CHAPTER II

BACKGROUND ON LOGIC AND GENERALIZATION UNDER θ-SUBSUMPTION

In this chapter we will describe the background on logic and the framework for generalization of clauses developed by Plotkin. This framework is based on a generality relation known as θ-subsumption.

We start with a brief review of clausal logic, derived from Appendix A in Idestam-Almquist [2].

Background

A **clausal logic** consists of a clausal alphabet, a clausal language, a set of axioms, and a single derivation rule, called resolution.

Definition 1. A **clausal alphabet** is a set of symbols $\mathbf{A} = V \cup F \cup P \cup O \cup U$ where V, F, P, O, U are disjoint and:

a) V is a non-empty set of variables,

b) F is a set of function symbols,

c) P is a non-empty set of predicate symbols,

d) O is the set of connectives $\{\neg\}$, and

e) U is the set of punctuation symbols {(,), {, }, ,} (the fifth symbol in U is a comma).

Definition 2. A **constant symbol** is a 0-ary function symbol.

Variables are normally denoted by the letters x, y, z, u, v, and w. Function symbols are normally denoted by f, g and h. Constant symbols are normally denoted by a, b, c, d and e. Predicate symbols are normally denoted by p, q and r. Note that by adding extra symbols if necessary, we may assume that there is an infinite number of constant symbols. We will need this assumption in Chapter III.

Definition 3. A **term** is defined as follows:

a) a variable is a term, and

b) $f(t_1, t_2,\ldots, t_n)$ is a term if f is an n-ary function symbol and $t_1, t_2,\ldots, t_n$ are terms ($n \geq 0$).

The term $a(\,)$, where a is a constant symbol, is denoted by a. In this work we will use the convention that $f^n(t)$ denotes n applications of the function f to t. For example the term $f(f(f(f(a))))$ is written as $f^4(a)$.

Definition 4. If p is an n-ary predicate symbol and $t_1, t_2,\ldots, t_n$ are terms then $p(t_1, t_2,\ldots, t_n)$ is an **atom** ($n \geq 0$).

The atom $p(\,)$, where p is a 0-ary predicate symbol, will be denoted by p.

Definition 5. A **positive literal** is an atom A, and a **negative literal** is the negation $\neg B$ of an atom B. A **literal** is either a positive literal or a negative literal.

Definition 6. A **clause** is a finite set of literals. The **empty clause**, denoted by $\perp$, is the clause containing no literals.

A clause $\{A_1, A_2,\ldots, A_k, \neg B_1, \neg B_2,\ldots, \neg B_n\}$, where $A_1, A_2,\ldots, A_k$ are positive literals and $\neg B_1, \neg B_2,\ldots, \neg B_n$ are negative literals, is for convenience denoted by

$$(A_1, A_2,\ldots, A_k \leftarrow B_1, B_2,\ldots, B_n).$$

This clause corresponds to the formula

$$\forall x_1\, \forall x_2 \ldots \forall x_m\, (A_1 \vee A_2 \vee \ldots \vee A_k \vee \neg B_1 \vee \neg B_2 \vee \ldots \vee \neg B_n),$$

or the equivalent formula

$$\forall x_1 \forall x_2 \ldots \forall x_m ((A_1 \vee A_2 \vee \ldots \vee A_k) \leftarrow (B_1 \wedge B_2 \wedge \ldots \wedge B_n)),$$

where $x_1, x_2, \ldots, x_m$ are all of the variables in the clause.

Definition 7. A **simple expression** is either a term or a literal. An **expression** is either a simple expression or a finite set of simple expressions.

Definition 8. An expression E is **ground** if and only if it contains no variables.

Definition 9. A **substitution** is a finite set of the form $\{x_1/t_1, x_2/t_2, \ldots, x_n/t_n\}$, where $x_1, x_2, \ldots, x_n$ are variables and $t_1, t_2, \ldots, t_n$ are terms, such that t_i is distinct from x_i for every $1 \le i \le n$. An element x/t in a substitution is called a **binding** for x. A substitution $\theta = \{x_1/t_1, x_2/t_2, \ldots, x_n/t_n\}$ is a **ground substitution** if and only if t_i is ground for every $1 \le i \le n$.

