Conditional Rewriting

Bernhard Gramlich

ISR 2009, Brasilia, Brazil, June 22-26, 2009
Outline

- Introduction
- Basics in Conditional Rewriting
- Termination
- Confluence
- Transforming CTRSs Into TRSs
- Systems with Extra Variables (3-Ctrss)
- Transforming CTRSs Into TRSs
- Further Topics (expressive power, modularity, ... )
Lecture 1
Introduction

Examples

- $x \leq z = true \iff x \leq y = true, y \leq z = true$
- $a = b \iff a \neq c, a = c \iff a \neq b \ (a = b \lor a = c)$

Hence: minimal models non-unique. Consequence (here): no negation in conditions.

- $f(x, y) \rightarrow g(x) \iff P(x, y)$ (built-in predicates, not here)
- $a \rightarrow b \iff c = d$
- $a \rightarrow b \iff c \leftrightarrow* d$
- $a \rightarrow b \iff c \downarrow d$
- $a \rightarrow b \iff c \rightarrow* d$
- $a \rightarrow b \iff a \rightarrow* c$
- $x \leq z \rightarrow true \iff x \leq y \rightarrow* true, y \leq z \rightarrow* true$

- $s(x) + y \rightarrow s(z) \iff x + y \leftrightarrow* z$
- $s(x) + y \rightarrow s(z) \iff x + y \rightarrow* z$
Basics in Conditional Rewriting

Format of CESs / CTRSs

- **conditional (term) equational system (CES)** $R$: axioms of shape $l = r \iff s_1 = t_1, \ldots, s_n = t_n$; leads to

- **conditional equational logic (CEL)**: $\equiv_R$

- **conditional (term) rewrite system (CTRS)** $R$: rules of shape $l \rightarrow r \iff s_1 \Box t_1, \ldots, s_n \Box t_n$; interpretation of equality $\Box$ in conditions:
  - $\Box = \leftrightarrow^*$: semi-equational system
  - $\Box = \downarrow$: join system
  - $\Box = \rightarrow^*$: oriented system
  - $\Box = \rightarrow^*$, all $t_1$ $R_u$-irreducible ground terms: normal (join) system (where $R_u = \{ l \rightarrow r \mid l \rightarrow r \implies c \in R \}$)
Basics in Conditional Rewriting

Join CTRSs as non-left-linear normal (join) CTRSs

- replace every $l \rightarrow r \leftarrow s_1 \downarrow t_1, \ldots, s_n \downarrow t_n$ by
  $l \rightarrow r \leftarrow eq(s_1, t_2) \rightarrow^* true, \ldots, eq(s_n, t_n) \rightarrow^* true$
- add $eq(x, x) \rightarrow true$
- $true$ fresh boolean constant, $eq$ fresh binary predicate
- $s \rightarrow_R t$ iff $eq(s, t) \rightarrow_R^* true$ (in many-sorted setting)

Syntactical properties of CTRSs $R$ (via $R_u$)

- $R_u = \{ l \rightarrow r \mid l \rightarrow r \Rightarrow c \in R \}$
- $R$ left-linear, right-linear, linear, (non-)collapsing, (non-)duplicating, non-overlapping, orthogonal, overlaying, \ldots, if $u$ is left-linear, \ldots, respectively
- advantage of these syntactic definitions: easily decidable!
- disadvantage (partially): semantically misleading (Ex.: $f(x, y) \rightarrow a \leftarrow x \leftrightarrow^* y$)
Basics in Conditional Rewriting

Classification of CTRSs according to extra variables

\( \text{VAR}(s) \) set of all variables occurring in \( s \), \( l \rightarrow r \leftarrow c \in R \)

- **type 1**: \( \text{VAR}(l) \supseteq \text{VAR}(r) \cup \text{VAR}(c) \) (no extra variables)
- **type 2**: \( \text{VAR}(l) \supseteq \text{VAR}(r) \) (extra variables at most in cond.)
- **type 3**: \( \text{VAR}(l) \cup \text{VAR}(c) \supseteq \text{VAR}(r) \) (extra variables in right-hand sides must appear in conditions)
- **type 4**: no restrictions.
- **R \( n \)-CTRS** if all its rules are of type \( n \)

