An Axiomatic Approach to Feature Term Generalization

Hassan Aït-Kaci[1] and Yutaka Sasaki[2]

[1] Simon Fraser University
Burnaby, BC, Canada V5A 1S6, Canada
hak@sfu.ca
[2] NTT Communication Science Laboratories
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan
sasaki@cslab.kecl.ntt.co.jp

Abstract. This paper presents a missing link between Plotkin's least general generalization formalism and generalization on the *Order Sorted Feature (OSF)* foundation. A Feature Term (or ψ-term) is an extended logic term based on *ordered sorts* and is a normal form of an OSF-term. An *axiomatic* definition of ψ-term generalization is given as a set of OSF clause generalization rules and the least generality of the axiomatic definition is proven in the sense of Plotkin's least general generalization (lgg). The correctness of the definition is given on the basis of the axiomatic foundation. An operational definition of the least general generalization of clauses based on ψ-terms is also shown that it is a realization of the axiomatic definition.

1 Introduction

A Feature Term (or ψ*-term*) is an *order-sorted logic term*, *i.e.*, an extended form of a logic term where functor symbols are *ordered sorts*. In addition, *features* (or attribute labels) are added to a sort as argument indicators. [1–3].

For example, the following two ψ-terms describe George and his mother as having the same last name "Bush" and Al and his mother as having the same last name "Gore":

$$\begin{aligned} &George(\ last \quad\ \Rightarrow Y_1 : Bush,\\ &\qquad\qquad mother \Rightarrow Barbara(last \Rightarrow Y_1)),\end{aligned}$$

$$\begin{aligned} &Al(\ last \quad\ \Rightarrow Y_2 : Gore,\\ &\qquad mother \Rightarrow Pauline(last \Rightarrow Y_2)).\end{aligned}$$

In this example, *George*, *Barbara*, *Bush*, *Al*, *Pauline*, and *Gore* are sort symbols, while *last* and *mother* are feature symbols. The variables Y_1 and Y_2 link George's and Al's last names to their mother's last names.

A goal of a ψ*-term generalization* is to calculate (or *induce*) the following generic knowledge from the two examples:

$$\begin{aligned} &person(\ last \quad\ \Rightarrow Y : name,\\ &\qquad\qquad mother \Rightarrow person(last \Rightarrow Y))\end{aligned}$$

Formally, without syntax sugaring, the schema is represented as

$$
\begin{aligned}
X : person(\ & last && \Rightarrow Y : name, \\
& mother && \Rightarrow Z : person(last \Rightarrow Y : \top))
\end{aligned}
$$

where $\top$ is the *universal*—*i.e.*, the most general—sort, and variable tags are systematically used for each sort.

The ψ-term is useful in the fields of Artificial Intelligence (AI) and Natural Language Processing (NLP). For instance, the feature term [4], equivalent to the ψ-term, is used to represent the syntax and semantics of natural language sentences. In case-based reasoning, feature terms are used as the data structure of cases [9] and the generalization of cases is a key to the reuse of cases. Inductive Logic Programming (ILP) is extended to an induction (*i.e.* generalization) of logic programs based on ψ-terms [11, 12].

While feature terms play an essential role in AI and NLP, there is a missing link between Plotkin's least general generalization formalism of classic logic terms and generalization of ψ-terms on the basis of the OSF foundation. This paper presents the missing link.

2 Preliminaries on ψ-terms

This section introduces ψ-terms on the basis of the *Order-Sorted Feature (OSF)* formalism [2, 3].

2.1 Syntax

Definition 1 (OSF Signature) *An OSF Signature is given by*

$$\Sigma_{OSF} = \langle \mathcal{S}, \preceq, \sqcap, \sqcup, \mathcal{F} \rangle, \ \ s.t. :$$

- *$\mathcal{S}$ is a set of sort symbols with the sorts $\top$ and $\bot$;*
- *$\preceq$ is a partial order on $\mathcal{S}$ such that $\top$ is the greatest element and $\bot$ is the least element;*
- *$\langle \mathcal{S}, \preceq, \sqcap, \sqcup \rangle$ is a lattice, where $s \sqcap t$ is defined as the infimum (or glb) of s and t and $s \sqcup t$ is the supremum (or lub) of sorts s and t;*
- *$\mathcal{F}$ is a set of feature symbols.*

For sorts $s_1, s_2 \in \mathcal{S}$, we denote $s_1 \prec s_2$ iff $s_1 \preceq s_2$ and $s_1 \neq s_2$.

Let $\mathcal{V}$ be a countable infinite set of variables.

Definition 2 (OSF-terms) *Given $\Sigma_{OSF} = \langle \mathcal{S}, \preceq, \sqcap, \sqcup, \mathcal{F} \rangle$, if $s \in \mathcal{S}$, $l_1, \ldots, l_n \in \mathcal{F}$, $X \in \mathcal{V}$, $n \geq 0$, and $t_1, ..., t_n$ are OSF-terms, then an OSF-term has the form*

$$X : s(l_1 \Rightarrow t_1, ..., l_n \Rightarrow t_n).$$

Let $\psi = X : s(l_1 \Rightarrow t_1, ..., l_n \Rightarrow t_n)$. X is called the root variable of ψ, which is described as $Root(\psi)$, and s is called the root sort of ψ, which is described as $Sort(\psi)$.

