Term Generalization for Idempotent Equational Theories

David M. Cerna and Temur Kutsia

Research Institute for Symbolic Computation (RISC)
Johannes Kepler University, Linz, Austria

July 7^{th}, 2018

Term Generalization

- Let Σ be a signature of ranked function and constant symbols, $\mathcal{V}$ a countably infinite set of variables, and $\mathcal{L}$ the language constructible from $\Sigma \cup \mathcal{V}$.
- Generalization: given $t, s \in \mathcal{L}$ find an $r \in \mathcal{L}$ s.t. $\exists$ s σ_1 and σ_2 and $r\sigma_1 = t$ and $r\sigma_2 = s$.
- We extend the generalization problem by adding idempotent functions to Σ and considering equality modulo idempotency.
- In particular we are looking for least general generalizers (lggs), i.e. $\nexists$ σ and r' s.t. for a given r, $r' = r\sigma$.

Two Idempontent functions $\equiv$ Infinite Set of LGGs

- In "Generalisation de termes en theorie equationnelle. Cas associatif-commutatif" by L. Pottier, an example using two idempotent function symbols whose solutions contains an infinite number of incomparable generalizations was given.
- Let $\Sigma = \{f(\cdot,\cdot), g(\cdot,\cdot), a, b\}$ where f and g are idempotent. We refer to the equational theory as $I_{\{f,g\}}$.

$$f(a,b) \triangleq g(a,b)$$

A Infinite Sequence of Generalizations

- The following terms generalize the anti-unification problem:

$$g(f(a,x), f(y,b)) \qquad f(g(a,x), g(y,b))$$

$$g(f(a,x), f(y,b))\,\{x \mapsto a\ ,\ y \mapsto b\} =_{I_{\{f,g\}}} g(a,b)$$

$$g(f(a,x), f(y,b))\,\{x \mapsto b\ ,\ y \mapsto a\} =_{I_{\{f,g\}}} f(a,b)$$

- This is not a complete set, but enough for constructing an infinite incomparable sequence.

$$S_0 = g(f(a,x), f(y,b))$$

$$S_{n+1} = f(g(f(a,x), f(y,b)), g(S_n, f(g(a,x), g(y,b))))$$

$$f(g(f(a,x), f(y,b)), g(S_n, f(g(a,x), g(y,b)))) \neq_{I_{\{f,g\}}} f(g(f(a,x), f(y,b)), g(S_{n+1}, f(g(a,x), g(y,b))))$$

One Idempontent function $\equiv$ Infinite Set of LGGs?

- If one idempotent function symbol turns out to be finitary, then the above result would imply that joining two finitary theories can result in an infinitary theory. Unstable behavior!
- But does the above example really need both f and g?
- Considering f and g to be functions it is easy to imagine an h such that $h'(a, a, b) = f(a, b)$ and $h'(b, a, b) = g(a, b)$.
- What if we apply this reasoning to our problem and look at the anti-unification problem:

$$h(a, h(a, b)) \triangleq h(b, h(a, b))$$

One Function, Infinite Solutions

- The following terms generalize the anti-unification problem:

$$h(h(x, h(x, b)), h(a, h(x, b))) \quad h(f(x, h(a, x)), h(h(x, b), h(a, b)))$$

$$h(h(x, h(x, b)), h(a, h(x, b)))\,\{x \mapsto a\} =_{I_{\{h\}}} h(a, h(a, b))$$

$$h(h(x, h(x, b)), h(a, h(x, b)))\,\{x \mapsto b\} =_{I_{\{h\}}} h(b, h(a, b))$$

- Notice that the solutions are in some sense simpler and thus more fundamental. Less variables.
- Using the Pottier construction we can produce an infinite set of incomparable LGGs.

Minimal Complete Pottier construction

- Given that one idempotent function symbol is enough for infinitely many solutions, What is the simplest example of this behavior?
- It turns out that

  $$f(a,b) \triangleq f(b,a)$$

  is enough

  $$f(f(x_1,a), f(b,x_2)) \qquad f(f(x_1,b), f(a,x_2))$$

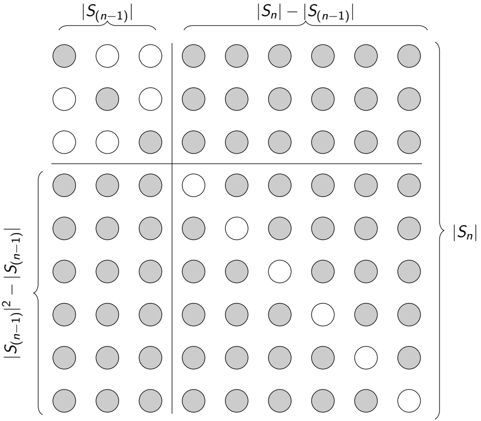
- We can generalize of Pottier's construction to illustrate this:

