proofs. We pursue this analogy with linear logic. First a discrete time can be defined on trips. The proof of Lemma 2.3 conveys the idea of *empire* of an occurrence (subformula) of Definition 2.9.3 of [7]. The trips have a Question/Response structure, as in linear logic. If a path is "drawn" on a proof, we see an *alternation* of upwards and downwards moves. Moving upwards can be interpreted as a question, moving downwards as a response. We have inversion of Q/R on axioms, and when identifying the two occurrences of the proper variable of a two-premiss rule. This allows the definition of Positive/Negative or Left/Right occurrences along a path. Remember first that to each vertex of the graph $\mathcal{D}^+(\mathcal{E})$ corresponds a unique term occurrence in $\mathcal{E}$.

Definition 2.3 *Let $\mathcal{D} \vdash^{\mathcal{E}}_{LE_1} x = M$ be a deduction and p be some path in $\mathcal{T}(\mathcal{D})$ associated to an occurrence $O \in \mathcal{O}(M)$. Without loss of generality, we assume that p is not a short-circuit (otherwise consider the minimum subdeduction containing the path). We define Positive and Negative occurrences in $\mathcal{A}(\mathcal{D})$ according to p by induction on $\mathcal{D}$:*

- *$\mathcal{D}$ is an axiom instance $x = M$. The first vertex of p defines a positive occurrence in $\mathcal{E}$, the last vertex a negative occurrence in $\mathcal{E}$.*
- *$\mathcal{D}$ ends with a symmetry rule:*

$$s\,\dfrac{\begin{matrix}\mathcal{D}'\\ x = y\end{matrix}}{y = x}$$

  To p is associated a unique path $q \in \mathcal{T}(\mathcal{D}')$. Define the positive/negative occurrences of p as in q.

- *$\mathcal{D}$ ends with a transitivity rule:*

$$t\,\dfrac{\begin{matrix}\mathcal{D}' & \mathcal{D}''\\ x = y & y = M\end{matrix}}{x = M}$$

  The path p defines a path p' in $\mathcal{D}'$ and a path p'' in $\mathcal{D}''$. The positive (negative) occurrences of p are the positive (negative) occurrences of p' and p''.

- *$\mathcal{D}$ ends with a unification rule:*

$$u\,\dfrac{\begin{matrix}\mathcal{D}' & \mathcal{D}''\\ x = C[y] & x = D[M]\end{matrix}}{y = M}$$

  The path p defines a path p' in $\mathcal{D}'$ and a path p'' in $\mathcal{D}''$. The positive (negative) occurrences of p are the positive (negative) occurrences of p' and p''. Notice that the path in $\mathcal{D}'$ goes from $V(y)$ to $V(x)$.

- *$\mathcal{D}$ ends with a substitution rule:*

$$su\,\dfrac{\begin{matrix}\mathcal{D}' & \mathcal{D}''\\ x = M & y = C[x]\end{matrix}}{y = C[M]}$$

  We have two cases: if p defines a unique path in $\mathcal{T}(\mathcal{D}'')$, same as the symmetry rule; if p defines a path in $\mathcal{T}(\mathcal{D}')$ and a path in $\mathcal{T}(\mathcal{D}'')$, same as the transitivity rule.

Notice that it is possible that some occurrence receives multiple positive and negative labels. Some immediate properties (established by easy inductions):

- The first occurrence of a path is positive, its last one is negative.
- The number of positive occurrences is equal to the number of negative occurrences.
- Assume that the occurrences of the proper variable of a binary rule belongs to some path. One of the two occurrences of this proper variable is labeled positively, the other one negatively.

The intuition behind this definition is the usual meaning of Positivity and Negativity in predicate calculi: positive occurrences contribute to the "strength" of the first occurrence of a path, negative ones to the last one'strength. Examples can be found at the end of the paper. Naturally, unification logic, sharing the symmetries of linear logic, discovered in an independant context, confirms the accuracy of the intuitions behind these logics.

In connection with the so-called *characterization* problem for theorems in LE_1 [23], having disclosed the trip structure on proofs, we would like to know if these trips contain enough information in order to recover the proof. In fact, we take the trips as the (operational) semantics of proofs and give an answer in the next section, in the spirit of proof-nets and Theorem 2.9 of [7].

Before extending this system, we apply it to our original motivation, which was to find a complete equational characterisation of the cycles in $\mathcal{G}(\mathcal{E})$, $\mathcal{E}$ a set of equations. Such a cycle is "represented" by a deduction $\mathcal{D} \vdash^{\mathcal{E}} x = C[x]$, where the path in $\mathcal{G}(\mathcal{E})$ specified by the vertex of x and the occurrence O_C is a fundamental cycle in the graph-theoretic sense [1]. In order to find these deductions $\mathcal{D}$, a notion of minimality for proofs, subtler than the syntactic rewriting, is required.

