2014年度 SCCP s1200191 古河智弥

論理型プログラミング言語 Prolog の学習

目的

宣言型言語であり、探索などに利用することができるプログラミング言語 Prolog の基本を習得し、機械学習の研究への応用および、データベースの問い合せ言語として Prolog を記述する方法を学ぶ。

概要

The Art of Prolog [1] の全24章の内、始めの2章を読み、内容の要約を発表する形式で学習した。ここでは Prolog についてと、発表資料を元に学習した内容についてまとめた。

Prolog とは

Prolog とは記号処理を目的につくられた、非数値計算用プログラミング言語である。一階述語論理をベースとした宣言型言語であり、プログラムは複数の節と呼ばれる構造からなる。Prolog のプログラムの流れは、事実や規則の集合に対して質問をし、その質問にあてはまるものを Prolog が自動で返し、それを使ってまた質問をする、といったくりかえしで構成される。プログラムを作成する目的は通常なにかの目的(10の階上を知りたい)があり、Prolog はその目的を記述することに集中できるように作られている。他の手続型言語のように手順の詳細を記述する必要はない。

キーワード

- **一階述語論理**
  数理論理学における論理の数学的モデルのひとつである述語論理の中でも、固体の量化のみを許すもの。つまり一階述語論理では述語を量化することはできないので、これを元にしている Prolog でも量化することはできない。

- **宣言型言語**
  代表的な手続き型言語である C 言語などのように処理の流れではなく、それがなんであるかをプログラムに記述するというプログラミングのパラダイムを表す。

- **リテラル**
  原子論理式と呼ばれる論理式における最小の単位、またはそれの否定をあらわす。

- **節**
  命題論理においてリテラルを選言で結合した命題であり、Prolog のコードは式の変形を行うことで全て節の形になる。詳細については The Art of Prolog のキーワードの項で記す。

・事実
animal(dog).
のような、変数を含まずそれが常に真であるような節である。

・規則
規則は
A :- B1,B2,…,Bn. (n >= 0)
のような形をした節であり、A の部分をヘッド, B1,B2,…,Bn (n >= 0) の部分をボディと言う。例えば、
creature(X) :- animal(X).
のようなものであり、論理学における A → B を逆向きにしたものをあらわしている。つまり animal(X) が真であるならば creature(X) は真である、という意味である。
また、n = 0 である規則は事実とみなすことができる。

・質問
?- animal(dog).
のように、Prolog に対して式を渡し、それとマッチするようなものを検索してもらうこと。上記のような事実(animal(dog).)が宣言されている場合 Prolog は true のように結果を返す。変数を含むこともでき
?- animal(X).
とすると、Prolog は X = dog のように、マッチする項目を探し結果を返す。

The Art of Prolog の概要

ここより、[1]の要約と用語の解説をする。

1章

この章では Prolog のプログラムにおいて一つの質問や、規則を表す "節" に焦点を当てて解説している。節、規則、事実の関係は、 節 ⊃ 規則 ⊃ 事実 であり、 Prolog のプログラムは節の集合で成りたっている。

キーワード

・基底節
節の中でも変数を含まないものを基底節と言い、そうでないものを非基底節という。

・ホーン節
数理論理学において、以下の条件を満す節を差し、 Prolog での自動推論を行う際の基本となっている。

否定でない原子命題のリテラルを Ln (n > 0), C とする。否定でないリテラルがただひとつの節 (¬L1 ∨ ¬L2 ∨ … ∨ ¬Ln ∨ C) を 確定節という。また、全てが否定を含むのリテラ

ルからなる節 (¬L0 ∨ ¬L1 ∨ … ∨ ¬Ln) を ゴール節という。ここで命題論理において、P,Qを原子命題とすれば、¬P ∨ Q = P → Q になるのでゴール節は、
= ¬L1 ∨ ¬L2 ∨ … ∨ ¬Ln ∨ C
= ¬(L1 ∧ L2 ∧ … ∧ Ln) ∨ C
= L1 ∧ L2 ∧ … ∧ Ln → C
という変形をすることができ, これは Prolog の規則である C :- L1, L2, …, Ln と対応する。これを利用することで、機械的な推論を効率的に行うことが可能になっている。

・手続き
Prolog における手続きとは 同じ名前のヘッドを持つ規則の集合をさす。例えば、
```
member(X,[X|_]).
member(X,[_|T]) :-
  member(X,T).
```
のような一連の手続きを指す。

2章

この章では Prolog の基本的な利用として、論理データベースの定義と、データ構造の処理を取り上げ、それについて紹介している。論理データベースは事実と規則の集合で構成され, 関係データベースにおけるように、関係を事実の集合から定義する。また、関係代数のように、複雑な関係の質問をルールを用いて定義する方法を示しめしている。

