Seminar Decision Procedures and Applications

Background Information: Part II

Viorica Sofronie-Stokkermans

University Koblenz-Landau

25 June 2019

Brief Introduction to Term Rewriting

Equality is the most important relation in mathematics and functional programming.

In principle, problems in first-order logic with equality can be handled by, e.g., resolution theorem provers.

Handling Equality Naively

$F \mapsto \tilde{F}$ ($\approx \mapsto \sim$). Encode properties of equality $\mapsto Eq(\Sigma)$

$$\begin{gathered}
\forall x\,(x \sim x) \\
\forall x, y\,(x \sim y \rightarrow y \sim x) \\
\forall x, y, z\,(x \sim y \wedge y \sim z \rightarrow x \sim z) \\
\forall \vec{x}, \vec{y}\,(x_1 \sim y_1 \wedge \cdots \wedge x_n \sim y_n \rightarrow f(x_1, \ldots, x_n) \sim f(y_1, \ldots, y_n)) \\
\forall \vec{x}, \vec{y}\,(x_1 \sim y_1 \wedge \cdots \wedge x_n \sim y_n \wedge p(x_1, \ldots, x_n) \rightarrow p(y_1, \ldots, y_n))
\end{gathered}$$

F is satisfiable if and only if $Eq(\Sigma) \cup \{\tilde{F}\}$ is satisfiable.

Handling Equality Naively

By giving the equality axioms explicitly, first-order problems with equality can in principle be solved by a standard resolution or tableaux prover.

But this is unfortunately not efficient
(mainly due to the transitivity and congruence axioms).

Roadmap

How to proceed:

- Arbitrary binary relations.
- Equations (unit clauses with equality):

  Term rewrite systems.

  Expressing semantic consequence syntactically.

  Entailment for equations.

- Equational clauses:

  The superposition calculus

Roadmap

How to proceed:

- Arbitrary binary relations.
- Equations (unit clauses with equality):

  Term rewrite systems.

  Expressing semantic consequence syntactically.

  Entailment for equations.

- Equational clauses:

  The superposition calculus

Roadmap

How to proceed:

- Arbitrary binary relations.
- Equations (unit clauses with equality):

  Term rewrite systems.

  Expressing semantic consequence syntactically.

  Entailment for equations.

- Equational clauses:

  The superposition calculus

Roadmap

How to proceed:

- Arbitrary binary relations.
- Equations (unit clauses with equality):

  Term rewrite systems.

  Expressing semantic consequence syntactically.

  Entailment for equations.

- Equational clauses:

  The superposition calculus

Abstract Reduction Systems

Abstract reduction system: $(A, \to)$, where

A is a set,

$\to \subseteq A \times A$ is a binary relation on A.

Abstract Reduction Systems

$\to^0$	$= \{ (x, x) \mid x \in A \}$	identity
$\to^{i+1}$	$= \to^i \circ \to$	$i + 1$-fold composition
$\to^+$	$= \bigcup_{i>0} \to^i$	transitive closure
$\to^*$	$= \bigcup_{i\geq 0} \to^i \; = \; \to^+ \cup \to^0$	reflexive transitive closure
$\to^=$	$= \to \cup \to^0$	reflexive closure
$\to^{-1}$	$= \leftarrow \; = \{ (x, y) \mid y \to x \}$	inverse
$\leftrightarrow$	$= \to \cup \leftarrow$	symmetric closure
$\leftrightarrow^+$	$= (\leftrightarrow)^+$	transitive symmetric closure
$\leftrightarrow^*$	$= (\leftrightarrow)^*$	refl. trans. symmetric closure

Abstract Reduction Systems

$x \in A$ is reducible, if there is a y such that $x \to y$.

x is in normal form (irreducible), if it is not reducible.

y is a normal form of x, if $x \to^* y$ and y is in normal form.
Notation: $y = x{\downarrow}$ (if the normal form of x is unique).

x and y are joinable, if there is a z such that $x \to^* z \leftarrow^* y$.
Notation: $x \downarrow y$.

Abstract Reduction Systems

A relation $\rightarrow$ is called

Church-Rosser, if $x \leftrightarrow^* y$ implies $x \downarrow y$.

confluent, if $x \leftarrow^* z \rightarrow^* y$ implies $x \downarrow y$.

locally confluent, if $x \leftarrow z \rightarrow y$ implies $x \downarrow y$.

terminating, if there is no infinite decreasing chain
$x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \ldots$.

normalizing, if every $x \in A$ has a normal form.

convergent, if it is confluent and terminating.

