# Rewriting

## Part 6. Confluence of Term Rewriting Systems

Temur Kutsia

RISC, JKU Linz

# Confluence is undecidable

The following problem is undecidable:

Given: A finite TRS $R$.

Question: Is $R$ confluent or not?

# Confluence is undecidable

The following problem is undecidable:

Given: A finite TRS $R$.

Question: Is $R$ confluent or not?

## Proof.

Idea:

- Given a set of identities $E$ such that $\mathcal{V}ar(l) \approx \mathcal{V}ar(r)$ for all $l \approx r \in E$, $l$ and $r$ not being variables.
- Construct a TRS whose confluence problem is equivalent to the ground word problem for $E$.
- Undecidability of the ground word problem for $E$ (see e.g. Examples 4.1.3 and 4.1.4 from the book of Baader and Nipkow) will imply undecidability of the confluence problem.

# Confluence is undecidable

The following problem is undecidable:

Given: A finite TRS $R$.

Question: Is $R$ confluent or not?

**Proof.**

Construction of a TRS:

1. $R := E \cup E^{-1}$ is a confluent TRS.
2. $R_{st} := R \cup \{a \to s, a \to t\}$, where $s$ and $t$ are given ground terms and $a$ is a new constant.
3. $R_{st}$ is confluent iff $s \approx_E t$.

Hence, the ground word problem for $E$ reduces to the confluence problem for $R_{st}$. □

# A decidable subcase

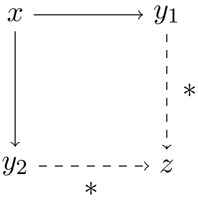
**Theorem 6.1**
*For terminating TRSs, confluence is decidable.*

Proof idea:

- By Newman's lemma, if a TRS is terminating and locally confluent, then it is confluent.
- To prove the theorem, we need to prove that local confluence is decidable for terminating TRSs.

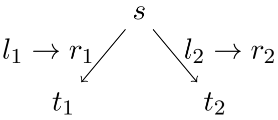
# How to test local confluence?

Local confluence:



# How to test local confluence?

To test for local confluence of $\to_R$, consider reductions:




That means, there are rules $l_1 \to r_1, l_2 \to r_2 \in R$, positions $p_1, p_2 \in \mathcal{P}os(s)$, and substitutions $\sigma_1, \sigma_2$ such that

- $s|_{p_1} = \sigma_1(l_1)$ and $t_1 = s[\sigma_1(r_1)]_{p_1}$.
- $s|_{p_2} = \sigma_2(l_2)$ and $t_2 = s[\sigma_2(r_2)]_{p_2}$.

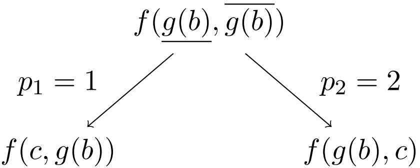
Consider several cases, depending on the relative positions of $p_1$ and $p_2$.

# How to test local confluence?

Case 1: $p_1$ and $p_2$ are parallel positions.

Example: $R := \{f(a, g(x)) \to f(x, x),\ g(b) \to c\}$

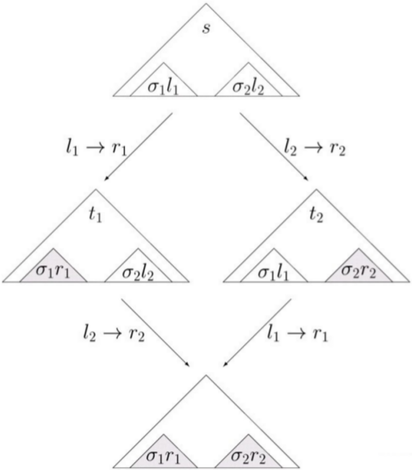
Peak:



# How to test local confluence?

Case 1: $p_1$ and $p_2$ are parallel positions.

Outcome: The reducts are joinable.



# How to test local confluence?

Case 1: $p_1$ and $p_2$ are parallel positions.

Outcome: The reducts are joinable.