Definition 2.4 *Let $\mathcal{D} \vdash^{\mathcal{E}} x = C[x]$ be some cyclic deduction. The deduction $\mathcal{D}$ is minimal if no inference rule identifies two equal occurrences of some variable in the axioms $\mathcal{E}$ and the two occurrences of x in the conclusion of the deduction are distinct.*

Notice that this definition applies to non-cyclic proofs when we remove its last clause. An example of a non-minimal deduction is given below for the equations $x = f(x,y)$ and $x = f(z,t)$, with the following normal derivation:

$$\dfrac{\dfrac{\dfrac{}{x = f(x,y)} \quad \dfrac{}{x = f(z,t)}}{x = z}\, u \quad \dfrac{\dfrac{\dfrac{}{x = f(z,t)} \quad \dfrac{}{x = f(x,y)}}{z = x}\, u \quad \dfrac{}{x = f(x,y)}}{z = f(x,y)}\, t}{z = f(z,y)}\, su \qquad (\mathcal{D}_2)$$

Both occurrences of x, proper variable of the substitution rule, and both occurrences of z in the conclusion are equal occurrences in the axioms. The intuition underlying this definition assumes that each inference step encodes an elementary computation or communication. In case of an inference step performed by identifying the *same* occurrences of a variable, the result of the computation does not give any essentially new information. Alternatively, consider the positive/negative interpretation: the positive occurrence asks some question... which is answered by the *same* occurrence.

Proposition 2.4 *Let $\mathcal{D} \vdash^{\mathcal{E}} x = C[x]$ be some cyclic deduction, one can associate to $\mathcal{D}$ a minimal deduction $\mathcal{D}' \vdash^{\mathcal{E}} y = D[y]$ such that the cycles $(V(x), O_C)$ and $(V(y), O_D)$ are equal in $\mathcal{G}(\mathcal{E})$.*

Proof. See [18]. The proof is constructive and rests upon a delicate analysis of the trips associated to proofs and the removal of unneeded détours. Notice that the conclusion may not be preserved. Something else is preserved that should be close to the trips. □

Next we want to collect the set of deductions in normal form and minimal, given the set of axioms $\mathcal{E}$.

Proposition 2.5 *Let $\mathcal{E}$ be a finite set of equations. The set $C(\mathcal{E})$ of minimal and normal cyclic deductions $\mathcal{D}$ associated to fundamental cycles of $\mathcal{A}(\mathcal{D})$ is finite and can be effectively computed. Every deduction in $C(\mathcal{E})$ is sequential.*

Proof. The proof proceeds by showing that there can be no repetition of proper variables in any branch of such a proof. □

For an application of this Proposition to second-order unification, see the end of the paper. To conclude this section, we explain how this relates to the sequentiality of unification.

Definition 2.5 *The set E' is a one-step linearization of a set E of equations iff E' is obtained from E by renaming some occurrence of a variable that occurs at least twice in E. The set E' is a linearization of E iff there is a non-void sequence of one-step linearizations from E to E'. An elementary cyclic set E is a set of equations such that:*

1. *$\mathcal{G}(E)$ contains at least one cycle,*
2. *the graph $\mathcal{G}(E')$ is a dag for every linearization E' of E,*

It is established in [18] that the graph of an elementary cyclic set contains a unique cycle, and there exists a *canonical* minimal and normal deduction associated to any elementary cyclic set E. This canonicity is modulo a cyclic permutation of the auxiliary deductions, which is interpreted by the fact that the correct invariant is the associated long-trip: it is immediate that two cyclic deductions obtained by a cyclic permutation of their auxiliary deductions define the same long-trip which is actually a circuit.

Let $\mathcal{E}$ be a set of equations. An exhaustive search gives all the elementary cyclic sets imbedded in $\mathcal{E}$. Call the set of their canonical deductions $\mathcal{B}(\mathcal{E})$. Now trivially, $\mathcal{E}$ is unifiable iff $\mathcal{B}(\mathcal{E})$ is empty. But we have something more: the set $\mathcal{B}(\mathcal{E})$ encodes the minimal amount of computation that has to be performed in order to prove that $\mathcal{E}$ is ununifiable. Formalizing this argument would lead us too far. We conjecture that some results about the complexity of subclasses of the unification problem [3] follow easily from the description of their base $\{\mathcal{B}(\mathcal{E}) \mid \mathcal{E} \in C\}$. For example, if the number of repeated variables is bounded, the argument in the proof of Proposition 2.5 shows that the height of the deductions in the base $\mathcal{B}(\mathcal{E})$ is bounded uniformly on the class. Hence an efficient parallel algorithm can be designed, showing that this class is in NC while unification is P-complete. We now turn to the generalization of the logic to full equational logic.