Abstract Reduction Systems

Lemma 2: If $\rightarrow$ is terminating, then it is normalizing.

Note: The reverse implication does not hold.

Abstract Reduction Systems

Theorem 3: The following properties are equivalent:

(i) $\rightarrow$ has the Church-Rosser property ($x \leftrightarrow^* y$ implies $x \downarrow y$)

(ii) $\rightarrow$ is confluent ($x \leftarrow^* z \rightarrow^* y$ implies $x \downarrow y$)

Proof:

(i)$\Rightarrow$(ii): trivial.

(ii)$\Rightarrow$(i): by induction on the number of peaks in the derivation $x \leftrightarrow^* y$.

Abstract Reduction Systems

Lemma 4:
If $\rightarrow$ is confluent, then every element has at most one normal form.

Corollary 5:
If $\rightarrow$ is normalizing and confluent, then every element x has a unique normal form.

Proposition 6:
If $\rightarrow$ is normalizing and confluent, then $x \leftrightarrow^* y$ if and only if $x{\downarrow} = y{\downarrow}$.

Well-Founded Orderings

Lemma 7:

If $\rightarrow$ is a terminating binary relation over A,
then $\rightarrow^+$ is a well-founded partial ordering.

Lemma 8:

If $>$ is a well-founded partial ordering and $\rightarrow \subseteq >$,
then $\rightarrow$ is terminating.

Proving Confluence

Theorem 9 ("Newman's Lemma"):
If a terminating relation $\rightarrow$ is locally confluent ($x \leftarrow z \rightarrow y$ implies $x \downarrow y$), then it is confluent ($x \leftarrow^* z \rightarrow^* y$ implies $x \downarrow y$).

Proof:

Let $\rightarrow$ be a terminating and locally confluent relation.
Then $\rightarrow^+$ is a well-founded ordering.
Define $P(z) \;\Leftrightarrow\; \bigl(\forall x, y : x \leftarrow^* z \rightarrow^* y \Rightarrow x \downarrow y\bigr)$.
Prove $P(z)$ for all $z \in A$ by well-founded induction over $\rightarrow^+$:
Case 1: $x \leftarrow^0 z \rightarrow^* y$: trivial.
Case 2: $x \leftarrow^* z \rightarrow^0 y$: trivial.
Case 3: $x \leftarrow^* x' \leftarrow z \rightarrow y' \rightarrow^* y$: use local confluence, then use the induction hypothesis.

Roadmap

How to proceed:

- Arbitrary binary relations.
- Equations (unit clauses with equality):

  Term rewrite systems.

  Expressing semantic consequence syntactically.

  Entailment for equations.

- Equational clauses:

  The superposition calculus

Rewrite Systems

Notation:

Positions of a term s:

$$\mathrm{Pos}(x) = \{\varepsilon\},$$
$$\mathrm{Pos}(f(s_1, \ldots, s_n)) = \{\varepsilon\} \cup \bigcup_{i=1}^{n} \{\, ip \mid p \in \mathrm{Pos}(s_i) \,\}.$$

Size of a term s: $|s|$ = cardinality of $\mathrm{Pos}(s)$.

Subterm of s at a position $p \in \mathrm{Pos}(s)$:

$$s/\varepsilon = s,$$
$$f(s_1, \ldots, s_n)/ip = s_i/p.$$

Replacement of the subterm at position $p \in \mathrm{Pos}(s)$ by t:

$$s[t]_\varepsilon = t,$$
$$f(s_1, \ldots, s_n)[t]_{ip} = f(s_1, \ldots, s_i[t]_p, \ldots, s_n).$$

Rewrite Relations

Let E be a set of equations.

The rewrite relation $\rightarrow_E \subseteq \mathsf{T}_\Sigma(X) \times \mathsf{T}_\Sigma(X)$ is defined by

$$
\begin{array}{lll}
s \rightarrow_E t & \text{iff} & \text{there exist } (l \approx r) \in E,\ p \in \mathrm{Pos}(s), \\
 & & \text{and } \sigma : X \rightarrow \mathsf{T}_\Sigma(X), \\
 & & \text{such that } s/p = l\sigma \text{ and } t = s[r\sigma]_p.
\end{array}
$$

An equation $l \approx r$ is also called a rewrite rule, if l is not a variable and $\mathrm{Var}(l) \supseteq \mathrm{Var}(r)$.