Example: $R \ := \ \{f(a, g(x)) \to f(x, x), \ g(b) \to c\}$

Peak:

$$f(c, g(b)) \xleftarrow{\ p_1 = 1\ } f(\underline{g(b)}, \overline{g(b)}) \xrightarrow{\ p_2 = 2\ } f(g(b), c)$$

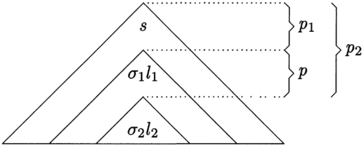
Joinability:

$$f(c, \underline{g(b)}) \to f(c, c)$$

$$f(\underline{g(b)}, c) \to f(c, c)$$

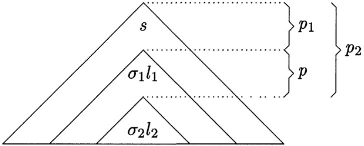
# How to test local confluence?

Case 2: One position is a prefix of another.
Say, $p_1$ is a prefix of $p_2$: $p_2 = p_1 p$ for some $p$.



# How to test local confluence?

Case 2: One position is a prefix of another.
Say, $p_1$ is a prefix of $p_2$: $p_2 = p_1 p$ for some $p$.




We restrict our attention to $\sigma_1(l_1)$, because

$$\begin{array}{ccc} & \sigma_1(l_1) & \\ \swarrow & & \searrow \\ \sigma_1(r_1) & & \sigma_1(l_1)[\sigma_2(r_2)]_p \\ {}^*\searrow & & \swarrow{}^* \\ & t & \end{array}$$

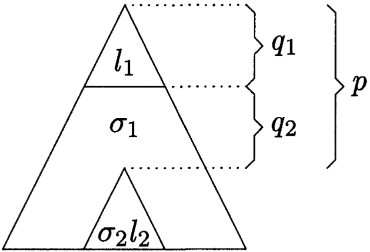
implies $s[\sigma_1(r_1)]_{p_1} \xrightarrow{*} s[t] \xleftarrow{*} s[\sigma_1(l_1)[\sigma_2(r_2)]_p]_{p_1} = s[\sigma_2(r_2)]_{p_2}$.

# How to test local confluence?

Case 2.1: The redex $\sigma_2(l_2)$ does not overlap with $l_1$ itself, but is contained in $\sigma_1$.

$p = q_1 q_2$ such that $q_1$ is a variable position in $l_1$.

$\sigma_1(l_1)$ has the form:




Non-critical overlap.

# How to test local confluence?

Case 2.1: The redex $\sigma_2(l_2)$ does not overlap with $l_1$ itself, but is contained in $\sigma_1$.

$p = q_1 q_2$ such that $q_1$ is a variable position in $l_1$.

Example: $R \ := \ \{f(a, g(x)) \to f(x, x), \ g(b) \to c\}$

Peak:

$$\underline{f(a, g(g(\overline{g(b)})))}$$

$$p_1 = \epsilon \swarrow \qquad \searrow p_2 = 211$$

$$f(g(g(b)), g(g(b))) \qquad\qquad f(a, g(g(c)))$$

$l_1 = f(a, g(x))$, $\sigma_1 = \{x \mapsto g(g(b))\}$, $l_2 = g(b)$, $\sigma_2 = \varepsilon$.

$p = 211$, $q_1 = 21$, $q_2 = 1$.

# How to test local confluence?

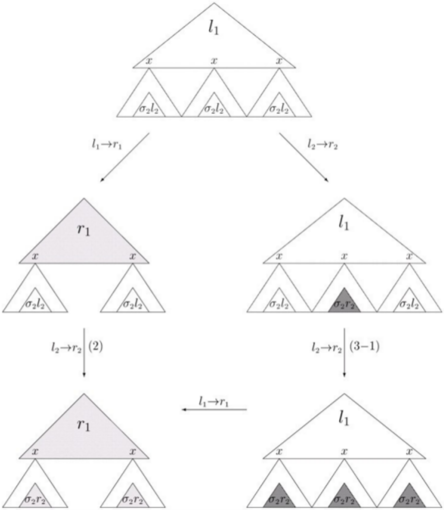
Case 2.1: The redex $\sigma_2(l_2)$ does not overlap with $l_1$ itself, but is contained in $\sigma_1$.
$p = q_1 q_2$ such that $q_1$ is a variable position in $l_1$.

Outcome: The reducts are joinable.

The analysis is complicated by the fact that $x = l_1|_{q_1}$ may occur repeatedly both in $l_1$ and $r_1$.

# How to test local confluence?

Case 2.1: Instance: $x$ appears three times in $l_1$ and twice in $r_1$.



# How to test local confluence?

Case 2.1: The redex $\sigma_2(l_2)$ does not overlap with $l_1$ itself, but is contained in $\sigma_1$.
$p = q_1 q_2$ such that $q_1$ is a variable position in $l_1$.

