GENERALIZATION: A SURVEY

Temur Kutsia
RISC, Johannes Kepler University Linz

JOHANNES KEPLER
UNIVERSITY LINZ

Generalization: an abstract view

$\mathcal{O}$: a set of syntactic objects.

- Typically, expressions (e.g., terms, formulas, ...) in some formal language.

$\mathcal{M}$: a set of mappings from $\mathcal{O}$ to $\mathcal{O}$.

- Typically, variable substitutions.

Generalization: an abstract view

$\mathcal{O}$: a set of syntactic objects.

- Typically, expressions (e.g., terms, formulas, $\ldots$) in some formal language.

$\mathcal{M}$: a set of mappings from $\mathcal{O}$ to $\mathcal{O}$.

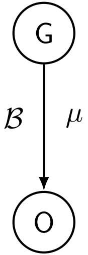
- Typically, variable substitutions.

$\mu(\mathrm{O})$ is called an instance of the object O with respect to $\mu \in \mathcal{M}$.

Generalization: an abstract view

A base relation $\mathcal{B}$ is a binary reflexive relation on $\mathcal{O}$.

An object $\mathsf{G} \in \mathcal{O}$ is a generalization of the object $\mathsf{O} \in \mathcal{O}$ with respect to $\mathcal{B}$ and $\mathcal{M}$ (briefly, $\mathcal{B}_{\mathcal{M}}$-generalization) if $\mathcal{B}(\mu(\mathsf{G}), \mathsf{O})$ holds for some mapping $\mu \in \mathcal{M}$.



Generalization: an abstract view

A preference relation $\mathcal{P}$: a binary reflexive transitive relation on $\mathcal{O}$.

$\mathcal{P}(\mathsf{O}_1, \mathsf{O}_2)$ indicates that the object O_1 is preferred over O_2.

It induces an equivalence relation $\equiv_{\mathcal{P}}$:

$$\mathsf{O}_1 \equiv_{\mathcal{P}} \mathsf{O}_2 \text{ iff } \mathcal{P}(\mathsf{O}_1, \mathsf{O}_2) \text{ and } \mathcal{P}(\mathsf{O}_2, \mathsf{O}_1).$$

Generalization: an abstract view

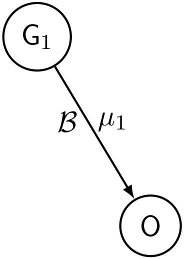
The base relation $\mathcal{B}$ and the preference relation $\mathcal{P}$ are consistent on $\mathcal{O}$ with respect to $\mathcal{M}$ or, shortly, $\mathcal{M}$-consistent, if the following holds:

- If G_1 is a $\mathcal{B}_\mathcal{M}$-generalization of O and $\mathcal{P}(\mathsf{G}_1, \mathsf{G}_2)$ holds for some G_2, then G_2 is also a $\mathcal{B}_\mathcal{M}$-generalization of O.

Generalization: an abstract view

The base relation $\mathcal{B}$ and the preference relation $\mathcal{P}$ are consistent on $\mathcal{O}$ with respect to $\mathcal{M}$ or, shortly, $\mathcal{M}$-consistent, if the following holds:

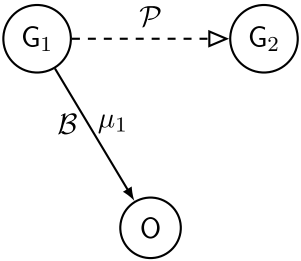
- If G_1 is a $\mathcal{B}_{\mathcal{M}}$-generalization of O and $\mathcal{P}(\mathsf{G}_1, \mathsf{G}_2)$ holds for some G_2, then G_2 is also a $\mathcal{B}_{\mathcal{M}}$-generalization of O.



