A brief history of rewriting with *extensionality*

Roberto Di Cosmo

LIENS (CNRS) - DMI
Ecole Normale Supérieure
45, Rue d'Ulm
75005 Paris - France
E-mail: dicosmo@dmi.ens.fr
WWW: http://www.ens.fr/~dicosmo

September 12, 1996

- A survey of confluence, decidability and normalization results for

  typed λ-calculi with *extensional* rules

- A survey of the proof techniques

- Some applications

Extensionality and the lambda calculus

Extensional *axioms* (or equalities) are customary in lambda calculus: they give us

- categoricity (universal property) of the associated data type
- a sort of minimal observational equivalence

Typical examples are:

- η-equality:

$$(\eta) \quad \lambda x.Mx \;=\; M \quad (x \notin FV(M))$$

- surjective pairing:

$$(SP) \quad \langle \pi_1(M), \pi_2(M)\rangle \;=\; M$$

- case uniqueness:

$$(+!) \quad case(P, M \circ in_A, M \circ in_B) \;=\; MP$$

Extensionality and universal properties

SP

$$
\begin{tikzcd}
& C \arrow[dl, "f"'] \arrow[d, dashed, shift right, "h"'] \arrow[d, shift left, "\langle f,g\rangle"] \arrow[dr, "g"] & \\
A & A \times B \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & B
\end{tikzcd}
$$

That is:

$$h = \langle f, g\rangle = \langle \pi_1 \circ h, \pi_2 \circ h\rangle$$

Case

$$
\begin{tikzcd}
& C & \\
A \arrow[ur, "f"] \arrow[r, "in_1"'] & A + B \arrow[u, dashed, shift left, "h"] \arrow[u, shift right, "{[f,g]}"'] & B \arrow[ul, "g"'] \arrow[l, "in_2"]
\end{tikzcd}
$$

That is:

$$h = [f, g] = [h \circ in_1, h \circ in_2]$$

From equations to rewriting

Two choices to orient extensional equalities:

$$
\begin{array}{lrcll}
\eta & \lambda x.Mx & \overrightarrow{\longleftarrow} & M & x \notin FV(M) \\
SP & \langle \pi_1(M), \pi_2(M) \rangle & \overrightarrow{\longleftarrow} & M & \\
(+!) & case(P, M \circ in_A, M \circ in_B) & \overrightarrow{\longleftarrow} & MP &
\end{array}
$$

as contractions

+ the rules do not depend on types
- the rules are non-local: require search in FV(M) or equality testing
- the rules are not left-linear (except η)
- do not mix well with other rules like Top (lost CR)

as expansions

+ the rules are local
+ do mix well with other rules like Top
- depend on types to make sense
- need some restrictions to preserve normalization

Normalization and conditional expansion rules

Two kind of loops can arise using expansions naïvely, let's see the case of η:

Structural:

$$\lambda x.M \longrightarrow \lambda y.(\lambda x.M)y \longrightarrow \lambda y.M[y/x] =_\alpha \lambda x.M$$

Contextual:

$$MN \longrightarrow (\lambda x.Mx)N \longrightarrow MN$$

To break the loops we turn expansions into *conditional* rules:

$$(\eta) \quad M \longrightarrow \lambda x : A.Mx \text{ if } \begin{cases} x fresh \\ M : A \rightarrow B \\ M \text{ is not a } \lambda\text{-abstraction} \\ M \text{ is not applied} \end{cases}$$

Then, *restricted expansion is no longer a congruence*, but we can show that:

- no equality is lost
- normalization is preserved

Chronology I

1970s: the first expansion

1971 Prawitz suggests to reverse η [Pra71]

1976 Huet uses $\beta\eta$-long normal forms for higher-order unification [Hue76]

1979 Mints reverses η and SP [Min79]

197- Many people suggest expansions: Martin-Löf, Meyer, Statman, etc.