A minimal complete set

$$S_0 = \{f(f(x_1, a), f(b, x_2)), f(f(x_1, b), f(a, x_2))\}.$$
$$S_k = \{f(s_1, s_2) \mid s_1, s_2 \in S_{k-1}, s_1 \neq s_2\} \cup S_{k-1},\ \ k > 0.$$

- The limit S_∞ can be proven minimal complete for the above problem and bounds on the growth can be computed.

Proof: Growth of S_n is $O\left(2^{2^n}\right)$



$$|S_{(n+1)}| = |S_n|^2 - |S_{(n-1)}|^2 + |S_{(n-1)}| \qquad |S_1| = m \qquad |S_0| = 1$$

Is the S-hierarchy Enough?

- While the S-hierarchy works for $f(a,b) \triangleq f(b,a)$ it fails for slightly more complex problems, i.e.

$$f(a, f(a,b)) \triangleq f(a, f(b,a))$$

It captures an infinite number of incomparable generalizations, but it also misses an infinite number because $f(a,b) \triangleq f(b,a)$ is embedded within this problem:

$$f(a, f(f(x,a), f(b,y))) \in S_\infty \qquad f(a, f(f(x,b), f(a,y))) \in S_\infty$$

$$f(f(a, f(f(x,a), f(b,y))), f(a, f(f(x,b), f(a,y)))) \in S_\infty$$

$$f(a, f(f(f(x,a), f(b,y)), f(f(x,b), f(a,y)))) \notin S_\infty$$

Capturing Minimal Completeness

- Idempotent Generalization has more structure than the S-hierarchy can capture.
- Consider, instead of computing generalizers:

$$f(a, f(f(x, a), f(b, y))) \quad f(a, f(f(x, a), f(b, y)))$$

we construct a binding list

$$[x \mapsto f(a, x_2)]\,[x_2 \mapsto f(f(z_1, a), f(b, x_4))]$$

$$[x \mapsto f(a, x_2)]\,[x_2 \mapsto f(f(z_2, b), f(a, x_5))]$$

- using these binding list we can construct a larger set of binding:

$$[x \mapsto f(a, y)] \quad \begin{array}{l} [y \mapsto f(f(z, b), f(a, w))] \\ [y \mapsto f(f(w, a), f(b, z))] \end{array}$$

algorithmically.

Capturing Minimal Completeness

- This set can be extended to the following:

$$
\begin{array}{l}
[x \mapsto f(a, y)] \\
[x \mapsto f(f(a, y), f(a, y))] \\
\quad [y \mapsto f(f(z, b), f(a, w))] \\
\quad [y \mapsto f(f(w, a), f(b, z))] \\
\quad [y \mapsto f(f(f(w, a), f(b, z)), f(f(z, b), f(a, w)))]
\end{array}
$$

- If we collapse the bindings, both

$$f(a, f(f(f(w, a), f(b, z)), f(f(z, b), f(a, w))))$$

and

$$f(f(a, f(f(w, a), f(b, z))), f(a, f(f(z, b), f(a, w))))$$

are induced terms.

Its a Set, its a Substitution, its a Tree Grammar?!