Definition 10. For any substitution $\sigma = \{x_1/t_1, x_2/t_2, \ldots, x_n/t_n\}$, let the **domain** of σ be the set $\{x_1, x_2, \ldots, x_n\}$.

Definition 11. The substitution given by the empty set is called the **identity substitution**, and is denoted by ε.

Definition 12. Let $\theta = \{x_1/t_1, x_2/t_2, \ldots, x_n/t_n\}$ be a substitution and E an expression. Then $E\theta$, the **instance** of E by θ, is the expression obtained from E by simultaneously replacing each occurrence of variable x_i in E by the term t_i, for every $1 \le i \le n$. If $E\theta$ is ground then $E\theta$ is called a **ground instance** of E.

Definition 13. Let E and F be expressions. Then E and F are **variants**, denoted $E \cong F$, if and only if there exist substitutions θ and σ such that $E = F\theta$ and $F = E\sigma$. We also say that E is a **variant** of F.

Definition 14. Let $\theta = \{x_1/t_1, x_2/t_2, \ldots, x_n/t_n\}$ and $\sigma = \{y_1/s_1, y_2/s_2, \ldots, y_m/s_m\}$ be substitutions. Then the **composition** $\theta\sigma$ of θ and σ is the substitution obtained from the set $\{x_1/t_1\sigma, x_2/t_2\sigma,\ldots, x_n/t_n\sigma, y_1/s_1, y_2/s_2,\ldots, y_m/s_m\}$ by deleting any binding $x_i/t_i\sigma$ for which $x_i = t_i\sigma$ and deleting any binding y_j/s_j for which $y_j \in \{x_1, x_2,\ldots, x_n\}$ $(1 \le i \le n, 1 \le j \le m)$.

We denote an n-fold composition of a substitution θ by θ^n. For example $\theta^3 = \theta\theta\theta$ and $\theta^0 = \varepsilon$. Note that if E is an expression and θ and σ are substitutions, then $(E\theta)\sigma = E(\theta\sigma)$.

Definition 15. A substitution θ is a **unifier** for a finite set of simple expressions S if and only if $S\theta$ is a singleton. A unifier θ for S is a **most general unifier** (mgu) for S if and only if for each unifier σ of S there exists a substitution γ such that $\sigma = \theta\gamma$. A set of simple expressions S is **unifiable** if and only if there exists a unifier for S.

Definition 16. Let C be a clause, $\Gamma \subseteq C$ and γ an mgu of Γ. Then $C\gamma$ is a **factor** of C.

Example. The clause $(p(a) \leftarrow q(a))$ is a factor of $(p(x) \leftarrow q(x), q(a))$, since $\{x/a\}$ is an mgu of $\{\neg q(x), \neg q(a))\}$. The clause $(r(y) \leftarrow)$ is a factor of $(r(x) \leftarrow)$, since $\{x/y\}$ is an mgu of $\{r(x)\}$.

Every clause is a factor of itself and every clause is a factor of its variants. Thus it is always possible to find factors of two clauses such that the factors do not have any common variables, which is important for the definition of a resolvent.

Definition 17. A clause R is a **resolvent** of two clauses C and D if and only if there are $C\gamma$, $D\mu$, A, B and θ such that:

a) $C\gamma$ is a factor of C and $D\mu$ is a factor of D,

b) $C\gamma$ and $D\mu$ have no variables in common,

c) A is a literal in $C\gamma$ and B is a literal in $D\mu$,

d) θ is an mgu of $\{A, \overline{B}\}$, and

e) R is the clause $((C\gamma - \{A\}) \cup (D\mu - \{B\}))\theta$.

The clause C and D are called **parent clauses** of R, and the literals A and B are called the **literals resolved upon** in the resolution of C and D.

