COMBINING GENERALIZATION ALGORITHMS

Temur Kutsia

Joint work with Mauricio Ayala (University of Brasilia), David Cerna (Czech Academy) and Christophe Ringeissen (INRIA/LORIA)

JOHANNES KEPLER UNIVERSITY LINZ

Generalization problem

Given: An equational theory E and two terms t_1 and t_2.

Find: A term r such that for some substitutions σ_1 and σ_2,

$$r\sigma_1 =_E t_1 \quad \text{and} \quad r\sigma_2 =_E t_2.$$

r is more general than t_1 and t_2 (notation: $r \preceq_E t_i$, $i = 1, 2$).

σ_1 and σ_2: witness substitutions.

r: a common generalization of t_1 and t_2.

Generalization problem

Given: An equational theory E and two terms t_1 and t_2.

Find: A term r such that for some substitutions σ_1 and σ_2,

$$r\sigma_1 =_E t_1 \quad \text{and} \quad r\sigma_2 =_E t_2.$$

r is more general than t_1 and t_2 (notation: $r \preceq_E t_i$, $i = 1, 2$).

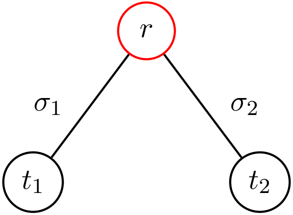
σ_1 and σ_2: witness substitutions.

r: a common generalization of t_1 and t_2.

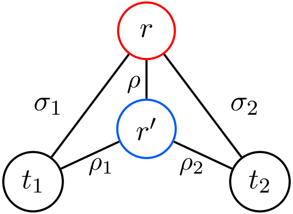
r is a least general generalization (lgg) of t_1 and t_2 if there is no r' such that

- r' is a generalization of t_1 and t_2, and
- r' is strictly less general than r.

