Equality Part 3

Christoph Wirnsperger

4th December 2016

Aim for today?

Equation rules:

$$a = bc$$
$$a = c$$
$$c = bbc$$
$$c = e$$

Does this equation holds or not? $bbe = bbbe$

$$\begin{array}{ccc} bbe & = & bbbe \\ \downarrow & & \downarrow \\ ? & & ? \end{array}$$

Overview

1. Introduction
2. Crititcal Pairs
3. Completion

Introduction

- Consider the following axioms for groups:

$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	associative
$1 \cdot x = x$	identity
$i(x) \cdot x = 1$	inverse
$x \cdot 1 = x$	Does this also holds?

Introduction

- Example interpretation for given terms:

$$\begin{array}{ccc} \cdot & \textit{as} & + \\ 1 & \textit{as} & 0 \\ i & \textit{as} & - \end{array}$$

- Use these interpretations for given laws (integers):

$$(x + y) + z = x + (y + z)$$
$$0 + x = x$$
$$-x + x = 0$$

- Second example interpretation for given terms:

$$\begin{array}{lll} \cdot & \textit{as} & \oplus \\ 1 & \textit{as} & \textit{false} \\ i & \textit{as} & \textit{identity} \end{array}$$

- Use these interpretations for given laws:

$$(x \oplus y) \oplus z = x \oplus (y \oplus z)$$
$$\textit{false} \oplus x = x$$
$$x \oplus x = \textit{false}$$

Remark

Many other interpretations are possible too.

1. Introduction
2. Crititcal Pairs
3. Completion

Critical Pairs

Example Joinability (from the book)

- Case 1:

$$(1 \cdot x) \cdot y \xrightarrow{R}$$
$$1 \cdot (x \cdot y) \xrightarrow{R}$$
$$x \cdot y$$

- Case 2:

$$(1 \cdot x) \cdot y \xrightarrow{R}$$
$$x \cdot y$$

Remark: This peak is joinable because with different reduction steps we get the same result.

Lemma local confluence

A Term Rewriting System is called locally confluent iff all CPs are joinable and the TRS terminates.

Newman's Lemma

If a locally confluent TRS has no infinite reduction sequences (in which case it is said to be terminating), then it is confluent.

Critical Pairs

Question

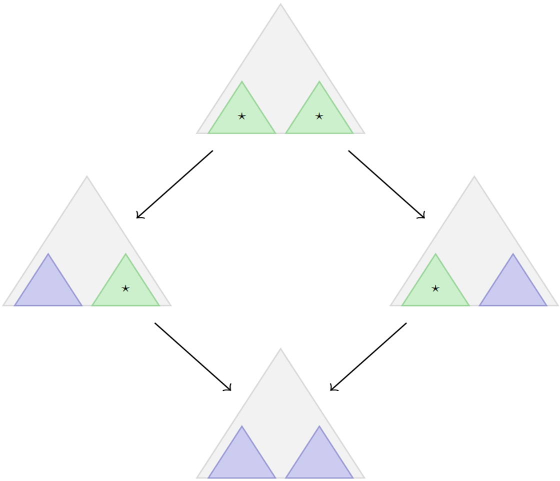
How can we prove local confluence?

Proof of Critical Pair Lemma - three local peaks

- case 1: parallel redexes
- case 2: variable overlap
- case 3: overlapping redex-pattern

Critical Pairs

Case 1: parallel redexes



Critical Pairs

Example: Case 1 parallel redexes

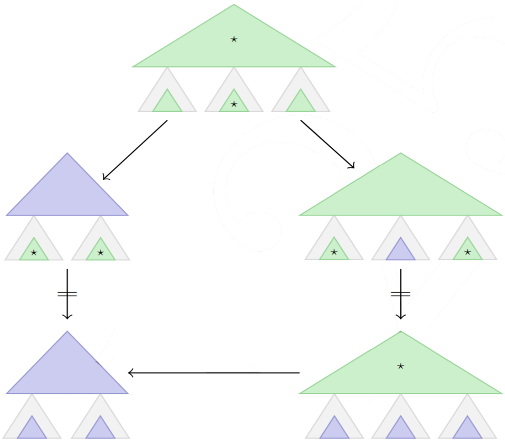
$$f(x,x,x) \to g(x,x)$$
$$a \to b$$

$$\begin{tikzcd} & g(a,a) \arrow[dl] \arrow[dr] & \\ g(a,b) \arrow[dr] & & g(b,a) \arrow[dl] \\ & g(b,b) & \end{tikzcd}$$