3 Generalization to Full Equational Logic

We now address the problem of normalization for equational logic. Naturally our main objective is to keep the properties of both LR_1 and the sequent calculus. This implies that we split the classical rules of equational logic into introductions and eliminations. Besides the three rules of reflexivity, symmetry and transitivity, we have the two rules of replacement and substitution [9]:

$$\frac{M = M' \quad N = N'}{f(M,N) = f(M',N')} \qquad \frac{M = N}{\sigma(M) = \sigma(N)}$$

Obviously these two rules are introduction rules in the sense that they create new terms; but from the view point of the subformula property, they are unsatisfactory, and should be written as:

$$\frac{f(M,N) = f(M,N) \quad M = M' \quad N = N'}{f(M,N) = f(M',N')} \qquad \frac{M = N \quad x_1 = M_1 \cdots x_n = M_n}{\sigma(M) = \sigma(N)}$$

This syntax is unsatisfactory as we would like to have a *single* introduction rule. But passing from the first rules to these ones renders the two rules quite similar. Also we drop the requirement that substitutions should substitute only variables, and that replacement should replace only under function symbols. Notice that the definition of the substitution is now included in the premissses, which after all is quite natural. This, together with the requirement of atomicity, gives the two symmetric rules:

$$IL\ \frac{M = N \quad C[N] = O}{C[M] = O} \qquad IR\ \frac{M = C[N] \quad N = O}{M = C[O]}$$

In order to establish that we get full equational logic, we simulate each group of rules by the other one. From the premisses of IL and the reflexivity axiom, we build by successive applications of the replacement rules

the equality $C[M] = C[N]$ and conclude with transitivity :

$$\frac{\dfrac{M = N}{C[M] = C[N]} \quad C[N] = O}{C[M] = O}$$

And similarly for the rule IR. The derivation simulating replacement is

$$IR\frac{IR\dfrac{f(M,N) = f(M,N) \quad M = M'}{f(M,N) = f(M',N)} \quad N = N'}{f(M,N) = f(M',N')}$$

We left to the reader the simulation of the substitution rule. As the mathematically meaningful properties of the system LR_1 follows from the duality introduction-elimination, we augment our logic with an elimination rule. Looking at the elimination rule of LR_1:

$$u\frac{x = C[y] \quad x = D[M]}{y = M}$$

we see that we should replace the variables x and y by arbitrary terms. But there is a subtle point in case here: the reader may check that in designing the rules IR and IL with a perfect symmetry, we have "untwisted" the trip through the (su)-rule. Also, in connection with the characterization problem, we may wonder if this untwisting can be applied to the elimination rule. This is true and gives us the rule:

$$E\frac{C[M] = N \quad N = D[O]}{M = O}$$

provided that the two contexts are equivalent. Naturally, this rules expresses the freeness of our logic, in the sense that unification solves equations in free algebras. The presence of this rule must be justified in an equational logic. First of all it is well-known that a logic will present some interest and strength iff it contains both introduction and elimination rules. Here, elimination means unification closure and Logicians were not interested in its properties. As is well-known, Computer Scientists have interest in unification since a long time [22]. By the way this explain why the present study was not already present in the literature. The rewriting system presented below may be restricted to classical equational logic (i.e. without rule E), however it looses much of its interest. A pleasant property of the rules IL, IR and E is that their *intersection* (trivialize all contexts) is the transitivity rule, which could be named the *neutral* rule. Hence our system is:

$$LE_2 \left\{ \begin{array}{lll} R\dfrac{}{M = M} & & S\dfrac{M = N}{N = M} \\ & E\dfrac{C[M] = N \quad N = D[O]}{M = O} & \\ IL\dfrac{M = N \quad C[N] = O}{C[M] = O} & & IR\dfrac{M = C[N] \quad N = O}{M = C[O]} \end{array} \right.$$

As in the previous section, we can define the set of trips associated to a deduction. The Reflexivity axiom will not be used in our investigations of unification. It is important to note the trivialization of the trips due to the untwisting of the rules. However, for practical computations on unification problems, the system LE_1 keeps its interest. We left the reader check that every proof in LE_1 can be translated into LE_2. The converse is false as the logic LE_2 allows us to prove equations of the form $M = N$, M not a variable.

We now give the rewriting system associated to the logic LE_2. As is usual for sequent calculi, we separate the rules according to specific groups, and enlarge the systems as we require stronger properties on

the normal forms. It should be already clear that some of these groups will break the L/R symmetry. To be convinced of this fact, remember that substitution is a highly parallel operation, while in LE_2, we are forced to sequentialize it. Our requirement to sequentialize the normal forms will lead us to an infinite rewriting system. Also the rules are rules schemes, the context notation should be read as a meta-level notation for a set of rules.

