Confluence of Term Rewriting Systems

Confluence

Allows us to forget about non-determinism:

Whatever rewriting is done we will converge later.

Back with the simple game

The rules of the game:

$$
\begin{aligned}
\bullet\bullet &\rightarrow \circ \\
\circ\circ &\rightarrow \circ \\
\bullet\circ &\rightarrow \bullet \\
\circ\bullet &\rightarrow \bullet
\end{aligned}
$$

A starting point:

$$\bullet \circ \bullet \circ \bullet \circ \bullet \bullet \bullet \bullet \circ \circ \bullet \circ \circ \bullet \bullet\circ$$

From a given start, is the result determinist?

Analysing the different cases

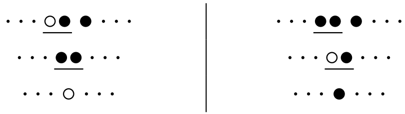
Disjoint redexes:

$$\begin{array}{c}\cdots \underline{\otimes\otimes} \cdots \otimes\ \otimes \cdots \\ \cdots \otimes \cdots \underline{\otimes\otimes} \cdots \\ \cdots \otimes \cdots \otimes \cdots\end{array}$$

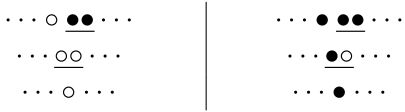
is the same as:

$$\begin{array}{c}\cdots \otimes\ \otimes \cdots \underline{\otimes\otimes} \cdots \\ \cdots \underline{\otimes\otimes} \cdots \otimes \cdots \\ \cdots \otimes \cdots \otimes \cdots\end{array}$$

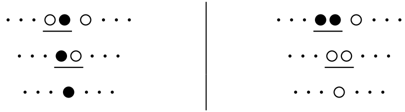
No disjoint redexes (central black):



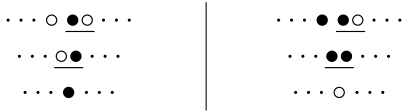
but



or



but



No disjoint redexes (central white):

$\cdots\ \underline{\bigcirc\bigcirc}\ \bullet\ \cdots$	$\cdots\ \underline{\bullet\bigcirc}\ \bullet\ \cdots$
$\cdots\ \underline{\bigcirc\bullet}\ \cdots$	$\cdots\ \underline{\bullet\bullet}\ \cdots$
$\cdots\ \bullet\ \cdots$	$\cdots\ \bigcirc\ \cdots$

but

$\cdots\ \bigcirc\ \underline{\bigcirc\bullet}\ \cdots$	$\cdots\ \bullet\ \underline{\bigcirc\bullet}\ \cdots$
$\cdots\ \underline{\bigcirc\bullet}\ \cdots$	$\cdots\ \underline{\bullet\bullet}\ \cdots$
$\cdots\ \bullet\ \cdots$	$\cdots\ \bigcirc\ \cdots$

or

$\cdots\ \underline{\bigcirc\bigcirc}\ \bigcirc\ \cdots$	$\cdots\ \underline{\bullet\bigcirc}\ \bigcirc\ \cdots$
$\cdots\ \underline{\bigcirc\bigcirc}\ \cdots$	$\cdots\ \underline{\bullet\bigcirc}\ \cdots$
$\cdots\ \bigcirc\ \cdots$	$\cdots\ \bullet\ \cdots$

but

$\cdots\ \bigcirc\ \underline{\bigcirc\bigcirc}\ \cdots$	$\cdots\ \bullet\ \underline{\bigcirc\bigcirc}\ \cdots$
$\cdots\ \underline{\bigcirc\bigcirc}\ \cdots$	$\cdots\ \underline{\bullet\bigcirc}\ \cdots$
$\cdots\ \bigcirc\ \cdots$	$\cdots\ \bullet\ \cdots$