1980s: the contraction

1980 Klop's counterexample to CR for λ+SP [Klo80]

1981 Pottinger shows CR for *typed* $\lambda\beta\eta$+SP [Pot81]

1986 Lambek - Scott, Obtulowicz: typed $\lambda\beta\eta$+SP+$\mathbf{T}$ is not CR [LS86]

1987 Poigné - Voss try completion for $\lambda\beta\eta$+SP+$\mathbf{T}$+sums and recursion [PV87]

1989 Nesmith: Klop's counterexample holds for simply typed λ-calculus+fixpoints [Nes89]

1991 Curien - Di Cosmo: completion for *polymorphic* $\lambda\beta\eta$+SP+$\mathbf{T}$ [CDC95]

1994 Necula: η is ok with algebraic non-currified TRS's [Nec94]

Chronology II

1990s: the second expansion

1991 Jay: SN for expansions+**T**+**N** [Jay92]

1992 Di Cosmo - Kesner: CR+SN for expansions+**T**+sums+weak extensional sums, CR with recursion [DCK93, DCK94b]

1992 Cubric: CR for expansions+**T** [Cub92]

1992 Ghani - Jay: CR+SN for expansions+**T**+**N** [JG92]

1992 Akama: SN+CR for expansions+**T** [Aka93]

1992 Dougherty: CR+SN for expansions+**T**+sums, CR with recursion [Dou93]

1993 Di Cosmo - Kesner: modularity of CR and SN for expansions + algebraic systems, of CR for recursion [DCK94a]

1993 Piperno, Ronchi Della Rocca: expansions for polymorphic type inference [PRDR94]

1994 Kesner: CR+SN for pattern calculus with η-expansion [Kes94]

1995 Ghani: expansion rules to decide equality for coproducts [Gha95]

1995 Di Cosmo - Piperno: SN+CR for polymorphic λ-calculus with η [DCP95]

1995 Di Cosmo - Kesner: SN+CR for polymorphic λ-calculus with η,η^2,SP,**T** via modified reducibility [DCK96]

1995 Danvy-Malmkjær-Palsberg: expansions in partial evaluation [DMP95]

1996 van Oostrom: CR for untyped η-expansion via developments [vO94]

1996 Kesner: η-expansion is the right choice for explicit substitutions [Kes96]

1996 Di Cosmo: CR and/or SN for η-expansions in various systems [DC96]

1996 Ghani: CR and SN for η-expansions in F^ω [Ghab]

1996 Ghani: CR for η-expansions in *Coc* [Ghaa]

1996 Di Cosmo-Ghani: CR and SN for combination of F^ω with η-expansions and algebraic TRS's, CR for combination of *Coc* with η-expansions and algebraic TRS's; SN *fails* for *Coc* with expansions [DCG96]

Summary of results

System	CR	SN	CR with TRS	CR+SN with TRS
			Property	
untyped	$\surd$	*no*	?	*no*
simply typed	$\surd$	$\surd$	$\surd$	$\surd$
NNO	$\surd$	$\surd$	$\surd$	$\surd$
recursion	$\surd$	n.a.	$\surd$	n.a.
weak case	$\surd$	$\surd$		
strong case	$\surd$	*no*?		*no*?
F with η	$\surd$	$\surd$	$\surd$	$\surd$
F with η, η^2, SP	$\surd$	$\surd$	$\surd$	$\surd$
F^ω	$\surd$	$\surd$	$\surd$	$\surd$
LF	?	*no*?	?	*no*?
Coc	$\surd$	*no*?	$\surd$	*no*?

Failure of SN with dependent types

$$x : X \to X \vdash \underline{x} : (\lambda z : X \to X.X)x \to X \quad \xrightarrow{\eta}$$
$$x : X \to X \vdash \underline{\lambda y : ((\lambda z : X \to X.X)x).xy} : (\lambda z : X \to X.X)x \to X \quad \longrightarrow \quad \cdots$$

Techniques

Many techniques have been used to show SN and/or CR with expansions:

- simulation/interpretation
  (techniques by Hardin/Tannen/Curien/DiCosmo/Kesner, etc.)

- decomposition
  (techniques by Akama/DiCosmo-Piperno/Geser/Kahrs, etc.)

- residuals/developments
  (Cubric/van Oostrom)

We focus here mainly on some example from the first two classes.