Example: $R := \{f(a, g(x)) \to f(x, x),\ g(b) \to c\}$

Peak:

$$\underline{f(a, g(g(\overline{g(b)})))}$$

$$p_1 = \epsilon \swarrow \qquad \searrow p_2 = 211$$

$$f(g(g(b)), g(g(b))) \qquad\qquad f(a, g(g(c)))$$

The reducts are joinable.

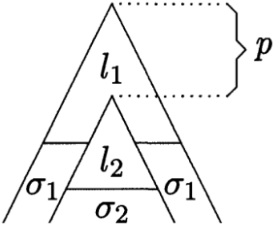
$f(g(g(b)), g(g(b)) \xrightarrow{2} f(g(c), g(c))$.

$f(a, g(g(c))) \to f(g(c), g(c))$.

# How to test local confluence?

Case 2.2: Two left-hand sides $l_1$ and $l_2$ overlap.
$p \in \mathcal{P}os(l_1)$, $l_1|_p$ is not a variable, and
$\sigma_1(l_1|_p) = \sigma_2(l_2)$.

$\sigma_1(l_1)$ has the form:



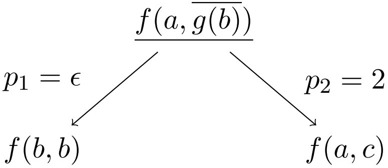

Critical overlap.

# How to test local confluence?

Case 2.2: Two left-hand sides $l_1$ and $l_2$ overlap.
$p \in \mathcal{P}os(l_1)$, $l_1|_p$ is not a variable, and
$\sigma_1(l_1|_p) = \sigma_2(l_2)$.

In the case of critical overlap, local confluence need not hold.

Example: $R \ := \ \{f(a, g(x)) \to f(x, x), \ g(b) \to c\}$




$l_1 = f(a, g(x))$, $\sigma_1 = \{x \mapsto b\}$, $l_2 = g(b)$, $\sigma_2 = \varepsilon$.
$p = 2$.

# How to test local confluence?

Case 2.2: Two left-hand sides $l_1$ and $l_2$ overlap.
$p \in \mathcal{P}os(l_1)$, $l_1|_p$ is not a variable, and
$\sigma_1(l_1|_p) = \sigma_2(l_2)$.

Problem: Critical overlaps must be checked for local confluence. How to do that?

Answer: It is enough to check finitely many critical pairs.

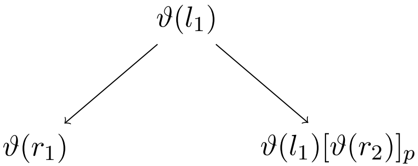
# How to test local confluence?

## Definition 6.1

Let

- $l_1 \to r_1$ and $l_2 \to r_2$ be two rules which do not share variables,
- $p \in \mathcal{P}os(l_1)$ be a position such that $l_1|_p$ is not a variable, and
- $\vartheta$ be an mgu of $l_1|p$ and $l_2$

Then the pair $\langle \vartheta(r_1), \vartheta(l_1)[\vartheta(r_2)]_p \rangle$ is called a critical pair.



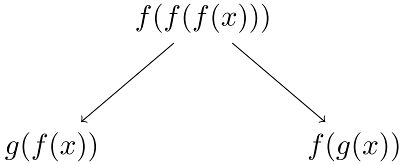
# How to test local confluence?

- The critical pairs of a TRS $R$ are the critical pairs between any of two of its renamed rules and are denoted by $CP(R)$.
- Includes overlaps of a rule with a renamed copy of itself.

# How to test local confluence?

## Example 6.1

- Let $R := \{f(f(x)) \to g(x)\}$.
- Take a critical pair between the rule and its renamed copy, $f(f(x)) \to g(x)$ and $f(f(y)) \to g(y)$




- The terms in the critical pair, $g(f(x))$ and $f(g(x))$, are not joinable.
- $R$ is not locally confluent.

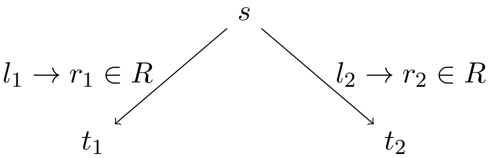
# How to test local confluence?