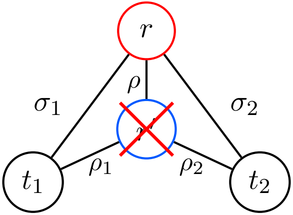
Lest General Generalizations



Lest General Generalizations



Lest General Generalizations



Examples

$E = \emptyset :$ syntactic equality

$t_1 = f(a, g(a))$

$t_2 = f(b, g(b))$

$r = f(x, g(x)) :$ a single lgg

$$r\{x \mapsto a\} =_E t_1$$

$$r\{x \mapsto b\} =_E t_2$$

Examples

$E = \{f(x,y) = f(y,x)\}$: f is commutative

$t_1 = g(a, f(a,b))$

$t_2 = g(b, f(b,a))$

$r_1 = g(x, f(x,y))$: an lgg

$$r_1\{x \mapsto a, y \mapsto b\} =_E t_1$$

$$r_1\{x \mapsto b, y \mapsto a\} =_E t_2$$

Examples

$$E = \{f(x,y) = f(y,x)\}: \ f \text{ is commutative}$$

$$t_1 = g(a, f(a,b))$$

$$t_2 = g(b, f(b,a))$$

$$r_1 = g(x, f(x,y)): \ \text{an lgg}$$

$$r_1\{x \mapsto a, y \mapsto b\} =_E t_1$$

$$r_1\{x \mapsto b, y \mapsto a\} =_E t_2$$

$$r_2 = g(z, f(a,b)): \ \text{another lgg}$$

$$r_2\{z \mapsto a\} =_E t_1$$

$$r_1\{z \mapsto b\} =_E t_2$$

Examples

$E = \{f(f(x,y),z) = f(x,f(y,z))\}$: f is associative

$t_1 = f(a,b,b,a)$

$t_2 = f(a,b,a)$

$r_1 = f(a,b,x)$: an lgg

$$r_1\{x \mapsto f(b,a)\} =_E t_1$$

$$r_1\{x \mapsto a\} =_E t_2$$

Examples

$E = \{f(f(x, y), z) = f(x, f(y, z))\}$: f is associative

$t_1 = f(a, b, b, a)$

$t_2 = f(a, b, a)$

$r_1 = f(a, b, x)$: an lgg

$r_1\{x \mapsto f(b, a)\} =_E t_1$

$r_1\{x \mapsto a\} =_E t_2$

$r_2 = f(y, b, a)$: another lgg

$r_2\{y \mapsto f(a, b)\} =_E t_1$

$r_1\{y \mapsto a\} =_E t_2$

Examples

$$E = \{f(f(x,y),z) = f(x,f(y,z)), f(x,e) = x, f(e,x) = x\}:$$

f is associative, e its unit element

$$t_1 = f(a,b,b,a)$$

$$t_2 = f(a,b,a)$$

$r_1 = f(a,b,x,a):$ one lgg

$$r_1\{x \mapsto b\} =_E t_1$$

$$r_1\{x \mapsto e\} =_E t_2$$

Examples

$$E = \{f(f(x,y),z) = f(x,f(y,z)), f(x,e) = x, f(e,x) = x\}:$$

f is associative, e its unit element

$t_1 = f(a,b,b,a)$

$t_2 = f(a,b,a)$

$r_1 = f(a,b,x,a):$ one lgg

$r_1\{x \mapsto b\} =_E t_1$

$r_1\{x \mapsto e\} =_E t_2$

$r_2 = f(a,y,b,a):$ another lgg

$r_2\{y \mapsto b\} =_E t_1$

$r_1\{y \mapsto e\} =_E t_2$

Examples

$$E = \{f(f(a,a),a) = f(a,a),\ f(f(b,b),b) = f(b,b)\}$$

$$t_1 = f(a,a)$$

$$t_2 = f(b,b)$$

$$r_1 = f(x_1, x_1),$$

$$r_2 = f(f(x_2, x_2), x_2),$$

$$r_3 = f(f(f(x_3, x_3), x_3), x_3),$$

$$\dots$$

infinitely many lggs

$r_i\{x_i \mapsto a\} =_E t_1$ for any $i \geq 1$

$r_i\{x_i \mapsto b\} =_E t_2$ for any $i \geq 1$

Examples

$$E = \{f(a) = a, f(b) = b\}$$

$$t_1 = a$$

$$t_2 = b$$

A chain of strictly decreasing generalizations

$r_1 = x_1 \prec_E r_2 = f(x_2) \prec_E r_3 = f(f(x_3)) \cdots$

$r_i\{x_i \mapsto a\} =_E t_1$ for any $i \geq 1$

$r_i\{x_i \mapsto b\} =_E t_2$ for any $i \geq 1$

Minimal Complete Set of Generalizations

Given an equational theory E, a set of terms $\mathcal{G}$ is called a complete set of E-generalizations of the given terms t_1 and t_2, if the following properties are satisfied:

- **Soundness:** each $r \in \mathcal{G}$ is an E-generalization of t_1 and t_2.
- **Completeness:** for each E-generalization r' of t_1 and t_2 there exists $r \in \mathcal{G}$ such that $r' \preceq_E r$.

Minimal Complete Set of Generalizations

Given an equational theory E, a set of terms $\mathcal{G}$ is called a complete set of E-generalizations of the given terms t_1 and t_2, if the following properties are satisfied:

- **Soundness:** each $r \in \mathcal{G}$ is an E-generalization of t_1 and t_2.
- **Completeness:** for each E-generalization r' of t_1 and t_2 there exists $r \in \mathcal{G}$ such that $r' \preceq_E r$.

The set $\mathcal{G}$ is called a minimal complete set of E-generalizations of t_1 and t_2, denoted by $\mathsf{mcsg}_E(t_1, t_2)$, if, in addition, the following holds:

- **Minimality:** no distinct elements of $\mathcal{G}$ are $\preceq_E$-comparable.

Generalization Algorithm in the Free Theory

A rule-base algorithm working on configurations $P; S; \sigma$, where

- P is a set of unsolved problems,
- S is a set of solved problems, and
- σ is a substitution representing the generalization computed so far.