Confluence via simulation/interpretation: DiCosmo-Kesner's lemma [DCK95]

Definition 1.1 (Simulation) *Given reductions R_1 and R_2, a translation $\_^\circ$ from $R_1 \cup R_2$ to R_2 has the* simulation property *if $M \xrightarrow{R_1 \cup R_2} N$ implies $M^\circ \overset{R_2}{\underset{+}{\twoheadrightarrow}} N^\circ$, i.e.*

$$\begin{tikzcd} M \arrow[r, "R_1 \cup R_2"] \arrow[d, dash, dotted] & N \arrow[d, dash, dotted] \\ M^\circ \arrow[r, "R_2", "+"', two heads] & N^\circ \end{tikzcd}$$

and the weak simulation property *if $M \xrightarrow{R_1 \cup R_2} N$ implies $M^\circ \overset{R_2}{\twoheadrightarrow} N^\circ$, i.e.*

$$\begin{tikzcd} M \arrow[r, "R_1 \cup R_2"] \arrow[d, dash, dotted] & N \arrow[d, dash, dotted] \\ M^\circ \arrow[r, "R_2", two heads] & N^\circ \end{tikzcd}$$

Proposition 1.2 (Confluence via simulation) *Given two reduction relations R_1, R_2 on a given set of terms, and a translation $\_^\circ$ s.t.*

- *Any reduction step $M \xrightarrow{R_1 \cup R_2} N$ can be (weakly) simulated by a reduction $M^\circ \overset{R_2}{\twoheadrightarrow} N^\circ$*
- *The translation is the identity on the R_1-normal-forms*

then if R_2 is confluent and R_1 is weakly normalizing, $R_1 \cup R_2$ is confluent.

Confluence via simulation/interpretation cont'd

Proof. Here is how to close any divergence of $R_1 \cup R_2$:

$$
\begin{tikzcd}
 & P_1 \arrow[r, "R_1^*"] & P_1^{R_1} \arrow[r, equal] & (P_1^{R_1})^\circ \arrow[dr, "R_2^*"] & \\
M \arrow[ur, "(R_1\cup R_2)^*"] \arrow[dr, "(R_1\cup R_2)^*"'] \arrow[rr, dotted, no head] & & M^\circ \arrow[ur, "R_2^*"'] \arrow[dr, "R_2^*"] & & Q \\
 & P_2 \arrow[r, "R_1^*"'] & P_2^{R_1} \arrow[r, equal] & (P_2^{R_1})^\circ \arrow[ur, "R_2^*"'] &
\end{tikzcd}
$$

□

This lemma has been used to show confluence for first order lambda calculi with expansions and algebraic TRS's, via a translation.

Related lemmas

- An instance of the previous lemma is used in [BTG94]
- A similar lemma is used in [CDC95]
- Hardin's interpretation method can be found in [Har89]
- Generalized interpretation methods are introduced by Kesner [Kes96], where it is used to show CR for λ_d, a calculus with explicit substitutions with a weak form of composition that preserves SN, and Kamareddine-Rios [KR95]
- A related method is due to Curien and Ghelli [CG91]

Confluence via simulation/interpretation: a general lemma

Here is how to subsume all the "interpretation/simulation" lemmas:

Proposition 1.3 (Generalized Confluence via simulation) *Given two reduction relations R_1, R_2 on a given set of terms, a reduction $S \subseteq (R_1 \cup R_2)^*$, and a translation $\_^\circ$ s.t.*

- *if $M \xrightarrow{R_1 \cup R_2} N$ then $M^\circ =_S N^\circ$ ctually, $\forall M^\circ \exists N^\circ . M^\circ =_S N^\circ$ suffices)*
- *any S divergence on $(|R_1 \cup R_2|)^\circ$ can be closed using S*
- *the translation is the identity on the R_1-normal-forms*

then if R_1 is weakly normalizing, $R_1 \cup R_2$ is confluent.

