Rewrite Techniques in Theorem Proving

Part II: Chaining

Leo Bachmair

Harald Ganzinger

Tutorial—CADE'94

Contents

Efficient handling of transitivity

$$x \not< y \;,\; y \not< z \;,\; x < z \qquad (\mathcal{T})$$

through rewrite techniques.

Applications

- equality, partial congruences
- orderings
- consequence relations, modal logics, higher-order logic, set theory
- non-deterministic computations

Chaining

Hyperresolution with transitivity

$$\frac{C\ ,\, u < v \quad D\ ,\, s < t \quad x \not< y\ ,\, y \not< z\ ,\, x < z}{(C\ ,\, D\ ,\, u < t)\sigma}$$

where $\sigma = \mathsf{mgu}(v, s)$.

Chaining [Slagle 72]:

$$\frac{C, u < v \quad D, s < t}{(C, D, u < t)\sigma} \qquad \sigma = \mathsf{mgu}(v, s)$$

Theorem 1 *Ordered resolution (not involving $\mathcal{T}$) + chaining is complete.*

Proof. Ordered resolution with selection is complete: select both negative literals of $\mathcal{T}$. $\square$

Problems

- the transitive closure of $<$ is naively and completely enumerated: $\Omega(n^3)$ in the finite case

- infinite transitive closures

- $s < t$ chains in both directions from s to t and from t to s

- chaining through variables: $x < s(x)$ and $p(y) < y$ generate $p^m(x) < s^n(x)$, for all n, m

- handling of additional properties such as monotonicity

  $$x < y \rightarrow f(\ldots, x, \ldots) < f(\ldots, y \ldots)$$

  or anti-symmetry

  $$x \leq y \; , \; y \leq x \rightarrow x \approx y$$

Infinitary Depth-First Search

Let $G = (V, E)$ be a possibly infinite graph.

DEPTH-FIRST SEARCH OF G

- define some well-founded, total ordering $\succ$ on V
- search
  $G_1 = (V, E \cap \succeq)$
  and $G_2 = (V, E^{-1} \cap \succeq)$
- combine the results

HERE:

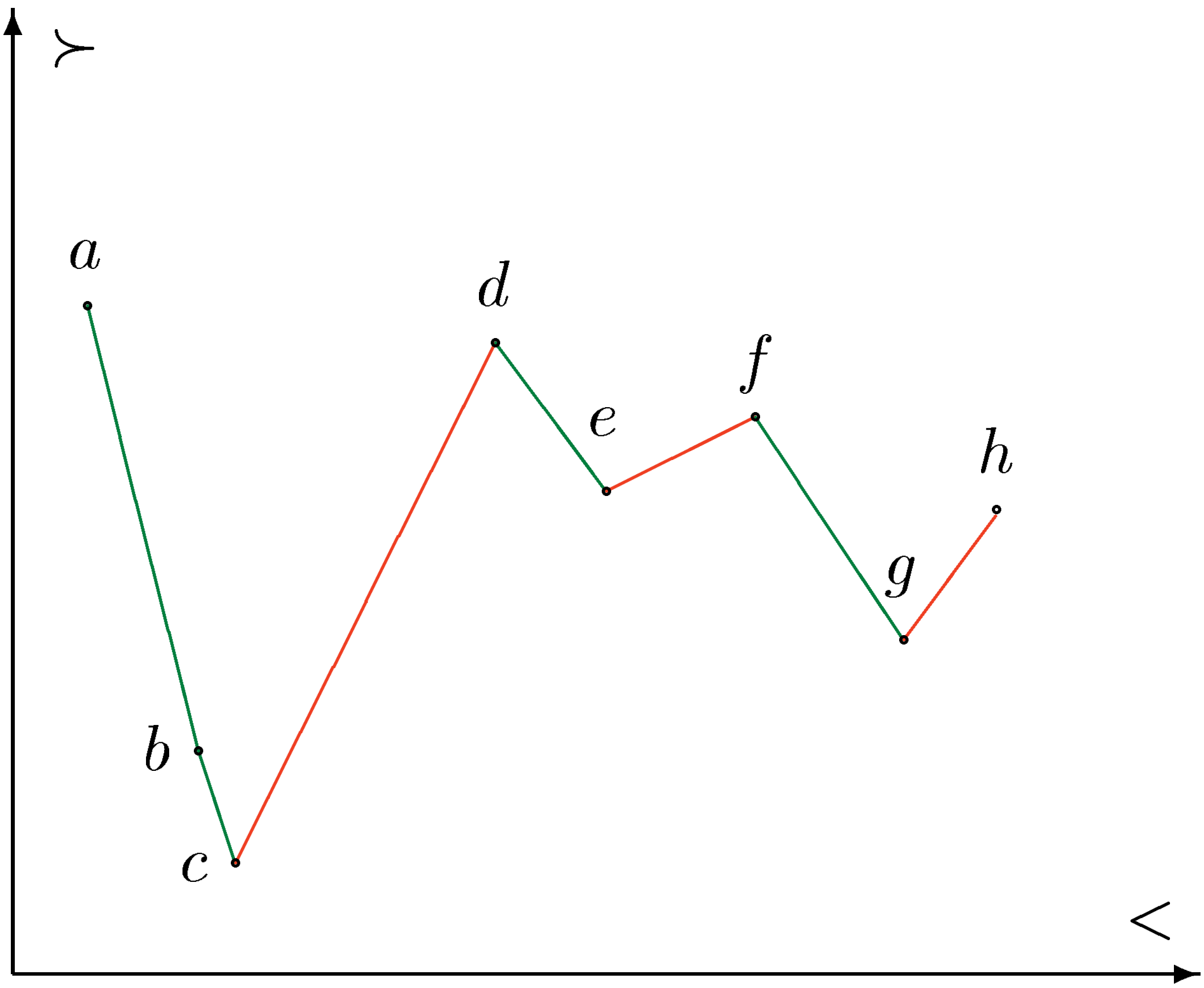
- V = set of ground terms
- E = set of inequations $s < t$
- $\succ$ = some complete reduction ordering