Generalization Algorithm in the Free Theory

A rule-base algorithm working on configurations $P; S; \sigma$, where

- P is a set of unsolved problems,
- S is a set of solved problems, and
- σ is a substitution representing the generalization computed so far.

To generalize t and s, take a fresh variable x, create the initial configuration $\{x : t \triangleq s\}; \emptyset; Id$ and apply the algorithm rules as long as possible.

Generalization Algorithm in the Free Theory

A rule-base algorithm working on configurations $P; S; \sigma$, where

- P is a set of unsolved problems,
- S is a set of solved problems, and
- σ is a substitution representing the generalization computed so far.

To generalize t and s, take a fresh variable x, create the initial configuration $\{x : t \triangleq s\}; \emptyset; Id$ and apply the algorithm rules as long as possible.

The process stops in some final configuration $\emptyset; S; \sigma$ such that

- $x\sigma$ is an lgg of t and s,
- S gives the corresponding witness substitutions.

Generalization Algorithm in the Free Theory

DEC: **Decomposition**

$$\{x : f(t_1, \ldots, t_n) \triangleq f(s_1, \ldots, s_n)\} \uplus P; S; \sigma \Longrightarrow$$
$$P \cup \{y_1 : t_1 \triangleq s_1, \ldots, y_n : t_n \triangleq s_n\}; S; \sigma\{x \mapsto f(y_1, \ldots, y_n)\},$$

where $n \geq 0$ and $y_1, \ldots, y_n$ are fresh.

SOL: **Solve**

$$\{x : t \triangleq s\} \uplus P; S; \sigma \Longrightarrow P; S \cup \{x : t \triangleq s\}; \sigma,$$

if $\mathsf{root}(t) \neq \mathsf{root}(s)$.

MER: **Merge**

$$\emptyset; \{x : t \triangleq s,\ x' : t \triangleq s\} \uplus S; \sigma \Longrightarrow \emptyset; S \cup \{x : t \triangleq s\}; \sigma\{x' \mapsto x\}.$$

Generalization Algorithm in the Free Theory

Example

Syntactic generalization of $f(a, g(a))$ and $f(b, g(b))$:

$$\{x : f(a, g(a)) \triangleq f(b, g(b))\}; \emptyset; Id \Longrightarrow_{\mathsf{Dec}}$$
$$\{y : a \triangleq b, z : g(a) \triangleq g(b)\}; \emptyset; \{x \mapsto f(y, z)\} \Longrightarrow_{\mathsf{Sol}}$$
$$\{z : g(a) \triangleq g(b)\}; \{y : a \triangleq b\}; \{x \mapsto f(y, z)\} \Longrightarrow_{\mathsf{Dec}}$$
$$\{u : a \triangleq b\}; \{y : a \triangleq b\}; \{x \mapsto f(y, g(u)), z \mapsto g(u)\} \Longrightarrow_{\mathsf{Sol}}$$
$$\emptyset; \{y : a \triangleq b,\ u : a \triangleq b\}; \{x \mapsto f(y, g(u)),\ z \mapsto g(u)\} \Longrightarrow_{\mathsf{Mer}}$$
$$\emptyset; \{y : a \triangleq b\}; \{x \mapsto f(y, g(y)), z \mapsto g(y), u \mapsto y\}.$$

The lgg: the instance of x under the computed subst.: $f(y, g(y))$.

The witness substitutions: $\{y \mapsto a\}$ and $\{y \mapsto b\}$.

Generalization Type

The type of the E-generalization problem between t_1 and t_2 is

- **unitary** $(\mathbf{1})$: if $\mathsf{mcsg}_E(t_1, t_2)$ is a singleton,
- **finitary** (ω): if $\mathsf{mcsg}_E(t_1, t_2)$ is finite (not a singleton),
- **infinitary** (∞): if $\mathsf{mcsg}_E(t_1, t_2)$ is infinite,
- **nullary** $(\mathbf{0})$: if $\mathsf{mcsg}_E(t_1, t_2)$ does not exist
  (i.e., minimality and completeness contradict each other).