- non-critical

Critical Pairs

Case 2: variable overlapping



Critical Pairs

Example: Case 2 variable overlapping

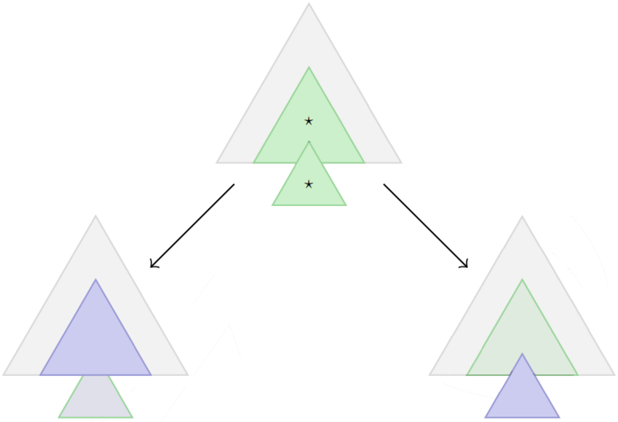
$$f(x,x,x) \rightarrow g(x,x)$$
$$a \rightarrow b$$

$$\begin{tikzcd}
 & & & f(a,a,a) \arrow[dl] \arrow[dr] & & & \\
 & & g(a,a) \arrow[dll] \arrow[d] & & f(a,b,a) \arrow[d] \arrow[drr] & & \\
g(a,b) \arrow[dr] & & g(b,a) \arrow[dl] & & f(b,b,a) \arrow[dr] & & f(a,b,b) \arrow[dl] \\
 & g(b,b) & & & & f(b,b,b) \arrow[llll] &
\end{tikzcd}$$

- non-critical

Critical Pairs

Case 3: overlapping redex-pattern



Critical Pairs

Example: Case 3 overlapping redex-pattern

$$f(g(x), y) \to y$$
$$g(x) \to f(x, x)$$

The term $f(g(x), y)$ could be reduced by the first rewrite rule or by the second:

1. $f(g(x), y) \xrightarrow{R} y$
2. $f(g(x), y) \xrightarrow{R} f(f(x, x), y)$

Critical Pairs

Example: Case 3 overlapping redex-pattern

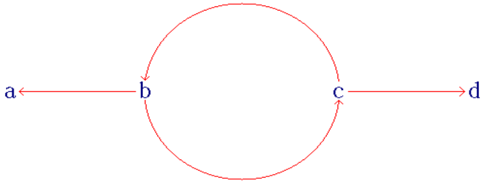
$$f(g(x), y) \to y$$
$$g(x) \to f(x, x)$$

The term $f(g(x), y)$ could be reduced by the first rewrite rule or by the second:

1. $f(g(x), y) \xrightarrow{R} y$
2. $f(g(x), y) \xrightarrow{R} f(f(x, x), y)$

- Overlapping at g
- CPs are $y, \quad f(f(x, x), y)$
- Is this joinable?

Critical Pairs



Remark Graph

- Cyclic, locally-confluent, but not globally confluent rewrite system
- CPs are (b, d) and (a, c) and they are joinable.

Two Critical Pairs

$$\begin{tikzcd} & c \arrow[dl] \arrow[dr] & \\ b \arrow[rr, two heads] & & d \end{tikzcd} \qquad \begin{tikzcd} & b \arrow[dl] \arrow[dr] & \\ a & & c \arrow[ll, two heads] \end{tikzcd}$$

Completion Algorithm

Definition

The Knuth–Bendix completion algorithm is an algorithm for transforming a set of equations into a confluent and terminating term rewriting system.

- Aim is to reach a canonical rewrite set for many algebraic theories, like groups
- Input is a set of equations and we want a TRS.
- When the algorithm succeeds, it effectively solves the problems
- This algorithm may
  1. terminates with success and yields a finitely terminating, confluent set of rules,
  2. terminates with failure, or
  3. loops without terminating (divergence)

Completion Algorithm

Definition

In TRS an object is in normal form if it cannot be rewritten any further.

Case 1: terminate with success

- rewrite rules

$$f(g(a)) \to h(b) \qquad f(g(a)) \to c$$

- two Critical Pairs

$$h(b) \approx c \qquad c \approx h(b)$$

- eventually orient and terminate

Orientation

Remark

The equation should always respect the orientation. It could happen that neither direction respects the ordering.