Syntactic Orderings $\succ$

Given $a < b < c < d < e < f < g < h$

define *syntactic* ordering

$a \succ d \succ f \succ e \succ h \succ g \succ b \succ c$

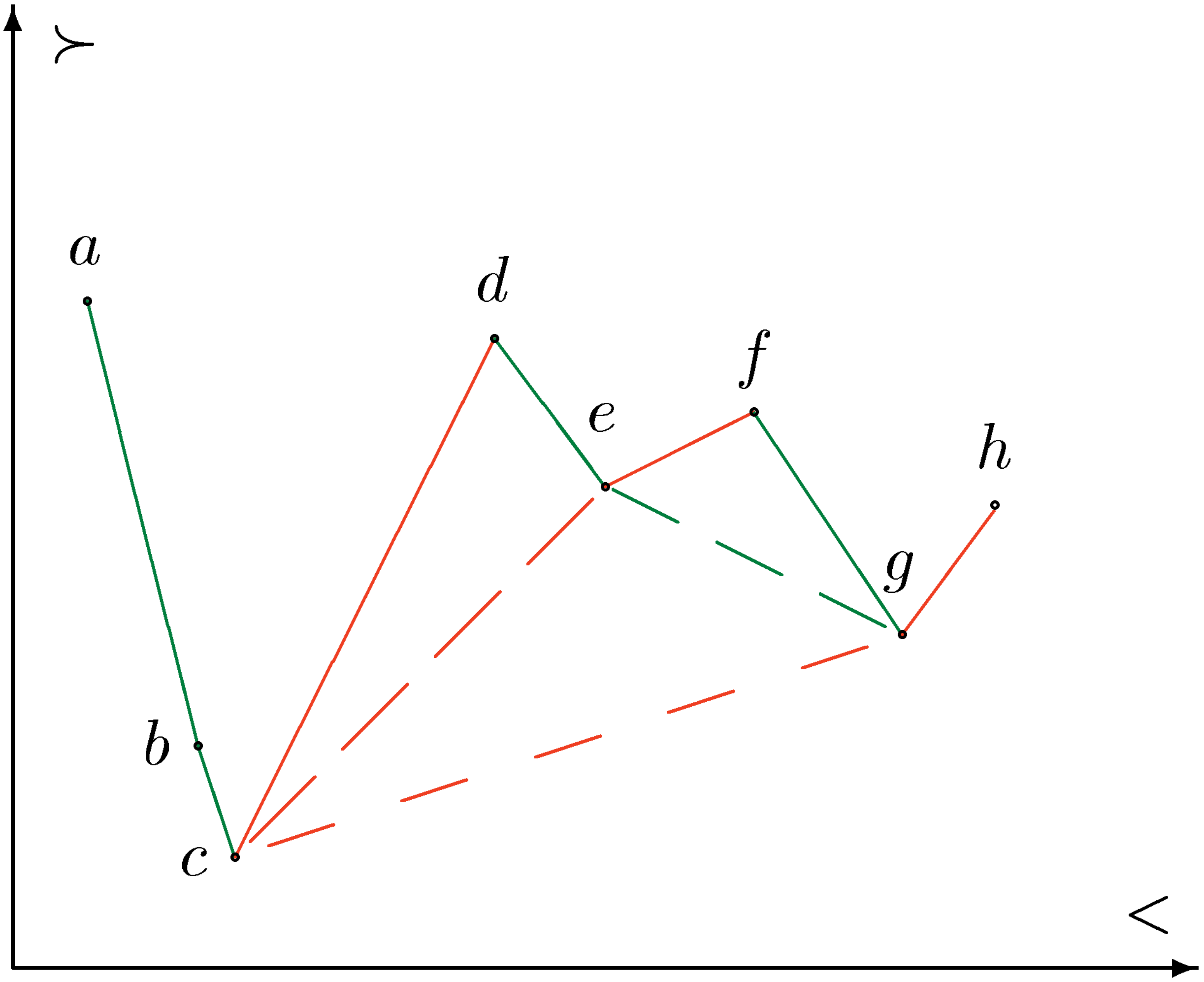


two *rewrite relations*

$$ {}^{\cdot}\!\!<\!\!{}_{\cdot} \; = \; < \cap \succeq \qquad \text{and} \qquad {}_{\cdot}\!\!<\!\!{}^{\cdot} \; = \; < \cap \preceq $$

Making Peaks Commute

. . . by adding chains through peaks.



Rᴇsᴜʟᴛ. $x <^{+} y$ iff

$$x = x_0 \;{}^{\cdot}\!<_{\cdot} \ldots {}^{\cdot}\!<_{\cdot}\; x_n = y_k \;{}_{\cdot}\!<^{\cdot} \ldots {}_{\cdot}\!<^{\cdot}\; y_0 = y,$$

where $n + k \geq 1$.

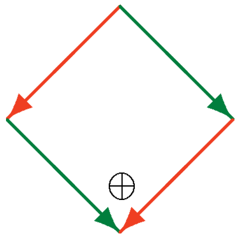
This is a *rewrite proof* of $x <^{+} y$.

Commutation

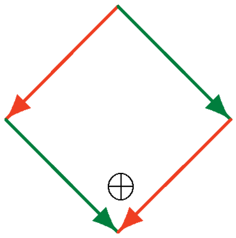