- Hence, local confluence test reduces to checking joinability of critical pairs.
- The analysis of the cases on the previous slides leads to the Critical Pair Lemma.

# How to test local confluence?

## Lemma 6.1 (Critical Pair Lemma)

*If $R$ is a TRS and*




*then $t_1 \downarrow_R t_2$, or $t_1 = s[u_1]_{p_1}$ and $t_2 = s[u_2]_{p_2}$ for some $p_1, p_2$, where $\langle u_1, u_2 \rangle$ or $\langle u_2, u_1 \rangle$ is an instance of a critical pair of $R$.*

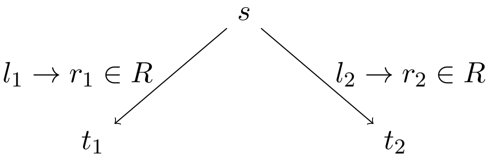
## Proof.

- When there is no overlap or a non-critical overlap, then $t_1 \downarrow_R t_2$.
- When there is a critical overlap, then $s|_{p_1} = \sigma(l_1)$ and $\sigma(l_1|_p) = \sigma(l_2)$.

# How to test local confluence?

## Lemma 6.1 (Critical Pair Lemma)

*If $R$ is a TRS and*




*then $t_1 \downarrow_R t_2$, or $t_1 = s[u_1]_{p_1}$ and $t_2 = s[u_2]_{p_2}$ for some $p_1, p_2$, where $\langle u_1, u_2 \rangle$ or $\langle u_2, u_1 \rangle$ is an instance of a critical pair of $R$.*

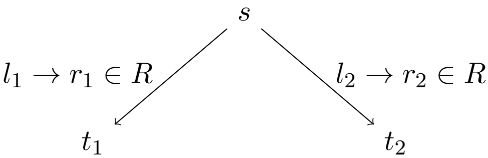
## Proof (cont.)

- Hence, $\sigma$ unifies $l_1|_p$ and $l_2$ and, therefore, is an instance of their mgu $\vartheta$.
- Therefore, $\langle \sigma(r_1), \sigma(l_1)[\sigma(r_2)]_p \rangle$ is an instance of the critical pair $\langle \vartheta(r_1), \vartheta(l_1)[\vartheta(r_2)]_p \rangle$

# How to test local confluence?

## Lemma 6.1 (Critical Pair Lemma)

*If $R$ is a TRS and*




*then $t_1 \downarrow_R t_2$, or $t_1 = s[u_1]_{p_1}$ and $t_2 = s[u_2]_{p_2}$ for some $p_1, p_2$, where $\langle u_1, u_2 \rangle$ or $\langle u_2, u_1 \rangle$ is an instance of a critical pair of $R$.*

## Proof (cont.)

- $t_1 = s[\sigma(r_1)]_{p_1}$, $t_2 = s[\sigma(l_1)[\sigma(r_2)]_p]_{p_1}$, $p_2 = p_1 p$.

□

# How to test local confluence?

### Theorem 6.2 (Critical Pair Theorem)

*A TRS is locally confluent iff all its critical pairs are joinable.*

### Proof.

($\Leftarrow$) Using the Critical Pair Lemma: Given $t_i = s[u_i]_p$, $i = 1, 2$, where $\langle u_1, u_2 \rangle$ (wlog) is an instance of some critical pair $\langle v_1, v_2 \rangle$ under a substitution $\varphi$, then $v_i \xrightarrow{*} t$ for some term $t$ implies $u_i \xrightarrow{*} \varphi(t)$ and, hence, $t_i \xrightarrow{*} s[\varphi(t)]_p$, $i = 1, 2$.

□

# How to test local confluence?

### Theorem 6.2 (Critical Pair Theorem)

*A TRS is locally confluent iff all its critical pairs are joinable.*

### Proof.

($\Leftarrow$) Using the Critical Pair Lemma: Given $t_i = s[u_i]_p$, $i = 1, 2$, where $\langle u_1, u_2 \rangle$ (wlog) is an instance of some critical pair $\langle v_1, v_2 \rangle$ under a substitution $\varphi$, then $v_i \xrightarrow{*} t$ for some term $t$ implies $u_i \xrightarrow{*} \varphi(t)$ and, hence, $t_i \xrightarrow{*} s[\varphi(t)]_p$, $i = 1, 2$.

($\Rightarrow$) Every critical pair is the product of a fork $\vartheta(r_1) \leftarrow \vartheta(l_1) \rightarrow \vartheta(l_1)[\vartheta(r_2)]_p$. Joinability follows from local confluence.