Thus:

$$
\begin{tikzcd}
 & t_0 \arrow[dl] \arrow[dr] & \\
t_1 \arrow[dr, dashed] & & t_2 \arrow[dl, dashed] \\
 & t_3 &
\end{tikzcd}
$$

but what about:

$$
\begin{tikzcd}
 & t_0 \arrow[dl, "*"'] \arrow[dr, "*"] & \\
t_1 \arrow[dr, dashed, "*"] & & t_2 \arrow[dl, dashed, "*"'] \\
 & t_3 &
\end{tikzcd}
$$

Definitions

Consider an ARS ($\mathcal{T}$,$\rightarrow$)

- An element $t \in \mathcal{T}$ is a $\rightarrow$-**normal form** if there exists no $t' \in \mathcal{T}$ such that $t \rightarrow t'$.
- The relation $\rightarrow$ is **terminating** (or **strongly normalizing**, or **noetherian**) if every reduction sequence is finite.
- The relation $\rightarrow$ is **weakly normalizing** (or weakly terminating) if every element $t \in \mathcal{T}$ has a normal form.
- The relation $\rightarrow$ has the **unique normal form property** if for any $t, t' \in \mathcal{T}$, $t \xleftrightarrow{*} t'$ and t, t' are normal forms imply $t = t'$.

Definitions

Localy confluent (LC)

$$\begin{tikzcd} & t_0 \arrow[dl] \arrow[dr] & \\ t_1 \arrow[dr, dotted, "*"'] & & t_2 \arrow[dl, dotted, "*"] \\ & t_3 & \end{tikzcd}$$

Diamond property (DP)

$$\begin{tikzcd} & t_0 \arrow[dl] \arrow[dr] & \\ t_1 \arrow[dr, dotted] & & t_2 \arrow[dl, dotted] \\ & t_3 & \end{tikzcd}$$

Church Rosser (CR)

$$\begin{tikzcd} t \arrow[rr, leftrightarrow, "+"] \arrow[dr, dotted, "*"'] & & s \arrow[dl, dotted, "*"] \\ & u & \end{tikzcd}$$

Confluent (C)

$$\begin{tikzcd} & t_0 \arrow[dl, "*"'] \arrow[dr, "*"] & \\ t_1 \arrow[dr, dotted, "*"'] & & t_2 \arrow[dl, dotted, "*"] \\ & t_3 & \end{tikzcd}$$

Local versus global confluence

1. $C \Rightarrow LC$
2. $LC \Rightarrow C$?

Local versus global confluence

1. $C \Rightarrow LC$
2. $LC \Rightarrow C$?
   - Consider four distinct elements a, b, c, d of $\mathcal{T}$ and the relation:
     $a \to b$
     $b \to a$
     $a \to c$
     $b \to d$

$$\begin{tikzcd} a \arrow[r, bend left] \arrow[d] & b \arrow[l, bend left] \arrow[d] \\ c & d \end{tikzcd}$$

Newman's lemma

Lemma[Newman 1942]

Provided the relation $\rightarrow$ is terminating

then

$\rightarrow$ is confluent iff it is locally confluent

Proof:

- locally confluent if confluent
  obvious
- confluent if locally confluent
  ?

Noetherian induction: a fondamental tool

Let $(\mathcal{T}, >)$ be an ordered set s.t. $>$ is well-founded.

Let $\mathcal{P}$ be a proposition:

1. $\forall t \in \mathcal{T},\ [\forall t' \in \{t' \mid t > t'\},\ \mathcal{P}(t')] \Rightarrow \mathcal{P}(t)$
2. $\mathcal{P}(t)$ is provable for all minimal element t,

then $\forall t \in \mathcal{T}, \mathcal{P}(t)$.

Noetherian induction: a fondamental tool

Consider $(\mathcal{T}, \rightarrow)$

$$
\begin{tikzcd}
 & & t \arrow[dl] \arrow[dr] & & \\
 & t_1' \arrow[dl, "m-1"'] \arrow[dr, "*"] & & t_2' \arrow[dl, "*"'] \arrow[dr, "n-1"] & \\
t_1 \arrow[dr, "*"] & & t' \arrow[dl, "*"'] & & t_2 \arrow[ddll, "*"'] \\
 & t_1'' \arrow[dr, "*"] & & & \\
 & & u & &
\end{tikzcd}
$$

Proving local confluence

Local confluence is equivalent to the confluence of the critical pairs of the relation.