For a lighter notation, hereafter we omit variables that are not shared and the sort of a variable when it is $\top$.

Definition 3 (ψ-terms) *An OSF-term*

$$\psi = X : s(l_1 \Rightarrow \psi_1, \ldots, l_n \Rightarrow \psi_n)$$

is in a normal form (and is called a ψ-term) if:

- *For any variables V_i in ψ, V_i is the root variable of at most one non-top ψ-term,* i.e., *one whose root sort is not $\top$;*
- *s is a nonbottom sort in $\mathcal{S}$;*
- *$l_1, \ldots, l_n$ are pairwise distinct feature symbols in $\mathcal{F}$;*
- *$\psi_1, \ldots, \psi_n$ are ψ-terms.*

We will see that OSF-terms can be normalized to ψ-terms by OSF clause normalization rules, which are given in Section 2.3, or thus proven to be inconsistent by being reduced to $\perp$.

Let $\psi = X : s(l_1 \Rightarrow \psi_1, ..., l_n \Rightarrow \psi_n)$. $s(l_1 \Rightarrow \psi_1, ..., l_n \Rightarrow \psi_n)$ is called an *untagged ψ-term*.

Definition 4 (Feature Projection) *Given a ψ-term $t = X : s(l_1 \Rightarrow t_1, \ldots, l_n \Rightarrow t_n)$, the l_i projection of t (written as $t.l_i$) is defined as $t.l_i = t_i$.*

The definitions of atoms, literals, clauses, Horn clauses, and definite clauses are as usual with the difference being that terms are ψ-terms. If features are non-zero integers $1, \ldots, n$, then a ψ-term $X : s(1 \Rightarrow t_1, 2 \Rightarrow t_2, \ldots, n \Rightarrow t_n)$ can be abbreviated to $X : s(t_1, t_2, \ldots, t_n)$.

2.2 Semantics

Definition 5 (OSF Algebras) *An OSF Algebra is a structure $\mathcal{A} = \langle D^{\mathcal{A}}, (s^{\mathcal{A}})_{s\in\mathcal{S}}, (l^{\mathcal{A}})_{l\in\mathcal{F}} \rangle$ s.t.:*

- *$D^{\mathcal{A}}$ is a non-empty set, called a domain of $\mathcal{A}$;*
- *for each sort symbol $s \in \mathcal{S}$, $s^{\mathcal{A}} \subseteq D^{\mathcal{A}}$; in particular, $\top^{\mathcal{A}} = D^{\mathcal{A}}$ and $\perp^{\mathcal{A}} = \emptyset$;*
- *$(s \sqcap s')^{\mathcal{A}} = s^{\mathcal{A}} \cap s'^{\mathcal{A}}$ for two sorts $s, s' \in \mathcal{S}$;*
- *$(s \sqcup s')^{\mathcal{A}} = s^{\mathcal{A}} \cup s'^{\mathcal{A}}$ for two sorts $s, s' \in \mathcal{S}$;*
- *for each feature symbol $l \in \mathcal{F}$, $l^{\mathcal{A}} : D^{\mathcal{A}} \rightarrow D^{\mathcal{A}}$.*

Definition 6 ($\mathcal{A}$-Valuation) *Given $\Sigma_{OSF} = \langle \mathcal{S}, \preceq, \sqcap, \sqcup, \mathcal{F} \rangle$, an $\mathcal{A}$-valuation is a function $\alpha : \mathcal{V} \rightarrow D^{\mathcal{A}}$.*

Definition 7 (Term Denotation) *Let t be a ψ-term of the form*

$$t = X : s(l_1 \Rightarrow t_1, \ldots, l_n \Rightarrow t_n).$$

Given an OSF Algebra $\mathcal{A}$ and an $\mathcal{A}$-valuation α, the term denotation of t is given by

$$[\![t]\!]^{\mathcal{A},\alpha} = \{\alpha(X)\} \cap s^{\mathcal{A}} \cap \bigcap_{1 \leq i \leq n} (l_i^{\mathcal{A}})^{-1}([\![t_i]\!]^{\mathcal{A},\alpha}).$$

$$[\![t]\!]^{\mathcal{A}} = \bigcup_{\alpha:\mathcal{V} \rightarrow D^{\mathcal{A}}} [\![t]\!]^{\mathcal{A},\alpha}.$$

2.3 Unification of ψ-terms

An alternative syntactic presentation of the information conveyed by OSF-terms can be translated into a constraint clause [2].

Definition 8 (OSF-Constraints) *An order-sorted feature constraint (OSF-constraint) is one of the following forms:*

- $X : s$
- $X \doteq Y$
- $X.l \doteq Y$

where X and Y are variables in $\mathcal{V}$, s is a sort in $\mathcal{S}$, and l is a feature in $\mathcal{F}$.