Generalization: an abstract view

The base relation $\mathcal{B}$ and the preference relation $\mathcal{P}$ are consistent on $\mathcal{O}$ with respect to $\mathcal{M}$ or, shortly, $\mathcal{M}$-consistent, if the following holds:

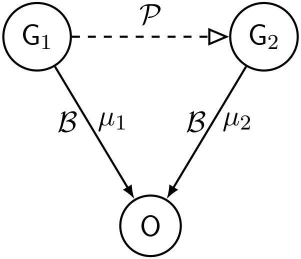
- If G_1 is a $\mathcal{B}_{\mathcal{M}}$-generalization of O and $\mathcal{P}(\mathsf{G}_1, \mathsf{G}_2)$ holds for some G_2, then G_2 is also a $\mathcal{B}_{\mathcal{M}}$-generalization of O.



Generalization: an abstract view

The base relation $\mathcal{B}$ and the preference relation $\mathcal{P}$ are consistent on $\mathcal{O}$ with respect to $\mathcal{M}$ or, shortly, $\mathcal{M}$-consistent, if the following holds:

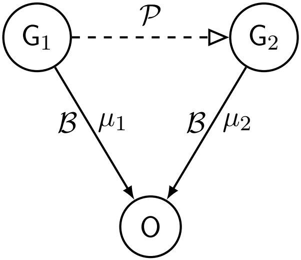
- If G_1 is a $\mathcal{B}_{\mathcal{M}}$-generalization of O and $\mathcal{P}(\mathsf{G}_1, \mathsf{G}_2)$ holds for some G_2, then G_2 is also a $\mathcal{B}_{\mathcal{M}}$-generalization of O.



Generalization: an abstract view

The base relation $\mathcal{B}$ and the preference relation $\mathcal{P}$ are consistent on $\mathcal{O}$ with respect to $\mathcal{M}$ or, shortly, $\mathcal{M}$-consistent, if the following holds:

- If G_1 is a $\mathcal{B}_{\mathcal{M}}$-generalization of O and $\mathcal{P}(\mathsf{G}_1, \mathsf{G}_2)$ holds for some G_2, then G_2 is also a $\mathcal{B}_{\mathcal{M}}$-generalization of O.



We consider only consistent base and preference relations.

Generalization: an abstract view

An object G is called a most $\mathcal{P}$-preferred common $\mathcal{B}_{\mathcal{M}}$-generalization of objects $\mathsf{O}_1, \ldots, \mathsf{O}_n$, $n \geq 2$ if

- G is a $\mathcal{B}_{\mathcal{M}}$-generalization of each O_i, and
- for any G' that is also a $\mathcal{B}_{\mathcal{M}}$-generalization of each O_i, if $\mathcal{P}(\mathsf{G}', \mathsf{G})$, then $\mathsf{G}' \equiv_{\mathcal{P}} \mathsf{G}$.
  (If G' is $\mathcal{P}$-preferred over G, then they are $\mathcal{P}$-equivalent.)

Generalization: an abstract view

$(\mathcal{B}_{\mathcal{M}}, \mathcal{P})$**-generalization problem over** $\mathcal{O}$:

Given:	Objects $\mathsf{O}_1, \ldots, \mathsf{O}_n \in \mathcal{O}$, $n \geq 2$.
Find:	An object $\mathsf{G} \in \mathcal{O}$ that is a most $\mathcal{P}$-preferred common $\mathcal{B}_{\mathcal{M}}$-generalization of $\mathsf{O}_1, \ldots, \mathsf{O}_n$.

Generalization: an abstract view

$(\mathcal{B}_{\mathcal{M}}, \mathcal{P})$**-generalization problem over** $\mathcal{O}$:

Given:	Objects $\mathsf{O}_1, \ldots, \mathsf{O}_n \in \mathcal{O}$, $n \geq 2$.
Find:	An object $\mathsf{G} \in \mathcal{O}$ that is a most $\mathcal{P}$-preferred common $\mathcal{B}_{\mathcal{M}}$-generalization of $\mathsf{O}_1, \ldots, \mathsf{O}_n$.

This problem may have zero, one, or more solutions.