The first group lifts the symmetry rule and deserves no more comment, it is obviously Church-Rosser and Noetherian. See the first three rules in Section 2, which gives here four rules. The second group is the principal one and is quite self-justified. It contains ten rules. A small justification based on the trip semantics presents each rule.

$$IL\dfrac{IL\dfrac{M=N\quad C[N]=O}{C[M]=O}\qquad D[O]=P}{DC[M]=P}\quad\Longrightarrow\quad IL\dfrac{M=N\qquad IL\dfrac{C[N]=O\quad D[O]=P}{DC[N]=P}}{DC[M]=P}\tag{5}$$

$$IR\dfrac{M=C[N]\qquad IR\dfrac{N=D[O]\quad O=P}{N=D[P]}}{M=CD[P]}\quad\Longrightarrow\quad IR\dfrac{IR\dfrac{M=C[N]\quad N=D[O]}{M=CD[O]}\qquad O=P}{M=CD[P]}\tag{6}$$

In accordance with the previous section, we consider the left (right) premiss of an IL (IR) rule as an intermediate result. The rules then decreases the complexity of auxiliary deductions. They have dual rules:

$$IL\dfrac{M=C[N]\qquad IL\dfrac{N=O\quad DC[O]=P}{DC[N]=P}}{D[M]=P}\quad\Longrightarrow\quad IL\dfrac{IR\dfrac{M=C[N]\quad N=O}{M=C[O]}\qquad DC[O]=P}{D[M]=P}\tag{7}$$

$$IR\dfrac{IR\dfrac{M=CD[N]\quad N=O}{M=CD[O]}\qquad D[O]=P}{M=C[P]}\quad\Longrightarrow\quad IR\dfrac{M=CD[N]\qquad IL\dfrac{N=O\quad D[O]=P}{D[N]=P}}{M=C[P]}\tag{8}$$

The left-hand sides perform two nested substitutions, but in *reverse* order: we substitute the imermost term, then the outermost one. Notice that these redexes do not exist in LE_1. As our rules IL, IR and E have a non-empty intersection, some conditions must be imposed on these four rules. Otherwise, say rules (5) and (7) yield infinite reductions. These conditions are $C[\_]$ non-trivial for rules (5) and (7), $D[\_]$ non-trivial for rules (6) and (8). These conditions do not make our system a conditional rewrite system [14] as they are rule schemes defining one rule per instance of $D[\_]$ (other finite presentations are possible by formalizing a context calculus). This was the subgroup of rules that orders the introduction rules. The next subgroup is the cut-elimination group. Our system LE_2 being expressed in natural deduction style [20] rather than in sequent style [5], a cut is an introduction immediatly followed by the corresponding elimination.

$$E\dfrac{IL\dfrac{C[M]=N\quad D[N]=O}{DC[M]=O}\qquad O=EF[P]}{M=P}\quad\Longrightarrow\quad E\dfrac{C[M]=N\qquad E\dfrac{D[N]=O\quad O=EF[P]}{N=F[P]}}{M=P}\tag{9}$$

$$E\dfrac{CD[M]=N\qquad IR\dfrac{N=E[O]\quad O=F[P]}{N=EF[P]}}{M=N}\quad\Longrightarrow\quad E\dfrac{E\dfrac{CD[M]=N\quad N=E[O]}{D[M]=O}\qquad O=F[P]}{M=P}\tag{10}$$

$$E\dfrac{IL\dfrac{M=N\quad CD[N]=O}{CD[M]=O}\qquad O=E[P]}{D[M]=P}\quad\Longrightarrow\quad IL\dfrac{M=N\qquad E\dfrac{CD[N]=O\quad O=E[P]}{D[N]=P}}{D[M]=P}\tag{11}$$

$$E\,\dfrac{C[M]=N \quad IR\,\dfrac{N=DE[O] \quad O=P}{N=DE[P]}}{M=E[P]} \quad\Longrightarrow\quad IR\,\dfrac{E\,\dfrac{C[M]=N \quad N=DE[O]}{M=E[O]} \quad O=P}{M=E[P]} \tag{12}$$

The conditions to be imposed to these rules by the coherence of the rewriting system are: rule (9): $D \neq \emptyset$, rule (10): $E \neq \emptyset$, rule (11): $CD \neq \emptyset$, rule (12): $DE \neq \emptyset$. This system (rules 1 up to 12) is not locally confluent. Several completions are possible. They will differ according to the semantics we have in mind, by identifying more or less proofs. Recall here that we want a data-flow semantics. Up to now, our rules possess a perfect L/R symmetry. An exhaustive search on two inference rules configurations gives us new equations. First:

$$IL\,\dfrac{M=N \quad IR\,\dfrac{C[N]=D[O] \quad O=P}{C[N]=D[P]}}{C[M]=D[P]} \quad=\quad IR\,\dfrac{IL\,\dfrac{M=N \quad C[N]=D[O]}{C[M]=D[O]} \quad O=P}{C[M]=D[P]} \tag{13}$$

$$E\,\dfrac{C[M]=D[N] \quad IL\,\dfrac{N=O \quad D[O]=E[P]}{D[N]=E[P]}}{M=P} \quad=\quad E\,\dfrac{IR\,\dfrac{C[M]=D[N] \quad N=O}{C[M]=D[O]} \quad D[O]=E[P]}{M=P} \tag{14}$$

Notice that these equations are not consequences of the previous ones. We therefore add these two equations and orientate them from left to right, quite arbitrarily. No conditions are imposed on the rules. Notice that the choice of this orientation breaks the L/R symmetry, but is unavoidable if we want a simple complete system. Further, the introduction of these rules is justified by our semantics.