Definition 9 (OSF-clauses) *An OSF-clause ϕ_1 & $\ldots$ & ϕ_n is a finite, possibly empty conjunction of OSF-constraints $\phi_1, \ldots, \phi_n (n \geq 0)$.* [1]

We can associate an OSF-term with a corresponding OSF-clause.
Let ψ be a ψ-term of the form

$$\psi = X : s(l_1 \Rightarrow \psi_1, ..., l_n \Rightarrow \psi_n).$$

An OSF-clause $\phi(\psi)$ corresponding to an OSF-term ψ has the following form:

$$\begin{aligned}\phi(\psi) = X : s \;\&\; X.l_1 \doteq X'_1 \;\&\; \ldots \;\&\; X.l_n \doteq X'_n \\ \&\; \phi(\psi_1) \qquad \&\; \ldots \;\&\; \phi(\psi_n),\end{aligned}$$

where $X, X'_1, \ldots, X'_n$ are the root variables of $\psi, \psi_1, \ldots, \psi_n$, respectively. We say $\phi(\psi)$ is dissolved from the OSF-term ψ.

Example 1 *Let $\psi = X : s(l_1 \Rightarrow Y : t, l_2 \Rightarrow Y : \top))$. The OSF-clause of ψ is:*
$\phi(\psi) = X : s \;\; \& \; X.l_1 \doteq Y \; \& \; Y : t \;\; \& \; X.l_2 \doteq Y \; \& \; Y : \top$

[1] We sometimes regard an OSF-clause as a set of OSF constraints.

Sort Intersection:

$$\frac{\phi \ \& \ X : s \ \& \ X : s'}{\phi \ \& \ X : s \sqcap s'}$$

Inconsistent Sort:

$$\frac{\phi \ \& \ X : \bot}{X : \bot}$$

Variable Elimination:

$$\frac{\phi \ \& \ X \doteq X'}{\phi[X/X'] \ \& \ X \doteq X'}$$

$$if \ X \neq X' \ and \ X \in Var(\phi)$$

Feature Decomposition:

$$\frac{\phi \ \& \ X.l \doteq X' \ \& \ X.l \doteq X''}{\phi \ \& \ X.l \doteq X' \ \& \ X' \doteq X''}$$

Fig. 1. OSF Clause Normalization Rules

On the other hand, an OSF-clause ϕ can be converted to an OSF-term $\psi(\phi)$ as follows: first complete it by adding as many $V{:}\top$ constraints as needed so that there is exactly one sort constraint for every occurrence of a variable V in an $X.l{=}V$ constraint, where X is a variable and l is a feature symbol; then convert it by the following ψ transform:

$$\psi(\phi) = X : s(l_1 \Rightarrow \psi(\phi(Y_1)), \ldots, l_n \Rightarrow \psi(\phi(Y_n)))$$

where X is a root variable of ϕ, ϕ contains $X : s$, and $X.l_1 \doteq Y_1, \ldots, X.l_n \doteq Y_n$ are all of the other constraints in ϕ with an occurrence of variable X on the left-hand side. $\phi(Y)$ denotes the maximal subclause of ϕ rooted by Y.

Definition 10 (Solved OSF-Constraint) *An OSF-clause ϕ is called solved if for every variable X, ϕ contains:*

- *at most one sort constraint of the form $X : s$, with $\bot \prec s$;*
- *at most one feature constraint of the form $X.l \doteq Y$ for each $X.l$;*
- *no equality constraint of the form $X \doteq Y$.*

Given ϕ in a normal form, we will refer to its part in a solved form as $Solved(\phi)$.

Example 2 *Let $\phi = X : s \ \& \ X.l_1 \doteq Y \ \& \ Y : t \ \& \ X.l_2 \doteq Y \ \& \ Y : \top$. The solved normal form of ϕ is :*
$Solved(\phi) = X : s \ \& \ X.l_1 \doteq Y \ \& \ Y : t \ \& \ X.l_2 \doteq Y$.

Theorem 1 *[2] The rules of Fig. 1 are solution-preserving, finite-terminating, and confluent (modulo variable renaming). Furthermore, they always result in a normal form that is either an inconsistent OSF clause or an OSF clause in a solved form together with a conjunction of equality constraints.*

Note that $Var(\phi)$ is the set of variables occurring in an OSF-clause ϕ and $\phi[X/Y]$ stands for the OSF-clause obtained from ϕ after replacing all occurrences of Y by X.

Sort Induction (SI):

$$\frac{\{X_1\backslash X\} \cup \Gamma_1, \{X_2\backslash X\} \cup \Gamma_2 \vdash \phi \ \& \ ((X_1 : s_1 \ \& \ \phi_1) \ \vee \ (X_2 : s_2 \ \& \ \phi_2))}{\{X_1\backslash X\} \cup \Gamma_1, \{X_2\backslash X\} \cup \Gamma_2 \vdash \phi \ \& \ (X : s_1 \sqcup s_2) \ \& \ ((X_1 : s \ \& \ \phi_1) \ \vee \ (X_2 : s_2 \ \& \ \phi_2))}$$

$$if \quad \neg \exists s \ (X : s \in \phi)$$

Feature Induction (FI):

$$\frac{\{X_1\backslash X\} \cup \Gamma_1, \{X_2\backslash X\} \cup \Gamma_2 \vdash \phi \ \& \ ((X_1.l \doteq Y_1 \ \& \ \phi_1) \ \vee \ (X_2.l \doteq Y_2 \ \& \ \phi_2))}{\{X_1\backslash X, Y_1\backslash Y\} \cup \Gamma_1, \{X_2\backslash X, Y_2\backslash Y\} \cup \Gamma_2 \vdash \phi \ \& \ X.l \doteq Y \ \& \ ((X_1.l \doteq Y_1 \ \& \ \phi_1) \ \vee \ (X_2.l \doteq Y_2 \ \& \ \phi_2))}$$

$$if \ \neg \exists \ y \ (Y_1\backslash y \in \{X_1\backslash X\} \cup \Gamma_1 \ and \ Y_2\backslash y \in \{X_2\backslash X\} \cup \Gamma_2)$$