Proof. Here is how to close any divergence of $R_1 \cup R_2$:

$$\begin{tikzcd}
 & P_1 \arrow[r, "R_1^*"] & P_1^{R_1} \arrow[r, equal] & (P_1^{R_1})^\circ \arrow[dr, "R_2^*"] \arrow[dl, equal, "S"] & \\
M \arrow[ur, "(R_1 \cup R_2)^*"] \arrow[dr, "(R_1 \cup R_2)^*"'] \arrow[rr, dotted, no head] & & M^\circ \arrow[dr, equal, "S"] & & Q \\
 & P_2 \arrow[r, "R_1^*"'] & P_2^{R_1} \arrow[r, equal] & (P_2^{R_1})^\circ \arrow[ur, "R_2^*"'] &
\end{tikzcd}$$

$\square$

Confluence via simulation/interpretation: a general lemma, cont'd

One then gets:

- Hardin's lemma:
  by requiring R_1 to be SN+CR, using R_1-n.f. as $\_^\circ$, and asking for S reduction instead of equality
- Di Cosmo-Kesner's lemma:
  by requiring S to be R_1, and asking for S reduction instead of equality
- Kamareddine-Rios' lemma:
  by using a fixed R_1-normalization strategy f as $\_^\circ$, and asking for S reduction instead of equality
- Kesner's lemma:
  by using R_1-normalization as $\_^\circ$

Confluence via simulation/interpretation: Curien-Ghelli's Lemma

Proposition 1.4 (Curien-Ghelli) *Given a confluent ARS* $< A, \xrightarrow{R} >$, *a weakly normalizing ARS* $< B, \xrightarrow{S} >$ *and a translation* $\phi : A \rightarrow B$ *s.t.*

- *if* $M \xrightarrow{S} N$ *then* $\phi(M) =_A \phi(N)$
- ϕ *sends* B*-normal forms to* A*-normal forms*
- ϕ *is injective on* B*-normal-forms*

then $< B, \xrightarrow{S} >$ *is confluent.*

Notice that lemma 1.3 is *not* an instance of 1.4: if R_1 is WN, and $B = R_1 \cup R_2$, then B is not WN in general.

Decomposition Lemmas

Here is one of the best known lemmas for getting CR via decomposition:

Lemma 1.5 (Hindley-Rosen ([Bar84], section 3))
If $\xrightarrow{R}$ *and* $\xrightarrow{S}$ *are confluent, and* $\xrightarrow{R}\!\!\!\!\rightarrow$ *and* $\xrightarrow{S}\!\!\!\!\rightarrow$ *commute with each other, i.e.*

$$\begin{tikzcd} \cdot \arrow[r, two heads, "S"] \arrow[d, two heads, "R"'] & \cdot \arrow[d, two heads, dashed, "R"] \\ \cdot \arrow[r, two heads, dashed, "S"'] & \cdot \end{tikzcd}$$

then $R \cup S$ *is confluent.*

Establishing the commutation may be complex, so one really uses:

Lemma 1.6 (usual sufficient condition for commutation)
If

$$\begin{tikzcd} \cdot \arrow[r, "S"] \arrow[d, "R"'] & \cdot \arrow[d, dashed, "R"] \\ \cdot \arrow[r, dashed, "S"', "="] & \cdot \end{tikzcd}$$

then $\xrightarrow{R}\!\!\!\!\rightarrow$ *and* $\xrightarrow{S}\!\!\!\!\rightarrow$ *commute with each other.*

Decomposition Lemmas: DPG

The restriction to *at most* one step of R to close the diagram is quite restrictive, and *not satisfied by expansion rules.* It is necessary in general:

$$
\begin{tikzcd}
\cdot \arrow[rr, "R"] \arrow[d, "S"] & & \cdot \arrow[d, "S"] \\
\cdot \arrow[r, "R"] \arrow[ddd, "S"] & \cdot \arrow[r, "R"] \arrow[d, "S"] & \cdot \\
 & \cdot \arrow[dd, "S"] & \vdots \\
 & & \\
\cdot \arrow[r, "R"] & \cdot
\end{tikzcd}
$$

But if R is a SN, then we can use a dual restriction to *at least* one R step.

Lemma 1.7 (dual condition from [DCP95])
If R is strongly normalizing, and the following diagram holds

$$
\text{(DPG)} \qquad
\begin{tikzcd}
a \arrow[r, "R"] \arrow[d, "S"'] & b \arrow[d, dashed, "S"] \\
c \arrow[r, dashed, "R", "+"'] & d
\end{tikzcd}
$$

Then $\overset{R}{\twoheadrightarrow}$ *and* $\overset{S}{\twoheadrightarrow}$ *commute.*

Alfons Geser remarked this very same property in his PhD Thesis (see [Ges90], page 38, remark after the proof sketch), where the (DPG) diagram is read as R *strictly locally commutes over* S^{-1}.