The next equations asserts the identity of two proofs that perform successive unnested introductions.

$$IR\,\dfrac{IR\,\dfrac{M=C[N_1,N_2] \quad N_2=O_2}{M=C[N_1,O_2]} \quad N_1=O_1}{M=C[O_1,O_2]} \quad=\quad IR\,\dfrac{IR\,\dfrac{M=C[N_1,N_2] \quad N_1=O_1}{M=C[O_1,N_2]} \quad N_2=O_2}{M=C[O_1,O_2]} \tag{15}$$

$$IL\,\dfrac{M_1=N_1 \quad IL\,\dfrac{M_2=N_2 \quad C[N_1,N_2]=O}{C[N_1,M_2]=O}}{C[M_1,M_2]=O} \quad=\quad IL\,\dfrac{M_2=N_2 \quad IL\,\dfrac{M_1=N_1 \quad C[N_1,N_2]=O}{C[M_1,N_2]=O}}{C[M_1,M_2]=O} \tag{16}$$

According to the trip semantics, the members of these equations are equal, provided that we dont want to order paths. Intuitively, we do not mind the exact order of a series of subtitutions in some term. We orientate these equations from left to right, arbitrarily but in accordance with the tradition of leftmost-outermost computation (which is of little meaning here).

Now the critical pair phenomenon gives us four new equations. As we give them, they are not equational consequences of the above theory. But non-confluent critical pairs become confluent if we add these rules.

$$E\,\dfrac{IL\,\dfrac{M=N \quad C[O,N]=P}{C[O,M]=P} \quad P=E[Q]}{O=Q} \quad\Longrightarrow\quad E\,\dfrac{C[O,N]=P \quad P=E[Q]}{O=Q} \tag{17}$$

$$E\,\dfrac{IL\,\dfrac{M=N \quad C[N,O]=P}{C[M,O]=P} \quad P=E[Q]}{O=Q} \quad\Longrightarrow\quad E\,\dfrac{C[N,O]=P \quad P=E[Q]}{O=Q} \tag{18}$$

$$E\,\dfrac{E[Q]=M \quad IR\,\dfrac{M=C[N,O] \quad N=P}{M=C[P,O]}}{Q=O} \quad\Longrightarrow\quad E\,\dfrac{E[Q]=M \quad M=C[N,O]}{Q=O} \tag{19}$$

$$E\dfrac{E[Q]=M \quad IR\dfrac{M=C[O,N] \quad N=P}{M=C[O,P]}}{Q=O} \quad\Longrightarrow\quad E\dfrac{E[Q]=M \quad M=C[O,N]}{Q=O} \tag{20}$$

Notice that we do not respect the semantics here, these four rules erase some paths, namely the short-circuits. But we are not interested in stupid proofs: all the left-hand sides above possess a short-circuit, or subproofs that undo (eliminate) what they had previously done (introduction). Also, it is natural at this point to add a set of rules rewriting any non-trivial proof $\mathcal{D} \vdash M = M$ into an instance of the reflexivity axiom. The remaining critical pairs are mere generalizations of the existing ones. It is important to note that these new rules, added by critical pair completion, are due to the symmetry breaking (rules 13 and 14). We first have the three rules:

$$IL\dfrac{IR\dfrac{IL\dfrac{M=N \quad C[N]=D[O]}{C[M]=D[O]} \quad O=Q}{C[M]=D[Q]} \quad ED[Q]=P}{EC[M]=P}$$

$$\Longrightarrow\quad IL\dfrac{M=N \quad IL\dfrac{IR\dfrac{C[N]=D[O] \quad O=Q}{C[N]=D[Q]} \quad ED[Q]=P}{EC[N]=P}}{EC[M]=P} \tag{21}$$

$$E\dfrac{IR\dfrac{IL\dfrac{C[M]=N \quad D[N]=E[P]}{DC[M]=E[P]} \quad P=O}{DC[M]=E[O]} \quad E[O]=FG[Q]}{M=Q}$$

$$\Longrightarrow\quad E\dfrac{C[M]=N \quad E\dfrac{IR\dfrac{D[N]=E[P] \quad P=O}{D[N]=E[O]} \quad E[O]=FG[Q]}{N=G[Q]}}{M=Q} \tag{22}$$

$$E\dfrac{IR\dfrac{IL\dfrac{M=N \quad CD[N]=E[P]}{CD[M]=E[P]} \quad P=O}{CD[M]=E[O]} \quad E[O]=F[Q]}{D[M]=Q}$$

$$\Longrightarrow\quad IL\dfrac{M=N \quad E\dfrac{IR\dfrac{CD[N]=E[P] \quad P=O}{CD[N]=E[O]} \quad E[O]=F[Q]}{D[N]=Q}}{D[M]=Q} \tag{23}$$