Coreference Induction (CI):

$$\frac{\{X_1\backslash X, Y_1\backslash Y\} \cup \Gamma_1, \{X_2\backslash X, Y_2\backslash Y\} \cup \Gamma_2 \vdash \phi \ \& \ ((X_1.l \doteq Y_1 \ \& \ \phi_1) \ \vee \ (X_2.l \doteq Y_2 \ \& \ \phi_2))}{\{X_1\backslash X, Y_1\backslash Y\} \cup \Gamma_1, \{X_2\backslash X, Y_2\backslash Y\} \cup \Gamma_2 \vdash \phi \ \& \ X.l \doteq Y \ \& \ ((X_1.l \doteq Y_1 \ \& \ \phi_1) \ \vee \ (X_2.l \doteq Y_2 \ \& \ \phi_2))}$$

$$if \quad X.l \doteq Y \notin \phi$$

Fig. 2. OSF Clause Generalization Rules

Theorem 2 (ψ-term Unification) *[2] Let ψ_1 and ψ_2 be two ψ-terms. Let ϕ be the normal form of the OSF-clause $\phi(\psi_1)$ & $\phi(\psi_2)$ & $X_1 \doteq X_2$, where X_1 and X_2 are the root variables of ψ_1 and ψ_2, respectively. Then,*

ϕ is an inconsistent clause iff their glb is $\perp$. If ϕ is not an inconsistent clause, then their glb $\psi_1 \sqcap \psi_2$ is given by the normal OSF-term $\psi(Solved(\phi))$.

3 Axiomatic ψ-term Generalization

As a dual of ψ-term unification, ψ-term generalization (or anti-unification) can be defined as OSF clause generalization rules.

To define generalization, we introduce a new constraint symbols $A \vee B$, which means the generalization of two OSF clauses A and B.

A ψ-term generalization rule is of the form:

$$\frac{\Gamma_1, \Gamma_2 \vdash \phi \ \& \ (\phi_1 \ \vee \ \phi_2)}{\Gamma_1', \Gamma_2' \vdash \phi' \ \& \ (\phi_1 \ \vee \ \phi_2)}$$

where Γ_1 and Γ_2 are sets of variable substitutions of the form $\{X_1\backslash X_1', \ldots, X_n\backslash X_n'\}$[2] , ϕ and ϕ' are OSF-clauses, and ϕ_1 and ϕ_2 are solved normal form of OSF-clauses of target ψ-terms ψ_1 and ψ_2, respectively.

Definition 11 (Axiomatic Generalization) *Let ϕ_1 and ϕ_2 be a solved normal from of ψ_1 and ψ_2, respectively, and Γ_1 and Γ_2 be variable substitutions. Then, a generalized OSF-clause ϕ of*

$$\Gamma_1, \Gamma_2 \vdash \phi \ \& \ (\phi_1 \vee \phi_2)$$

[2] This means that X_i' is substituted by X_i.

is obtained by applying OSF clause generalization rules (Fig. 2) until no rule is applicable, initiated with

$$\{X_1 \backslash X\}, \{X_2 \backslash X\} \vdash (\phi_1 \vee \phi_2)$$

where $X_1 = Root(\psi_2)$*,* $X_2 = Root(\psi_2)$*, and* X *is a fresh variable.*

A generalized ψ-term is given as $\psi(\phi)$.

Proposition 1 *The result of the axiomatic generalization is an OSF-clause in the normal form.*

Proposition 2 *The OSF clause generalization is finite terminating.*

Proof. Termination follows from the fact that the number of variables in ϕ_1 is finite because OSF-clauses are finite, and each of the three rules SI, FI, and CI strictly decreases the number of combinations of variables in ϕ_1 and ϕ_2 that satisfy the preconditions of the OSF generalization rules.

From the definition of $\phi(\cdot)$ and OSF clause normalization rules (Fig. 1), the number of variables in ϕ_i is finite since ψ_i is finite by the definition of OSF-terms. The Sort Induction (SI) strictly decreases the number of variable pairs that satisfy the conditions of the generalization rules. That is, the variable pair X_1 of $X_1 : s_1$ and X_2 of $X_2 : s_2$ does not satisfy the precondition of SI after its application. The Feature Induction (FI) strictly decreases the number of pairs of variable pairs that satisfy the rule conditions. The pair of variable pairs $\langle X_1, Y_1\rangle$ and $\langle X_2, Y_2\rangle$ does not satisfy the precondition of FI after its application. Since FI is only applicable a finite number of time, FI increases the finite number of pairs applicable to the SI and CI rules. Same as FI, the Coreference Induction (CI) strictly decreases the number of pairs of variable pairs that satisfy the rule conditions. The pair of variable pairs $\langle X_1, Y_1\rangle$ and $\langle X_2, Y_2\rangle$ does not satisfy the precondition of CI after its application.

