# 同義と排反を使った演繹について

坂間 千秋　　横田 一正
（財）新世代コンピュータ技術開発機構

## 1．はじめに

述語論理による知識表現を考えた場合，例えば Prolog のような一階述語論理の部分クラスである Horn 節に限った記述言語ではその表現力が乏しいため，これまで数多くの拡張が試みられている．
（例えば [Ait-Kaci 86],[Nukai 86],[Dincbas 86], [Yamaguchi 86], etc.）
本稿ではこのような拡張の一つとして，プログラムに含まれるリテラル間の関係を，知識として知識ベースに与えることを考える．ここで扱う関係とは知識の同義と排反関係であり，それらは対象世界の中にある自然な関係である．そして，これらの関係を使った演繹と否定について述べる．

## 2．同義による演繹

演繹過程に同値関係を取り込む試みとしては，[Robinson 69] を始めとする一階述語論理に等号論理を取り入れる方法などがあるが，一般に Horn 論理の枠組みにおいてリテラルの意味上の同値関係を演繹するためには，その関係を公理として宣言的に記述しておく必要がある．
以下では，Horn 論理からなる公理系におけるリテラルの集合を，意味上の同値関係によって分類して演繹を行うことを考える．そこで，先ずリテラルの同義関係と同義類を定義する．

**定義　同義，同義類**
ある公理系においてリテラル p，q が同義（synonymous）であるとは，

$$p \sim q = (p \wedge q) \vee (\neg p \wedge \neg q)$$

であることである．ここで，$p \sim q$ は p と q の同義関係を表す．
いま，あるリテラルの集合の任意の２要素について同義関係が成り立つとき，その集合を同義類（synonymy class）と呼ぶ．
このとき，ある公理系に表われるリテラルの集合Λは同義類の直和に分割される．この分割を Λ の同義に関する類別（classification）という．また，Λの類別において，各同義類から１つずつ取り出したΛの元をそれらの同義類の代表元（representative）という．　□

特に，グラウンドでないリテラルの集合からなる同義類 Φ を考えたとき，そのインスタンス Φθ（θ：代入）は同義類か，あるいはある同義類の部分集合となる．ここで Φ の代表元が φ のとき，Φθ を含む同義類の代表元を φθ とする．

**例**
Φ＝｛parent(X,Y), bring_up(X,Y), child(Y,X)｝
φ＝parent(X,Y)　　□

確定節の集合としてプログラムが与えられたとき，予め必要最小限のリテラルの集合（基本関係）と，同義関係などの他のリテラルとの関係（高階関係）を定義しておいて，そこでの問い合わせの処理過程において，この関係を利用しながら基本関係のみからなる問い合わせに変換して処理を行う方法に [Yokomori 86] があるが，ここでは確定節の集合からなるプログラムに対して，知識ベースによってプログラム中のリテラルの同義による類別の知識を与えることによって，プログラムをそのクラスの代表元のみからなるプログラムに変換することを考える．
ここで，プログラムの計算メカニズムとしては通常のＳＬＤ導出を適用する．

**例**
プログラム

```
ancestor(X,Y)←parent(X,Y).
ancestor(X,Y)←parent(X,Z),ancestor(Z,Y).
descendant(X,Y)←offspring(X,Y).
descendant(X,Y)←offspring(X,Z),descendant(Z,Y).
parent(taro,hanako).
parent(hanako,ichiro).
child(jiro,hanako).
```

に対して，同義類が

Φ1＝｛ancestor(X,Y), descendant(Y,X)｝
Φ2＝｛parent(X,Y), offspring(Y,X), child(Y,X)｝

で与えられたとき，Φ1,Φ2 の代表元をそれぞれ

φ1＝ancestor(X,Y)
φ2＝parent(X,Y)

とすると，もとのプログラムは以下のように変換される．

---

Deduction with Synonymy and Exclusion
by Chiaki SAKAMA and Kazumasa YOKOTA
(ICOT Research Center).

```
ancestor(X,Y)←parent(X,Y).
ancestor(X,Y)←parent(X,Z),ancestor(Z,Y).
parent(taro,hanako).
parent(hanako,ichiro).
parent(hanako,jiro).                    □
```

上の例で，例えばもとのプログラムでは ancestor(taro,jiro) は演繹されないが変換後では演繹される．なお，ここで変換の結果生じる冗長な節は除去している．