The first one is a critical pair between rules (6) and (7), the second one between rules (9) and (13) or (14), the last one between rules (11) and (13) or (14). We also have the two rules:

$$E\dfrac{IR\dfrac{IL\dfrac{M=N \quad C[O,N]=D[P]}{C[O,M]=D[P]} \quad P=Q}{C[O,M]=D[Q]} \quad D[Q]=E[R]}{O=R}$$

$$\Longrightarrow \quad E\,\dfrac{IR\,\dfrac{C[O,N]=D[P]\quad P=Q}{C[O,N]=D[Q]}\qquad D[Q]=E[R]}{O=R} \tag{24}$$

$$E\,\dfrac{IR\,\dfrac{IL\,\dfrac{M=N\quad C[N,O]=D[P]}{C[M,O]=D[P]}\qquad P=Q}{C[M,O]=D[Q]}\qquad D[Q]=E[R]}{O=R}$$

$$\Longrightarrow \quad E\,\dfrac{IR\,\dfrac{C[N,O]=D[P]\quad P=Q}{C[N,O]=D[Q]}\qquad D[Q]=E[R]}{O=R} \tag{25}$$

These critical pairs arose from superpositions between rules (15)–(16) and rules (13)–(14). Finally, each rule (21)–(25) has a generalization: for each integer n, we have a corresponding rule with a stack of n (IR)-rules, the conclusion of the ith-one being the left premiss of the $(i+1)$th one. Notice that these five sequences of rules are generalizations of the rules (5), (9), (11), (17) and (18), due to the symmetry breaking. These rules have critical pairs computed from deductions of the form

$$IL\,\dfrac{IL\,\dfrac{\vdots}{\cdots}\quad IR\,\dfrac{\vdots}{\cdots}}{\cdots}$$

Also, these rules can be noted $(5.n)$, $n \geq 0$ and our system RE_2 includes rules $(1) - -(4)$, $(6) - -(8)$, (10), $(12) - -(16)$, $(19) - -(20)$, $(5.n)$, $(9.n)$, $(11.n)$, $(17.n)$ and $(18.n)$, $n \geq 0$. The reader may check that the critical pairs from rules (8), (15) and (13), (14) are confluent.

Theorem 3.1 *The rewriting system RE_2 is strongly normalizing and Church-Rosser.*

Proof. The proof of local confluence is a boring (and long!) critical pair checking. The proof of well-foundedness follows the lines of the proof of Proposition 2.2, i.e. a lexicographic ordering does the job. □

We now describe the syntactical properties of the normal forms. A pattern is a sequence of two or more rules, but with the terms in the premisses and in the conclusion unspecified. A pattern is complete if all its possible instances appear in RE_2. It is almost immediate that all the patterns in left-hand sides are complete:

- Patterns of rules (5) and (6) are both self-complete.
- Patterns of rules (7) and (8), together with the right-hand sides of rules (5) and (6) and the left-hand sides of rules (15) and (16), are complete.
- Patterns of rules (9) – (12), together with the left-hand sides of rules (17) – (20), are complete.
- Patterns of rules (13) and (14) are self-complete.

Hence we have enumerated all possible patterns and examined all rules. This allows an easy description of the normal forms. First we observe that, in a normal form, an introduction whose conclusion is the premiss of an elimination (not *the* corresponding elimination, which would give a redex) occurs only as an instance of the right-hand side of rule (14). The leftmost branch of a normal form satisfies:

- No (IL)-rule occurs above an (E)-rule.
- No (IL)-rule occurs above an (IL)-rule and below such a rule we find only (IR)-rules.

Finally, the relative positions of introductions gives the following BNF description of normal forms, with the same conventions as in section 2:

$$\begin{array}{rcl} \mathcal{P} & ::= & \mathcal{R}_0|\mathcal{L}|\mathcal{A} \\ \mathcal{R}_0 & ::= & IR(\mathcal{R}_0,\mathcal{L})|IR(\mathcal{L},\mathcal{L})|IR(\mathcal{R},\mathcal{L}) \\ \mathcal{R} & ::= & IR(\mathcal{R},\mathcal{L})|\mathcal{A} \\ \mathcal{L} & ::= & IL(\mathcal{R},\mathcal{L})|\mathcal{A} \\ \mathcal{A} & ::= & \mathcal{E}|E(\mathcal{R},\mathcal{A})|E(\mathcal{A},\mathcal{A}) \end{array}$$

Of course, this is an incomplete description. We have the constraints on parallel substitutions that should be performed leftmost-outermost. Also, the auxiliary deductions have the properties described in Section 2: no neutral (transitivity) rule has its conclusion which is right premiss of an elimination. This achieves our description of the cut-free deductions. As is apparent from the above grammar, the introductions follow the eliminations, with the exception of an introduction relative to the proper term of an elimination (right-hand side of rule (14)). The rules preserve the trips, except the cancellative rules (17) – (20) and the reflexivity rules, which eliminate the short-circuits. A deduction in normal form does not possess short-circuits. We can define positive/negative occurrences according to a trip, with the same properties as trips of section 2. We also have the corresponding completeness theorem, where $\models^{\mathcal{E}} M = N$ is defined by the equality of the vertices of the terms M and N in the unification graph $\mathcal{G}(\mathcal{E})$.