Notation: $l \rightarrow r$.

A set of rewrite rules is called a term rewrite system (TRS).

Rewrite Relations

We say that a set of equations E or a TRS R is terminating, if the rewrite relation $\rightarrow_E$ or $\rightarrow_R$ has this property.

(Analogously for other properties of abstract reduction systems).

Note: If E is terminating, then it is a TRS.

Rewrite Relations

Corollary 10:
If E is convergent (i.e., terminating and confluent),
then $s \approx_E t$ if and only if $s \leftrightarrow_E^* t$ if and only if $s{\downarrow_E} = t{\downarrow_E}$.

Corollary 11:
If E is finite and convergent, then $\approx_E$ is decidable.

Reminder:
If E is terminating, then it is confluent if and only if it is locally confluent.

Rewrite Relations

Problems:

Show local confluence of E.

Show termination of E.

Transform E into an equivalent set of equations that is locally confluent and terminating.

Order $\succ$ on terms $\quad l \approx r,\, l \succ r \mapsto l \to r$

talk in this seminar: ground TRS (left and right hand side are ground terms)

Simple form: $f(c_1, \ldots, c_n) \to c$ or $c \to d$

Critical Pairs

Showing local confluence (Sketch for ground TRS):

Question:
Are there rewrite rules $l_1 \to r_1$ and $l_2 \to r_2$ such that some subterm l_1/p and l_2 are equal?

Let $l_i \to r_i$ $(i = 1, 2)$ be two rewrite rules in a TRS R
Let $p \in \text{Pos}(l_1)$ be a position such that $l_1/p = l_2$.

Then $r_1 \leftarrow l_1 \to (l_1)[r_2]_p$.

$\langle r_1, (l_1)[r_2]_p \rangle$ is called a critical pair of R.

The critical pair is joinable (or: converges), if $r_1 \downarrow_R (l_1)[r_2]_p$.

Critical Pairs

Theorem 12 ("Critical Pair Theorem"):
A TRS R is locally confluent if and only if all its critical pairs are joinable.

Proof (Here only for the case of ground TRS):
"only if": obvious, since joinability of a critical pair is a special case of local confluence.

"if": Suppose s rewrites to t_1 and t_2 using rewrite rules $l_i \rightarrow r_i \in R$ at positions $p_i \in \mathrm{Pos}(s)$, where $i = 1, 2$.
Then $s/p_i = l_i$ and $t_i = s[r_i]_{p_i}$.

We distinguish between two cases: Either p_1 and p_2 are in disjoint subtrees ($p_1 \parallel p_2$), or one is a prefix of the other (w.o.l.o.g., $p_1 \leq p_2$).

Critical Pairs

Case 1: $p_1 \parallel p_2$.

Then $s = s[l_1]_{p_1}[l_2]_{p_2}$,
and therefore $t_1 = s[r_1]_{p_1}[l_2]_{p_2}$ and $t_2 = s[l_1]_{p_1}[r_2]_{p_2}$.

Let $t_0 = s[r_1]_{p_1}[r_2]_{p_2}$.
Then clearly $t_1 \rightarrow_R t_0$ using $l_2 \rightarrow r_2$ and $t_2 \rightarrow_R t_0$ using $l_1 \rightarrow r_1$.

Case 2: $p_1 \leq p_2$.

Then $s/p_2 = l_2$ and $s/p_2 = (s/p_1)/p = l_1/p$; hence $l_2 = l_1/p$; and

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

By assumption, it is joinable, so $r_1 \rightarrow_R^* v \leftarrow_R^* (l_1)[r_2]_p$.

Consequently, $t_1 = s[r_1]_{p_1} = s[r_1]_{p_1} \rightarrow_R^* s[v]_{p_1}$ and
$t_2 = s[r_2]_{p_2} = s[(l_1)[r_2]_p]_{p_1} = s[(l_1)[r_2]_p]_{p_1} = s[((l_1)[r_2]_p)]_{p_1} \rightarrow_R^* s[v]_{p_1}$.

This completes the proof of the Critical Pair Theorem.

Critical Pairs

Note: Critical pairs between a rule and (a renamed variant of) itself must be considered – except if the overlap is at the root (i.e., $p = \varepsilon$).

Critical Pairs

Corollary 13:

A terminating TRS R is confluent if and only if all its critical pairs are joinable.