□

# How to test local confluence?

Theorem 6.2 (Critical Pair Theorem)

*A TRS is locally confluent iff all its critical pairs are joinable.*

Corollary 6.1

*A terminating TRS is confluent iff all its critical pairs are joinable.*

# How to test local confluence?

- The problem of testing local confluence reduces to critical pair joinability test.
- For terminating TRSs, the problem whether two terms are joinable can be decided.
- For finite TRSs, the number of critical pairs is finite.
- Hence, for terminating and finite TRSs local confluence is decidable.
- Therefore, for terminating and finite TRSs confluence is decidable.

# Deciding (local) confluence for terminating finite TRSs

Let $R$ be a terminating finite TRS.

Decision procedure:

# Deciding (local) confluence for terminating finite TRSs

Let $R$ be a terminating finite TRS.

Decision procedure:

- For each pair of rules $l_1 \to r_1$ and $l_2 \to r_2$ (there are $|R|^2$ of them) and for every $p \in \mathcal{P}os(l_1)$ with a nonvariable $l_1|_p$ (there are at most $|l_1|$ of them) try to generate critical pairs.

# Deciding (local) confluence for terminating finite TRSs

Let $R$ be a terminating finite TRS.

Decision procedure:

- For each pair of rules $l_1 \to r_1$ and $l_2 \to r_2$ (there are $|R|^2$ of them) and for every $p \in \mathcal{P}os(l_1)$ with a nonvariable $l_1|_p$ (there are at most $|l_1|$ of them) try to generate critical pairs.
- It involves unification of $l_1|_p$ and $l_2$ (decidable, unitary).

# Deciding (local) confluence for terminating finite TRSs

Let $R$ be a terminating finite TRS.

Decision procedure:

- For each pair of rules $l_1 \to r_1$ and $l_2 \to r_2$ (there are $|R|^2$ of them) and for every $p \in \mathcal{P}os(l_1)$ with a nonvariable $l_1|_p$ (there are at most $|l_1|$ of them) try to generate critical pairs.
- It involves unification of $l_1|_p$ and $l_2$ (decidable, unitary).
- Result: finitely many critical pairs.

# Deciding (local) confluence for terminating finite TRSs

Let $R$ be a terminating finite TRS.

Decision procedure:

- For each pair of rules $l_1 \to r_1$ and $l_2 \to r_2$ (there are $|R|^2$ of them) and for every $p \in \mathcal{P}os(l_1)$ with a nonvariable $l_1|_p$ (there are at most $|l_1|$ of them) try to generate critical pairs.
- It involves unification of $l_1|_p$ and $l_2$ (decidable, unitary).
- Result: finitely many critical pairs.
- For each critical pair $\langle u_1, u_2 \rangle$, reduce $u_i$, to some $R$-normal form $\hat{u}_i$, $i = 1, 2$.

# Deciding (local) confluence for terminating finite TRSs

Let $R$ be a terminating finite TRS.

Decision procedure:

- For each pair of rules $l_1 \to r_1$ and $l_2 \to r_2$ (there are $|R|^2$ of them) and for every $p \in \mathcal{P}os(l_1)$ with a nonvariable $l_1|_p$ (there are at most $|l_1|$ of them) try to generate critical pairs.
- It involves unification of $l_1|_p$ and $l_2$ (decidable, unitary).
- Result: finitely many critical pairs.
- For each critical pair $\langle u_1, u_2 \rangle$, reduce $u_i$, to some $R$-normal form $\hat{u}_i$, $i = 1, 2$.
- If $\hat{u}_1 = \hat{u}_2$ for all such pairs, $R$ is confluent (Corollary 6.1).

# Deciding (local) confluence for terminating finite TRSs

Let $R$ be a terminating finite TRS.

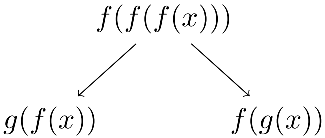
Decision procedure:

- For each pair of rules $l_1 \to r_1$ and $l_2 \to r_2$ (there are $|R|^2$ of them) and for every $p \in \mathcal{P}os(l_1)$ with a nonvariable $l_1|_p$ (there are at most $|l_1|$ of them) try to generate critical pairs.
- It involves unification of $l_1|_p$ and $l_2$ (decidable, unitary).
- Result: finitely many critical pairs.
- For each critical pair $\langle u_1, u_2 \rangle$, reduce $u_i$, to some $R$-normal form $\hat{u}_i$, $i = 1, 2$.
- If $\hat{u}_1 = \hat{u}_2$ for all such pairs, $R$ is confluent (Corollary 6.1).
- If $\hat{u}_1 \neq \hat{u}_2$ for such a pair, we have a non-confluent situation: $\hat{u}_1 \xleftarrow{*} u_1 \leftarrow u \to u_2 \xrightarrow{*} \hat{u}_2$.