Proving local confluence

Local confluence is equivalent to the confluence of the critical pairs of the relation.

A non-variable term t' and a term t *overlap* if there exists a position ω in t such that $t_{|\omega}$ and t' are unifiable (with $t_{|\omega}$ not a variable).

Proving local confluence

Local confluence is equivalent to the confluence of the critical pairs of the relation.

A non-variable term t' and a term t *overlap* if there exists a position ω in t such that $t_{|\omega}$ and t' are unifiable (with $t_{|\omega}$ not a variable).

Where does $(x + y) + z$ and $(x' + y') + z'$ overlap?

Proving local confluence

Local confluence is equivalent to the confluence of the critical pairs of the relation.

A non-variable term t' and a term t *overlap* if there exists a position ω in t such that $t_{|\omega}$ and t' are unifiable (with $t_{|\omega}$ not a variable).

Where does $(x + y) + z$ and $(x' + y') + z'$ overlap?

Where does $0 + x \rightarrow x$ and $s(x') + y \rightarrow s(x' + y)$ overlap?

Proving local confluence

Local confluence is equivalent to the confluence of the critical pairs of the relation.

A non-variable term t' and a term t *overlap* if there exists a position ω in t such that $t_{|\omega}$ and t' are unifiable (with $t_{|\omega}$ not a variable).

Where does $(x + y) + z$ and $(x' + y') + z'$ overlap?

Where does $0 + x \to x$ and $s(x') + y \to s(x' + y)$ overlap?

How about $H \vdash \overline{\overline{P}} \to H \vdash P$ and $H' \wedge P' \vdash P' \to T$?

Paranthesis

Unification problems

Solve an equation

Does it exist x, y, z such that

$$x + (z * y) = y + (x * z)$$

An infinity of solutions,
but a most general one

$$x = y = z$$

Unification problem.

Unification Problems

$\mathcal{F}$ a set of function symbols,
$\mathcal{X}$ a set of variables,
$\mathcal{A}$ an $\mathcal{F}$-algebra.
A $<\mathcal{F}, \mathcal{X}, \mathcal{A}>$-unification problem
is a disjunction of existentially quantified formulas

$$P_j = \exists \vec{z} \bigwedge_{i \in I_j} s_i =^?_{\mathcal{A}} t_i$$

sometimes abbreviated

$$P_j = \exists \vec{z} \ \{s_i =^?_{\mathcal{A}} t_i\}_{i \in I_j}.$$

A unifier to such a problem is a *substitution* σ such that
$\exists j, \forall i \in I_j, \quad \mathcal{A} \models \exists \vec{z} \ \sigma_{|\mathcal{X}-\vec{z}}(s_i) = \sigma_{|\mathcal{X}-\vec{z}}(t_i)$.

Rules for syntactic unification

$$
\begin{array}{lll}
\textit{Delete} & & P \wedge s =^? s \\
& \rightarrow & P
\end{array}
$$

Rules for syntactic unification

$$
\begin{array}{lrl}
\textit{Delete} & & P \wedge s =^{?} s \\
 & \to & P \\
 & & \\
\textit{Decompose} & & P \wedge f(s_1, \ldots, s_n) =^{?} f(t_1, \ldots, t_n) \\
 & \to & P \wedge s_1 =^{?} t_1 \wedge \ldots \wedge s_n =^{?} t_n
\end{array}
$$

Rules for syntactic unification

$$\begin{array}{lcll}
\textit{Delete} & & P \wedge s =^? s & \\
 & \to & P & \\
 & & & \\
\textit{Decompose} & & P \wedge f(s_1, \ldots, s_n) =^? f(t_1, \ldots, t_n) & \\
 & \to & P \wedge s_1 =^? t_1 \wedge \ldots \wedge s_n =^? t_n & \\
 & & & \\
\textit{Conflict} & & P \wedge f(s_1, \ldots, s_n) =^? g(t_1, \ldots, t_p) & \\
 & \to & \textit{Fail} & \text{if } f \neq g
\end{array}$$

