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:

- $\eta$-equality:
  \[(\eta) \lambda x.Mx = M \ (x \not\in FV(M))\]

- surjective pairing:
  \[(SP) \ \langle \pi_1(M), \pi_2(M) \rangle = M\]

- case uniqueness:
  \[(+!) \ case(P, M \circ in_A, M \circ in_B) = MP\]
**Extensionality and universal properties**

**SP**

That is:

\[ h = \langle f, g \rangle = \langle \pi_1 \circ h, \pi_2 \circ h \rangle \]

**Case**

That is:

\[ h = [f, g] = [h \circ in_1, h \circ in_2] \]
From equations to rewriting

Two choices to orient extensional equalities:

\[
\eta \quad \quad \lambda x. Mx \leftrightarrow M \quad x \notin \text{FV}(M)
\]

\[
SP \quad \quad \langle \pi_1(M), \pi_2(M) \rangle \leftrightarrow M
\]

\[
(+!) \quad \text{case}(P, M \circ \text{in}_A, M \circ \text{in}_B) \leftrightarrow MP
\]

as **contractions**

+ the rules do not depend on types
- the rules are non-local: require search in \text{FV}(M) or equality testing
- the rules are not left-linear (except \(\eta\))
- 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 $\eta$:

**Structural:**
\[
\lambda x. M \rightarrow \lambda y. (\lambda x. M) y \rightarrow \lambda y. M[y/x] =_{\alpha} \lambda x. M
\]

**Contextual:**
\[
MN \rightarrow (\lambda x. M x) N \rightarrow MN
\]

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

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

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 \( \eta \) [Pra71]

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

1979 Mints reverses \( \eta \) and SP [Min79]

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

1980s: the contraction

1980 Klop’s counterexample to CR for \( \lambda + SP \) [Klo80]

1981 Pottinger shows CR for typed \( \lambda\beta\eta + SP \) [Pot81]

1986 Lambek - Scott, Obtulowicz: typed \( \lambda\beta\eta + SP + T \) is not CR [LS86]

1987 Poigné - Voss try completion for \( \lambda\beta\eta + SP + T \) + sums and recursion [PV87]

1989 Nesmith: Klop’s counterexample holds for simply typed \( \lambda \)-calculus + fixpoints [Nes89]

1991 Curien - Di Cosmo: completion for \textit{polymorphic} \( \lambda\beta\eta + SP + T \) [CDC95]

1994 Necula: \( \eta \) is ok with algebraic non-curried 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 Dougherty: CR+SN for expansions + T + sums, CR with recursion [Dou93]
1992 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, η², 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 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

<table>
<thead>
<tr>
<th>System</th>
<th>CR</th>
<th>SN</th>
<th>CR with TRS</th>
<th>CR+SN with TRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>untyped</td>
<td>√</td>
<td>no</td>
<td>?</td>
<td>no</td>
</tr>
<tr>
<td>simply typed</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>NNO</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>recursion</td>
<td>√</td>
<td>n.a.</td>
<td>√</td>
<td>n.a.</td>
</tr>
<tr>
<td>weak case</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong case</td>
<td>√</td>
<td>no?</td>
<td></td>
<td>no?</td>
</tr>
<tr>
<td>$F$ with $\eta$</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>$F$ with $\eta, \eta^2, SP$</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>$F^\omega$</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>LF</td>
<td>?</td>
<td>no?</td>
<td>?</td>
<td>no?</td>
</tr>
<tr>
<td>Coc</td>
<td>√</td>
<td>no?</td>
<td>√</td>
<td>no?</td>
</tr>
</tbody>
</table>

### Failure of SN with dependent types

\[ x : X \rightarrow X \vdash \underline{x} : (\lambda z : X \rightarrow X.X)x \rightarrow X \]
\[ x : X \rightarrow X \vdash \lambda y : ((\lambda z : X \rightarrow X.X)x).xy : (\lambda z : X \rightarrow X.X)x \rightarrow X \]
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 $M^\circ \xrightarrow{R_2} N^\circ$, i.e.