Let $R \oplus S \;=\; (R^* \circ S^+) \;\cup\; (R^+ \circ S^*)$.

Definition $\;\cdot\!\!<\!\!.\;$ *commutes over* $\;.\!\!<\!\!\cdot\;$ iff

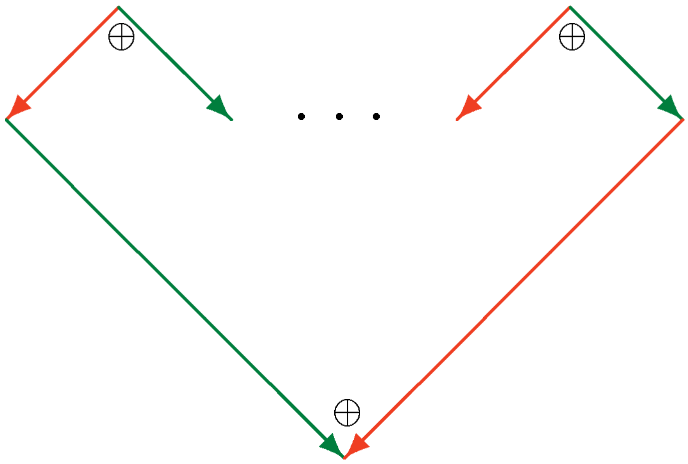
$.\!\!<\!\!\cdot \circ \cdot\!\!<\!\!. \quad \subseteq \quad \cdot\!\!<\!\!. \oplus .\!\!<\!\!\cdot\,.$



Proposition



iff



Hence: Commutation is equivalent with the existence of rewrite proofs for $<^+$.

If $.\!\!<\!\!\cdot \;=\; \cdot\!\!<\!\!.^{\,-1}$ we speak of the *local confluence* of $.\!\!<\!\!\cdot$ (and $\cdot\!\!<\!\!.$).

Ordered Chaining

One achieves commutation by computing all chaining inferences through peaks.

From now on let $\succ$ be a complete reduction ordering on terms, and let $\succ$ also denote its lexicographic extension to inequations.