- type 2 involves *existential search* in conditions
- type 3 involves existential search in conditions and rhs’s and may cause *infinite branching*
- type 4 very hard to understand / handle computationally (almost no results known for systems of this type)
Induced equality for CES \( E \)

Inference system for conditional equational logic (CEL): Given CES \( E \), CEL defines conditional equational derivability \( =_E \):

- **Reflexivity:** \( t = t \)
- **Symmetry:** \( t = s \) \( \Rightarrow s = t \)
- **Transitivity:** \( s = t, t = u \) \( \Rightarrow s = u \)
- **Congruence:** \( f(s, \ldots, s_n) = f(t_1, \ldots, t_n) \) \( \Rightarrow s_1 \sigma = t_1 \sigma, \ldots, s_n \sigma = t_n \sigma \)
- **Application:** \( l \sigma = r \sigma \) \( \iff l = r \iff s_1 = t_1, \ldots, s_n = t_n \in E \)
Basics in Conditional Rewriting

Induced rewrite relation for CTRS $R$

for $R$ consisting of rules of the form $l \rightarrow r \leftarrow s_1 \Box t_1, \ldots, s_n \Box t_n$

with $\Box \in \{\leftrightarrow^*, \downarrow, \rightarrow^*\}$:

- $R_0 = \emptyset$
- $R_{k+1} = \{l\sigma \rightarrow r\sigma \mid l \rightarrow r \leftarrow c \in R, s_i\sigma \Box R_k t_i\sigma \text{ for all } s_i \Box t_i \text{ in } c\}$.
- $\rightarrow = \rightarrow^R = \bigcup_{i \geq 0} \rightarrow^R_i$.
- If $s \rightarrow^*_R t$ with some concrete derivation $s \rightarrow^*_R n t$, the latter
  has level $n$.
- The depth of a reduction $s \rightarrow^*_R t$ is the minimal (level) $n$ with
  $s \rightarrow^*_R t$.

Relationships between different induced rewrite relations

- $\rightarrow R, \rightarrow^* \subseteq \rightarrow R, \downarrow \subseteq \rightarrow R, \leftrightarrow^* \subseteq = R$
- Logicality: $\leftrightarrow^*_R, \leftrightarrow^* \supseteq = R$ (hence $\leftrightarrow^*_R, \leftrightarrow^* \supseteq = R$)?
Basics in Conditional Rewriting

Examples

For \( R = \{ a \rightarrow b \leftarrow c \leftrightarrow^* d; e \rightarrow c; e \rightarrow d \} \):

- \( e \rightarrow_{R_1} c, \ e \rightarrow_{R_1} d \)
- \( a \rightarrow_{R_2} b, \ a \rightarrow_{R_k} b \) for every \( k \geq 2 \)
- \( a \not\rightarrow_{R_1} b, \ \text{depth}(a \rightarrow^* b) = 2 \)

For \( R = \{ a \rightarrow b \leftarrow c \downarrow d; e \rightarrow c; e \rightarrow d \} \):

- \( c, d \) in normal form w.r.t. every \( R_k \)
- \( a \rightarrow_{R_k} b \) for no \( k \), hence \( a \not\rightarrow_R b \)

For \( R = \{ f(x) \rightarrow a \leftarrow f(x) \downarrow x; b \rightarrow f(b) \} \):

- \( b \rightarrow_R f(b) \rightarrow_R a, \ \text{since} \ f(b) \downarrow_R b \ \text{due to} \ f(b) \leftarrow_R b \)
- \( f(a) \rightarrow_R a \ \text{iff} \ f(a) \downarrow_R a \ \text{iff} \ f(a) \rightarrow_R a \) (since \( a \) NF), hence \( f(a) \not\rightarrow_R a \)
Major Problems and Difficulties

effectiveness in testing conditions

- conditions are recursively(!) evaluated
- entails in general a possibly non-terminating search process
- consequence: basic notions like one-step reduction, one-step reducibility, being a normal form etc. are undecidable in general!

no modular decomposition of steps (non-locality)

- $s \rightarrow_R t$ using rule $\rho$ does not imply $s \rightarrow\{\rho\} t$, because verification of conditions may need more rules, possibly the whole system!
- major obstacle in extending modularity results form TRSs to CTRSs
Major Problems and Difficulties

confluence: variable overlaps may be critical!

