Little Engines of Proof

N. Shankar, L. de Moura, H. Ruess, A. Tiwari

shankar@csl.sri.com

URL: http://www.csl.sri.com/~shankar/LEP.html

Computer Science Laboratory
SRI International
Menlo Park, CA

III. Equality over Ground Terms: No Boolean Structure

$s_1 = t_1 \wedge s_2 = t_2 \wedge \cdots \wedge s_n = t_n \wedge s'_1 \neq t'_1 \wedge \cdots \wedge s'_m \neq t'_m$

Relevant equality axioms: 3 equivalence axioms + congruence

Closure under axioms does not terminate. Why?

Symmetric-Transitive-Congruence (STC) closure

Transitivity $\dfrac{s = t, t = u, \Gamma}{s = t, t = u, s = u, \Gamma}$ if $s = u \notin \Gamma$	
Contradiction $\dfrac{s = t, s \neq t, \Gamma}{\perp}$	Contradiction $\dfrac{s \neq s, \Gamma}{\perp}$
Congruence $\dfrac{s = t, \Gamma}{s = t, C[s] = C[t], \Gamma}$ if $C[s], C[t]$ occur in $\Gamma, s = t, \ldots$	

III. STC Inference System: Completeness-1

Partition ϕ into E and DE.

TPT: If E, DE is unsatisfiable, then $E, DE \vdash_{STC} \perp$

TPT: If $E \models s = t$ and $s \neq t \in DE$, then $E, DE \vdash_{STC} s = t$

Theorem: $E \models s = t$ iff $E \vdash s = t$

Define: $\rightarrow_R = \{C[l] \rightarrow C[r] : l \rightarrow r \in R\}$

Define: $\leftrightarrow_R = \leftarrow_R \cup \rightarrow_R$

Define: $\leftrightarrow^*_R = \leftrightarrow_R \circ \leftrightarrow_R \circ \ldots \leftrightarrow_R$

Theorem: $E \vdash s = t$ iff $s \leftrightarrow^*_E t$ Ex: Prove this.

TPT: If $s \leftrightarrow^*_E t$ and $s \neq t \in DE$, then $E, DE \vdash_{STC} s = t$

$$s = s_0 \leftrightarrow_E s_1 \leftrightarrow_E s_2 \leftrightarrow_E \cdots \leftrightarrow_E s_n = t$$

III. STC Inference System: Completeness-2

TPT: If $s \leftrightarrow^*_E t$ and s, t occur in E, DE, then $E, DE \vdash_{STC} s = t$

Prove by well-founded induction on pairs $\{s, t\}$

Ordering: multiset extension of depth ordering

$$s = s_0 \leftrightarrow s_1 \leftrightarrow s_2 \leftrightarrow \cdots \leftrightarrow s_n = t$$

Break proof at all TOP applications of E

$$s = s_0 \leftrightarrow^* s_i \leftrightarrow^{top} s_{i+1} \leftrightarrow^* s_j \leftrightarrow^{top} s_{j+1} \leftrightarrow^* s_n = t$$

If $t = f t_1 t_2 \ldots t_k$, let $(t)_p$ denote t_p

By induction hypothesis, $E, DE \vdash_{STC} (s_0)_p = (s_i)_p$ for all p

Similarly, $E, DE \vdash_{STC} (s_{i+1})_p = (s_j)_p$ for all p, and so on

$\therefore E, DE \vdash_{STC} s_0 = s_i$ and $E, DE \vdash_{STC} s_{i+1} = s_j$, and so on

$\therefore E, DE \vdash_{STC} s = t$

III. STC inference system: Exercises

Ex: Show that the STC system is sound and terminating.

Ex: Write up the completeness proof is full detail.

Ex: Show that the STC inference rules can be applied exponentially many times before terminating.

Ex: Show that the size of the final state can be exponentially large.

View the unordered STC calculus as an ordered calculus instantiated with the trivial (empty) ordering

Ex*: Optimize the STC rules using a well-founded ordering on terms. Is the worst case behavior, using a total ordering, any better than the worst case of STC?

Simplifying the Term Structure

Terms over Σ can be simplified by introducing new names from K.

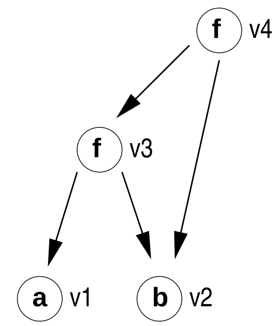
Extend	$\dfrac{s[fc_1 \ldots c_k] = t, \Gamma}{s[c] = t, fc_1 \ldots c_k \to c, \Gamma}$	if $f \in \Sigma,\ c \in K$
Simplify	$\dfrac{s[u] = t, u \to c, \Gamma}{s[c] = t, u \to c, \Gamma}$	

If we apply Extend and Simplify exhaustively, then the final configuration will look like

$$c_1 = d_1,\ \ldots,\ c_n = d_n,\ c_1' \neq d_1',\ \ldots,\ c_m' \neq d_m',$$
$$fe_1 \ldots e_k \to e, \ldots$$

Corresponds to DAG Representation

Example. Let $E_0 = \{a = fab \ \wedge\ f(fab, b) = b\}$.