Rules for syntactic unification

$$
\begin{array}{llll}
\textit{Delete} & & P \wedge s =^{?} s & \\
& \rightarrow & P & \\
& & & \\
\textit{Decompose} & & P \wedge f(s_1, \ldots, s_n) =^{?} f(t_1, \ldots, t_n) & \\
& \rightarrow & P \wedge s_1 =^{?} t_1 \wedge \ldots \wedge s_n =^{?} t_n & \\
& & & \\
\textit{Conflict} & & P \wedge f(s_1, \ldots, s_n) =^{?} g(t_1, \ldots, t_p) & \\
& \rightarrow & \textit{Fail} & \text{if } f \neq g \\
& & & \\
\textit{Coalesce} & & P \wedge x =^{?} y & \\
& \rightarrow & \{x \mapsto y\}P \wedge x =^{?} y & \text{if } x, y \in \mathcal{V}ar(P) \wedge x \neq y
\end{array}
$$

Rules for syntactic unification

$$\begin{array}{rll} \textit{Eliminate} & P \wedge x =^? s & \\ \to & \{x \mapsto s\}P \wedge x =^? s & \quad \text{if } x \notin \mathcal{V}ar(s), s \notin \mathcal{X}, x \in \mathcal{V}ar(P) \end{array}$$

Rules for syntactic unification

$$
\begin{array}{lll}
\textit{Eliminate} & P \wedge x =^? s & \\
\rightarrow & \{x \mapsto s\}P \wedge x =^? s & \text{if } x \notin \mathcal{V}ar(s), s \notin \mathcal{X}, x \in \mathcal{V}ar(P) \\
 & & \\
\textit{Merge} & P \wedge x =^? s \wedge x =^? t & \\
\rightarrow & P \wedge x =^? s \wedge s =^? t & \text{if } 0 < |s| \leq |t|
\end{array}
$$

Rules for syntactic unification

$$
\begin{array}{llll}
\textit{Eliminate} & P \wedge x =^? s & & \\
\rightarrow & \{x \mapsto s\}P \wedge x =^? s & & \text{if } x \notin \mathcal{V}ar(s), s \notin \mathcal{X}, x \in \mathcal{V}ar(P) \\
& & & \\
\textit{Merge} & P \wedge x =^? s \wedge x =^? t & & \\
\rightarrow & P \wedge x =^? s \wedge s =^? t & & \text{if } 0 < |s| \leq |t| \\
& & & \\
\textit{Check} & P \wedge x =^? s & & \\
\rightarrow & \textit{Fail} & & \text{if } x \in \mathcal{V}ar(s) \textit{ and } s \notin \mathcal{X}
\end{array}
$$

Rules for syntactic unification

$$
\begin{array}{lll}
\textit{Eliminate} & P \wedge x =^? s & \\
\rightarrow & \{x \mapsto s\}P \wedge x =^? s & \text{if } x \notin \mathcal{V}ar(s), s \notin \mathcal{X}, x \in \mathcal{V}ar(P) \\
 & & \\
\textit{Merge} & P \wedge x =^? s \wedge x =^? t & \\
\rightarrow & P \wedge x =^? s \wedge s =^? t & \text{if } 0 < |s| \leq |t| \\
 & & \\
\textit{Check} & P \wedge x =^? s & \\
\rightarrow & \textit{Fail} & \text{if } x \in \mathcal{V}ar(s) \textit{ and } s \notin \mathcal{X} \\
 & & \\
\textit{Check}* & P \wedge \;\; x_1 =^? s_1[x_2] \wedge \ldots & \\
 & \qquad \ldots \wedge x_n =^? s_n[x_1] & \\
\rightarrow & \textit{Fail} & \text{if } s_i \notin \mathcal{X} \textit{ for some } i \in [1..n]
\end{array}
$$