Theorem 3.2 $\vdash^{\mathcal{E}}_{LE_2} M = N$ *iff there exists some context* $C[\_,\ldots,\_]$ *such that* $M = C[M_1,\ldots,M_n]$, $N = C[N_1,\ldots,N_n]$ *and* $\models^{\mathcal{E}} M_i = N_i$, $i = 1,\ldots,n$.

Proof. Similar to theorem 2.1. □

Let us summarize the results of the paper. In order to classify fix-point equations, with the goal of understanding unification related to higher-order unification, we wanted a normal form result for a partial equational logic (proof-search is easier for proofs in normal form). This disclosed the trip structure which allowed us to generalize the normal form to a full equational logic, with a precise semantics in mind. Now the above theorems tells us that we succeeded in this task, at least syntactically. We hope to include in the final version of the paper the proof of:

- Two distinct sequential deductions in normal form have distinct associated long-trips.
- Given a long-trip, it is possible to recover its associated normal form deduction.

These statements are the analog in LE_2 of the lifting of proof-nets to sequent deductions in linear logic [7]. This will end the analysis of unification by a complete answer to the characterization problem.

Naturally, turning back to our fix-point equations, we want a smaller set than $\mathcal{C}(\mathcal{E})$. Namely we want a representative for each distinct trip [19]. To conclude the paper, we briefly mention how all the above material applies to second-order unification. The most important point here, especially for a Computer Scientist, is that the trips give us the *operational* semantics of deductions. Consider the following set of (1st-order) equations:

$$\mathcal{E}\left\{\begin{array}{rcl} x & = & f(x,y) \\ x & = & f(f(x,z),t) \end{array}\right. \qquad (\mathcal{D}_3)$$

The set $\mathcal{C}(\mathcal{E})$ is equal to the axioms (each axiom defines an elementary cyclic set) plus the derivation:

$$u\,\frac{x = f(x,y) \quad x = f(f(x,z),t)}{x = f(x,z)}$$

The positivity/negativity relative to the circuit of the (sequential) deduction $\mathcal{D}_3$ is displayed below:

$$\begin{array}{rcl} x^- & = & f(x^+,y) \\ x^+ & = & f(f(x^-,z),t) \end{array}$$

Consider now some associated higher-order equations:

1. $X(A) = f(\Pi(X), y)$, $X(B) = f(f(\Pi(X), z), t)$.
2. $X(A) = f(\Pi(X), y)$, $X(A) = f(f(\Pi(X), z), t)$.
3. $X = \lambda y.f(X(A), y)$, $X = \lambda t.f(f(\Pi(X), z), t)$.
4. $X = \lambda a.f(X(a), y)$, $X(A) = f(f(\Pi(X), z), t)$.

The reader can check that all sets, except 2., protect the deduction $\mathcal{D}_3$, but that only the first set protects simultaneously all elements of $\mathcal{C}(\mathcal{E})$. An higher-order cyclic equation is protected iff it has the form:

$$\lambda\overline{\alpha}.\Phi(\overline{X}) = C[\Phi(\overline{Y})]$$

where the arguments $\overline{X}$ and $\overline{Y}$ are pure applicative terms (no abstraction), with bound variables of atomic type (either some variable in $\overline{\alpha}$ or some variable bound by the context $C[\_]$); and iff we have $\overline{X} \gg \overline{Y}$ with: $\overline{A} \gg \overline{B}$ iff there exists an index i such that:

- A_i is not a bound variable while B_i is,
- or $A_i = A(\overline{U})$, $B_i = B(\overline{V})$ and either A is distinct from B or $\overline{U} \gg \overline{V}$.

The intuition behind this definition is that when Φ is an appropriate recursive projector, the equation becomes, e.g.:

$$\lambda\overline{\alpha}.A(\overline{U}) = C[B(\overline{V})]$$

which is no longer a fixed-point equation. A simple size-of-term argument shows that if some non-protected cyclic equation is derivable from a set of higher-order equations, then this set has no solution at all. Hence protecting all cyclic equations at 2nd-order is a *necessary* condition for the existence of solutions. The reader may play with the above examples and see how the operational semantics of proofs handle *communication*: the arguments of the premises are sent to the conclusion by first-order unification, the most general unifying substitution being passed as argument to λ-terms:

$$\cfrac{\cfrac{\lambda a.X(a) = \lambda a.f(X(a), y)}{X(A) = f(X(A), y)}\,\Lambda \qquad X(A) = f(f(\Pi(\lambda a.X(a)), z), t)}{X(A) = f(\Pi(\lambda a.X(a)), z)}\,u$$