Two reasons of zero solutions:

- either the objects $\mathsf{O}_1, \ldots, \mathsf{O}_n$ have no common $\mathcal{B}_{\mathcal{M}}$-generalization at all (i.e, $\mathsf{O}_1, \ldots, \mathsf{O}_n$ are not generalizable), or
- they are generalizable but have no most $\mathcal{P}$-preferred common $\mathcal{B}_{\mathcal{M}}$-generalization.

Generalization: an abstract view

To characterize “informative” sets of possible solutions, we introduce two notions: $\mathcal{P}$-complete and $\mathcal{P}$-minimal complete sets of common $\mathcal{B}_{\mathcal{M}}$-generalizations of multiple objects.

Generalization: an abstract view

A set of objects $\mathcal{G} \subseteq \mathcal{O}$ is called a $\mathcal{P}$-complete set of common $\mathcal{B}_\mathcal{M}$-generalizations of the given objects $\mathsf{O}_1, \ldots, \mathsf{O}_n$, $n \geq 2$, if the following properties are satisfied:

- **Soundness:** every $\mathsf{G} \in \mathcal{G}$ is a common $\mathcal{B}_\mathcal{M}$-generalization of $\mathsf{O}_1, \ldots, \mathsf{O}_n$, and
- **Completeness:** for each common $\mathcal{B}_\mathcal{M}$-generalization G' of $\mathsf{O}_1, \ldots, \mathsf{O}_n$ there exists $\mathsf{G} \in \mathcal{G}$ such that $\mathcal{P}(\mathsf{G}, \mathsf{G}')$.

Generalization: an abstract view

A set of objects $\mathcal{G} \subseteq \mathcal{O}$ is called a $\mathcal{P}$-complete set of common $\mathcal{B}_{\mathcal{M}}$-generalizations of the given objects $\mathsf{O}_1, \ldots, \mathsf{O}_n$, $n \geq 2$, if the following properties are satisfied:

- **Soundness:** every $\mathsf{G} \in \mathcal{G}$ is a common $\mathcal{B}_{\mathcal{M}}$-generalization of $\mathsf{O}_1, \ldots, \mathsf{O}_n$, and
- **Completeness:** for each common $\mathcal{B}_{\mathcal{M}}$-generalization G' of $\mathsf{O}_1, \ldots, \mathsf{O}_n$ there exists $\mathsf{G} \in \mathcal{G}$ such that $\mathcal{P}(\mathsf{G}, \mathsf{G}')$.

The set $\mathcal{G}$ is called $\mathcal{P}$-minimal complete set of common $\mathcal{B}_{\mathcal{M}}$-generalizations of $\mathsf{O}_1, \ldots, \mathsf{O}_n$ and is denoted by $\mathsf{mcsg}_{\mathcal{B}_{\mathcal{M}},\mathcal{P}}(\mathsf{O}_1, \ldots, \mathsf{O}_n)$ if, in addition, the following holds:

- **Minimality:** no distinct elements of $\mathcal{G}$ are $\mathcal{P}$-comparable: if $\mathsf{G}_1, \mathsf{G}_2 \in \mathcal{G}$ and $\mathcal{P}(\mathsf{G}_1, \mathsf{G}_2)$, then $\mathsf{G}_1 = \mathsf{G}_2$.

Generalization: an abstract view

The type of the $(\mathcal{B}_{\mathcal{M}}, \mathcal{P})$-generalization problem between generalizable objects $\mathsf{O}_1, \ldots, \mathsf{O}_n \in \mathcal{O}$ is

- **unitary** $(\mathbf{1})$: if $\mathsf{mcsg}_{\mathcal{B}_{\mathcal{M}},\mathcal{P}}(\mathsf{O}_1, \ldots, \mathsf{O}_n)$ is a singleton,
- **finitary** (ω): if $\mathsf{mcsg}_{\mathcal{B}_{\mathcal{M}},\mathcal{P}}(\mathsf{O}_1, \ldots, \mathsf{O}_n)$ is finite and contains at least two elements,
- **infinitary** (∞): if $\mathsf{mcsg}_{\mathcal{B}_{\mathcal{M}},\mathcal{P}}(\mathsf{O}_1, \ldots, \mathsf{O}_n)$ is infinite,
- **nullary** $(\mathbf{0})$: if $\mathsf{mcsg}_{\mathcal{B}_{\mathcal{M}},\mathcal{P}}(\mathsf{O}_1, \ldots, \mathsf{O}_n)$ does not exist (i.e., minimality and completeness contradict each other).