$$
\begin{aligned}
N &= \{\mathbf{x}_{\text{gen}}, \mathbf{x}_1, \ldots, \mathbf{x}_6\},\\
T &= \{f, a, b, y_1, y_2\},\\
R &= \{\mathbf{x}_{\text{gen}} \to \mathbf{x}_1,\ \mathbf{x}_{\text{gen}} \to f(\mathbf{x}_{\text{gen}}, \mathbf{x}_{\text{gen}}),\\
&\quad \mathbf{x}_1 \to f(a, \mathbf{x}_2),\ \mathbf{x}_1 \to f(f(a, f(y_1, a)), \mathbf{x}_3),\\
&\qquad \mathbf{x}_1 \to f(f(a, f(a, y_2)), \mathbf{x}_4),\ \mathbf{x}_1 \to f(\mathbf{x}_1, \mathbf{x}_1),\\
&\quad \mathbf{x}_2 \to f(f(y_1, a), f(b, y_2)),\ \mathbf{x}_2 \to f(f(y_1, b), f(a, y_2)),\\
&\qquad \mathbf{x}_2 \to f(\mathbf{x}_2, \mathbf{x}_2),\\
&\quad \mathbf{x}_3 \to f(a, f(b, y_2)),\ \mathbf{x}_3 \to f(f(a, y_2), \mathbf{x}_5),\\
&\qquad \mathbf{x}_3 \to f(f(a, f(y_1, a)), f(y_2, f(b, y_2))),\ \mathbf{x}_3 \to f(\mathbf{x}_3, \mathbf{x}_3),\\
&\quad \mathbf{x}_4 \to f(y_1, f(a, y_2)),\ \mathbf{x}_4 \to f(f(y_1, a), \mathbf{x}_6),\\
&\qquad \mathbf{x}_4 \to f(f(y_1, f(y_1, b)), f(a, f(a, y_2))),\ \mathbf{x}_4 \to f(\mathbf{x}_4, \mathbf{x}_4),\\
&\quad \mathbf{x}_5 \to f(f(y_1, a), f(b, y_2)),\ \mathbf{x}_5 \to f(f(y_1, b), f(a, y_2)),\\
&\qquad \mathbf{x}_5 \to f(\mathbf{x}_5, \mathbf{x}_5),\\
&\quad \mathbf{x}_6 \to f(f(y_1, a), f(b, y_2)),\ \mathbf{x}_6 \to f(f(y_1, b), f(a, y_2)),\\
&\qquad \mathbf{x}_6 \to f(\mathbf{x}_6, \mathbf{x}_6)\}.
\end{aligned}
$$

This is the tree grammar of $f(a,f(a,b)) \triangleq f(a,f(b,a))$

In addition to the expected rules we include rules of the form $\mathbf{x}_5 \to f(\mathbf{x}_5, \mathbf{x}_5)$

These rules capture the expansion presented earlier.

$$\text{STORE} = \{y_1 : a \triangleq b, y_2 : b \triangleq a\}$$

.

I-PreGen or How to Make the Grammars

I-PreGen is the generalization algorithm which provides the foundation for the tree grammar construction:

Dec: **Decomposition**

$\{x : f(s_1, \dots, s_n) \triangleq f(t_1, \dots, t_n)\} \uplus A;\ S;\ B \Longrightarrow \{y_1 : s_1 \triangleq t_1, \dots, y_n : s_n \triangleq t_n\} \cup A;\ S;\ B\{x \mapsto f(y_1, \dots, y_n)\}$

where f is a free function symbol, $n \geq 0$, and $y_1, \dots, y_n$ are fresh variables.

Solve: **Solve**

$\{x : s \triangleq t\} \uplus A;\ S;\ B \Longrightarrow A;\ \{x : s \triangleq t\} \cup S; B,$

where $head(s) \neq head(t)$ and neither $head(s)$ nor $head(t)$ is an idempotent symbol.

Id-Left: **Idempotent symbol in the left**

$\{x : f(s_1, s_2) \triangleq t\} \uplus A;\ S;\ B \Longrightarrow$
$\quad \{y_1 : s_1 \triangleq t,\ y_2 : s_2 \triangleq t\} \cup A;\ S;\ B\{x \mapsto f(y_1, y_2)\},$

where f is an idempotent function symbol, $head(t)$ is not idempotent, and y_1 and y_2 are fresh variables.

Id-Right: **Idempotent symbol in the right**

$\{x : s \triangleq f(t_1, t_2)\} \uplus A;\ S;\ B \Longrightarrow$
$\quad \{y_1 : s \triangleq t_1, y_2 : s \triangleq t_2\} \cup A;\ S;\ B\{x \mapsto f(y_1, y_2)\},$

where f is an idempotent function symbol, $head(s)$ is not idempotent, and y_1 and y_2 are fresh variables.

I-PreGen or How to Make the Grammars

Id-Both 1: **Idempotent symbol on both sides 1**

$\{x : f(s_1, s_2) \triangleq f(t_1, t_2)\} \uplus A;\ S;\ B \Longrightarrow$
$\quad \{y_1 : s_1 \triangleq t_1,\ y_2 : s_2 \triangleq t_2\} \cup A;\ S;\ B \cup \{x \mapsto f(y_1, y_2)\},$

where f is an idempotent function symbol and y_1 and y_2 are fresh variables.

Id-Both 2: **Idempotent symbol on both sides 2**

$\{x : f(s_1, s_2) \triangleq t\} \uplus A;\ S;\ B \Longrightarrow$
$\quad \{y_1 : s_1 \triangleq t,\ y_2 : s_2 \triangleq t\} \cup A;\ S;\ B \cup \{x \mapsto f(y_1, y_2)\},$

where f is an idempotent function symbol, $head(t)$ is idempotent, and y_1 and y_2 are fresh variables.