Here, the arguments of the proper variable X of the inference have been unified, A being a constant and a a variable. Notice that the set 1. protects the deduction $\mathcal{D}_3$ by ununifiability of the arguments A and B (two distinct constants) of the proper variable X of the inference. Finally, some quite involved examples show that we must consider the base $\mathcal{C}(\mathcal{E})$, protecting $\mathcal{B}(\mathcal{E})$ is not sufficient. Notice the use of the η-rule of λ-calculus above. Naturally, this operational semantics is well-behaved under syntactic conditions, they are met in the type-inference problem. These conditions will be setlled in [19]. Returning to the example of the introduction and applying our criterion, $M = (\lambda x.xx)\ (\lambda xy.xyx)$, we can now reject the structures, where X and $Y(\alpha)$ are 2nd-order types and α a type variable:

$$(\lambda x : \Phi.(xX)x)\ (\Lambda\alpha.\lambda x : \Psi(\alpha), y : \Theta(\alpha).xyx), \qquad (\lambda x : \Phi.xx)\ (\Lambda\alpha.\lambda x : \Psi(\alpha), y : \Theta(\alpha).((xy)Y(\alpha))x),$$

and accept the following ones:

$$(\lambda x : \Phi.(xX)x)\ (\Lambda\alpha.\lambda x : \Psi(\alpha), y : \Theta(\alpha).((xy)Y(\alpha))x),$$

$$(\lambda x : \Phi.(xX)x)\ (\Lambda\alpha.\lambda x : \Psi(\alpha), y : \Theta(\alpha).((xY(\alpha))y)x).$$

References

[1] BERGE C. *Graphes.* Gauthier-Villars (1983, 3rd ed.).

[2] BARENDREGT H.K. *The Lambda Calculus: Its Syntax and Semantics.* North-Holland (1983, 3rd ed.).

[3] DWORK C., KANELLAKIS P. AND MITCHELL J. On the Sequential Nature of Unification. *J. of Logic Programming* 1(1) (1984), 35–50.

[4] GENTZEN G. *The Collected Papers of Gerhard Gentzen.* Ed. E. Szabo, North-Holland, Amsterdam (1969).

[5] GIRARD J.Y. *Proof Theory and Logical Complexity.* Studies in Proof Theory 1, Bibliopolis, Napoli (1988).

[6] GIRARD J.Y. Interprétation fonctionnelle et élimination des coupures dans l'arithmétique d'ordre supérieur. Thèse d'Etat, Université Paris VII (1972).

[7] GIRARD J.Y. Linear Logic. *Theo. Comp. Sci.* 50 (1987) 1–102.

[8] GOLDFARB W. The Undecidability of the Second-Order Unification Problem. *Theo. Comp. Sci.* 13,2 (1981) 225–230.

[9] HENKIN L. The Logic of Equality. *The Amer. Math. Monthly* 1977, 597–612.

[10] HERBRAND J. Sur la Théorie de la Démonstration. in: *Logical Writings*, W. Goldfarb (ed.) Cambridge (1971).

[11] HOWARD W. The Formulas-as-Types Notion of Construction. In *To H.B. Curry: Essays on Combinatory Logic, Lambda-Calculus and Formalism.* Seldin J.P. and Hindley J.R. (Eds.) Academic Press (1980) 479–490.

[12] HUET G. Confluent Reductions: Abstract Properties and Application to Term Rewriting Systems. *J. ACM* 274 (1980) 797–821.

[13] HUET G. A Unification Algorithm for Typed λ-calculus. *Theo. Comp. Sci.* 1 (1975) 27–57.

[14] KAPLAN S. Positive/Negative Conditional Rewriting. Proc. MFCS, LNCS 324, Springer-Verlag (1988) 381–395.

[15] KOZEN D. Lower Bounds for Natural Proof Systems. Proc. 18th FOCS (1977) 254–266.

[16] KREISEL G. AND TAIT W. Finite Definability of Number Theoretic Functions and Parametric Completeness of Equational Calculi. *Z. Math. Logik Grundlagen Math.* 7 (1961) 28–38.

[17] LE CHENADEC PH. *Canonical Forms in Finitely Presented Algebras.* Research Notes in Theoretical Computer Science, Pitman–Wiley (1986).

[18] LE CHENADEC PH. On Positive Occur-Checks in Unification. Proc. MFCS, LNCS 324, Springer-Verlag (1988) 433–444.

[19] LE CHENADEC PH. On Unavoidable Polymorphism in λ-terms. in Preparation (1988).

[20] PRAWITZ D. *Natural Deduction.* Almqist and Wiskell, Stockolm (1965).

[21] REYNOLDS J.C. Towards a Theory of Type Structure. Paris Colloq. on Programming (1974) 408–425.

[22] ROBINSON J.A. A Machine Oriented Logic Based on the Resolution Principle. *J. ACM* 12,1 (1965) 23–41

[23] SHOENFIELD J.R. *Mathematical Logic.* Addison-Wesley (1967).

[24] STATMAN R. Herbrand's Theorem and Gentzen's Notion of a Direct Proof. In *Handbook of Mathematical Logic.* Barwise J. (Ed.), Studies in Logic 90, North-Holland (1977).

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique