

## 労働大臣賞（特選） 産業技術短大生のためのProlog入門

神奈川県立産業技術短期大学校 吉田 玉緒

### 1. はじめに

Prolog(プロログ)はプログラミング言語である。一般にはあまり知られていないが、人工知能のプログラミングに関しては、PrologとLISP(リスプ)がポピュラーな言語である。日本ではPrologの方が使われる割合が高いと言われている。

人工知能というとコンピュータの分野でも最先端の部分で、職業訓練には無関係なのではないかと思われることが多い。確かに、当校を卒業して人工知能システム開発に従事している者はいない。しかし、システム開発という仕事は手順の定められた作業を繰り返しているわけではない。多くの場合は、開発者が自分で考え、工夫を凝らしてシステムを構築している。「今まで手作業で行っていたが、これからはコンピュータで処理できるようにしたい」という顧客の要望を満足させるために、「手作業」の内容をコンピュータ化する作業に四苦八苦することも多い。このような場合に人工知能に関する知識・技術が役に立つことがある。

人工知能とは人間が考えて処理するようにコンピュータに処理させようという仕組みである。われわれ人間がものを考えるとき、どのように考えているだろうか。「雨が降っているから傘を持って行こう」などは、比較的わかりやすい。このような「もしならば である」という形式の知識を集めて、三段論法のように推論を行う人工知能システムにプロダクションシステムがある。また、「このケース

は のケースや のケースに似ているから同じように対処しよう」と考えることもある。大量の事例を集めて、現在問題となっている状況との類似点から推論を行うシステムに事例ベース推論がある。他にもさまざまな方法で人間の思考をコンピュータ上に擬似的に実現しようという試みがある。人工知能システムを開発する場合以外でも、人工知能の学習が一般のシステム開発に役立つことは多い。

しかしながら、2年間という限られた期間で情報技術を学習することを考えると、人工知能の学習に費やせる時間は限られたものとなる。また、学生の中には「どうせ研究者にはならないのだから人工知能の知識など不要だ」など学習意欲に欠ける者もてくる。学生たちが短い時間で興味を持って学べるようにするためにはどうしたらよieldろうか、という観点から授業に工夫を凝らしてきた。今回出品したテキスト「産業技術短大生のためのProlog入門」はそのような授業に対する工夫の一つである。

### 2. 授業内容

情報技術科2年次前期の「記号論理学」の授業で人工知能について学習する。学生の定員は40名、使用マシンはUNIXワークステーションである。

単に人工知能システムの紹介を行う講義では学生が身近な問題とは感じない。そこで、最初に知識とは何かということを考え、次にPrologについて学習し、実際に学生が簡単な人工知能システムを構築するようにした。

本テキスト「産業技術短大生のためのProlog入門」は、のPrologとの人工知能システム構築から構成されている。～の概要について以下に述べる。

## 2.1 知識とは何か

学生が最初に取り組むのは、人工知能の世界では古典的な課題「猿とバナナ」である。その内容を図1に示す。学生は、バナナを取ろうとする猿をシミュレートするプログラムをC言語で作成する。ほとんどの学生が図2に示すようなアルゴリズムを考える。この場合、猿がどこにいても、箱がどこに置いてあっても、猿は一目散に箱のところへ行き、箱をバナナの下へ押して行き、箱に登ってバナナを取る。大変に賢い猿である。しかし、その猿は本当に賢いのだろうか。賢いというのはどういうことなのか、知識とは何なのかを、学生に考えさせる。

## 2.2 Prolog実習

学生が、講義を聞いたときには理解したつもりになっていても、実習で実行してみようとしたら十分理解できていない、というのはどの教科でもあるが、Prologでは特に多いように感じられる。これはPrologプログラミングの「理解しにくさ」が原因と考えられる。

Prologは、手続き型言語ではないので、処理手続きをそのまま順に記述することができない、分岐や繰り返しなどの制御構造を明示的に記述しない、再帰を多用する、などの特徴があり、学生に「わかりにくい」という印象を与えてしまう。

そこで、授業では講義よりも実習に比重を置くこ

猿は部屋の中のある場所(仮にa地点とする)にいる。部屋のb地点にはバナナが天井からつり下げられている。猿は空腹でバナナがほしい。しかし、猿が精一杯背伸びしても届かないところにバナナはある。部屋のc地点には猿が使えるような箱がある。猿には次の行動が可能である。すなわち、床を歩く、箱を押して歩く、箱に乗る。もし、バナナの下で箱の上に立てれば、バナナをつかむことができるだろう。さて、猿はバナナを取ることができるだろうか。

図1 「猿とバナナ」の課題

ととした。学生は学習の段階ごとに例題プログラムに取り組むことでPrologの動作を理解する。次に理解をより深めるために演習課題に取り組む。この作業を繰り返して学習を進めていく。時間的には演習問題に取り組んでいる時間が最も長い。

学習内容に関しては、4.テキストの内容で詳述する。

## 2.3 人工知能システム

Prologの文法について学習した後で、実際に人工知能システムを構築する。授業時間が限られているので短時間で作成できるものにした。ここで取り上げたのはプロダクションシステムである。

プロダクションシステムは「もしならばである」というif-then型の知識とあらかじめ与えられた事実から新たな事実を導き出す。例えば、

(i) もしA市に出張ならば、A市内のホテルに泊

```

Procedure 猿とバナナ;
Begin
  % 初期設定
  猿の位置を示す変数に値を入力
  バナナの位置を示す変数に値を入力
  箱の位置を示す変数に値を入力
  if 猿の位置 = 箱の位置 then
    変数に猿が箱に乗っているかどうか入力
  else 変数に猿は箱に乗っていないと設定
  % main
  if 猿の位置 箱の位置 then
    猿の位置に箱の位置を代入
    print (猿が箱の位置へ歩く)
  if (猿の位置 = 箱の位置) and
    (猿の位置 バナナの位置) then
    猿の位置にバナナの位置を代入
    箱の位置にバナナの位置を代入
    print (猿が箱をバナナの位置へ押す)
  if (猿の位置 = 箱の位置) and (猿の位置 = バナナの位置) and (猿は箱に乗っていない) then
    変数に猿は箱に乗っていると設定
    print (猿が箱に乗る)
  if (猿の位置 = 箱の位置) and (猿の位置 = バナナの位置) and (猿は箱に乗っている) then
    print (猿がバナナをつかむ)
end
end { 猿とバナナ };

```

図2 学生が考えたアルゴリズム

まる。

(ii) もしA市内のホテルに泊まるつもりで、Bホテルに空室があればBホテルに泊まる。などの知識があるとする。このとき、「A市出張」という事実が生じれば、(i)を適用して、「A市内のホテルに泊まる」という事実が生じる。ここで、「Bホテルに空室がある」という事実が新たに加わると、(ii)を適用して「Bホテルに泊まる」という事実が生じる。

このような考え方は、人工知能システムの中では比較的簡単で理解しやすい部類に入る。if-then型の身近な例として、雑誌の記事でYes/Noで分岐して「あなたはタイプです」というものがある。どの学生もこのような記事を目にしたことがあり、これに類似しているため比較的簡単で理解しやすいと思われる。

### 3. テキスト作成の目的

前項で述べたようにPrologプログラミング学習では実習を重視している。実習において特に力を入れているのが、学生が問題を解けずにいる場合、すなわちプログラムがうまく動かない場合である。プログラミング技術の未熟な学生は、目に見える現象から、プログラムに潜むバグ(プログラムのミス)を推察することが苦手である。ここで正解を教えると学生は「そうだったのか」で終わってしまいがちで、今後はどのような点に注意してデバッグ(バグの除去)を行ったらよいかをあまり考えない。

そこで、正解は教えずに、学生が自分で問題を解けるようにバグ推察のヒントとなる現象に気づくよう働きかける。そうすると学生は自力で問題を解いた状態に近いので、バグを推察する手法を学ぶとともに自信を持つことにつながる。このような指導方法は毎回成功するわけではないが、成功すると学生は達成感を得て学習に対してより意欲的になる。

しかし、学生が間違いに気づくよう働きかけるためには、学生の思考パターンを把握し、どこに注目させたらよいかを考えなければならない。一人の学生に30分近くかかることもあり、40名の学生を掌握

して授業を行うには多大な労力を要する。

そこで、学習者が自律的に学習することと、教授者による学習者の状態把握を容易にするために支援システム<sup>\*1</sup>を構築した。本テキストは支援システムのオンラインテキストを冊子にしたものである。

習熟度の高い学生はより多くを学べるように、また、欠席などで授業の進捗が遅れても自分一人で遅れを取り戻せるように、自習が可能な内容のテキストとした。自習に近い体制で実施できるのであれば、問題を解決できずに悩んでいる学生に十分な時間を割けると考えたのである。

このため、テキストの文章は図7に示すような優しい語りかけ口調とした。また、学習の段階ごとに演習課題を設定した。演習課題はレポートとして電子メールで提出させるため、テキストに解答は記載していない。学生の進捗は支援システムの進捗データファイルに蓄積されるので、学生の進捗状況の把握は容易になる。これにより、自習に近い体制といっても、教授者が進捗の止まっている学生に声をかけたり、学生の考え違いや思い込みを見つけて注意を促したりすることが可能である。

### 4. テキストの内容

本テキストの内容を図4に示す。第1章では、マンガ「サザエさん」<sup>\*2</sup>の磯野家の家族関係を図3に示すようにPrologで記述し、Prologインタプリタで動かしてみる。図3の3行のプログラムはそれぞれ、「波平とサザエ」「波平とカツオ」「サザエとタラ」の3組の親子関係を記述したものである。ここで、

```
parent (namihei, sazae ).  
parent (namihei, katuo ).  
parent (sazae, tara ).  
(以下略)
```

図3 磯野家の家族関係

\*1 詳しくは『技能と技術』1998年1月号の「Prolog学習支援システム」をご参照ください。

\*2 長谷川町子「サザエさん」姉妹社

マンガの登場人物を用いたのは、親しみやすさとプログラムのミスを見つけやすさのためである。

磯野家の家族関係は学生誰もが知っているので、例えば、「ワカメはサザエの子か？」という質問をPrologインタプリタに行い、結果がyesとなった場合に間違いがあることがすぐにわかる。これが、「花子は洋子の子か？」ではピンとこない。

第2章～第4章ではユニファイやバックトラックなどPrologインタプリタ特有の動作について学習す

る。Prologインタプリタはプログラムに変数があった場合、その変数のとり得る値を便宜的に代入して実行する。これをユニファイという。便宜的に代入したのだから、値が妥当でなくプログラムの実行が行き詰まることがある。その場合は、実行の制御を戻して、変数の値を設定し直す。これをバックトラックという。第5章～第8章ではPrologプログラムの分岐・繰り返しなどの制御構造について学習する。これらはPrologに特有の部分なので小規模なプログラムで繰り返し学習する。

第9章～第12章ではPrologの文法について学習する。第13章では30～50ステップ程度の規模のプログラムを解析する。ここでは、無向グラフの探索と簡単な構文解析プログラムをあげた。グラフの探索は2年次前期の「グラフ理論」の授業で講義を行っている。構文解析は2年次後期の「卒業研究」で日英翻訳を希望する学生のために行う。学生は英文にwhatがあったら翻訳文として「...は、何ですか？」を出力すれば翻訳できると考えがちである。しかし、実際にはwhatが疑問詞か、関係代名詞かを認識するためには前後の単語との関連を調べる必要がある。そのために構文解析を行うのである。第14章では与えられたプロダクションシステムのプログラムをもとにif-then型の知識を追加してシステムを完成させる。

ほとんどの章は、例題プログラムを入力し、説明どおりに操作することによってPrologインタプリタの動作を確認する構成をとっている。各章の終わりには演習課題がある。