Definition 18. Let T be a set of clauses. Then, the $\boldsymbol{n}^{\text{th}}$ **resolution** of T, denoted $R^n(T)$, is defined as:

a) $R^0(T) = T$, and

b) $R^n(T) = R^{n-1}(T) \cup \{R \mid C, D \in R^{n-1}(T)$ and R is a resolvent of C and $D\}$ $(n > 0)$.

Definition 19. The **complement** $\overline{A}$ of a positive literal A is $\neg A$, and the **complement** $\overline{\neg A}$ of a negative literal $\neg A$ is A.

Semantics is concerned with the meaning attached to the formulas in a language.

Definition 20. Let $\mathcal{L}$ be a clausal language given by $\mathcal{A}$. Then the **Herbrand universe** $\mathcal{U}(\mathcal{L})$ for $\mathcal{L}$ is the set of all ground terms which can be formed out of the function symbols in $\mathcal{A}$. Note that our assumption that $\mathcal{A}$ contains an infinite number of constant symbols guarantees $\mathcal{U}(\mathcal{L})$ is not empty.

Definition 21. Let $\mathcal{L}$ be a clausal language given by $\mathcal{A}$. The **Herbrand base** $\mathcal{B}(\mathcal{L})$ for $\mathcal{L}$ is the set of all ground atoms which can be formed out of predicate symbols in $\mathcal{A}$ and terms in $\mathcal{U}(\mathcal{L})$.

Definition 22. An **interpretation** I of a clausal language $\mathcal{L}$ is a subset of the Herbrand base $\mathcal{B}(\mathcal{L})$ for $\mathcal{L}$.

Definition 23. Let I be an interpretation of a clausal language $\mathcal{L}$. Then a clause C in $\mathcal{L}$ is given a **truth value** (true or false) with respect to I as follows:

a) A ground clause $C = (A_1, A_2,..., A_m \leftarrow B_1, B_2,..., B_n)$, is true if $A_i \in I$ for some $1 \leq i \leq m$ or $B_j \notin I$ for some $1 \leq j \leq n$, otherwise it is false.

b) A non-ground clause C is true if every ground instance of C is true, otherwise it is false.

Definition 24. Let I be an interpretation of a clausal language $\mathcal{L}$, and C a clause in $\mathcal{L}$. Then I is a **model** for C if and only if C is true with respect to I.

Definition 25. Let I be an interpretation of a clausal language $\mathcal{L}$, and T a set of clauses in $\mathcal{L}$. Then I is a **model** for T if and only if I is a model for each clause in T.

Definition 26. Let X be a clause or a set of clauses in a clausal language $\mathcal{L}$. Then:

a) X is **satisfiable** if there exists an interpretation of $\mathcal{L}$ which is a model for X,

b) X is **valid** if every interpretation of $\mathcal{L}$ is a model for X,

c) X is **unsatisfiable** if no interpretation of $\mathcal{L}$ is a model for X, and

d) X is **nonvalid** if there exists an interpretation of $\mathcal{L}$ which is not a model for X.

Definition 27. Let $\mathcal{H}$ be a set of clauses, and C a clause. Then C is a **logical consequence** of $\mathcal{H}$, denoted by $\mathcal{H} \models C$, if and only if every model for $\mathcal{H}$ is a model for C.

We are particularly interested in the following syntactical properties of terms, literals, and clauses.

Definition 28. A clause is:

a) **a definite clause** if and only if it contains one positive literal and any number of negative literals ($A \leftarrow B_1, B_2,..., B_n$),

b) **a Horn clause** if and only if it is a definite clause, or it contains no positive literals and any number of negative literals ($\leftarrow B_1, B_2,..., B_n$), and

c) a **unit clause** if and only if it contains exactly one literal, positive ($A \leftarrow$) or negative ($\leftarrow B$).

Example. Consider the following clauses:

$$C = (\, p(x) \leftarrow q(y)),$$
$$D = (\, p(a) \leftarrow p(b),\, q(a)),$$
$$E = (\, \leftarrow p(b),\, q(b)),$$
$$F = (\, p(x) \leftarrow), \text{ and}$$
$$G = (\, \leftarrow p(a)).$$

C and D are definite clauses, C, D, E and G are Horn clauses, and F and G are unit clauses.