Generalization: an abstract view

The type of $(\mathcal{B}_{\mathcal{M}}, \mathcal{P})$-generalization over $\mathcal{O}$ is

- **unitary** $(\mathbf{1})$: if each $(\mathcal{B}_{\mathcal{M}}, \mathcal{P})$-generalization problem between generalizable objects from $\mathcal{O}$ is unitary,
- **finitary** (ω): if each $(\mathcal{B}_{\mathcal{M}}, \mathcal{P})$-generalization problem between generalizable objects from $\mathcal{O}$ is unitary or finitary, and there exists a finitary problem,
- **infinitary** (∞): if each $(\mathcal{B}_{\mathcal{M}}, \mathcal{P})$-generalization problem between generalizable objects from $\mathcal{O}$ is unitary, finitary, or infinitary, and there exists an infinitary problem,
- **nullary** $(\mathbf{0})$: if there exists a nullary $(\mathcal{B}_{\mathcal{M}}, \mathcal{P})$-generalization problem between generalizable objects from $\mathcal{O}$.

Generalization: an abstract view

Let $\mathcal{S} \subseteq \mathcal{O}$.

$\mathcal{S}$-**fragment** of the generalization problem:

- the given objects $\mathsf{O}_1, \ldots, \mathsf{O}_n$ are restricted to belong to $\mathcal{S}$:

$$\mathsf{O}_1 \in \mathcal{S}, \ldots, \mathsf{O}_n \in \mathcal{S}$$

$\mathcal{S}$-**variant** of the generalization problem:

- the desired generalizations G are restricted to belong to $\mathcal{S}$:

$$\mathsf{G} \in \mathcal{S}$$

It also makes sense to consider an $\mathcal{S}_1$-variant of an $\mathcal{S}_2$-fragment of the problem, where $\mathcal{S}_1$ and $\mathcal{S}_2$ are not necessarily the same.

Generalization: an abstract view

Interesting questions:

- **Generalization type:** What is the $(\mathcal{B}_{\mathcal{M}}, \mathcal{P})$-generalization type over $\mathcal{O}$?
- **Generalization algorithm/procedure:** How to compute (or enumerate) a complete set of generalizations (preferably, $\mathsf{mcsg}_{\mathcal{B}_{\mathcal{M}},\mathcal{P}}$) for objects from $\mathcal{O}$.

Some concrete cases: FOSG

First-order syntactic generalization:

Generic	Concrete (FOSG)
$\mathcal{O}$	First-order terms
$\mathcal{M}$	First-order substitutions
$\mathcal{B}$	$\doteq$ (syntactic equality)
$\mathcal{P}$	$\succeq$: $\quad s \succeq t$ iff $s \doteq t\sigma$ for some σ
$\equiv_{\mathcal{P}}$	Equi-generality: $\succeq$ and $\preceq$
Type	Unitary
Algorithm	[Plotkin70, Reynolds70, Huet76]

Example

$$\mathsf{mcsg}(f(a, f(a, c)), f(b, f(b, c))) = \{f(x, f(x, c))\}.$$

Some concrete cases: FOEG

First-order equational generalization modulo an equational theory E:

Generic	Concrete (FOEG)
$\mathcal{O}$	First-order terms
$\mathcal{M}$	First-order substitutions
$\mathcal{B}$	$\doteq_{\mathsf{E}}$ (equality modulo the theory E)
$\mathcal{P}$	$\succeq_{\mathsf{E}}$: $s \succeq_{\mathsf{E}} t$ iff $s \doteq_{\mathsf{E}} t\sigma$ for some σ.
$\equiv_{\mathcal{P}}$	Equi-generality modulo E: $\succeq_{\mathsf{E}}$ and $\preceq_{\mathsf{E}}$
Type	Depends on a particular E
Algorithm	Depends on a particular E

