

# Automated Reasoning

## Lecture 15: Rewriting II

Jacques Fleuriot  
`jdf@inf.ed.ac.uk`

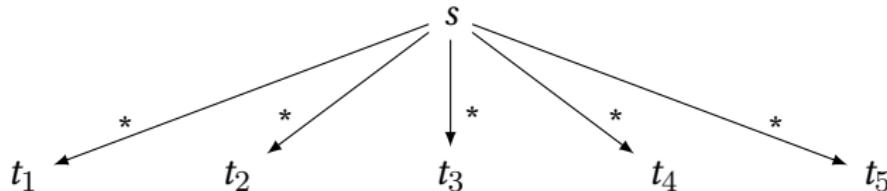
# Recap

- ▶ Previously: Rewriting
  - ▶ Definition of Rewrite Rule of Inference
  - ▶ Termination
  - ▶ Rewriting in Isabelle
- ▶ This time: More of the same!
  - ▶ Canonical normal forms
  - ▶ Confluence
  - ▶ Critical Pairs
  - ▶ Knuth-Bendix Completion

## Canonical Normal Form

For some rewrite rule sets, order of application might affect result.

We might have:



where all of  $t_1, t_2, t_3, t_4, t_5$  are in normal form after multiple (zero or more) rewrite rule applications.

If all the normal forms are identical we can say we have a **canonical** normal form for  $s$ .

This is a very nice property!

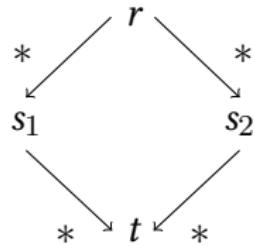
- ▶ Means that order of rewrite rule application doesn't matter
- ▶ In general, means our rewrites are simplifying the expression in a canonical (safe) way.

# Confluence and Church-Rosser

How do we know when a set of rules yields canonical normal forms?

A set of rewrite rules is **confluent** if for all terms  $r$ ,  $s_1$ ,  $s_2$  such that  $r \rightarrow^* s_1$  and  $r \rightarrow^* s_2$  there exists a term  $t$  such that  $s_1 \rightarrow^* t$  and  $s_2 \rightarrow^* t$ .

A set of rewrite rules is **Church-Rosser** if for all terms  $s_1$  and  $s_2$  such that  $s_1 \leftrightarrow^* s_2$ , there exists a term  $t$  such that  $s_1 \rightarrow^* t$  and  $s_2 \rightarrow^* t$ .



## Theorem

*Church-Rosser is equivalent to confluence.*

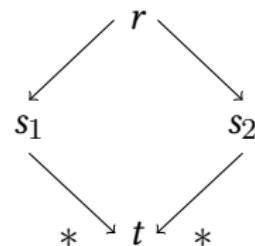
## Theorem

*For terminating rewrite sets, these properties mean that any expression will rewrite to a canonical normal form.*

# Local Confluence

The properties of Church-Rosser and confluence can be difficult to prove. A weaker definition is useful:

A set of rewrite rules is **locally confluent** if for all terms  $r, s_1, s_2$  such that  $r \rightarrow s_1$  and  $r \rightarrow s_2$  there exists a term  $t$  such that  $s_1 \rightarrow^* t$  and  $s_2 \rightarrow^* t$ .



## Theorem (Newman's Lemma)

$$\text{local confluence} + \text{termination} = \text{confluence}$$

Also: local confluence is decidable (due to Knuth and Bendix)

Both theorem and the decision procedure use idea of **critical pairs**

# Choices in Rewriting

How can choices arise in rewriting?

- ▶ Multiple rules apply to a single redex: **order might matter**
- ▶ Rules apply to multiple redexes:
  - ▶ if they are separate: **order does not matter**
  - ▶ if one contains the other: **order might matter**

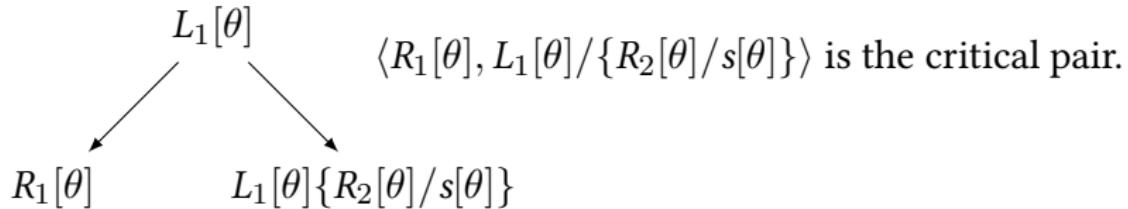
We are interested in cases where the order matters:

Rules	Rewrites	Critical Pair
$X^0 \Rightarrow 1$	$0^0$ rewrites to 0 and to 1	$\langle 0, 1 \rangle$
$0^Y \Rightarrow 0$		
$X \cdot e \Rightarrow X$	$(x \cdot e) \cdot z$ rewrites to	$\langle x \cdot z, x \cdot (e \cdot z) \rangle$
$(X \cdot Y) \cdot Z \Rightarrow X \cdot (Y \cdot Z)$	$x \cdot z$ and $x \cdot (e \cdot z)$	

## Critical Pairs

Given two rules  $L_1 \Rightarrow R_1$  and  $L_2 \Rightarrow R_2$ , we are concerned with the case when there exists a *non-variable* sub-term  $s$  of  $L_1$  such that  $s[\theta] = L_2[\theta]$ , with most general unifier  $\theta$ .