The generalization type of an equational theory E:

- the $<$-maximal type of E-generalization problems, where
  $\mathbf{1} < \omega < \infty < \mathbf{0}$.

Generalization Type: Examples

- **Unitary**:

  $E = \emptyset$.

- **Finitary**:

  commutativity, associativity, associativity with unit.

- **Infinitary**:

  $E = \{f(f(a,a),a) = f(a,a),\ f(f(b,b),b) = f(b,b)\}$

- **Nullary**:

  $E = \{f(a) = a,\ f(b) = b\}$.

Combination Problem

Is it possible to derive a generalization algorithm for a union of equational theories from the existing generalization algorithms for the component theories?

Combination Problem

Is it possible to derive a generalization algorithm for a union of equational theories from the existing generalization algorithms for the component theories?

Given (complete) algorithms for theory E_1 and for theory E_2, can we obtain a (complete) algorithm for $E_1 \cup E_2$?

Modularity.

Combination Problem

For theories sharing function symbols, the combination problem is very challenging.

Generalization type might change.

Each of $E_1 = \{f(a) = a\}$ and $E_2 = \{f(b) = b\}$ are theories with the unitary generalization type, but $E_1 \cup E_2$ is nullary.

Combination Problem

For theories sharing function symbols, the combination problem is very challenging.

Generalization type might change.

Each of $E_1 = \{f(a) = a\}$ and $E_2 = \{f(b) = b\}$ are theories with the unitary generalization type, but $E_1 \cup E_2$ is nullary.

We focus on signature-disjoint union (no shared symbols in the axioms).

Combination Problem

Even signature-disjoint union can be problematic.

Each of

- $E_1 = \{f(x,0) = x,\ f(0,x) = x\}$ and
- $E_2 = \{g(x,1) = x,\ g(1,x) = x\}$

are theories with the finitary generalization type, but $E_1 \cup E_2$ is nullary.

Combination Problem

Even signature-disjoint union can be problematic.

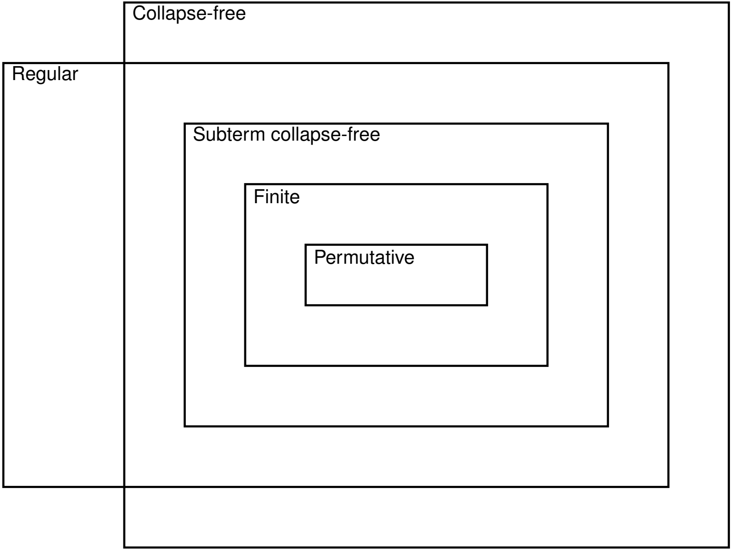
Each of

- $E_1 = \{f(x, 0) = x,\ f(0, x) = x\}$ and
- $E_2 = \{g(x, 1) = x,\ g(1, x) = x\}$