$a = fab,$	$ffabb = b$
$c_1 = fc_1b,$ $a \to c_1$	$ffc_1bb = b,$
$c_1 = fc_1c_2,$ $a \to c_1,$	$ffc_1c_2c_2 = c_2,$ $b \to c_2$
$c_1 = c_3,$ $a \to c_1,$ $fc_1c_2 \to c_3$	$fc_3c_2 = c_2,$ $b \to c_2$
$c_1 = c_3,$ $a \to c_1,$ $fc_1c_2 \to c_3,$	$c_4 = c_2,$ $b \to c_2$ $fc_3c_2 \to c_4$

III. Completing the Rules

Now, ϕ can be partitioned as $\phi_1 \ \wedge\ \phi_2$

ϕ_1: conjunction of D-equations of the form $fc_1 \ldots c_k = c$

ϕ_2: conjunction of C-equations $c = d$ and $c \neq d$

Handling ϕ_2: Recall Ordered-Transitive closure rules: Orient, Simplify, Collapse, Compose (Union-Find)

But we are still missing the congruence axiom

Superpose	$\dfrac{fc_1 \ldots c_k \to c,\ \ fc_1 \ldots c_k \to d,\ \ \Gamma}{c = d,\ \ fc_1 \ldots c_k \to c,\ \ \Gamma}$
Collapse	$\dfrac{f \ldots c \ldots \to d,\ \ c \to c',\ \ \Gamma}{f \ldots c' \ldots \to d,\ \ c \to c',\ \ \Gamma}$
Compose	$\dfrac{f \ldots \to c,\ \ c \to d,\ \ \Gamma}{f \ldots \to d,\ \ c \to d,\ \ \Gamma}$

III. Abstract Congruence Closure (ACC)

Extend + Simplify +

Orient $\dfrac{c = d, \Gamma}{c \to d, \Gamma}$ if $c \succ d$	Delete $\dfrac{c = c, \Gamma}{\Gamma}$
Simplify $\dfrac{c = d, c \to d', \Gamma}{d' = d, c \to d', \Gamma}$	Simplify $\dfrac{c \neq d, c \to d', \Gamma}{d' \neq d, c \to d', \Gamma}$
Superpose $\dfrac{fc_1 \ldots c_k \to c, \;\; fc_1 \ldots c_k \to d, \;\; \Gamma}{c = d, \;\; fc_1 \ldots c_k \to c, \;\; \Gamma}$	
Collapse $\dfrac{f \ldots c \ldots \to d, \;\; c \to c', \;\; \Gamma}{f \ldots c' \ldots \to d, \;\; c \to c', \;\; \Gamma}$	Collapse $\dfrac{c \to d, c \to c', \Gamma}{c' = d, c \to c', \Gamma}$
Compose $\dfrac{f \ldots \to c, \;\; c \to d, \;\; \Gamma}{f \ldots \to d, \;\; c \to d, \;\; \Gamma}$	Compose $\dfrac{c' \to c, c \to d, \Gamma}{c' \to d, c \to d, \Gamma}$
Contradict $\dfrac{c \neq c, \Gamma}{\perp}$	

III. Abstract Congruence Closure: Termination

Termination: Each rule, except Extend, is size nonincreasing

Extend decreases size of the set of equations

Number of Extend steps $< n$; hence size of system $=$ O(n)

Number of applications of other rules $=$ O$(n\delta)$

δ: length of longest chain $d_1 \to d_2 \to \cdots$

Clearly, $\delta \leq n$, hence maximum length of derivation $=$ O(n^2)

Ex: Implement a quadratic time congruence closure algorithm.

III. Abstract Congruence Closure: Basic Strategies

Union-Find strategy on C-equations guarantees $\delta < \log(n)$

Efficient congruence closure: O$(n\log(n))$ inference steps

Certain strategies can make certain rules inapplicable

Ex: Which ACC inference rules are optional in this sense?

Ex: Implement a O$(n\log(n))$ congruence closure algorithm using the above inference rules.

Ex: Can you interpret your strategy as suitable manipulations on the term DAG data-structure?