Applying these rules in different orders gives rise to a **critical pair**, where  $L_1[\theta]\{R_2[\theta]/s[\theta]\}$  denotes replacing  $s[\theta]$  by  $R_2[\theta]$  in  $L_1[\theta]$ .



**Note:** the variables in the two rules should be *renamed* so they do **not** share any variable names.

**Note:** A rewrite rule may have critical pairs with itself e.g. consider the rule  $f(f(x)) \Rightarrow g(x)$ .

With  $W \cdot e \Rightarrow W$  and  $(X \cdot Y) \cdot Z \Rightarrow X \cdot (Y \cdot Z)$ , where  $X, Y$  and  $Z$  are variables, we can have  $\theta = [W/X, e/Y]$ , **any other?**

## Critical Pairs: Example

Consider the rewrite rules:

$$\begin{array}{ccc} \overbrace{f(f(x, y), z)}^{L_1} & \Rightarrow & \overbrace{f(x, f(y, z))}^{R_1} \\ s & & \\ \overbrace{f(i(x_1), x_1)}^{L_2} & \Rightarrow & \overbrace{e}^{R_2} \end{array}$$

The mgu  $\theta$ , given our choice of non-variable subterm  $s$  of  $L_1$ , is given by  $\theta = \{i(x_1)/x, x_1/y\}$  and by considering:

$$\begin{array}{ccc} f(f(i(x_1), x_1), z) & & \\ \searrow & & \searrow \\ f(i(x_1), f(x_1, z)) & & f(e, z) \end{array}$$

We get the critical pair  $\langle f(i(x_1), f(x_1, z)), f(e, z) \rangle$ .

## Testing for Local Confluence

If we can **conflate** (join) all the critical pairs, then have **local confluence**.

**Conflation** for a critical pair  $\langle s_1, s_2 \rangle$  is when there is a  $t$  such that  $s_1 \longrightarrow^* t$  and  $s_2 \longrightarrow^* t$ .

An algorithm to test for local confluence (assuming termination):

1. Find all the critical pairs in set of rewrite rules  $R$
2. For each critical pair  $\langle s_1, s_2 \rangle$ :
  - 2.1 Find a normal form  $s'_1$  of  $s_1$ ;
  - 2.2 Find a normal form  $s'_2$  of  $s_2$ ;
  - 2.3 Check  $s'_1 = s'_2$ , if not then fail.

## Establishing Local Confluence

Sometimes a set of rules is not locally confluent

$X \cdot e \Rightarrow X$   
 $f \cdot X \Rightarrow X$  is not locally confluent:  $\langle f, e \rangle$  does not conflate.

We can add the rule  $f \Rightarrow e$  to make this critical pair joinable.

However, adding new rules requires **care**:

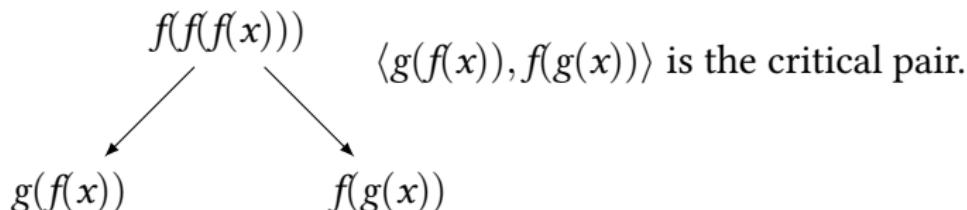
- ▶ Must preserve termination
- ▶ Might give rise to *new* critical pairs and so we may need to check local confluence again.

## Establishing Local Confluence: Example

Consider the set  $R$  consisting of just one rewrite rule, with  $x$  a variable:

$$f(f(x)) \Rightarrow g(x)$$

which has exactly one critical pair (CP) when it is overlapped with a *renamed* copy of itself  $f(f(y)) \Rightarrow g(y)$ . The lhs  $f(f(x))$  unifies with the subterm  $f(y)$  of the renamed lhs to produce the mgu  $\{f(x)/y\}$ :



- ▶ This CP is not joinable, so  $R$  is not locally confluent.
- ▶ Adding the rule  $f(g(x)) \Rightarrow g(f(x))$  to  $R$  makes the pair joinable.
- ▶ The enlarged  $R$  is terminating (how?), but
- ▶ (After renaming) new CP:  $\langle g(g(z)), f(g(f(z))) \rangle$  arises (how?);
- ▶ LC test: it is joinable,  $f(g(f(z))) \rightarrow g(f(f(z))) \rightarrow g(g(z))$ .

# Knuth-Bendix (KB) Completion Algorithm

Start with a set  $R$  of terminating rewrite rules

While there are non-conflatable critical pairs in  $R$ :

1. Take a critical pair  $\langle s_1, s_2 \rangle$  in  $R$
2. Normalise  $s_1$  to  $s'_1$  and  $s_2$  to  $s'_2$  (and we know  $s'_1 \neq s'_2$ )
3. if  $R \cup \{s'_1 \Rightarrow s'_2\}$  is terminating then

$$R := R \cup \{s'_1 \Rightarrow s'_2\}$$

else if  $R \cup \{s'_2 \Rightarrow s'_1\}$  is terminating then

$$R := R \cup \{s'_2 \Rightarrow s'_1\}$$

else Fail

- ▶ If KB succeeds then we have a locally confluent and terminating (and hence confluent) rewrite set (KB may run forever!)
- ▶ Depends on the termination check: define a measure and use that to test for termination.

# Summary

- ▶ Rewriting (Bundy Ch. 9)
  - ▶ Local confluence
  - ▶ Local confluence + Termination = Confluence
  - ▶ Canonical Normal Forms
  - ▶ Critical Pairs and Knuth-Bendix Completion