- TRS: \(l\sigma \rightarrow r\sigma, \sigma \rightarrow \sigma' \Rightarrow l\sigma' \rightarrow r\sigma'\) for \(l \rightarrow r \in R\)
- CTRS: \(l\sigma \rightarrow r\sigma, \sigma \rightarrow \sigma' \Rightarrow l\sigma' \rightarrow r\sigma'\) for \(l \rightarrow r \Leftarrow c \in R\) ?
  In general NO!
  - \(s_i\sigma \rightarrow^* t_i\sigma, \sigma \rightarrow \sigma',\) hence: \(s_i\sigma \rightarrow^* s_i\sigma', t_i\sigma \rightarrow^* t_i\sigma'\)
  - but: \(s_i\sigma' \downarrow t_i\sigma'\)?

termination: may be non-effective!

- \(R = \{a \rightarrow b \Leftarrow a \rightarrow^* c\}\) is
  - terminating (has empty rewrite relation \(\rightarrow_R\))
  - but trying to apply the rule leads to a cycle
- Hence, from a practical point of view,
  - \(R\) should not only be terminating,
  - but also the evaluation of conditions (leading to stronger notions of termination)
Major Problems and Difficulties

realization / implementation of conditional rewriting

▶ how to realize / implement rewriting with conditional rewrite rules?
▶ strategies for sequentialization / backtracking?
▶ how to avoid complicated rewrite machine architecture?

why CTRSs instead of transformed unconditional TRSs?

▶ conceptually more adequate, intuitive
▶ any encoding / transformation entails a loss of structural information!
▶ appropriate encodings / transformations into TRSs are not easy!
More tomorrow!
Lecture 2
Termination $\neq$ “operational termination”

problems

- effective computation (reduction, normalization)
  example: $a \rightarrow b \iff a \downarrow c$

- confluence criteria for TRSs do not directly generalize to CTRSs (see later)

- idea for recovery:
  - require well-founded decrease not only from $l$ to $r$, but also from $l$ to $u$, $u$ condition term in $c$, $l \rightarrow r \iff c \in R$
  - yields stronger notions of termination
Termination strengthened

approaches [Kaplan'84-87, Dershowitz et al. ’88-90]

- **R simplifying** if there exists simplification ordering $\succ$ (rewrite ordering with subterm property) s.t.
  $l \succ r, s_i, t_i$ for all $l \rightarrow r \Leftarrow s_1 \downarrow t_1, \ldots, s_n \downarrow t_n \in R$

- **R reductive** if there exists reduction ordering $\succ$ s.t. $l \succ r, s_i, t_i$
  for all $l \rightarrow r \Leftarrow s_1 \downarrow t_1, \ldots, s_n \downarrow t_n \in R$

- **R decreasing** if there exists well-founded term ordering s.t.
  
  - $\rightarrow_R \subseteq \succ$
  - $l\sigma \succ_{st} s_1\sigma, t_1\sigma, \ldots, s_n\sigma, t_n\sigma$ for all
  - $l \rightarrow r \Leftarrow s_1 \downarrow t_1, \ldots, s_n \downarrow t_n \in R (\succ_{st} = (\succ \cup \triangleright)^{+})$
Termination strengthened

relationships

- simplifying $\Rightarrow$ reductive $\Rightarrow$ decreasing $\Rightarrow$ terminating
- all implications are proper:
  - $f(f(x)) \rightarrow a \iff f(g(f(x))) \downarrow b$ reductive, but not simplifying
  - $f(b) \rightarrow f(a); a \rightarrow c \iff b \downarrow d$ decreasing, but not reductive
  - $a \rightarrow b \iff a \downarrow c$ terminating, but not decreasing
Termination strengthened

remarks

- oriented + normal CTRSs: for decreasingness e.g.:
  \[ l\sigma \triangleright_{st} s_1\sigma, \ldots, s_n\sigma \] instead of
  \[ l\sigma \triangleright_{st} s_1\sigma, t_1\sigma, \ldots, s_n\sigma, t_n\sigma \]

- decreasingness depends on interpretation of “=” in conditions
  (\(\downarrow\) or \(\rightarrow^*\))

- simplifyingness, reductivity and decreasingness can be refined:
  check conditions sequentially e.g. from left to right, instead of simultaneously)

- 2-CTRSs with extra variables in conditions cannot be simplifying, . . .