Ordered Chaining (Unit Case) $\mathsf{B}_<$

$$\frac{u < s \qquad t < v}{u\sigma < v\sigma}$$

where $\sigma = \mathsf{mgu}(s, t)$, $u\sigma \not\succeq s\sigma$, and $v\sigma \not\succeq s\sigma$.

Redundancy of Clauses

A ground clause C is called $\mathcal{T}$-redundant in a set N of clauses iff

1. $C_1, \ldots, C_n, \mathcal{T} \models C$
2. $C \succ C_i$,

where the C_i are ground instances of N.

A clause is $\mathcal{T}$-redundant in N if all its ground instances are.

Redundancy of Inferences

An inference from ground clauses

$$\frac{D_1, \ldots, D_n}{C}$$

is called $\mathcal{T}$-redundant in N if one of the D_i is T-redundant in N, or else if

1. $C_1, \ldots, C_n, \mathcal{T} \models C$
2. $D \succ C_i$, with $D = \mathsf{max}_\succ D_i$,

where the C_i are ground instances of N.

An inference is $\mathcal{T}$-redundant in N if all its ground instances are.

Saturation up to Redundancy

Let $\mathcal{I}$ be an inference system.

N is saturated (with respect to $\mathcal{I}$) up to redundancy if all inferences from N in $\mathcal{I}$ are $\mathcal{T}$-redundant in N.

NOTE:

- If an inference is monotone, i.e., $D \succ C$, it becomes redundant when we add its conclusion to N.
- All inference systems that we consider are monotone.
- For more on how to saturate a set of clauses see part III.
- The given notion of redundancy is not compatible with arbitrary inference systems. In particular, inference systems for clauses with constraints come with a modified notion of redundancy.

Model Construction

Given a set of ground inequations and an ordering $\succ$, define, for each inequation $C = s < t$, Herbrand interpretations I_C and I^C and an auxiliary set E_C as follows:

$$
\begin{aligned}
I_C &= \Big(\bigcup_{C \succ D} E_D\Big)^{\vee} \\
E_C &= \begin{cases} \{s < t\}, & \text{if } s < t \text{ not in } I_C \\ \emptyset, & \text{otherwise} \end{cases} \\
I^C &= (I_C \cup E_C)^{\vee}
\end{aligned}
$$

where $R^{\vee}$ is the rewrite closure of R.

Finally $I = (\bigcup_C E_C)^{\vee}$.

C is called productive if $E_C \neq \emptyset$.

$R_C = \bigcup_{C \succ D} E_D$

Main Lemma

Lemma 1 *Let N be a set of ground inequations that is saturated under $\mathsf{B}_<$ up to $\mathcal{T}$-redundancy. Then for every inequation C in N we have:*

1. *If C is $\mathcal{T}$-redundant in N then C is in I_C.*
2. *If C is a productive inequation it is not $\mathcal{T}$-redundant in N.*
3. *I_C and I^C are transitive, i.e. satisfy $\mathcal{T}$.*
4. *C is in I^C.*

Proof of the Lemma

By induction over $\succ$.

(1) If C is redundant, then $C_1, \ldots, C_n, \mathcal{T} \models C$, for inequations C_i in N such that $C \succ C_i$. By induction hypothesis for (4), C_i is in I_C and, by induction hypothesis for (3), $\mathcal{T}$ is true in I_C. Hence C is in I_C.

(2) If C is productive then C is not in I_C. By (1) it is therefore not redundant.

(3) By induction hypothesis, all peaks involving inequations in R_C commute. The cases $E_C = \emptyset$ or $s = t$ are trivial. Otherwise and if $C = s \,.{<}^{\cdot}\, t$ R_C cannot contain any inequation $t \,^{\cdot}{<}.\, u$ as inequations of this form are bigger than C. In the remaining case $E_C = \{s \,^{\cdot}{<}.\, t\}$. Suppose there is an inequation $C' = u \,.{<}^{\cdot}\, s$ in R_C. As N is saturated, the inference from C' and C by ordered chaining produces $u < t$ and is redundant in N. Being productive, by (2) neither C nor C' can be redundant, hence $u < t$ must follow from $\mathcal{T}$ and from clauses C_i in N which are smaller than C. Applying the induction hypothesis for (4) we conclude that $u < t$ is in I_C.

(4) $C \in I_C \cup E_C$ by construction.