Id-Both 3: **Idempotent symbol on both sides 3**

$\{x : s \triangleq f(t_1, t_2)\} \uplus A;\ S;\ B \Longrightarrow$
$\quad \{y_1 : s \triangleq t_1, y_2 : s \triangleq t_2\} \cup A;\ S;\ B \cup \{x \mapsto f(y_1, y_2)\},$

where f is an idempotent function symbol, $head(s)$ is idempotent, and y_1 and y_2 are fresh variables.

Merge: **Merge**

$\emptyset;\ \{x_1 : s_1 \triangleq t_1,\ x_2 : s_2 \triangleq t_2\} \uplus S;\ B \Longrightarrow \emptyset;\ \{x_1 : s_1 \triangleq t_1\} \cup S;\ B\{x_2 \mapsto x_1\},$

where $s_1 \approx_I s_2$ and $t_1 \approx_I t_2$.

Two Idempotent Heads, Better than One?

- The rules Id-Both 1,2, and 3 add bindings which are later used in the grammar.
- Essentially the occurrence of idempotent symbols on both sides of an AUP results in branching in the solution set. Consider, $f(a,b) \triangleq f(b,a)$

$$x : f(a,b) \triangleq f(b,a);\ [x_{gen} \rightarrow x] \Longrightarrow_{Id\text{-}Both\ 2}$$

$$y_1 : f(a,b) \triangleq b, y_2 : f(a,b) \triangleq a;\ \emptyset;\ [x_{gen} \rightarrow x] \cup [x \rightarrow f(y_1,y_2)]$$

$$x : f(a,b) \triangleq f(b,a);\ [x_{gen} \rightarrow x]\} \Longrightarrow_{Id\text{-}Both\ 3}$$

$$y_1 : a \triangleq f(b,a), y_2 : b \triangleq f(b,a);\ \emptyset;\ [x_{gen} \rightarrow x] \cup [x \rightarrow f(y_1,y_2)]$$

Building the Grammar

- After exhausting all derivations of I-PreGen for a given AUP and join them, we can simplify the resulting set of bindings by removing I-comparable bindings. This is the base set of grammar rule.
- The full set is constructed by adding the duplication rules

$Duplicate(x, G_b) :=$

$\{x \to f(x, x) \mid f$ is an idempotent symbol and R_b contains two different rules $y \to r_1$ and $y \to r_2$ for $y \in reach(G_b, x)\}$

$$reach(G, \nu) := \{\mu \mid \nu \to_G^+ t \text{ and } \mu \in nter(G, t)\}$$

- We also join all of the individual stores together in **STORE** and remove duplicate variable renaming.

Complete and Minimal

- The resulting Grammar is both complete and minimal after idempotent normalization of its language.

Theorem (Completeness)

Let t_1 and t_2 be two terms and $G(t_1, t_2)$ their regular tree grammar. Let r be a common I-generalization of t_1 and t_2. Then there exists $s \in L(G(t_1, t_2))$ such that $r \preceq_I s$.

Proof.

Structural induction on r. □

- Minimality requires an induction on the number of duplication rules (i.e. $\boldsymbol{x} \to f(\boldsymbol{x}, \boldsymbol{x})$) used for the construction of a given term.

L_n and Minimality

- $L_n(G(t_1, t_2)) \subset L(G(t_1, t_2))$ where $t \in L_n(G(t_1, t_2))$ is constructed using n duplication rules. I.e. it is a finite language.

Theorem

Let s and t be two terms and $G(s, t)$ be their grammar. Then for all $n \geq 0$, $L_n(G(s, t))$ is minimal.

Proof.

We first perform an induction on n. Then we perform on another induction on the m, where $L_m(G(t_1, t_2))$ is the language where a subterm used in a generalizer from $L_{n+1}(G(t_1, t_2))$ comes from. Then we perform an induction on the length of the derivation constructing this subterm. Then finally we perform an induction on the number of idempotent symbols occurring in the top most rule constructing the subterm. □

Interesting Corollaries and Conclusions

Corollary

For all $n, m \geq 0$ *and* $n \leq m$*, if* $r \in L_n(G(s,t))$ *then* $r \in L_m(G(s,t))$.

Corollary

$L(G(s,t))$ *is minimal complete modulo Idempotent Equivalence.*

Corollary

Idempotent generalization is infinitary.

- Future work will investigate the uses for proof transformation methods, cut introduction, and induction theorem proving based on tree grammar construction and minimization.

The End

Thank you for your time!