Decomposition Lemmas: Proof of DPG

Proof. Since R is a strongly normalizing rewriting system, we have a well-founded order $<$ on $\mathcal{A}$ by setting $a_1 < a_2$ if $a_2 \xrightarrow{R} a_1$. Also, let us denote $dist(a_1, a_2)$ the length of a given S-reduction sequence from a_1 to a_2. The proof then proceeds by well-founded induction on pairs $(b, dist(a, b))$, ordered lexicographically. Indeed, if b is an R-normal form and $dist(a, b) = 0$, then the lemma trivially holds. Otherwise, by hypothesis, there exist a', a'', a''' as in the following diagram.

$$
\begin{tikzcd}
a \arrow[r, "R"] \arrow[d, "S"] & a' \arrow[r, two heads, "R"] \arrow[d, "S"] & c \arrow[dd, dashed, "S"] \\
a'' \arrow[r, "R"] \arrow[d, "S"] \arrow[dr, phantom, "D_1"] & a''' \arrow[d, dashed, "S"] \arrow[dr, phantom, "D_2"] & \\
b \arrow[r, dashed, two heads, "R"] & b' \arrow[r, dashed, two heads, "R"] & d
\end{tikzcd}
$$

We can now apply the inductive hypothesis to the diagram D_1, since

$$(b, dist(a'', b)) <_{lex} (b, dist(a, b)).$$

Finally, we observe that $b \xrightarrow{R}_{+} b'$, just composing the diagram in the hypothesis down from a.

Hence we can apply the inductive hypothesis to the diagram D_2, since

$$(b', dist(a', b')) <_{lex} (b, dist(a, b)),$$

and we are done.

$\square$

Rewriting Lemmas: Confluence and Normalization

If one wants both confluence and normalization, here is a nice useful lemma by Akama:

Lemma 1.8 (Akama) *Let R, S be CR+SN. If*

$$\begin{tikzcd} M \arrow[r, "R"] \arrow[d, "S"] & N \arrow[d, "S"] \\ M^S \arrow[r, dashed, "R", "+"'] & N^S \end{tikzcd}$$

then $R \cup S$ is CR+SN.

Again, difficult application, so here is how DPG helps:

Lemma 1.9 (Simplified Akama's Lemma) *Let S and R be confluent and strongly normalizing reductions, s.t.*

$$\begin{tikzcd} a \arrow[r, "R"] \arrow[d, "S"] & c \arrow[d, dashed, "S"] \\ b \arrow[r, dashed, "R", "+"'] & d \end{tikzcd}$$

and R preserves S-normal forms: then $S \cup R$ is also confluent and strongly normalizing.

This lemma has been applied in a variety of systems, see [DC96].

Other Rewriting Lemmas: Confluence via decomposition

Another interesting lemma due to Kahrs has been used in [JG92] to show confluence of first order typed lambda calculus with expansion rules.

Definition 1.10 *S satisfies the diamond lemma relative to R if*

$$\begin{tikzcd} t \arrow[r, "S"] \arrow[d, "S"'] & t_2 \arrow[r, dashed, "S"] & t_3 \arrow[d, dashed, "R^*"] \\ t_1 \arrow[r, dashed, "S"'] & t_4 \arrow[r, dashed, "R^*"'] & t_5 \end{tikzcd}$$

Definition 1.11 *S is* R-extendable *if*

$$\begin{tikzcd} t \arrow[r, "R"] \arrow[d, "S"'] & t_2 \arrow[r, dashed, "R^*"] & t_3 \arrow[d, dashed, "S"] \\ t_1 \arrow[rr, dashed, "R^*"'] & & t_4 \end{tikzcd}$$

Proposition 1.12 (Kahrs) *If R is CR+SN, S, is R-extendable and satisfies the diamond lemma relative to R, then $R \cup S$ is CR.*

This lemma is used with S as parallel η in [JG92].