3.1 Least General Generalization

This section newly introduces the *least general generalization of* ψ*-terms* along the line with Plotkin's *least general generalization (lgg)* [10].

Definition 12 (Sorted Substitution) *A sorted substitution has the form* $\{X_1{:}s_1/Y_1{:}t_1, \ldots, X_n{:}s_n/Y_n{:}t_n\}$*, where* $X_1, \ldots, X_n$ *are pairwise distinct variables and* $Y_1, \ldots, Y_n$ *are variables in* $\mathcal{V}$*,* $s_1, \ldots, s_n$ *and* $t_1, \ldots, t_n$ *are sort symbols with* $\bot \prec s_i \preceq t_i$ *for every* i*. If expression* E *is a term, a literal, or a clause,* $E\theta$ *is the result of replacing all occurrences of* $Y_i{:}t_i$ *by* $X_i{:}s_i$ *and* Y_i *by* X_i *simultaneously for every* i*.*

Note that the sorted substitution changes only variable names and sorts; it does not add or remove constraint of the form $X.l \doteq Y$. This means that the sorted substitution preserves the structure of an original expression.

Definition 13 (Sorted Ordering of ψ-terms) *Let ψ_1 and ψ_2 be two ψ-terms. Let ϕ_1 and ϕ_2 be a solved normal form of OSF clauses of ψ_1 and ψ_2, respectively. $\psi_1 \leq \psi_2$ iff there exists a sorted substitution θ such that $\phi_1\theta \subseteq \phi_2$* [3] *and $(Root(\psi_2) : Sort(\psi_2)/Root(\psi_1) : Sort(\psi_1)) \in \theta$.*

We read $\psi_1 \leq \psi_2$ as meaning that ψ_1 *is more general than* ψ_2.

Example 3 *$(X{:}s) \leq Y{:}t(l \Rightarrow Z{:}u)$ with $t \preceq s$ because for $\theta=\{Y{:}t/X{:}s\}$, $(X{:}s)\theta = \{Y{:}t\} \subseteq (Y{:}t \;\&\; Y.l \doteq Z \;\&\; Z{:}u)$.*

Proposition 3 *If ϕ is the result of the ψ-term generalization of the OSF-clauses of ψ-terms ψ_1 and ψ_2 and $\psi = \psi(\phi)$, then $\psi \leq \psi_1$ and $\psi \leq \psi_2$ in terms of sorted ordering $\leq$.*

Proof. Prove $\psi \leq \psi_1$. Let ϕ_1 be a solved normal form of ψ_1. Let the final result of the ψ-term generalization be $\Gamma_1, \Gamma_2 \vdash \phi \;\&\; (\phi_1 \vee \phi_2)$ with $\Gamma_1 = \{X'_1 \backslash X_1, \ldots, X'_n \backslash X_n\}$. Let s'_i be the sort of $X'_i{:}s'_i \in \phi(\psi_1)$ and s_i be the sort of $X_i{:}s_i \in \phi(\psi)$. A sorted substitution $\theta = \{X'_1{:}s'_1/X_1{:}s_1, \ldots, X'_n{:}s'_n/X_n{:}s_n\}$ clearly satisfies the relation $\phi\theta \subseteq \phi_1$ according to the OSF generalization rules. The proof of $\psi \leq \psi_2$ is the same.

Definition 14 (Least General Generalization) *Let ψ_1 and ψ_2 be ψ-terms. ψ is the least general generalization (lgg) of ψ_1 and ψ_2 iff*

(1) $\psi \leq \psi_1$ and $\psi \leq \psi_2$.
(2) If $\psi' \leq \psi_1$ and $\psi' \leq \psi_2$, then $\psi' \leq \psi$.

Theorem 3 (Least Generality of Generalization) *The axiomatic ψ-term generalization is a least general generalization with respect to sorted ordering of ψ-terms.*

Proof. (1) $\psi \leq \psi_1$ and $\psi \leq \psi_2$ are immediate from Proposition 3. (2) Let ψ_1 and ψ_2 be ψ-terms and ψ be the result of ψ-term generalization of ψ_1 and ψ_2. Assume that there exists a ψ-term ψ' such that $\psi' \leq \psi_1$, $\psi' \leq \psi_2$, and $\psi < \psi'$, *i.e.*, ψ is strictly more general than ψ'. Let ϕ, ϕ_1, ϕ_2 and ϕ' be a solved normal form of ψ, ψ_1, ψ_2, and ψ', respectively. The assumption $\psi < \psi'$ requires that there exist an OSF constraint C' in ϕ' such that no sorted substitution θ satisfies $C'\theta \in \phi$ and $\phi'\theta \subseteq \phi$. There are two cases to be considered: (case 1) C' is of the form $X' : s'$; (case 2) C' is of the form $X'.l \doteq Y'$.