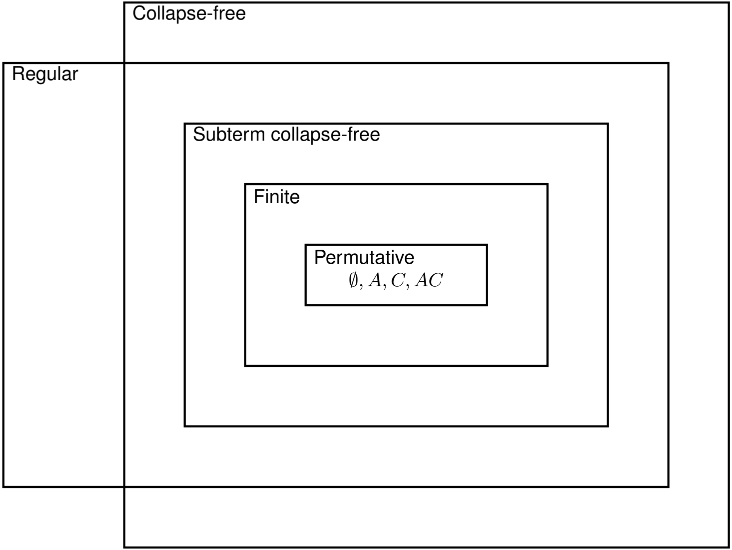
are theories with the finitary generalization type, but $E_1 \cup E_2$ is nullary.

$\longrightarrow$ Signature-disjoint union for a restricted class of equational theories.