Some concrete cases: FOEG, AC

First-order equational generalization, the AC case:

Generic	Concrete (FOEG: AC)
$\mathcal{O}$	First-order terms
$\mathcal{M}$	First-order substitutions
$\mathcal{B}$	$\doteq_{\mathsf{AC}}$ (equality modulo AC)
$\mathcal{P}$	$\succeq_{\mathsf{AC}}$: $\quad s \succeq_{\mathsf{AC}} t$ iff $s \doteq_{\mathsf{AC}} t\sigma$ for some σ.
$\equiv_{\mathcal{P}}$	Equi-generality modulo AC: $\succeq_{\mathsf{AC}}$ and $\preceq_{\mathsf{AC}}$
Type	Finitary
Algorithm	[Alpuente et al, 2014]

Example

If f is an AC symbol, then

$$\mathsf{mcsg}(f(f(a,a),b),\ f(f(b,b),a)) = \{f(f(x,x),y),\ f(f(x,a),b)\}.$$

Some concrete cases: FOEG, Abs

First-order equational generalization, the absorption case.
Axioms: $f(x, e) = e, \quad f(e, x) = e$.

Generic	Concrete (FOEG: Abs)
$\mathcal{O}$	First-order terms
$\mathcal{M}$	First-order substitutions
$\mathcal{B}$	$\doteq_{\mathsf{Abs}}$ (equality modulo Abs)
$\mathcal{P}$	$\succeq_{\mathsf{Abs}}$: $s \succeq_{\mathsf{Abs}} t$ iff $s \doteq_{\mathsf{Abs}} t\sigma$ for some σ.
$\equiv_{\mathcal{P}}$	Equi-generality modulo Abs: $\succeq_{\mathsf{Abs}}$ and $\preceq_{\mathsf{Abs}}$
Type	Infinitary
Algorithm	Andres Gonzalez et al, ongoing work

Some concrete cases: FOEG, GSC

First-order equational generalization for a ground subterm-collapsing theory, axiomatized with two equalities $f(a) = a, \;\; f(b) = b$.

Generic	Concrete (FOEG: GSC)
$\mathcal{O}$	First-order terms
$\mathcal{M}$	First-order substitutions
$\mathcal{B}$	$\doteq_{\mathsf{GSC}}$ (equality modulo GSC)
$\mathcal{P}$	$\succeq_{\mathsf{GCS}}$: $\;\; s \succeq_{\mathsf{GSC}} t$ iff $s \doteq_{\mathsf{GSC}} t\sigma$ for some σ.
$\equiv_{\mathcal{P}}$	Equi-generality modulo GSC: $\succeq_{\mathsf{GSC}}$ and $\preceq_{\mathsf{GSC}}$
Type	Nullary
Algorithm	TBD

Example

The problem $a \triangleq^{?}_{\mathsf{GSC}} b$ has no mcsg: the complete set of generalizations contains an infinite chain $x \preceq_{\mathsf{GCS}} f(x) \preceq_{\mathsf{GCS}} f(f(x)) \cdots$.

Summary for some FOEG theory types

- A, C, AC: finitary [Alpuente et al, 2014]
- $U^{>1}$, $(ACU)^{>1}$, $(CU)^{>1}$, $(AU)^{>1}$, (AU)(CU): nullary
  Their single-symbol versions as well as linear variants are finitary [Cerna&Kutsia, FSCD'20];
- I, AI, CI: infinitary [Cerna&Kutsia, TOCL, 2020];
- $(UI)^{>1}$, $(AUI)^{>1}$, $(CUI)^{>1}$, $(ACUI)^{>1}$, semirings: nullary [Cerna 2020];
- Commutative theories: unitary [Baader 1991].