Case 1: From the assumption $\psi' \leq \psi_1$ and $\psi' \leq \psi_2$, $X' : s'$ can be substituted to $X_1 : s_1$ in ϕ_1 and $X_2 : s_2$ in ϕ_2. Therefore, $s_1 \preceq s'$ and $s_2 \preceq s'$. Since sorted substitutions preserve the structure of ψ-terms, according to ψ-term generalization rules, if $X_1 : s_1$ and $X_2 : s_2$ correspond to the same constraint $X' : s$, then $X : s$ should be included in ϕ. By SI, sort s in ψ is the least upper bound (lub) of a sort s_1 in ψ_1 and a sort s_2 in ψ_2. This contradicts $s_1 \preceq s'$, $s_2 \preceq s'$, and $s' \prec s$.

[3] We regard a clause as a set of constraints here.

Case 2: Similarly, from the assumption of $\psi' \leq \psi_1$ and $\psi' \leq \psi_2$, $X'.l \doteq Y'$ can be substituted to $X_1.1 \doteq Y_1$ in ϕ_1 and $X_2.l \doteq Y_2$ in ϕ_2. Since sorted substitutions preserve the structure of ψ-terms, if $X_1.l \doteq Y_1$ and $X_2.l \doteq Y_2$ correspond to the same constraint $X'.l \doteq Y'$, then $X.l \doteq Y$ should be in ϕ. This is a contradiction.

4 Operational ψ-term Generalization

On the other hand, an operational definition of ψ-term generalization [11] has been defined as an extension of Plotkin's *least general generalization (lgg)* using the following notations. a and b represent untagged ψ-terms. s, t, and u represent ψ-terms. f, g, and h represent sorts. X, Y, and Z represent variables in $\mathcal{V}$.

Definition 15 (lgg of ψ-terms) *Let ψ_1 and ψ_2 be ψ-terms. lgg(ψ_1,ψ_2) is defined as follows with the initial history $Hist = \{\}$.*

1. *$lgg(X : a, X : a) = X : a$.*
2. *$lgg(X : a, Y : b) = Z : \top$, where $X \neq Y$ and the tuple (X, Y, Z) is already in the history $Hist$.*
3. *If $s = X{:}f(l_1^s \Rightarrow s_1, \ldots, l_n^s \Rightarrow s_n)$ and $t = Y{:}g(l_1^t \Rightarrow t_1, \ldots, l_m^t \Rightarrow t_m)$, then $lgg(s,t)$=$Z{:}(f \sqcup g)(l_1 \Rightarrow lgg(s \cdot l_1, t \cdot l_1), \ldots, l_{|L|} \Rightarrow lgg(s \cdot l_{|L|}, t \cdot l_{|L|}))$, where $l_i \in L = \{l_1^s, \ldots, l_n^s\} \cap \{l_1^t, \ldots, l_m^t\}$. Then, (X, Y, Z) is added to $Hist$.*

Note that in this definition $s \cdot l$ is defined as $s \cdot l = X : a$ if $s.l = X : \top$ and $X : a \in \psi_1$ with $a \neq \top$ else $s \cdot l = s.l$. $t \cdot l$ is defined similarly.

For example, the lgg of $X{:}passenger(of{\Rightarrow}X'{:}10)$ and $Y{:}man(of{\Rightarrow}Y'{:}2)$ is

$$Z : person(of \ \Rightarrow Z' : number),$$

if $passenger \sqcup man = person$ and $10 \sqcup 2 = number$.

Theorem 4 (Correctness) *The result of the operational ψ-term generalization ψ is the least general generalization of ψ-terms ψ_1 and ψ_2 in terms of the sorted ordering.*

Proof. (Sketch) Each step of the operational definition can be translated into OSF generalization rules. Step 1 is a special case of Sort Induction.

Step 2 is Coreference Induction where tuple (X,Y,Z) in *Hist* corresponds to $X \backslash Z$ in Γ_1 and $Y \backslash Z$ in Γ_2.

Step 3 is Sort Induction of $X : f \sqcup g$ and Feature Induction, where tuple (X, Y, Z) added to *Hist* corresponds to $X \backslash Z$ and $Y \backslash Z$ which are added to variable substitutions. All of the steps of the operational definition are realizations of the OSF clause generalization. Therefore, the result of the operational generalization is the least general generalization of ψ-terms.

5 Generalization of Clauses based on ψ-terms

This section presents a least general generalization of logic programs based on ψ-terms along the line with Plotkin's lgg of atom and clauses [10].

Definition 16 (Ordering of Atoms) *Let $A_1 = p(\psi_1, \ldots, \psi_n)$ and $A_2 = q(\psi'_1, \ldots, \psi'_n)$ be atomic formulae based on ψ-terms. $A_1 \leq A_2$ iff $A_1\theta = A_2$ for some sorted substitution θ which includes substitutions replacing the root variable of ψ_i by the root variable of ψ'_i.*

Definition 17 (Ordering of Clauses) *Let C_1 and C_2 be clauses based on ψ-terms. $C_1 \leq C_2$ iff $C_1\theta \subseteq C_2$ for some sorted substitution θ which includes substitutions replacing the root variables of ψ-terms in C_1 by the corresponding root variables of ψ-terms in C_2.*