Some applications of expansions

decision procedures for category theory via expansive rewriting systems

higher order unification here terms are reduced to expansive normal form for unification

partial evaluation expansions allow to get much improved residual programs

flexible typing working up to η-expansion can give better typings

isomorphisms of types their study needs CR reductions for various calculi, best given with expansions

algebraic functional system the combination with TRS's is best done using expansions

pattern calculi need expansions to rewrite with extensionality

explicit substitutions extensionality is only reasonable as an expansion

References

[Aka93] Yohji Akama. On Mints' reductions for ccc-Calculus. In *Typed Lambda Calculus and Applications*, number 664 in LNCS, pages 1–12. Springer Verlag, 1993.

[Bar84] Henk Barendregt. *The Lambda Calculus; Its syntax and Semantics (revised edition)*. North Holland, 1984.

[BTG94] Val Breazu-Tannen and Jean Gallier. Polymorphic rewiting preserves algebraic confluence. *Information and Computation*, 114:1–29, 1994.

[CDC91] Pierre-Louis Curien and Roberto Di Cosmo. A confluent reduction system for the λ-calculus with surjective pairing and terminal object. In Leach, Monien, and Artalejo, editors, *Intern. Conf. on Automata, Languages and Programming (ICALP)*, volume 510 of *Lecture Notes in Computer Science*, pages 291–302. Springer-Verlag, July 1991.

[CDC95] Pierre-Louis Curien and Roberto Di Cosmo. A confluent reduction system for the λ-calculus with surjective pairing and terminal object. *Journal of Functional Programming*, 1995. To appear. This is an extended and revised version of [CDC91].

[CG91] Pierre-Louis Curien and Giorgio Ghelli. On the confluence of weakly normalizing systems. In *Rewriting Techniques and Applications*, volume 488 of *LNCS*, 1991.

[Cub92] Djordje Cubric. On free CCC. Distributed on the `types` mailing list, 1992.

[DC96] Roberto Di Cosmo. On the power of simple diagrams. In *Rewriting Techniques and Applications*, number 1103 in Lecture Notes in Computer Science, pages 200–214, July 1996.

[DCG96] Roberto Di Cosmo and Neil Ghani. Combining algebric rewriting with higher order extensional lambda calculi. 1996. submitted.

[DCK93] Roberto Di Cosmo and Delia Kesner. A confluent reduction for the extensional typed λ-calculus with pairs, sums, recursion and terminal object. In Andrzej Lingas, editor, *Intern. Conf. on Automata, Languages and Programming (ICALP)*, volume 700 of *Lecture Notes in Computer Science*, pages 645–656. Springer-Verlag, July 1993.

[DCK94a] Roberto Di Cosmo and Delia Kesner. Combining first order algebraic rewriting systems, recursion and extensional lambda calculi. In Serge Abiteboul and Eli Shamir, editors, *Intern. Conf. on Automata, Languages and Programming (ICALP)*, volume 820 of *Lecture Notes in Computer Science*, pages 462–472. Springer-Verlag, July 1994.

[DCK94b] Roberto Di Cosmo and Delia Kesner. Simulating expansions without expansions. *Mathematical Structures in Computer Science*, 4:1–48,

1994. A preliminary version is available as Technical Report LIENS-93-11/INRIA 1911.

[DCK95] Roberto Di Cosmo and Delia Kesner. Combining algebraic rewriting, extensional lambda calculi and fixpoints. *Theoretical Computer Science*, 1995. To appear.

[DCK96] Roberto Di Cosmo and Delia Kesner. Rewriting with polymorphic extensional λ-calculus. In *CSL'95*, volume 1092 of *Lecture Notes in Computer Science*, pages 215–232. Springer-Verlag, 1996. Extended abstract presented in Paderborn, September 1995.

[DCP95] Roberto Di Cosmo and Adolfo Piperno. Expanding extensional polymorphism. In Mariangiola Dezani-Ciancaglini and Gordon Plotkin, editors, *Typed Lambda Calculus and Applications*, volume 902 of *Lecture Notes in Computer Science*, pages 139–153, April 1995.

[DMP95] O. Danvy, K. Malmkjær, and J. Palsberg. The essence of eta-expansion in partial evaluation. *Lisp and Symbolic Computation*, 8(3):209–228, September 1995.

[Dou93] Daniel J. Dougherty. Some lambda calculi with categorical sums and products. In *Proc. of the Fifth International Conference on Rewriting Techniques and Applications (RTA)*, 1993.