Some concrete cases: FOVG

First-order variadic generalization:

Generic	Concrete (FOVG)
$\mathcal{O}$	Variadic terms and their sequences
$\mathcal{M}$	Substitutions (for terms and for sequences)
$\mathcal{B}$	$\doteq$ (syntactic equality)
$\mathcal{P}$	$\succeq$: $s \succeq t$ iff $s \doteq t\sigma$ for some σ.
$\equiv_{\mathcal{P}}$	Equi-generality: $\succeq$ and $\preceq$
Type	Finitary (also for the rigid variant)
Algorithm	[Kutsia et al, 2014]

Example

mcsg$(g(f(a), f(a)),\ g(f(a), f))$ for the unrestricted case is

$$\{g(f(a), f(X)),\ \ g(f(X,Y), f(X)),\ \ g(f(X,Y), f(Y))\}.$$

For the rigid variant, it is $\{g(f(a), f(X))\}$.

Some concrete cases: FOCG

First-order clausal generalization.

Generic	Concrete (FOCG)
$\mathcal{O}$	First-order clauses
$\mathcal{M}$	First-order substitutions
$\mathcal{B}$	$\subseteq$
$\mathcal{P}$	$\succeq_{\text{Cl}}$: $s \succeq_{\text{Cl}} t$ iff $s \supseteq t\sigma$ for some σ. (t σ-subsumes s)
$\equiv_{\mathcal{P}}$	Equi-generality modulo Cl: $\succeq_{\text{Cl}}$ and $\preceq_{\text{Cl}}$
Type	Unitary
Algorithm	[Plotkin, 1970]

Example

Let $C_1 := p(a) \leftarrow q(a), q(b)$ $\quad C_2 := p(b) \leftarrow q(b), q(x)$

$G_1 := p(y) \leftarrow q(y), q(b)$ $\quad G_2 := p(y) \leftarrow q(y), q(b), q(z)$

Then G_1 and G_2 both are lggs of C_1 and C_2, and $G_1 \equiv_{\mathcal{P}} G_2$.

Some concrete cases: HOG$_{\alpha\beta\eta}$

Higher-order $\alpha\beta\eta$-generalization

Generic	Concrete (HOG$_{\alpha\beta\eta}$)
$\mathcal{O}$	Simply-typed λ terms
$\mathcal{M}$	Higher-order substitutions
$\mathcal{B}$	$\approx$ (equality modulo $\alpha\beta\eta$)
$\mathcal{P}$	$\succsim$: $s \succsim t$ iff $s \approx t\sigma$ for a substitution σ.
$\equiv_{\mathcal{P}}$	Equi-general ($\succsim$ and $\precsim$) modulo $\alpha\beta\eta$
Type	nullary in general [Buran&Cerna, to appear] unitary for the TMS variant [Cerna&Kutsia, 2019]
Algorithm	TMS variant [Cerna&Kutsia, 2019], patterns [Baumgartner et al, 2017]

Some concrete cases: HOG$_{\alpha\beta\eta}$

Example

Various top-maximal shallow lggs for

$\lambda x.\, f(h(g(g(x))), h(g(x)), a)$ and $\lambda x.f(g(g(x)), g(x), h(a))$

Projection-based:

$\lambda x.f(X(h(g(g(x))), g(g(x))), X(h(g(x)), g(x)), X(a, h(a))),$

Common subterms:

$\lambda x.f(X(g(g(x))), X(g(x))), Z(a)),$

Patterns:

$\lambda x.f(X(x), Y(x), Z).$

Some concrete cases: DLs

Description logics.

Decidable fragments of first-order logic.

The basic syntactic building blocks in DLs:

- (primitive) concept names $P, Q, \ldots$ (unary predicates),
- role names $r, q, \ldots$ (binary predicates),
- individual names $a, b, \ldots$ (constants).