キーワードと関係代数の定義

・関係スキーマ
スキーマは関係 (表) と関係内の属性 (フィールド) 、属性や関係の関連の定義である。Prolog ではこれを述語で行う。

・定義
データベースが parent, male, female という事実からのみ成り立っているとすると、father, mother という関係をルールにより定義することができる。

```
father(Dad,Child) :-
    parent(Dad,Child), male(Dad).
mother(Mum,Child) :-
    parent(Mum,Child), female(Mum).
```

このように、データベース中に暗黙に存在する関係を明示的に表すことができる。

・データの構造化
データの構造化は、プログラミング一般、とくに論理プログラミングにおいて重要であり、データの構造化を行うことにより、データを意味豊かに表現することができ、ルールについてはより抽象的に記述することができる。例えば、月曜の午前9時から12時、講義棟 M8教室で大井先生による OS の講義、というものを、構造化をしない場合とする場合

で比較してみる。

course(os,mon,9,12,lecture_building,m8,oi).
course(os,date(mon,9,12),place(lecture_building,m8),oi).

後者の方がデータまとまりがはっきりしており、また質問する時も、例えば詳細な場所が必要で、講義の日時や授業の講師についての情報が不要な場合は

?- course(os,_,place(Bulding,Room),_).

というように、必要な部分だけを取り出すことができる。(_ は 無視することを示す特殊な変数)構造化がされていない場合、

?- course(os,_,_,_,LecturerPlace,ClassRoom,_).

のように、必要な _ が増え煩雑になり、どれが何をあらわすかの判断が難しくなる。

・論理プログラムと関係データベース

論理プログラムは、関係データベースの強力な拡張とみることができる。新しいルールを記述できることにより、より高い記述力を得ることができる。論理プログラムに導入された多くの概念が、データベースにおける概念との意味深い類似性をもち、関係代数の基本演算は論理プログラムの範疇で容易に記述することができる。関係代数は集合和、集合差、直積、射影、および選択という5つの基本演算により定義される。それらのプログラムを以下に示す。

・集合和演算

それぞれ項数が n であるような2つの関係 r, s とから 項数 n の関係を新しくつくる。

```
r_union_s(X1,…,Xn) :- r(X1,…,Xn).
r_union_s(X1,…,Xn) :- s(X1,…,Xn).
```

・集合差

これを定義するには否定が必要であり、述語 not が定義されているとすると、

```
r_diff_s(X1,…,Xn) :- r(X1,…,Xn),not s(X1,…,Xn).
r_diff_s(X1,…,Xn) :- s(X1,…,Xn),not r(X1,…,Xn).
```

(処理系によっては not がビルドインでない可能性がある。)

・直積

直積は1つの rule で定義できる。 r を m 関係、 s を n 関係とすると、その直積は m + n 項関係となる。

```
r_x_s(X1,…,Xm,Xm+1,…,Xm+n) :- r(X1,…,Xm),s(Xm+1,…,Xm+n).
```

・射影

射影は既存の関係が持つ属性の中からその一部だけを使って新しい関係を形成する。例

えば、項数が3である関係 r の第1および第3引数を選択するような射影 r13 は、次のようになる。
r13(X1,X3) :- r(X1,X2,X3).

- **選択**
  選択は一部に条件を加える。第2要素が第3要素よりも大きい n 個の項からなる関係は、
  r1(X1,X2,X3,...,Xn) :- r(X1,X2,X3,...,Xn),X2 > X3.
  次に、第1要素が Smith あるいは Jones であるような関係は、宣言的関係
  smith_or_jones(smith).
  smith_or_jones(jones).
  が必要となり、次のようになる。
  r2(X1,X2,X3),smith_or_jones(X1).
  smith_or_jones(smith).
  smith_or_jones(jones).

まとめ

今回の学習では、Prolog で用いられる基本的な用語の解説を中心に学んだ。Prolog の独特な記法に慣れ、簡単なアルゴリズムや問題に対するプログラムを作成できるようになった。また今回の目的である Prolog のデータベースへの利用に向けて関係代数の基本演算を Prolog で記述する方法を学んだ。今後この本ではさらに人工知能の分野の話(探索、非決定性問題)などが続くため、更に読み進め理解を深め、最終的には卒論、修論における機械学習の研究に利用していきたい。

参考文献

[1]. Leon Sterling, Ehud Shapiro, *The Art of Prolog*, Massachusetts Institute of Technology 1994.
[2]. Ivan Bratko, 安部 憲広, *Prologへの入門*, 近代科学社, 1992.