[Ges90] Alfons Geser. *Relative termination*. Dissertation, Fakultät für Mathematik und Informatik, Universität Passau, Germany, 1990. Also available as: Report 91-03, Ulmer Informatik-Berichte, Universität Ulm, 1991.

[Ghaa] Neil Ghani. Eta-expansions in dependent type theory - the calculus of constructions. Submitted to TLCA '97.

[Ghab] Neil Ghani. Eta-expansions in F^ω. To appear in CSL'96.

[Gha95] Neil Ghani. $\beta\eta$-equality for coproducts. In Mariangiola Dezani-Ciancaglini and Gordon Plotkin, editors, *Typed Lambda Calculus and Applications*, volume 902 of *Lecture Notes in Computer Science*, April 1995.

[Har89] Thérèse Hardin. Confluence results for the pure strong categorical logic C.C.L.; λ-calculi as subsystems of C.C.L. *Theoretical Computer Science*, 65(2):291–342, 1989.

[Hue76] Gérad Huet. Résolution d'équations dans les langages d'ordre $1, 2, \ldots, \omega$. *Thèse d'Etat, Université Paris VII*, 1976.

[Jay92] Colin Barry Jay. Long $\beta\eta$ normal forms and confluence (revised). Technical Report 44, LFCS - University of Edinburgh, August 1992.

[JG92] Colin Barry Jay and Neil Ghani. The Virtues of Eta-expansion. Technical Report ECS-LFCS-92-243, LFCS, 1992. University of Edimburgh, preliminary version of [JG95].

[JG95] Colin Barry Jay and Neil Ghani. The Virtues of Eta-expansion. *Journal of Functional Programming*, 5(2):135–154, April 1995.

[Kes94] Delia Kesner. Reasoning about layered, wildcard and product pat terns. In Rodíguez-Artalejo Levi, editor, *Int. Conf. on Algebraic and Logic Programming (ALP)*, number 850 in LNCS. Springer-Verlag, 1994.

[Kes96] Delia Kesner. Confluence properties of extensional and non-extensional λ-calculi with explicit substitutions. In Harald Ganzinger, editor, *Proc. of the Seventh International Conference on Rewriting Techniques and Applications (RTA)*, number 1103 in LNCS, pages 184–199, 1996.

[Klo80] Jan Willem Klop. Combinatory reduction systems. *Mathematical Center Tracts*, 27, 1980.

[KR95] Fairouz Kamareddine and Alejandro Ríos. The confluence of the λs_e-calculus via a gen eralized interpretation method, 1995. Draft.

[LS86] Joachim Lambek and Philip J. Scott. *An introduction to higher order categorical logic*. Cambridge University Press, 1986.

[Min79] Gregory Mints. Teorija categorii i teoria dokazatelstv.I. *Aktualnye problemy logiki i metodologii nauky*, pages 252–278, 1979.

[Nec94] Ciprian Necula. Algebraic rewriting preserves (β, η) confluence in the typed lambda calculus. Draft Manuscript, Pol. Inst. of Bucharest, e-mail: `George_Ciprian_Necula @ PL1.FOX.CS.CMU.EDU`, 1994.

[Nes89] Dan Nesmith. An application of klop's counterexample to a higher-order rewrite system. Draft Paper, 1989.

[Pot81] Garrel Pottinger. The Church Rosser Theorem for the Typed lambda-calculus with Surjective Pairing. *Notre Dame Journal of Formal Logic*, 22(3):264–268, 1981.

[Pra71] D. Prawitz. Ideas and results in proof theory. *Proceedings of the 2nd Scandinavian Logic Symposium*, pages 235–307, 1971.

[PRDR94] Adolfo Piperno and Simonetta Ronchi Della Rocca. Type inference and extensionality. In *Symp. on Logic in Computer Science (LICS)*, Paris, France, July 1994. IEEE computer society Press.

[PV87] Axel Poigné and Josef Voss. On the implementation of abstract data types by programming language constructs. *Journal of Computer and System Science*, 34(2-3):340–376, April/June 1987.

[vO94] Vincent van Oostrom. Developing developments. Submitted to Theoretical Computer Science should appear in volume 145, 1994.