Starting from these constructions, complex concept descriptions and roles are built using constructors, which determine the expressive power of the DL.

$$\begin{aligned} \mathcal{EL}: &\quad C, D := P \mid \top \mid C \sqcap D \mid \exists r.C. \\ \mathcal{FLE}: &\quad C, D := P \mid \top \mid C \sqcap D \mid \exists r.C \mid \forall r.C. \\ \ldots & \end{aligned}$$

Some concrete cases: DLs

An interpretation $\mathcal{I} = (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of

- a non-empty set $\Delta^{\mathcal{I}}$, called the interpretation domain, and
- a mapping $\cdot^{\mathcal{I}}$, called the extension mapping.

Some concrete cases: DLs

An interpretation $\mathcal{I} = (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of

- a non-empty set $\Delta^{\mathcal{I}}$, called the interpretation domain, and
- a mapping $\cdot^{\mathcal{I}}$, called the extension mapping.

The mapping maps

- every concept name P to a set $P^{\mathcal{I}} \subseteq \Delta_{\mathcal{I}}$,
- every role name r to a binary relation $r^{\mathcal{I}} \subseteq \Delta_{\mathcal{I}} \times \Delta_{\mathcal{I}}$.

Some concrete cases: DLs

An interpretation $\mathcal{I} = (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of

- a non-empty set $\Delta^{\mathcal{I}}$, called the interpretation domain, and
- a mapping $\cdot^{\mathcal{I}}$, called the extension mapping.

The mapping maps

- every concept name P to a set $P^{\mathcal{I}} \subseteq \Delta_{\mathcal{I}}$,
- every role name r to a binary relation $r^{\mathcal{I}} \subseteq \Delta_{\mathcal{I}} \times \Delta_{\mathcal{I}}$.

For the other concept descriptions:

- $\top^{\mathcal{I}} = \Delta_{\mathcal{I}}$,
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
- $(\exists r.C)^{\mathcal{I}} = \{d \in \Delta_{\mathcal{I}} \mid \exists e.\, (d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$,
- $(\forall r.C)^{\mathcal{I}} = \{d \in \Delta_{\mathcal{I}} \mid \forall e.\, (d, e) \in r^{\mathcal{I}} \Rightarrow e \in C^{\mathcal{I}}\}$.

Some concrete cases: DLs

A concept description C is subsumed by D, written $C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all interpretations $\mathcal{I}$.

$C \equiv D$: if C and D subsume each other.

A concept description D is called a least common subsumer of C_1 and C_2, if

- $C_1 \sqsubseteq D$ and $C_2 \sqsubseteq D$ and
- if there exists D' such that $C_1 \sqsubseteq D'$ and $C_2 \sqsubseteq D'$, then $D \sqsubseteq D'$.

Some concrete cases: DLs

A concept description C is subsumed by D, written $C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all interpretations $\mathcal{I}$.

$C \equiv D$: if C and D subsume each other.

A concept description D is called a least common subsumer of C_1 and C_2, if

- $C_1 \sqsubseteq D$ and $C_2 \sqsubseteq D$ and
- if there exists D' such that $C_1 \sqsubseteq D'$ and $C_2 \sqsubseteq D'$, then $D \sqsubseteq D'$.

The problem of computing the least common subsumer of two or more concept descriptions is a version of the problem of computing generalizations in DLs.

Some concrete cases: DLs

DLs $\mathcal{EL}$ and $\mathcal{FLE}$:

Generic	Concrete (DL)
$\mathcal{O}$	Concept descriptions
$\mathcal{M}$	Contains only the identity mapping
$\mathcal{B}$	$\sqsupseteq$
$\mathcal{P}$	$\sqsubseteq$
$\equiv_{\mathcal{P}}$	$\equiv$: $\sqsubseteq$ and $\sqsupseteq$
Type	Unitary
Algorithm	[Baader et al, 1999]