また、テキスト本文中にTea Breakとしてコラムを掲載した。内容は、

- ・ Prologの由来など
- ・ 人工知能の逸話（ELIZAシステム）
- ・ ACM<sup>\*3</sup>カリキュラム（人工知能のモデル）

である。テキストは本文のみで78ページである。

## 5．訓練への適用と効果

本テキストを適用した授業は、平成9年度および

第1章 プログラムを動かしてみる Prologインタプリタでプログラム実行の仕方について学習する。
第2章 トレースしてみる ルールの定義方法とPrologインタプリタのトレース機能について学習する。
第3章 バックトラック Prologインタプリタがバックトラックするときの処理の流れについて学習する。
第4章 Prologプログラミングの構文 データの構造とPrologが変数をユニファイする（変数にある値を当てはめる動作）仕組みについて学習する。
第5章 Prologプログラムの意味 Prologプログラムの分岐とループについて学習する。
第6章 猿はバナナを取れるか？ 「猿とバナナ」のプログラムをPrologで実行する。
第7章 プログラムの実行順序 意図しない無限ループとプログラムの効果的な記述法について学習する。
第8章 いろいろなプログラム この後の学習に有益な小規模のプログラムについて学習する。
第9章 リスト処理 リスト処理について学習する。
第10章 算術計算 算術計算について学習する。
第11章 入出力 Prologのread, write関数について学習する。
第12章 いろいろな組み込み関数 integer, assert, fail, true, repeat, bagofなどの組み込み関数について学習する。
第13章 Prolog応用プログラム 無向グラフの探索を行うプログラムと簡単な構文解析プログラムについて学習する。
第14章 人工知能を作ってみよう プロダクションシステムのプログラムにルールを追加してシステムを完成させる。

図4 テキストの内容

\* 3 The Association Computing Machinery

10年度の情報技術科2年次前期の「記号論理学」で、週1回90分17週で実施した。進捗が早い学生は最後のプロダクションシステムまで学習を進めることができたが、第8章や第9章で終わった学生もいた。

学生を対象とした授業アンケートのうち、「配布資料」の項目を以下にあげる。アンケートは「非常に良い」「良い」「悪い」「非常に悪い」の選択と自由意見で答えるものである。「普通」という選択肢は設けていない。

平成9年度生に比べて10年度生のほうが「非常に良い」「良い」が少なく、「悪い」が17%（6名）いる。配布資料の自由意見欄は平成9年度生では以下のようなものが多かった（原文のまま記載）

- ・面白い授業だった。サザエさんとかYU<sup>2</sup>白書\*<sup>4</sup>など実に身近（笑）。
- ・実行例などが多かったのがよかった。
- ・前のページをもう一度みれば、すぐ、ああそーだったっけと思うことができる点でとてもよいです。
- ・説明がたくさん書いてあったので読み進んでいけたのでわかりやすかったです。

平成10年度生でも自由意見には「わかりやすい」というものが多かったが、以下のようなものが現れた。

- ・理解するのに苦戦した。
- ・最初のほうでテキストのどの部分がプログラムなのかがよくわからなかった。最初の部分くらいもっと詳しく書いてほしかった。

この2年間はテキストの内容も授業のやり方も変えていない。他項目のアンケート結果、課題レポートの内容、学生の反応等を考えると、10年度生では自分で考えて問題を解くよりも、とにかく早く正解を教えてほしい、実習での作業をもっと細かく指示してほしいという要求が強いように感じる。年度によって学生の気質に差があるため一概には言えないが、自分で考えることの重要性を学生が認識するよう導く必要があると考えている。