## 3．排反による否定

論理型言語における否定は閉世界仮説，あるいはNegation as failure が代表的であるが，これらの否定は証明不能としての消極的な否定である．これに対して，プログラム中に直接否定的知識を書くことにより積極的に否定を証明しようとする試みに，例えば〔Sakai 84〕，〔Gabbay 86〕などがある．

以下では，Horn 論理からなる公理系にリテラルの意味上の排反関係を公理として取り込み，ＳＬＤ導出の下でこの排反関係に基づいて否定を証明することを考える．そのために，先ずリテラルの排反関係と排反類を定義する．

**定義**　排反，排反類

ある公理系において，リテラル ｐ，ｑが排反（exclusive）であるとは

$$p \dotplus q = \neg(p \wedge q)$$

であることである．ここで，$p \dotplus q$は ｐ と ｑ の排反関係を表す．

いま，あるリテラルの集合の任意の２要素について排反関係が成り立つとき，その集合を排反類（exclusion class）と呼ぶ．　□

特に，グラウンドでないリテラルの集合からなる排反類 Ψ を考えたとき，そのインスタンス Ψθ（θ：代入）もまた排反類をなす．

**例**

Ψ＝｛parent(X,Y),grandparent(X,Y),child(X,Y)｝　□

次に，このような排反関係を使った否定を定義する．

**定義**　排反による否定

ある公理系Ｓに対して，命題ｐ，ｑが

$$S \vdash p \quad \text{かつ} \quad p \dotplus q$$

であるとき，ｑ は排反によって否定（negation as exclusion）されたという．　□

**例**

公理系Ｓが｛parent(taro,hanako)，parent(X,Y)＋child(X,Y)｝のとき，child(taro,hanako) は排反によって否定された．　□

ある命題の排反による否定は，その命題と排反する命題のＳＬＤ導出による反駁の証明によってなされる．

またゴール節に変数を含む場合，例えば ←parent(X,Y)．と排反するゴール ←grandparent(X,Y)．に対して X=taro,Y=ichiro が導出された場合，" taro は ichiro の grandparent であって parent ではない " という事実が証明されたことになる．

## 4．おわりに

実際のシステムを考えた場合，知識ベース中に同義，排反関係が知識として貯えられており，そこに Horn 論理で記述されたプログラムが入力されると，そのプログラムは代表元からなるプログラムにコンパイルされ，排反関係は代表元に対して定義されることになる．

このようなモデルにおいては，代表元をもとのプログラムに対するアセンブラと見做し，知識の処理はこのアセンブラに対して行われると考えると，もとのプログラム中の知識の表現方法には依存しない柔軟なシステムが構築出来ると考えられる．

しかし，ここで代表元の選択方法，コンパイル後のプログラムの冗長性の除去，あるいは排反関係の無矛盾性の保証などの幾つかの問題点がある．

今後，これらの理論的問題点と共に実現方法について検討していく．

**謝辞**

貴重なコメントを頂いた松本裕治，坂井公 各氏，並びに ＫＢＭ プロジェクトの諸兄に感謝致します．

**参考文献**

〔Ait-Kaci 86〕 Ait-Kaci,H. and Nasr,R.:"LOGIN: A Logic Programming Language with Built-in Inheritance", Jour. of Logic Programming, vol.3, No.3, pp.185-215, 1986.

〔Dincbas 86〕 Dincbas,M.:"Constraints, Logic Programming and Deductive Databases", Proc. of France-Japan AI Sym.'86, Tokyo, pp.1-27, 1986.

〔Nukai 86〕 Nukai,K.:"CIL reference mannual", ICOT-TR, 1986.

〔Yamaguchi 86〕 Yamaguchi,J.:"Boolean-valued Prolog", NEC-TR, 1986.

〔Robinson 69〕 Robinson,G. and Wos,L.:"Paramodulation and Theorem-proving in First-order Theories with Equality", Machine Intelligence, vol.4, pp.135-150, 1969.

〔Yokomori 86〕 Yokomori,T.:"On Analogical Query Processing in Logic Database", Proc. of 12th Int. Conf. on VLDB, Kyoto, pp.376-383, 1986.

〔Sakai 84〕 Sakai,K. and Miyachi,T.:"Incorporating Naive Negation into Prolog", Lecture Note in Computer Science, pp.130-143, vol.220, 1984.

〔Gabbay 86〕 Gabbay,D.M. and Sergot,M.J.:"Negation as Inconsistency, I", Jour. of Logic Programming, vol.3, No.1, pp.1-35, 1986.