- 3-CTRSs with extra variables in right-hand sides:
  sequentialization becomes essential to make sense (see later)!
Confluence

criteria for TRSs

► without termination
  ► orthogonality implies confluence (CR)

► with termination
  ► **Critical Pair Theorem:**
    local confluence (WCR) ⇐⇒ joinability of CPs (JCP)
  ► **Newman’s Lemma:** SN ⇒ (WCR ⇐⇒ CR)
  ► hence: SN ⇒ (JCP ⇐⇒ CR)

critical pairs (peaks), CPs

► $l_1\sigma[r_2\sigma] \leftarrow_p l_1\sigma[l_2\sigma] \rightarrow_\epsilon r_1\sigma$ where
  ► $l_i \rightarrow r_i \in R$, $i = 1, 2$, variable disjoint
  ► $l_1|_p \in F$ (no overlap into variables)
  ► $\sigma$ mgu of $l_1|_p$ and $l_2$
Confluence

conditional critical pairs (peaks), CCPs

- \( l_1 \sigma [r_2 \sigma] \leftarrow_p l_1 \sigma [l_2 \sigma] p \rightarrow_\epsilon r_1 \sigma \) with
  - \( l_1 \rightarrow r_1 \leftarrow c_1 \in R, \sigma \) at \( \epsilon \) and \( l_2 \rightarrow r_2 \leftarrow c_2 \in R, \sigma \) at \( p \)
  - \( l_1|_p \in F \)
  - \( \sigma \) mgu of \( l_1|_p \) and \( l_2 \)

- yielding CCP \( l_1 \sigma [r_2 \sigma] = r_1 \sigma \leftarrow c_1 \sigma, c_2 \sigma \)

- joinability: \( s = t \leftarrow c \in CCP(R) \) joinable if
  \( \forall \sigma: s\sigma \downarrow t\sigma \leftarrow c\sigma \)

- (in)feasibility: \( s = t \leftarrow c \) feasible if there exists \( \sigma \) with \( c\sigma \)

- even for terminating systems:
  - (in)feasibility and joinability of CCPs undecidable

- \( R \) non-overlapping CTRS (via \( R_u \)) \( \longrightarrow \) syntactic criterion for “non-overlappingness”

- \( R \) has only infeasible CCPs \( \longrightarrow \) semantic criterion for “non-overlappingness”
Confluence of (Join 2-)CTRSs

criteria without termination

- orthogonality? No!
- \( b \rightarrow f(b); f(x) \rightarrow a \iff f(x) \downarrow x \)
  is orthogonal (left-linear and non-overlapping)
- \( b \rightarrow f(b) \rightarrow a \), due to \( f(b) \downarrow b \) due to \( f(b) \leftarrow b \), hence:
  \( f(a) \leftarrow f(b) \rightarrow a \), but
  \( f(a) \not\downarrow a \):
  \( f(a) \downarrow a \iff f(a) \rightarrow a \iff f(a) \downarrow a \ldots \) No!

sufficient criterion

- orthogonality + normality
- example: \( b \rightarrow f(b); f(x) \rightarrow a \iff f(x) \downarrow c \)
- proof idea: show that \( \leftarrow \rightarrow_m \) commutes over \( \rightarrow \rightarrow_n \), via
  induction on \( m+n \) (here: \( \rightarrow \rightarrow_m = \rightarrow \rightarrow R_m, \rightarrow m = \rightarrow R_m \))
Confluence of (Join 2-)CTRSs

criteria with termination

- for Newman’s Lemma we need: WCR
- does WCR $\iff$ JCP hold for CTRSs? No!
- counterexample $R$:
  
  (1) $h(x) \rightarrow k(b) \iff k(x) \downarrow h(b)$
  (2) $k(a) \rightarrow h(a)$
  (3) $a \rightarrow b$

  CCPs: $k(b) \leftarrow (3) k(a) \rightarrow (2) h(a)$,
  - due to $k(a) \downarrow h(b)$ via $k(a) \rightarrow h(a) \rightarrow h(b)$
  - joinable via: $k(b) \leftarrow h(a)$
  - however, the variable overlap $h(b) \leftarrow h(a) \rightarrow k(b)$ is NOT joinable anymore!