Definition 29. A clause C is **recursive** if and only if there exist literals $A, \neg B \in C$ such that A is unifiable with a variant of B.

Example. The following clause:

$$C = (\, p(x),\, q(a) \leftarrow\, p(f(x)),\, r(b))$$

is recursive since $p(x)$ is unifiable with $p(f(z))$ and $p(f(z))$ is a variant of $p(f(x))$.

All recursive clauses, and only recursive clauses, can be resolved with themselves.

θ-subsumption

Here we define the generality relation θ-subsumption.

Definition 30. Let C and D be clauses. Then C **θ-subsumes** D, denoted by $C \prec D$, if and only if there exists a substitution θ such that $C\theta \subseteq D$. We also say that C is a **generalization under θ-subsumption** of D.

Proposition 31. θ-subsumption is reflexive and transitive.

Proof. Let C, D and E be clauses.

1) We must show that $C \prec C$. Since $C \subseteq C$, we have $C\varepsilon \subseteq C$, where ε is the identity substitution. Thus, $C \prec C$. So, θ-subsumption is reflexive.

2) We must show that if $C \prec D$ and $D \prec E$, then $C \prec E$. Assume that $C \prec D$ and $D \prec E$. Then there exist substitutions θ and σ such that $C\theta \subseteq D$ and $D\sigma \subseteq E$. Hence $C\theta\sigma \subseteq E$, and $C \prec E$. Thus, θ-subsumption is transitive. □

Example. Consider the following clauses:

$C = (\leftarrow p(f(y)), p(x))$,

$D = (q(y) \leftarrow p(f(y)))$, and

$E = (q(a) \leftarrow p(f(a)), r(b))$.

We have $C \prec C$ since every clause is a subset of itself. We also have $C \prec D$ since $C\theta \subseteq D$ where $\theta = \{x/f(y)\}$, and $D \prec E$ since $D\sigma \subseteq E$, where $\sigma = \{y/a\}$. Then $C \prec E$, since $C\theta\sigma \subseteq E$.

Two clauses may θ-subsume each other without being variants. In other words, θ-subsumption is not anti-symmetric.

Definition 32. Let C and D be clauses. Then C and D are **equivalent under θ-subsumption**, denoted $C \sim D$, if and only if $C \prec D$ and $D \prec C$.

Example. Consider the following clauses:

$C = (p(a) \leftarrow q(a), q(x))$,

$D = (p(a), p(y) \leftarrow q(a))$, and

$E = (p(a) \leftarrow q(a))$.

Then we have $C \sim D$ since $C\{x/a\} \subseteq D$ and $D\{y/a\} \subseteq C$. We also have $C \sim E$ and $D \sim E$. Hence all three clauses are equivalent under θ-subsumption. Note that no two of these clauses are variants.

We are particularly interested in least general generalizations. As already mentioned in Chapter I, least general generalization under θ-subsumption is the most commonly used form of generalization of clauses.

Definition 33. A clause C is a **generalization under θ-subsumption** of a set of clauses $S = \{D_1, D_2, \dots, D_n\}$ if and only if $C \prec D_i$ for every $1 \le i \le n$. A generalization under θ-subsumption C of S is a **least general generalization under θ-subsumption** (LGGθ) of S if and only if $C' \prec C$ for every generalization under θ-subsumption C' of S.

Example. Consider the following clauses:

$C = (\,p(a) \leftarrow q(a), q(b)),$

$D = (\,p(b) \leftarrow q(b), q(x)),$

$E = (\,p(y) \leftarrow q(y), q(b)),$ and

$F = (\,p(y) \leftarrow q(y), q(b), q(z), q(w)).$

Both clauses E and F are LGGθs of $\{C, D\}$.

In general, an LGGθ is not unique, as shown by the example above. Plotkin showed that there exists an LGGθ of every finite set of clauses, a result which is not obvious.