Proof:

By Newman's Lemma and the Critical Pair Theorem.

Critical Pairs

Corollary 14:

For a finite terminating TRS, confluence is decidable.

Proof:

For every pair of rules and every non-variable position in the first rule there is at most one critical pair $\langle u_1, u_2 \rangle$.

Reduce every u_i to some normal form u_i'. If $u_1' = u_2'$ for every critical pair, then R is confluent, otherwise there is some non-confluent situation $u_1' \leftarrow_R^* u_1 \leftarrow_R s \rightarrow_R u_2 \rightarrow_R^* u_2'$.

Roadmap

How to proceed:

- Arbitrary binary relations.
- Equations (unit clauses with equality):

  Term rewrite systems.

  Expressing semantic consequence syntactically.

  Entailment for equations.

- Equational clauses:

  The superposition calculus "ordered resolution with built-in rewriting"

The resolution calculus

Resolution

$$\frac{C \vee L \quad D \vee \neg L'}{(C \vee D)\sigma}$$

$\sigma = \mathrm{mgu}(L, L')$

Factoring

$$\frac{C \vee L \vee L'}{(C \vee L)\sigma}$$

$\sigma = \mathrm{mgu}(L, L')$

The ordered resolution calculus

Ordered Resolution $\succ$ order on ground literals

$$\frac{C \vee A \quad D \vee \neg A'}{(C \vee D)\sigma}$$

$\sigma = \text{mgu}(L, L'),\ A\sigma \succ C\sigma, \neg A\sigma \succeq D\sigma$

Ordered Factoring

$$\frac{C \vee A \vee A'}{(C \vee L)\sigma}$$

$\sigma = \text{mgu}(A, A'),\ A\sigma \succeq C\sigma$

The superposition calculus

Handling equality: Ordered resolution with “built-in” term rewriting

$\succ$ ordering on terms $\mapsto$ ordering on atoms of the form $l \approx r$

The superposition calculus

Handling equality: Ordered resolution with “built-in” term rewriting

$\succ$ ordering on terms $\mapsto$ ordering on atoms of the form $l \approx r$

Superposition left

$$\frac{C \vee l[u'] \approx r \quad D \vee u \approx v}{(C \vee D \vee l[v] \approx r)\sigma}$$

Paramodulation

$$\frac{C \vee \neg l[u'] \approx r \quad D \vee u \approx v}{(C \vee D \vee \neg l[v] \approx r)\sigma}$$

$\sigma = \mathrm{mgu}(u, u')$,

(i)	$\sigma(u) \succ \sigma(v)$,	(iii)	$\sigma(l) \succ \sigma(r)$
(ii)	$\sigma(u \approx v) \succ \sigma(D)$	(iv)	$\sigma(l \approx r) \succ \sigma(C)$

The superposition calculus

Reflection

$$\frac{C \vee \neg u' \approx u}{C\sigma}$$

$\sigma = \mathrm{mgu}(u, u')$, $\sigma(u \approx u') \succeq \sigma(C)$

Factoring

$$\frac{C \vee u \approx v \vee u' \approx v'}{(\neg v \approx v' \vee C \vee u \approx v')\sigma}$$

$\sigma = \mathrm{mgu}(u, u')$,

(i) $\sigma(u) \succ \sigma(v)$,

(ii) $\sigma(u \approx v) \succeq \sigma(\mathrm{positive}(C) \cup \{u' \approx v'\})$

(iii) $\sigma(u) \succ \sigma(\mathrm{negative}(C))$

The superposition calculus

Subsumption	$\mapsto$	subsumed clauses are deleted
Simplification	$\mapsto$	in the presence of a unit clause $l \approx r$ with $l \succ r$,
		the rule is used as a "rewriting rule" for simplification
Deletion	$\mapsto$	Clauses containing $t \approx t$ are always true and are deleted

The superposition calculus

Theorem The superposition calculus is sound and refutationally complete:

A set N of clauses in FOL with equality is unsatisfiable iff $N \vdash_{\text{Superposition}} \bot$.

Stefan Strüder: Situations in which the superposition calculus terminates.

Overview

- Arbitrary binary relations.

- Equations (unit clauses with equality):

  Term rewrite systems.
  Expressing semantic consequence syntactically.
  Entailment for equations.

- Equational clauses:

  The superposition calculus "ordered resolution with built-in rewriting"