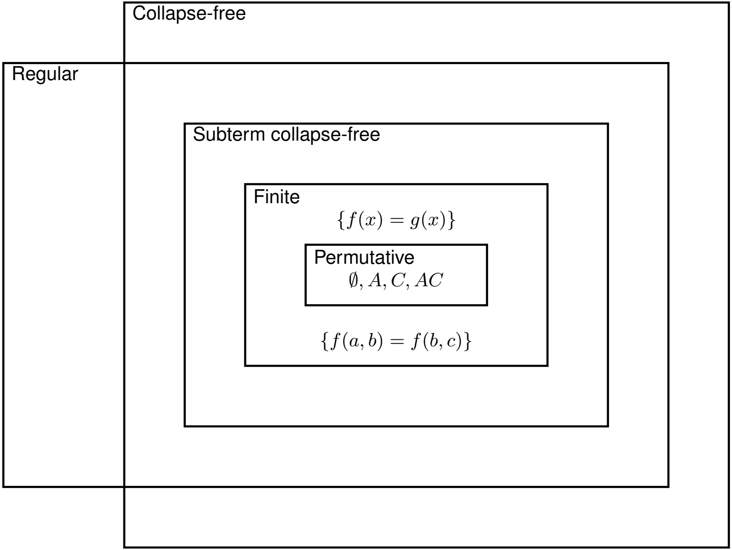
Classes of Equational Theories



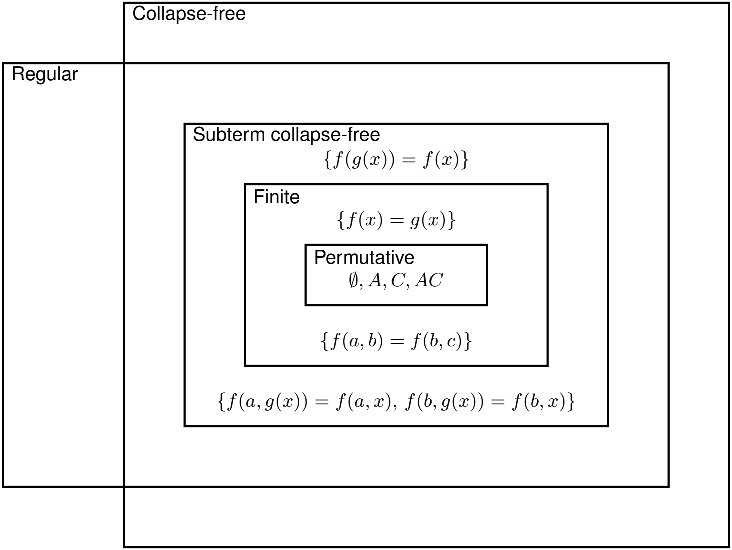
Classes of Equational Theories



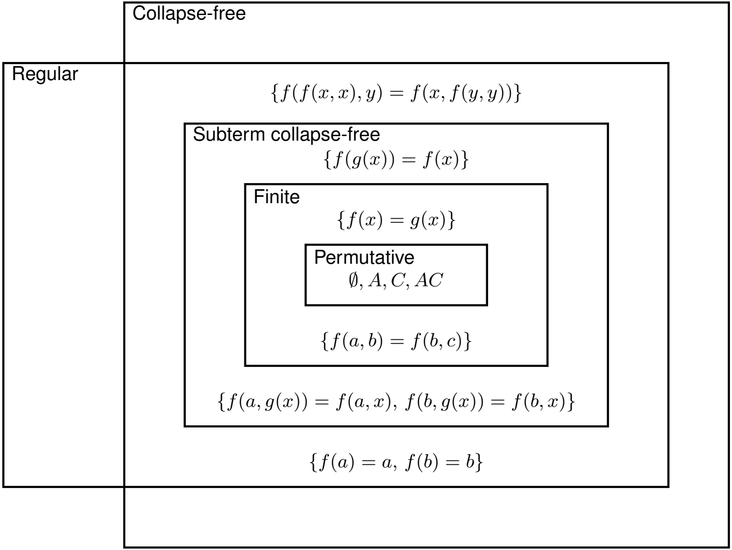
Classes of Equational Theories



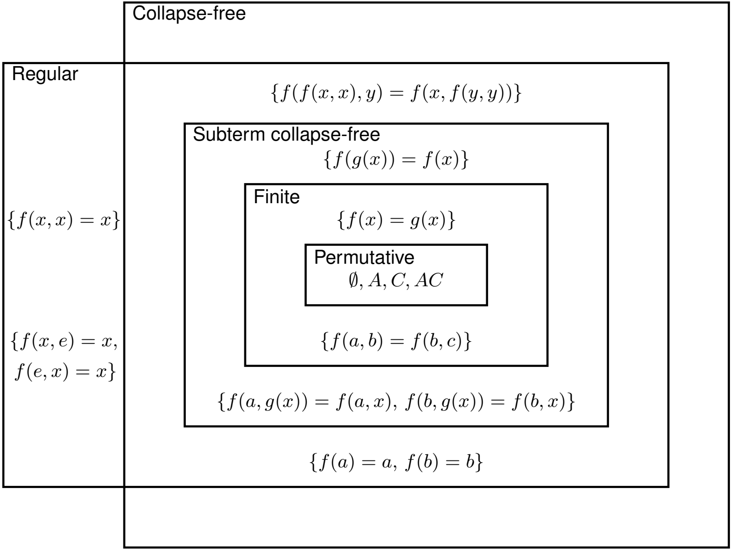
Classes of Equational Theories



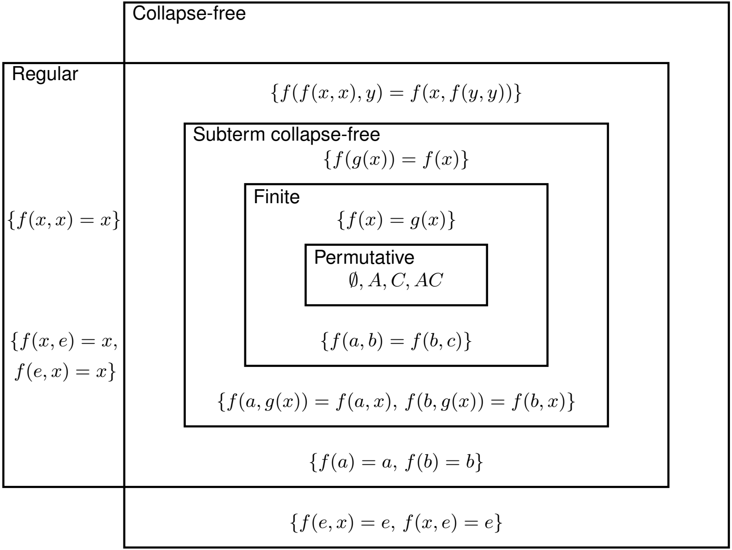
Classes of Equational Theories



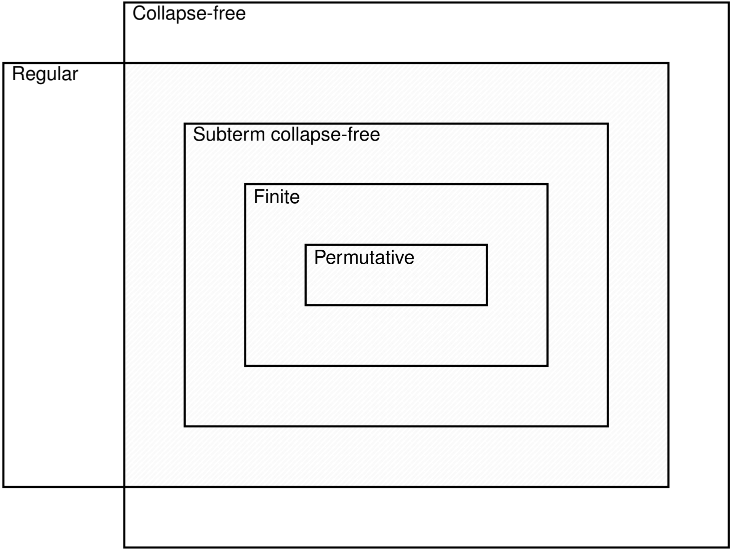
Classes of Equational Theories



Classes of Equational Theories



Classes of Equational Theories



Regular Collapse-Free Theories

We designed a combination method for generalization in the union of signature-disjoint regular collapse-free theories, where

- free (uninterpreted) constants are allowed in generalization problems together with symbols from the component signatures,
- for each component theory (enriched with free constants), we have a procedure that computes a minimal complete set of generalizations together with their witness substitutions,
- equality is decidable in the combined theory.

Combination Algorithms: Two Versions

Black-box combination:

- does not require the knowledge of how the component algorithms work,
- only an input-output interface with the them is needed,
- more generic, but less transparent.