III. Abstract Congruence Closure: Example

$a = fab, f(fab)b = b$

$a \to c_1, b \to c_2, fc_1c_2 \to c_3, fc_3c_2 \to c_4, c_1 = c_3, c_4 = c_2$

$a \to c_1, b \to c_2, fc_1c_2 \to c_3, fc_3c_2 \to c_4, c_3 \to c_1, c_4 \to c_2$

$a \to c_1, b \to c_2, fc_1c_2 \to c_3, fc_1c_2 \to c_4, c_3 \to c_1, c_4 \to c_2$

$a \to c_1, b \to c_2, fc_1c_2 \to c_3, c_4 = c_3, c_3 \to c_1, c_4 \to c_2$

$a \to c_1, b \to c_2, fc_1c_2 \to c_3, c_2 = c_1, c_3 \to c_1, c_4 \to c_2$

$a \to c_1, b \to c_2, fc_1c_2 \to c_3, c_2 \to c_1, c_3 \to c_1, c_4 \to c_2$

$a \to c_1, b \to c_2, fc_1c_1 \to c_3, c_2 \to c_1, c_3 \to c_1, c_4 \to c_2$

$a \to c_1, b \to c_1, fc_1c_1 \to c_3, c_2 \to c_1, c_3 \to c_1, c_4 \to c_1$

III. Abstract Congruence Closure: Soundness

Soundness: Each inference rule preserves satisfiability

Ignore disequations presently

If $E \vdash_{ACC} E'$, then $\leftrightarrow^*_E$ is identical to $\leftrightarrow^*_{E'}$ restricted to the terms over the original signature

Ex: Prove!

If $E \vdash^*_{ACC} E'$ and E' is a final state, then $s \leftrightarrow^*_E t$ iff $s \leftrightarrow^*_{E'} t$, for all terms s, t over Σ

There are no equations in the final state, only rules

Final state R: contains D-rules and C-rules

III. Abstract Congruence Closure: Completeness-1

TPT: If E, DE is unsatisfiable, then $E, DE \vdash^*_{ACC} \perp$

TPT: If $E \models (s = t)$ and $s \neq t \in DE$, then $E, DE \vdash^*_{ACC} s = t$

Suppose: $E, DE \vdash^*_{ACC} R, DE'$, where R, DE' is a final state

WPT: whenever $s \leftrightarrow^*_E t$, then $s \rightarrow^*_R \circ \leftarrow^*_R t$

We have

$$s \leftrightarrow_E \circ \leftrightarrow_E \cdots \leftrightarrow_E t$$

Therefore, there is a proof of the form

$$s \leftrightarrow_R \circ \leftrightarrow_R \cdots \leftrightarrow_R t$$

III. Abstract Congruence Closure: Completeness-2

$$s \leftrightarrow_R \circ \leftrightarrow_R \cdots \leftrightarrow_R t$$

$\rightarrow_R$ is terminating

The following patterns cannot occur:

- Pattern $d \leftarrow_R f \ldots c \ldots \rightarrow_R f \ldots c' \ldots$ (Collapse)
- Pattern $d \leftarrow_R f \ldots \rightarrow_R c$ where $c \not\equiv d$ (Superpose)
- Pattern $d \leftarrow_R c \rightarrow_R c'$ where $c' \not\equiv d$ (Collapse)

Local confluence of R: If $s \leftarrow_R u \rightarrow_R t$, then $s \rightarrow^*_R v \leftarrow^*_R t$

(Get new proof by commuting the two steps)

Therefore, R is terminating and locally confluent

III. Abstract Congruence Closure: Completeness-3

Confluence of R: If $s \leftarrow^*_R u \rightarrow^*_R t$, then $s \rightarrow^*_R v \leftarrow^*_R t$

Newman's Lemma: If R is terminating and locally confluent, then R is confluent

Hence, R is confluent

We had $s \leftrightarrow_R \circ \leftrightarrow_R \cdots \leftrightarrow_R t$

By repeated applications of confluence, we get $s \rightarrow^*_R \circ \leftarrow^*_R t$

Convergent: confluence + termination

Normal Form of s is s' where $s \rightarrow^*_R s'$ and $s' \not\rightarrow_R$

R is convergent. Convergent R induce unique normal forms and all reductions lead to it.

III. Abstract Congruence Closure: Completeness Summary

Started with equations E

The ACC rules transformed E into R such that

- (for all terms s, t over Σ,) $E \models s = t$ iff $s \rightarrow_R^* \circ \leftarrow_R^* t$
- $\rightarrow_R$ is terminating

I.e., equal terms w.r.t E have the same normal form w.r.t R

Hence, inconsistency of any $s \neq t$ can be detected by normalizing s and t by R to get $u \neq u$

The process of transforming E to R is called (Knuth-Bendix) completion

This was a very special case: Only ground equations

And we had one unusual rule: Extend

III. Completion: Illustration