# Deciding (local) confluence for terminating finite TRSs

Example 6.2

Recall the TRS $\{f(f(x)) \to g(x)\}$, which is not locally confluent.
The only critical pair $\langle g(f(x)), f(g(x)) \rangle$ is not joinable.



# Deciding (local) confluence for terminating finite TRSs

Example 6.2

Recall the TRS $\{f(f(x)) \to g(x)\}$, which is not locally confluent.
The only critical pair $\langle g(f(x)), f(g(x)) \rangle$ is not joinable.

# Deciding (local) confluence for terminating finite TRSs

Example 6.2

Recall the TRS $\{f(f(x)) \to g(x)\}$, which is not locally confluent. The only critical pair $\langle g(f(x)), f(g(x)) \rangle$ is not joinable.

This example illustrates that the two conditions in the definition of the critical pairs are necessary:

# Deciding (local) confluence for terminating finite TRSs

## Example 6.2

Recall the TRS $\{f(f(x)) \to g(x)\}$, which is not locally confluent. The only critical pair $\langle g(f(x)), f(g(x)) \rangle$ is not joinable.

This example illustrates that the two conditions in the definition of the critical pairs are necessary:

- Rules are to be renamed. Otherwise $f(f(x))$ and $f(x)$ are not unifiable.

# Deciding (local) confluence for terminating finite TRSs

Example 6.2

Recall the TRS $\{f(f(x)) \to g(x)\}$, which is not locally confluent. The only critical pair $\langle g(f(x)), f(g(x)) \rangle$ is not joinable.

This example illustrates that the two conditions in the definition of the critical pairs are necessary:

- Rules are to be renamed. Otherwise $f(f(x))$ and $f(x)$ are not unifiable.
- The critical pair of a rule and (a renamed copy of) itself has to be taken into account. Otherwise all one-rule systems would appear to be locally-confluent.

# Deciding (local) confluence for terminating finite TRSs

### Example 6.2

Recall the TRS $\{f(f(x)) \to g(x)\}$, which is not locally confluent. The only critical pair $\langle g(f(x)), f(g(x)) \rangle$ is not joinable.

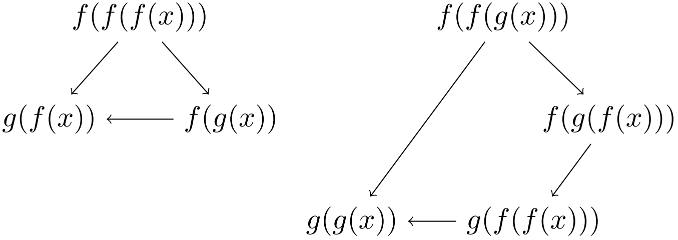
This example illustrates that the two conditions in the definition of the critical pairs are necessary:

- Rules are to be renamed. Otherwise $f(f(x))$ and $f(x)$ are not unifiable.
- The critical pair of a rule and (a renamed copy of) itself has to be taken into account. Otherwise all one-rule systems would appear to be locally-confluent.
- Critical pairs can be helpful lemmas: $g(f(x)) \approx_R f(g(x))$ is an interesting consequence of $f(f(x)) \to_R g(x)$ which may not be apparent at first sight.

# Deciding (local) confluence for terminating finite TRSs

Example 6.3

The TRS $\{f(f(x)) \to g(x), f(g(x)) \to g(f(x))\}$ is locally confluent. Both critical pairs are joinable:




Since the TRS is also terminating (use LPO with $f > g$), it is also confluent.

# Deciding (local) confluence for terminating finite TRSs

- Because critical pairs are equational consequences, adding a critical pair as a new rewrite rule does not change the induced equality.
- If $R$ is a TRS and $R'$ is obtained from $R$ by adding a critical pair as a new rule, then $\approx_R = \approx_{R'}$.
- The idea of adding a critical pair as a new rule is called "completion".