Definition 18 (Lgg of Atoms) *Given a signature $\Sigma_{OSF} = \langle \mathcal{S}, \preceq, \sqcap, \sqcup, \mathcal{F} \rangle$ and a set of predicate symbols $\mathcal{P}$, let P and Q be atomic formulae. An operational definition of a function $lgg(P, Q)$ that computes the least general generalization of P and Q is as follows.*

1. *If $P = p(s_1, \ldots, s_n)$ and $Q = p(t_1, \ldots, t_n)$,*
$$lgg(P, Q) = p(lgg(s_1, t_1), \ldots, lgg(s_n, t_n))$$
with the sharing of history Hist.
2. *Otherwise, $lgg(P, Q)$ is undefined.*

Definition 19 (Lgg of Literals) *Let P and Q be atoms and L_1 and L_2 be literals. The lgg of literals is defined as follows [8].*

1. *If L_1 and L_2 are atoms, then $lgg(L_1, L_2)$ is the lgg of the atoms.*
2. *If L_1 and L_2 are the form $\neg P$ and $\neg Q$, respectively, then $lgg(L_1, L_2) = lgg(\neg P, \neg Q) = \neg lgg(P, Q)$.*
3. *Otherwise, $lgg(L_1, L_2)$ is undefined.*

Definition 20 (Lgg of Clauses) *Let clauses $C = \{L_1, \ldots, L_n\}$ and $D = \{K_1, \ldots, K_m\}$. Then $lgg(C, D) = \{\ lgg(L_i, K_j) \mid L_i \in C, K_j \in D$ and $lgg(L_i, K_j)$ is not undefined$\}$.*

The least general generality of lggs of atoms, literals, and clauses is conservative extension of Plotkin's lgg since the operational ψ-term generalization is a lgg of terms.

6 Related Work

The definition of the *least general generalization (lgg)* was first investigated in [10]. The lgg of ψ-terms has already been illustrated [1]; however, axiomatic and operational definitions have been left untouched. The lgg of a subset of *description logics*, called the *least common subsumer (LCS)*, was studied in [5]. The lgg of *feature terms*, which are equivalent to ψ-terms, can be found in [9]. The generalization for *Sorted First Order Predicate Calculus (SFOPC)* [7] is presented in [6].

7 Conclusion and Remarks

Two generalization approaches have been presented and related. An *axiomatic* definition of ψ-term generalization was presented as ψ-term generalization rules. The definition is proven to be a *least general generalization (lgg)* in terms of Plotkin's lgg. The correctness of an *operational* definition of ψ-term generalization was provided on the basis of the generalization rules. The operational definition was shown to be one realization of the axiomatic generalization. The lgg of clauses based on ψ-terms was presented, and a fundamental bridge between ψ-term generalization and the lgg useful for inductive logic programming was given. The main benefit of this paper is that it expresses generalization (and hence induction) as an OSF constraint *construction* process. This approach may lead to other axiomatic constraint systems provided with inductive algorithms.

References

1. Hassan Aït-Kaci and Roger Nasr. LOGIN: A logic programming language with built-in inheritance. *Journal of Logic Programming*, 3:185–215, 1986.
2. Hassan Aït-Kaci and Andreas Podelski. Towards a meaning of LIFE. *Logic Program.*, 16(3-4):195–234, July-August 1993.
3. Hassan Aït-Kaci, Andreas Podelski, and Seth Copen Goldstein. Order-sorted feature theory unification. *Journal of Logic Programming*, 30(2):99–124, 1997.
4. Bob Carpenter. *The Logic of Typed Feature Structures*, volume 32 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 1992.
5. William W. Cohen, Alexander Borgida, and Helen Hirsch. Computing least common subsumers in description logics. In *Twelfth National Conference on Artificial Intelligence*, pages 754–760, San Jose, CA, 1992. AAAI-92.
6. Alan M. Frisch and C. David Page Jr. Generalization with taxonomic information. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 755–761, Boston, MA, 1990. AAAI-90.
7. Alan M. Frisch. A general framework for sorted deduction: Fundamental results on hybrid reasoning. In *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning*, pages 126–136, 1989.
8. Nada Lavrač and Sašo Džeroski. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, 1994.
9. Enric Plaza. Cases as terms: A feature term approach to the structured representation of cases. In *Proceedings of the 1st International Conference on Case-Based Reasoning*, pages 263–27, 1995.
10. Gordon Plotkin. A note on inductive generalization. In *Machine Intelligence*, pages 153–163. Edinburgh University Press, 1969.
11. Yutaka Sasaki. Induction of logic programs based on ψ-terms. In *Proceedings of the 10th International Conference on Algorithmic Learning Theory*, pages 169–181, Tokyo, Japan, 1999. ALT-99, Springer-Verlag LNAI 1720.
12. Yutaka Sasaki. *Hierarchically Sorted Inductive Logic Programming and Its Application to Information Extraction*. Ph.D thesis, Graduate School of Systems and Information Engineering, University of Tsukuba, Japan, September 2000.