Combination Algorithms: Two Versions

White-box combination:

- both component algorithms work on the same data-structure: $P; S; \sigma$, where P is a set of unsolved problems, S is a set of solved ones, σ is a substitution representing the generalization computed so far.
- PSS: problem, solved set, substitution,
- less generic than black-box, but more transparent and provides more flexibility,
- many existing algorithms for particular theories are PSS algorithms.

Alien Subterm Abstraction

A technique used in various combination algorithms.

Idea:

- If the root of a term t is a symbol from the signature Σ_1 (resp. Σ_2), replace all maximal subterms of t rooted in Σ_2 (resp. Σ_1) by fresh free constants.
- Equal subterms are replaces by the same constants.
- Hence, if the original term contains symbols from Σ_1, Σ_2, and C (free constants), and is rooted in Σ_1, then the abstracted term contains symbols from Σ_1 and C.

Alien Subterm Abstraction

Idea by example.

Assume

- $\Sigma_1 = \{f, h\}$,
- $\Sigma_2 = \{g, a, b\}$ with $g(a) =_{E_2} b$,
- c is a free constant.

Then

- Blue subterms of $f(g(a), b, h(g(b)), c)$ are alien in it.
- Abstraction of $f(g(a), b, h(g(b)), c)$ is $f(c_1, c_1, h(c_2), c)$, where c_1 and c_2 are new free constants.