Example ($\mathcal{EL}$)

$$C = P \sqcap \exists r.(\exists r.(P \sqcap Q) \sqcap \exists s.Q) \sqcap \exists r.(P \sqcap \exists s.P)$$

$$D = \exists r.(P \sqcap \exists r.P \sqcap \exists s.Q)$$

$$LCS(C, D) = \exists r.(\exists r.P \sqcap \exists s.Q) \sqcap \exists r.(P \sqcap \exists s.\top)$$

Some concrete cases: ProxGen

Quantitative generalization modulo fuzzy proximity relations:

Generic	Concrete (ProxGen)
$\mathcal{O}$	First-order terms
$\mathcal{M}$	First-order substitutions
$\mathcal{B}$	$\approx_{\mathcal{R},\lambda}$ (approximate equality)
$\mathcal{P}$	$\succeq$: $\quad s \succeq t$ iff $s \doteq t\sigma$ for some σ.
$\equiv_{\mathcal{P}}$	Equi-generality: $\succeq$ and $\preceq$
Type	Finitary
Algorithm	[Kutsia&Pau, 2022]

Some concrete cases: ProxGen

If we defined $\mathcal{P}$ as $\succsim_{\mathcal{R},\lambda}$ where $s \succsim_{\mathcal{R},\lambda} t$ iff $s \approx_{\mathcal{R},\lambda} t\sigma$ for some σ, then it would not be consistent with $\mathcal{B}$.

If $\mathcal{R}(a, b) = 0.7$ and $\mathcal{R}(b, c) = 0.7$, then both a and b are $(\mathcal{R}, 0.7)$-generalizations of a and b, but c is not.

But taking $\mathcal{P} = \succsim_{\mathcal{R},\lambda}$, we would get that c is also a $(\mathcal{R}, 0.7)$-generalization of a and b, which is wrong.

Some more concrete cases

Clausal generalization:

- based on relative θ-subsumption,
- based on T-implication.

Order-sorted generalization:

- syntactic,
- modulo equational theories.

Variadic generalization:

- for commutative (orderless) theories,
- for term-graphs.

Generalization in the description logic $\mathcal{EL}$:

- an approach that allows variables in the generalization

Some more concrete cases

Nominal generalization:

- allowing finitely many atoms
- using atom variables
- …

Higher-order generalization:

- simple types, modulo $\alpha\beta\eta$ and equational theories,
- polymorphic lambda-calculus ($\lambda 2$),
- second order variadic terms,
- …

Applications

Typical applications fall into one of the following areas:

- learning and reasoning,
- synthesis and exploration,
- analysis and repair.

Future directions

- Studying the influence of the signature of equational theories on the generalization type
- Investigating methods of combining generalization algorithms over disjoint equational theories
- Characterization of equational theories exhibiting similar behavior and properties for generalization problems

Future directions

- Studying the influence of the signature of equational theories on the generalization type
- Investigating methods of combining generalization algorithms over disjoint equational theories
- Characterization of equational theories exhibiting similar behavior and properties for generalization problems
- Studying generalization in more expressive theories (higher-order, quantitative, . . .)

Future directions

- Studying the influence of the signature of equational theories on the generalization type
- Investigating methods of combining generalization algorithms over disjoint equational theories
- Characterization of equational theories exhibiting similar behavior and properties for generalization problems
- Studying generalization in more expressive theories (higher-order, quantitative, . . .)
- Studying the influence of the preference relation choice on the type and solution set of generalization problems

Future directions

- Studying the influence of the signature of equational theories on the generalization type
- Investigating methods of combining generalization algorithms over disjoint equational theories
- Characterization of equational theories exhibiting similar behavior and properties for generalization problems
- Studying generalization in more expressive theories (higher-order, quantitative, ...)
- Studying the influence of the preference relation choice on the type and solution set of generalization problems
- Combination with other kind of generalization and abstraction techniques + new applications

Reference

David Cerna and Temur Kutsia.
Anti-unification and Generalization: a Survey.
In: E. Elkind, editor. *Proceedings of IJCAI 2023 -
32nd International Joint Conference on Artificial Intelligence.*
ijcai.org, 2023. 6563–6573