Examples

$$x+(z*y) =^? y+(x*z)$$

$$\Rightarrow_{\text{decompose}} x =^? y \;\wedge\; z*y =^? x*z$$

$$\Rightarrow_{\text{decompose}} x =^? y \;\wedge\; z =^? x \;\wedge\; y =^? z$$

$$\Rightarrow_{\text{coalesce}} y =^? z \;\wedge\; x =^? z \;\wedge\; z =^? x$$

$$\Rightarrow_{\text{coalesce}} z =^? x \;\wedge\; y =^? x \;\wedge\; x =^? x$$

$$\Rightarrow_{\text{delete}} z =^? x \;\wedge\; y =^? x$$

Examples

$$f(a, f(x, x)) =^? f(x, f(a, y))$$

Examples

$$f(a, f(x, x)) =^? f(x, f(a, y))$$
$$f(x, f(y, z)) =^? f(f(a, b), f(x, b)$$

Examples

$$
\begin{aligned}
&f(a, f(x, x)) =^? f(x, f(a, y)) \\
&f(x, f(y, z)) =^? f(f(a, b), f(x, b) \\
&f(x, f(f(x, b), a)) =^? f(f(a, y), f(y, a))
\end{aligned}
$$

A *tree solved form* for P is any conjonction of equations

$$x_1 =^? t_1 \wedge \cdots \wedge x_n =^? t_n$$

equivalent to P such that $\forall i, x_i \in \mathcal{X}$ and:

$$\begin{array}{ll} (i) & \forall 1 \leq i \leq n, x_i \in \mathcal{V}ar(P), \\ (ii) & \forall 1 \leq i, j \leq n, i \neq j \Rightarrow x_i \neq x_j, \\ (iii) & \forall 1 \leq i, j \leq n, x_i \notin \mathcal{V}ar(t_j). \end{array}$$

Example: $x =^? f(f(y)) \wedge z =^? g(a)$.

Theorem: Starting with a unification problem P and using the above rules repeatedly until none is applicable
— results in *Fail* iff P has no solution, or else it
— results in a tree solved form $x_1 =^? t_1 \wedge \cdots \wedge x_n =^? t_n$ with the same set of solutions than P.
Moreover

$$\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$$

is a most general unifier of P.

Critical pair

A non-variable term t' and a term t overlap if there exists a position ω in t such that $t_{|\omega}$ and t' are unifiable (with $t_{|\omega}$ not a variable).

Where does $(x + y) + z$ and $(x' + y') + z'$ overlap?

Where does $0 + x \to x$ and $s(x') + y \to s(x' + y)$ overlap?

How about $H \vdash \overline{\overline{P}} \to H \vdash P$ and $H' \wedge P' \vdash P' \to T$?

Critical Pairs

Critical pair

$$\mathbf{l_1} \to \mathbf{r_1} \qquad\qquad \mathbf{l_2}[\mathbf{u}] \to \mathbf{r_2}$$

$$(\mathbf{l_2}[\mathbf{r_1}]\sigma, \mathbf{r}_2\sigma)$$

u is a non-variable sub-term of l_2
σ is the $mgu(u, l_1)$

Critical Pair Lemma

R is locally confluent iff all critical pair satisfies:

$$\mathbf{l}_2[r_1]\sigma \xrightarrow{*}_R \quad \otimes \quad {}_R\xleftarrow{*} r_2\sigma$$

Orthogonal systems

A rewrite system that is both linear (the left-hand side of each rule is a linear term) and non-overlapping is called orthogonal.

Orthogonal systems

A rewrite system that is both linear (the left-hand side of each rule is a linear term) and non-overlapping is called orthogonal.

Theorem

If a rewrite system is orthogonal, then it is confluent.

Orthogonal systems

A rewrite system that is both linear (the left-hand side of each rule is a linear term) and non-overlapping is called orthogonal.

Theorem

If a rewrite system is orthogonal, then it is confluent.

Linearity is needed for non-terminating rewriting system:

$$\begin{array}{lcl} d(x,x) & \to & t \\ d(x,c(x)) & \to & f \\ a & \to & c(a) \end{array}$$

Orthogonal systems

Exemple:

$$\begin{array}{ccc} \bullet\bullet & \longrightarrow & \circ \\ \circ\circ & \longrightarrow & \circ \\ \bullet\circ & \longrightarrow & \bullet \\ \circ\bullet & \longrightarrow & \bullet \end{array}$$

Other systems

Theorem Consider a reduction relation $\rightarrow_R$ and let $\rightarrow_D$ s.t.