Black-Box Combination Algorithm: Rules

E_i-GEN: **Generalization in the E_i-theory**

$$\{x : t \triangleq s\} \uplus A; S; \sigma \Longrightarrow$$
$$(P \setminus S_i) \cup A; S_i \cup S; \sigma\{x \mapsto \pi^{-1}(r)\},$$

where

- $\mathsf{root}(t) \in \Sigma_i \cup C$ and $\mathsf{root}(s) \in \Sigma_i \cup C$,
- π is the alien subterm abstraction mapping,
- $r \in \mathsf{mcsg}_{E_i}(\pi(t), \pi(s))$, and ϕ, ψ are its witnesses,
- $P = \{y : \pi^{-1}(y\phi) \triangleq \pi^{-1}(y\psi) \mid y \in \mathsf{var}(r)\}$,
- $S_i = \{y : t' \triangleq s' \mid (y : t' \triangleq s') \in P,$
  $\mathsf{root}(t') \in \Sigma_i \cup C,\ \mathsf{root}(s') \in \Sigma_i \cup C\}$.

Black-Box Combination Algorithm

$E_{1,2}$-SOL: **Solving in the combined theory**

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

if $\mathrm{root}(t) \in \Sigma_i$ and $\mathrm{root}(s) \in \Sigma_{3-i}$, $i = 1, 2$.

$E_{1,2}$-MER: **Merging in the combined theory**

$$\emptyset; \{x : t \triangleq s,\ x' : t' \triangleq s'\} \uplus S; \sigma \Longrightarrow$$
$$\emptyset; S \cup \{x : t \triangleq s\}; \sigma\{x' \mapsto x\},$$

if $t =_{E_1 \cup E_2} t'$ and $s =_{E_1 \cup E_2} s'$.

White-Box Combination Algorithm: Rules

E_i-STEP: **Step in the E_i-theory**

$$\{x : t \triangleq s\} \uplus P;\, S;\, \sigma \Longrightarrow$$
$$P \cup \pi^{-1}(P_i);\, S \cup \pi^{-1}(S_i);\, \sigma(\pi^{-1}(\sigma_i)),$$

where

- $\mathsf{root}(t) \in \Sigma_i \cup C$ and $\mathsf{root}(s) \in \Sigma_i \cup C$,
- π is the alien subterm abstraction mapping, and
- $\{x : \pi(t) \triangleq \pi(s)\}; \emptyset; Id \Longrightarrow_{\mathfrak{G}_i} P_i; S_i; \sigma_i$, where $\Longrightarrow_{\mathfrak{G}_i}$ is a step performed by the $\mathfrak{G}_i$ algorithm for theory E_i.

White-Box Combination Algorithm

$E_{1,2}$-SOL: **Solving in the combined theory**

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

if $\mathrm{root}(t) \in \Sigma_i$ and $\mathrm{root}(s) \in \Sigma_{3-i}$, $i = 1, 2$.

$E_{1,2}$-MER: **Merging in the combined theory**

$$\emptyset; \{x : t \triangleq s,\ x' : t' \triangleq s'\} \uplus S; \sigma \Longrightarrow$$
$$\emptyset; S \cup \{x : t \triangleq s\}; \sigma\{x' \mapsto x\},$$

if $t =_{E_1 \cup E_2} t'$ and $s =_{E_1 \cup E_2} s'$.

Properties

We designed a combination method for generalization in the union of signature-disjoint regular collapse-free theories, where

- free (uninterpreted) constants are allowed in generalization problems together with symbols from the component signatures,
- for each component theory (enriched with free constants), we have a procedure that computes a minimal complete set of generalizations together with their witness substitutions,
- equality is decidable in the combined theory.

Both black-box and white-box combined algorithms are complete.

If at least one of the component theories is not finitary, the white-box combination is preferred over the black-box, since the latter may generate an infinitely branching derivation tree.

Next Steps

Relaxing the disjointness restriction: shared constructors?

Replacing substitutions by a certain kind of tree grammars (finitary representation of an infinite set of generalizations) $\longrightarrow$ terminating algorithms for some non-finitary problems?

Relaxing regularity and collapse-freeness restrictions.