Appendix (Example of Axiomatic ψ-term generalization)

Suppose that we have two ψ-terms ψ_1 and ψ_2, and $u{=}s \sqcup t$.

$$\psi_1 = X : s(a \Rightarrow Z : s, b \Rightarrow Z)$$
$$\psi_2 = Y : t(a \Rightarrow W : t, b \Rightarrow U : u)$$

The normal form of OSF clauses of these ψ-terms are:

$$\phi(\psi_1) = X : s \mathbin{\&} X.a \doteq Z \mathbin{\&} Z : s \mathbin{\&} X.b \doteq Z,$$
$$\phi(\psi_2) = Y : t \mathbin{\&} Y.a \doteq W \mathbin{\&} W : t \mathbin{\&} Y.b \doteq U \mathbin{\&} U : u.$$

A generalization of of these two OSF clauses is obtained by applying generalization rules to the OSF clause:
$C = (X{:}s \mathbin{\&} X.a \doteq Z \mathbin{\&} Z{:}s \mathbin{\&} X.b \doteq Z)$
$\quad \vee \quad (Y{:}t \mathbin{\&} Y.a \doteq W \mathbin{\&} W{:}t \mathbin{\&} Y.b \doteq U \mathbin{\&} U{:}t)$
The following steps show the process to achieve a generalization.

$\{X\backslash V\}, \{Y\backslash V\} \vdash$
$\quad ((\underline{X{:}s} \mathbin{\&} X.a \doteq Z \mathbin{\&} Z{:}s \mathbin{\&} X.b \doteq Z) \vee$
$\quad (\underline{Y{:}t} \mathbin{\&} Y.a \doteq W \mathbin{\&} W{:}t \mathbin{\&} Y.b \doteq U \mathbin{\&} U{:}t))$

$\{X\backslash V\}, \{Y\backslash V\} \vdash \quad (V{:}u)$
$\quad \mathbin{\&} \ ((X{:}s \mathbin{\&} \underline{X.a \doteq Z} \mathbin{\&} Z{:}s \mathbin{\&} X.b \doteq Z) \vee$
$\quad (Y{:}t \mathbin{\&} \underline{Y.a \doteq W} \mathbin{\&} W{:}t \mathbin{\&} Y.b \doteq U \mathbin{\&} U{:}t))$ (by SI)

$\{X\backslash V, Z\backslash V'\}, \{Y\backslash V, W\backslash V'\} \vdash \quad (V{:}u \mathbin{\&} V.a \doteq V')$
$\quad \mathbin{\&} \ ((X{:}s \mathbin{\&} X.a \doteq Z \mathbin{\&} \underline{Z{:}s} \mathbin{\&} X.b \doteq Z) \vee$
$\quad (Y{:}t \mathbin{\&} Y.a \doteq W \mathbin{\&} \underline{W{:}t} \mathbin{\&} Y.b \doteq U \mathbin{\&} U{:}t))$ (by FI)

$\{X\backslash V, Z\backslash V'\}, \{Y\backslash V, W\backslash V'\} \vdash \quad (V{:}u \mathbin{\&} V.a \doteq V' \mathbin{\&} V'{:}u)$
$\quad \mathbin{\&} \ ((X{:}s \mathbin{\&} X.a \doteq Z \mathbin{\&} Z{:}s \mathbin{\&} \underline{X.b \doteq Z}) \vee$
$\quad (Y{:}t \mathbin{\&} Y.a \doteq W \mathbin{\&} W{:}t \mathbin{\&} \underline{Y.b \doteq U} \mathbin{\&} U{:}t))$ (by SI)

$\{X\backslash V, Z\backslash V', Z\backslash V''\}, \{Y\backslash V, W\backslash V', U\backslash V''\} \vdash \quad (V : u \mathbin{\&} V.a \doteq V' \mathbin{\&} V' : u \mathbin{\&} V.b \doteq V'')$
$\quad \mathbin{\&} \ ((\underline{X{:}s} \mathbin{\&} X.a \doteq Z \mathbin{\&} Z{:}s \mathbin{\&} X.b \doteq Z) \vee$
$\quad (Y{:}t \mathbin{\&} Y.a \doteq W \mathbin{\&} W{:}t \mathbin{\&} Y.b \doteq U \mathbin{\&} \underline{U{:}t}))$ (by FI)

$\{X\backslash V, Z\backslash V', Z\backslash V''\}, \{Y\backslash V, W\backslash V', U\backslash V''\} \vdash (\underline{V{:}u \mathbin{\&} V.a \doteq V' \mathbin{\&} V'{:}u \mathbin{\&} V.b \doteq V'' \mathbin{\&} V''{:}u})$
$\quad \mathbin{\&} \ ((X : s \mathbin{\&} X.a \doteq Z \mathbin{\&} Z : s \mathbin{\&} X.b \doteq Z) \vee$
$\quad (Y : t \mathbin{\&} Y.a \doteq W \mathbin{\&} W : t \mathbin{\&} Y.b \doteq U \mathbin{\&} U : t))$ (by SI)

Therefore, an OSF clause ψ_3 of a ψ-term generalization of ψ_1 and ψ_2 is:

$$\phi_3 = V : u \mathbin{\&} V.a \doteq V' \mathbin{\&} V' : u \mathbin{\&} V.b \doteq V'' \mathbin{\&} V'' : u$$

The ψ-term of ψ_3 is:

$$\psi(\phi_3) = V : u(a \Rightarrow V' : u, b \Rightarrow V'' : u)$$