Completeness

Theorem 2 *Let N be a set of inequations and saturated under $\mathsf{B}_{<}$ up to $\mathcal{T}$-redundancy. Let s and t be ground terms. Then $s < t$ follows from $N \cup \{\mathcal{T}\}$ iff $s < t$ has a rewrite proof in ground instances of N.*

Proof. We apply the lemma to the ground instances of N. By standard lifting arguments it follows that if N is saturated so is the set of its ground instances. Let I be the interpretation constructed from the latter. Then:

$s < t$ follows from $N \cup \{\mathcal{T}\}$

> $\Rightarrow$ $s < t$ true in I *[lemma: (3), (4)]*
>
> $\Rightarrow$ $s < t$ has rewrite proof consisting of (productive) instances of N.

$\square$

Applications

- testing $N \cup \{\mathcal{T}\} \models s < t$ by rewriting
- solving inequations $s < t$ over $N \cup \{\mathcal{T}\}$ by narrowing-like procedures: negative chaining
- extension of the calculus to Horn clauses

Calculus $\mathsf{B}_<$ for Horn Clauses

Ordered Chaining

$$\frac{C \to \boxed{u \mathrel{.{<}^{\cdot}} s} \qquad D \to \boxed{t \mathrel{{}^{\cdot}{<}.} v}}{\{C\ ,\ D \to u < v\}\sigma}$$

where

(i) $\sigma = \mathsf{mgu}(s, t)$,

(ii) $C\sigma \not\succeq (u < s)\sigma$,

(iii) $D\sigma \not\succeq (t < v)\sigma$,

(iv) $u\sigma \not\preceq s\sigma$, and

(v) $v\sigma \not\preceq s\sigma$.

In what follows we specify maximality constraints for literals in clauses by boxes, and those for terms in literals by using $.{<}^{\cdot}$ and ${}^{\cdot}{<}.$ as indicated above.

Horn Clauses II

Negative chaining

$$\frac{\Gamma\ ,\ \boxed{u \mathrel{\dot{<}\!.} s} \to \Delta \qquad \Lambda \to \boxed{t \mathrel{\dot{<}\!.} v}}{\{\Gamma\ ,\ \Lambda\ ,\ v < s \to \Delta\}\sigma}$$

where $\sigma = \mathsf{mgu}(u, t)$; and

$$\frac{\Gamma\ ,\ \boxed{u \mathrel{.\!\dot{<}} s} \to \Delta \qquad \Lambda \to \boxed{t \mathrel{.\!\dot{<}} v}}{\{\Gamma\ ,\ \Lambda\ ,\ u < t \to \Delta\}\sigma}$$

where $\sigma = \mathsf{mgu}(s, v)$.

Ordered resolution

$$\frac{\Gamma\ ,\ \boxed{A} \qquad \boxed{\neg B}\ ,\ \Lambda}{\Gamma\sigma\ ,\ \Lambda\sigma}$$

where $\sigma = \mathsf{mgu}(A, B)$.

Completeness of $\mathsf{B}_<$

Theorem 3 *Let N be a set of Horn clauses and saturated under $\mathsf{B}_<$ up to $\mathcal{T}$-redundancy. Then N is satisfiable iff the empty clause is not in N.*

Proof. We combine the criteria for productivity for general clauses (cf. part I) with those for inequations as follows:

$$E_C = \begin{cases} \{s < t\}, & \text{if (i) -- (iii)} \\ \emptyset, & \text{otherwise} \end{cases}$$

where

(i) C is false in I_C,

(ii) C is of the form $\Gamma \rightarrow s < t$,

(iii) $(s < t) \succ \Gamma$,

and, as before,

$$I_C = \Big(\bigcup_{C \succ D} E_D\Big)^{\vee}.$$

□

Variable Chaining

Proposition 1 *Chaining through shielded variables is not needed.*

Proposition 2 *Chaining into non-linear, unshielded variables is required.*

Proof. Consider this unsatisfiable set

$$
\begin{array}{rcl}
 & \rightarrow & a \mathrel{.{<}^{\cdot}} a' \\
 & \rightarrow & a' \mathrel{.{<}^{\cdot}} c \\
 & \rightarrow & b \mathrel{.{<}^{\cdot}} b' \\
 & \rightarrow & b' \mathrel{.{<}^{\cdot}} c \\
a < x, b < x & \rightarrow &
\end{array}
$$

where $c \succ b' \succ b \succ a' \succ a$.