- properties of $R$:
  - $\neg$ decreasing, “normal”, $\neg$ overlaying, $\neg$ shallow joinable
Confluence of (Join 2-)CTRSs

strengthened notions of confluence for CTRSs

- shallow joinable CCP $t \leftarrow s \rightarrow u \leftarrow c$:
  $\forall \sigma$ with $c\sigma$: $t\sigma \leftarrow m s\sigma \rightarrow n u\sigma$ implies $t\sigma \rightarrow^* n \nu \leftarrow^* m u\sigma$ for some $\nu$ (notation: $\rightarrow n \Rightarrow \rightarrow R_n$)

- shallow confluence: $\leftarrow^* m \circ \rightarrow^* n \subseteq \rightarrow^* n \circ \leftarrow^* m$

- level confluence: $\leftarrow^* n \circ \rightarrow^* n \subseteq \rightarrow^* n \circ \leftarrow^* n$

- shallow confluence $\Rightarrow$ level confluence $\Rightarrow$ confluence
Confluence of (Join 2-)CTRSs

confluence criteria with termination [Dershowitz et al.’88-90]
A terminating (join 2-) CTRS with joinable CCPs is confluent if it is

(a) decreasing; or
(b) left-linear, normal, shallow joinable; or
(c) overlaying

proof ideas

(a) by well-founded induction (as for TRSs), but using the stronger decreasing order
(b) by proving shallow confluence
(c) by using $\rightarrow^+$ as induction order and exploiting the overlay property
Confluence of (Join 2-)CTRSs

beyond overlay systems

- consider $R$ given by

\[
\begin{align*}
    a & \rightarrow b \\
    f(a, a) & \rightarrow b \\
    f(b, x) & \rightarrow f(x, x) \iff f(x, x) \downarrow x \\
    f(x, b) & \rightarrow f(x, x) \iff f(x, x) \downarrow x
\end{align*}
\]

- CCPs (the non-trivial ones) joinable:
  - $f(b, a) \leftarrow f(a, a) \rightarrow b$: $f(b, a) \rightarrow f(a, a) \rightarrow b$
  - $f(a, b) \leftarrow f(a, a) \rightarrow b$: $f(a, b) \rightarrow f(a, a) \rightarrow b$

- but: $f(b, b) \parallel f(a, a) \rightarrow a$ not joinable anymore

- shared parallel critical peaks play a crucial role

- this can be exploited for generalizing the overlay confluence criterion (c) beyond overlay systems [Gramlich/Wirth’96]
Exercises A – Confluence of CTRS

Test the following join (1-) CTRSs for confluence (via constructing conditional critical pairs and applying known confluence criteria):

(a) $R = \{ \begin{align*}
    h(g(x), y) & \rightarrow h(g(x), g(y)) & \iff & h(x, x) \downarrow a \\
    h(a, g(y)) & \rightarrow y
\end{align*} \} \) 

(b) $R = \{ \begin{align*}
    f(x) & \rightarrow k(x) & \iff & h(x) \downarrow x \\
    h(a) & \rightarrow a \\
    g(f(x)) & \rightarrow b \\
    g(k(a)) & \rightarrow b \\
    g(k(h(x))) & \rightarrow g(k(x))
\end{align*} \} \) 

(c) $R = \{ \begin{align*}
    \text{ins}(x, c(y, \_)) & \rightarrow \text{c}(x, c(y, l)) & \iff & x \leq y \downarrow \text{true} \\
    \text{ins}(x, c(y, \_)) & \rightarrow \text{c}(y, \text{ins}(x, \_)) & \iff & x \leq y \downarrow \text{false} \\
    0 & \leq y \rightarrow \text{true} \\
    s(x) & \leq 0 \rightarrow \text{false} \\
    s(x) & \leq s(y) \rightarrow \text{false}
\end{align*} \} \)
Lecture 4
Selected literature on conditional rewriting