\[
\begin{array}{c}
M \xrightarrow{R_1 \cup R_2} N \\
M^\circ \xrightarrow{+} R_2 \xrightarrow{\circ} N^\circ
\end{array}
\]

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

\[
\begin{array}{c}
M \xrightarrow{R_1 \cup R_2} N \\
M^\circ \xrightarrow{R_2} N^\circ
\end{array}
\]

**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 $M^\circ \xrightarrow{R_2} N^\circ$

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

\[ P_1 \xrightarrow{R_1^*} P_1^{R_1} \xrightarrow{\ast} (P_1^{R_1})^\circ \]
\[ (R_1 \cup R_2)^* \]
\[ M \]

\[ Q \]
\[ (R_1 \cup R_2)^* \]
\[ M^\circ \]
\[ (R_2^*) \]
\[ R_2^* \]

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 $\lambda_d$, 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]
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 $M^\circ = S N^\circ$ (actually, $\forall M^\circ \exists N^\circ. M^\circ = S N^\circ$ 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 $R_1 \cup R_2$:

![Diagram](https://via.placeholder.com/150)

$\Box$
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 $\phi(M) =_A \phi(N)$
- $\phi$ sends $B$-normal forms to $A$-normal forms
- $\phi$ 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.
Here is one of the best known lemmas for getting CR via decomposition:

**Lemma 1.5 (Hindley-Rosen ([Bar84], section 3))**

If $\rightarrow_R$ and $\rightarrow_S$ are confluent, and $\rightarrow_R$ and $\rightarrow_S$ commute with each other, i.e.

\[
\begin{array}{c}
\xymatrix{ R \ar[r] & S \\
R_1 \\
\ar[u] S \\
\ar[uu] }
\end{array}
\]

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

\[
\begin{array}{c}
\xymatrix{ R \ar[r] & S \\
R_1 \\
\ar[u] S \\
\ar[uu] }
\end{array}
\]

then $\rightarrow_R$ and $\rightarrow_S$ 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:

\[
\begin{array}{cc}
\cdot & R \\
S & \downarrow \\
R & R \\
S & \downarrow \\
S & \downarrow \\
R & \\
\end{array}
\]

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*

\[
\begin{array}{cc}
a & \xrightarrow{R} b \\
S & \downarrow \\
& S \\
& c \xrightarrow{R} d \\
\end{array}
\]

*Then* $\longrightarrow$ and $\xrightarrow{s}$ *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 $\prec$ on $\mathcal{A}$ by setting $a_1 \prec a_2$ if $a_2 \overrightarrow{R} a_1$. Also, let us denote $\text{dist}(a_1, a_2)$ the length of a given $S$-reduction sequence from $a_1$ to $a_2$. The proof then proceeds by well-founded induction on pairs $(b, \text{dist}(a, b))$, ordered lexicographically. Indeed, if $b$ is an $R$-normal form and $\text{dist}(a, b) = 0$, then the lemma trivially holds. Otherwise, by hypothesis, there exist $a', a'', a'''$ as in the following diagram.

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

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

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

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

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

and we are done.

$\Box$
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

\[
\begin{align*}
M \xrightarrow{R} N \\
\downarrow S \\
M \xrightarrow{S} R
\end{align*}
\]

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.

\[
\begin{align*}
a \xrightarrow{R} c \\
\downarrow S \\
b \xrightarrow{R} d
\end{align*}
\]

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

\[
\begin{align*}
  t & \xrightarrow{S} t_2 \xrightarrow{S} t_3 \\
  S & \xrightarrow{R^*} t_1 \xrightarrow{S} t_4 \xrightarrow{R^*} t_5
\end{align*}
\]

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

\[
\begin{align*}
  t & \xrightarrow{R} t_2 \xrightarrow{R^*} t_3 \\
  S & \xrightarrow{S} t_1 \xrightarrow{R^*} t_4
\end{align*}
\]

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


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