Without chaining into variables, $\mathsf{B}_<$ only produces $a \mathrel{.{<}^{\cdot}} b' \rightarrow$ and $a \mathrel{.{<}^{\cdot}} b \rightarrow$. $\square$

Calculus $\mathsf{C}_<$ for Clauses

$\mathsf{C}_<$ is $\mathsf{B}_<$ together with:

Ordered factoring

$$\frac{\Gamma\ ,\ \boxed{A}\ ,\ B}{\Gamma\sigma\ ,\ A\sigma}$$

where $\sigma = \mathsf{mgu}(A, B)$

Transitivity resolution

$$\frac{C\ ,\ \boxed{u \mathrel{\dot{<}\!.} v}\ ,\ u' \mathrel{\dot{<}\!.} v' \quad D\ ,\ \boxed{s \mathrel{\dot{<}\!.} t}}{\{D\ ,\ t \not< v'\ ,\ s < v'\}\sigma}$$

where $\sigma = \mathsf{mgu}(u, u', s)$.

TR specializes further in the presence of additional logical properties.

Theorem 4 *Let N be a set of clauses and saturated under $\mathsf{C}_<$ up to $\mathcal{T}$-redundancy. Then N is satisfiable iff the empty clause is not in N.*

Monotonicity

Subterm Chaining

$$\frac{C\ ,\ \boxed{u \mathrel{.\!\!<\!\!\cdot}\ s} \qquad D\ ,\ \boxed{w[t] \mathrel{\cdot\!\!<\!\!.}\ v}}{\{C\ ,\ D\ ,\ w[u] < v\}\sigma}$$

where

- $\sigma = \mathsf{mgu}(s, t)$,
- t a subterm of w, including subterms at or below variables,
- the functions above t in w are monotonic wrt. $<$.

Similarly for anti-monotonicity.

Composition of Relations

Composition laws of the form

$$x \oplus y \,,\, y \ominus z \to x \otimes z$$

can be translated into

Ordered composition

$$\frac{u \oplus s \qquad t \ominus v}{u\sigma \otimes v\sigma}$$

where $\sigma = \mathsf{mgu}(s,t)$, $u\sigma \not\succeq s\sigma$, and $v\sigma \not\succeq s\sigma$.

Examples

$$\begin{array}{lcl} x < y \,,\, y < z & \to & x < z \\ x \leq y \,,\, y < z & \to & x < z \\ x \approx y \,,\, y \oplus z & \to & x \oplus z \\ x \in y \,,\, y \subseteq z & \to & x \in z \end{array}$$

Summary

- commutation yields rewrite proofs for transitive relations
- similarly for almost arbitrary composition laws
- chaining required for peaks only
  $\Rightarrow$ ordered inferences
- monotonicity properties require chaining through subterms
- straightforward extension to Horn clauses; disjunctions somewhat more involved
- abstract notion of redundancy
- inference systems compatible with redundancy
- problem of chaining at or below variables: ok for symmetric relations (cf. part III) and for dense total orderings (cf. CADE conference)

Literature I

L. Bachmair, H. Ganzinger, 1993. Rewrite Techniques for Transitive Relations. TR MPI-I-93-249, MPI Informatik, Saarbrücken. Short version in Proc. LICS'94.

- describes system $C_<$ with completeness proof
- discusses influence of symmetry and monotonicity

L. Bachmair, H. Ganzinger, 1993. Ordered Chaining for Total Orderings. TR MPI-I-93-250, MPI Informatik, Saarbrücken. Short version in Proc. CADE'94.

- describes maximal chaining for total orderings, together with completeness proof
- chaining for equality and inequality
- variable elimination as simplification
- reports experimental results (Saturate system)

Literature II

J. Levy, J. Agustí, 1993. Bi-rewriting, a Term Rewriting Technique for Monotonic Order Relations. In C. Kirchner, editor, *Rewriting Techniques and Applications*, Lecture Notes in Computer Science, vol. 690, pp. 17–31.

- bi-rewriting = 2 commuting rewrite systems
- considers unit inequations and monotonic transitive relations
- application to lattice theory with discussion of variable chaining

P. Nivela, R. Nieuwenhuis, H. Ganzinger, 1994. Overview of the Saturate System. Hypertext document on the world-wide web at "ftp://ftp.mpi-sb.mpg.de/pub/SATURATE/Saturate.html".

- describes the usage of the Saturate system, featuring
  - orderd chaining (with selection) for arbitrary composition laws
  - constraint solving for lpo constraints
  - simplification techniques, including tautology deletion by ground chaining, subsumption, reduction, case analysis, clausal rewriting
- applications of Saturate to problems in equational logic, modal logic, verification of quicksort