- $\rightarrow_R \subseteq \rightarrow_D \subseteq \xrightarrow{*}_R$
- $\rightarrow_D$ has the diamond property

Then, $\rightarrow_R$ is confluent.

Exercices

Show the confluence of the two rewriting systems:

$$x + zero \to x$$
$$zero + x \to x$$
$$suc(y) + x \to suc(x + y)$$
$$x + suc(y) \to suc(x + y)$$

Show the confluence of the two rewriting systems:

$$x + zero \to x$$
$$zero + x \to x$$
$$suc(y) + x \to suc(x + y)$$
$$x + suc(y) \to suc(x + y)$$

CP:

$x + zero = zero + x \Longrightarrow zero + zero$

$x + zero = suc(y) + x \Longrightarrow suc(y) + zero$

$\ldots$

Show the confluence of the two rewriting systems:

$$x + zero \to x$$

$$x * zero \to zero$$

$$suc(y) + x \to suc(x + y)$$

$$x * suc(y) \to x + (x * y)$$

Show the confluence of the two rewriting systems:

$$x + zero \to x$$
$$x * zero \to zero$$
$$suc(y) + x \to suc(x + y)$$
$$x * suc(y) \to x + (x * y)$$

CP:

$x * zero = zero * x \Longrightarrow zero * zero$

$\ldots$

Syntactic filtering

$$
\begin{array}{lll}
 & t \ll^? t \wedge P & \\
\to & P & \\
 & f(t_1, \ldots, t_n) \ll^? f(t'_1, \ldots, t'_n) \wedge P & \\
\to & \bigwedge_{i=1,\ldots,n} t_i \ll^? t'_i \wedge P & \\
 & f(t_1, \ldots, t_n) \ll^? g(t'_1, \ldots, t'_m) \wedge P & \\
\to & \mathcal{F} & \text{if } f \neq g \\
 & x \ll^? t \wedge x \ll^? t' \wedge P & \\
\to & \mathcal{F} & \text{if } t \neq t' \\
 & f(t_1, \ldots, t_n) \ll^? x \wedge P & \\
\to & \mathcal{F} & \text{if } x \in \mathcal{X}
\end{array}
$$

Confluence?

Groups

$$
\begin{aligned}
x + e &\to x \\
e + x &\to x \\
x + (y + z) &\to (x + y) + z \\
x + i(x) &\to e \\
i(x) + x &\to e
\end{aligned}
\qquad\qquad
\begin{aligned}
i(e) &\to e \\
(y + i(x)) + x &\to y \\
(y + x) + i(x) &\to y \\
i(i(x)) &\to x \\
i(x + y) &\to i(y) + i(x)
\end{aligned}
$$

Groups

$$
\begin{array}{lcl}
x + e & \to & x \\
e + x & \to & x \\
x + (y + z) & \to & (x + y) + z \\
x + i(x) & \to & e \\
i(x) + x & \to & e
\end{array}
\qquad
\begin{array}{lcl}
i(e) & \to & e \\
(y + i(x)) + x & \to & y \\
(y + x) + i(x) & \to & y \\
i(i(x)) & \to & x \\
i(x + y) & \to & i(y) + i(x)
\end{array}
$$

Confluence:

Groups

$$
\begin{aligned}
x + e &\to x & \qquad i(e) &\to e \\
e + x &\to x & (y + i(x)) + x &\to y \\
x + (y + z) &\to (x + y) + z & (y + x) + i(x) &\to y \\
x + i(x) &\to e & i(i(x)) &\to x \\
i(x) + x &\to e & i(x + y) &\to i(y) + i(x)
\end{aligned}
$$

Confluence:

$x + e = e + x$

$x + e = i(x) + x$

$x + e = (y + i(x)) + x$

$x + e =_{x+y} i(x + y)$

$e + x =_{y+i(x)} (y + i(x)) + x$