Example - ordering by size

- $|t|$ = number of symbols in t

$$c \approx h(b)$$
$$|c| > |h(b)|$$
$$1 > 2 \qquad \text{mis-oriented}$$

- another TRS available

$$h(b) \approx c$$
$$|h(b)| > |c|$$
$$2 > 1 \qquad \text{oriented}$$

Completion Algorithm

Case 2: terminate with failure

- rewrite rules

$$f(g(a)) \to h(b) \qquad f(g(a)) \to h(y)$$

- two Critical Pairs

$$h(b) \approx h(y) \qquad h(y) \approx h(b)$$

- no orientation possible $\to$ failure

Completion Algorithm

Case 3: infinite loop

- rewrite rules:

$$f(g(x)) \to g(h(x)) \qquad g(a) \to b$$

$$g(h(a)) \to f(b) \qquad g(h(h(a))) \to f(f(b)) \qquad \dots$$

- LPO with precedence $a > f > g > h > b$
- Critical Pairs

$$f(b) \approx g(h(a))$$
$$f(f(b)) \approx g(h(h(a)))$$
$$f(f(f(b))) \approx g(h(h(h(a)))) \qquad \dots$$

- infinite loop, not terminating (divergence)

Mis-orientation

Remark

If Knuth–Bendix does not succeed, it will either run forever, or fail when it encounters an unorientable equation. The enhanced completion without failure will not fail on unorientable equations and provides a semi-decision procedure for the word problem.

Ocaml Code Example

```
let normalize_and_orient ord eqs (Atom(R("=",[s;t]))) =
  let s' = rewrite eqs s and t' = rewrite eqs t in
  if ord s' t' then (s',t') else if ord t' s' then (t',s')
  else failwith "Can't orient equation"
```

Completion Algorithm Rules

Six rules:

$$Delete \quad = \frac{< E \cup \{s = s\}, R >}{< E, R >}$$

$$Compose \quad = \frac{< E, R \cup \{s \rightarrow t\} >}{< E, R \cup \{s \rightarrow u\} >} \quad t \xrightarrow{R} u$$

$$Collapse \quad = \frac{< E, R \cup \{s \rightarrow t\} >}{< E \cup \{u = t\}, R >} \quad s \xrightarrow{R} u$$

$$Simplify \quad = \frac{< E \cup \{s = t\}, R >}{< E \cup \{s = u\}, R >} \quad t \xrightarrow{R} u$$

$$Deduce \quad = \frac{< E, R >}{< E \cup \{s = t\}, R >} \quad \textit{if } (s, t) \text{ is a CP of R}$$

$$Orient \quad = \frac{< E \cup \{s = t\}, R >}{< E, R \cup \{s \rightarrow u\} >} \quad s > t$$

Knuth-Bendix Completion Procedure

Input: an E and a reduction order

Output: a complete TRS R that represents E

$R := \emptyset; \quad C := E;$

while C $\neq \emptyset$ do

  choose a pair $s \approx t \in C$;

  $C := C \setminus \{s = t\}$;

  rewrite s and t to normal forms s' and t' with respect to R;

  if $s' \neq t'$ then

    if $s' > t'$ then

      $S := \{s' \rightarrow t'\}$

    else if $t' > s'$ then

      $S := \{t' \rightarrow s'\}$

    else

      failure

    $C := C \cup CP(R, S) \cup CP(S, R) \cup CP(S)$

    $R := R \cup S$

Beginning Example

Equation rules:

$$a = bc, \quad a = c,$$
$$c = bbc, \quad c = e,$$

Does this equation holds? $bbe = bbbe$

Beginning Example

Equation rules:

$$a = bc, \quad a = c,$$
$$c = bbc, \quad c = e,$$

Does this equation holds? $bbe = bbbe$

$$\begin{tikzcd}
 & bbc \arrow[dl] \arrow[dr] & \\
c \arrow[d] & & ba \arrow[d] \\
e & & be \arrow[ll]
\end{tikzcd}$$

Beginning Example

Equation rules:

$$a = bc, \quad a = c,$$
$$c = bbc, \quad c = e,$$

Does this equation holds? $bbe = bbbe$

$$\begin{tikzcd}[column sep=small] & bbc \arrow[dl] \arrow[dr] & \\ c \arrow[d] & & ba \arrow[d, two heads] \\ e & & be \arrow[ll] \end{tikzcd} \qquad \begin{tikzcd} bbe \arrow[d, two heads] \arrow[r, phantom, "="] & bbbe \arrow[d, two heads] \\ e \arrow[r, phantom, "="] & e \end{tikzcd}$$

- John Harrison. Handbook of Practical Logic and Automated Reasoning. Cambridge University Press, 2009
- Aart Middeldorp Vincent van Oostrom. Term Rewriting. http://cl-informatik.uibk.ac.at/teaching/ss16/trs/material.php
  Computational Logic Group, University of Innsbruck, 2016

Demo

MKBtt is a completion tool for rewrite systems, which uses a termination tool to orient equations together with a special data structure to sequentialize the parallel execution of the processes that derive from choices in the orientation of equations.
(from the website)
`http://cl-informatik.uibk.ac.at/software/mkbtt/`