\* 4 富樫義博「幽遊白書」集英社

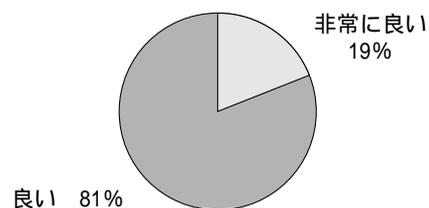


図5 平成9年度配布資料アンケート結果

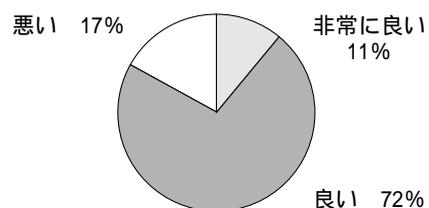


図6 平成10年度配布資料アンケート結果

## 6. おわりに

当校で実施している知識処理の授業の内容、自作テキストの内容およびその適用等について述べた。知識処理に関する技術や知識は高度で難解なものも多く、訓練時間の少ない短期大学校には向かない点もある。また、研究者になるわけではないから常識的なことだけ知っていればよいということもあるだろう。

もちろん、学生を人工知能のエキスパートにしようなどとは考えていない。ただ、今までのコンピュータ技術者や研究者が新たな問題に取り組もうとしたときに、どのように対処してきたのか、実現できていなかったものを実現しようとしたときに、どのように対処してきたのかを学ぶことによって、新たなものを作り出す喜びを感じてもらえればと思う。

本テキストにはさまざまな工夫を盛り込んでみたが、しかし、内容の面で不十分な点も多く、今後も常に改善を心がけていくつもりである。

### 【謝 辞】

本テキストの作成を含めて日常の業務についてご指導・ご意見をいただいた産業技術短期大学校の専任講師の皆さま、忌憚のない意見を聞かせてくれた学生諸君に深く感謝いたします。

この質問は、「サザエはタラの親か」という意味です。Prologは読み込んだプログラムの中に同一の節を発見し、次のように答えます。

```
yes
```

また、

```
?- parent(fune,wakame).
```

という質問に対して、prologは

```
no
```

と答えます。実際にワカメちゃんが舟さんの娘かどうかというよりも、プログラムの中にそのような記述がないことから、Prologはnoと答えるのです。だから、

```
?- parent(masuo,tara).
```

という質問に対しても、Prologはnoと答えます。noというのは「違う」というよりも「知らない」という意味に近いのです。Prologはそのような知識を持っていません、というのがnoなのです。

Prologはyes/noを答えるだけではありません。

```
?- parent(X,tara).
```

という質問は、「タラちゃんの親は誰か」という意味です。この質問に対してPrologは

```
X = sazae
```

と答えます。いま、Prologはマスオさんのことを知りませんから、タラちゃんの親はサザエさんだけです。では、カツオの場合はどうでしょう。

```
?- parent(X,katuo).
```

「カツオの親は誰か」という問に対して、波平と舟という複数の解答が可能です。このような場合、Prologは先ず1つの解答を返します。

```
X = namihei
```

これは、parent(namihei,katuo)という記述がparent(fune,katuo)より先に書かれているからです。他の解を求めるためには、セミコロンを入力します。この場合のセミコロンは「その解ではなくて、もっと別の解が欲しい」との意味になります。ですから、ここでセミコロンを入力すると、Prologは

```
X = fune
```

## Chapter1

### プログラムを動かしてみる

Prologは記号処理向けのプログラミング言語です。もっと具体的な言い方をすると、何らかの「もの(対象)」と「もの(対象)」との関係に関する問題解決に適しています。この章では例題を通してPrologの基本的なメカニズムについて考えてみます。何はともあれ、Prologを動かしてみましょう。

#### 1.1 磯野家の家族関係

先ず、人と人の関係を考えてみましょう。「波平がサザエの親である」という関係をPrologでは以下のように表現することができます。

```
parent(namihei,sazae).
```

ここでは、parentが関係の名称であり、namiheiとsazaeはその引数となっています。namiheiとsazaeは、小文字で書かなければなりません。後述しますが、大文字で始める名称は「変数名」になるからです。それでは、磯野家の家族関係の一部をPrologで定義してみましょう。

```
parent(namihei,sazae).
parent(namihei,katuo).
parent(namihei,wakame).
parent(fune,sazae).
parent(fune,katuo).
parent(sazae,tara).
```

このプログラムの1行を節と言います。文頭またはピリオドの次の文字からピリオドまでが節です。各節はparent関係について「ある人物がある人物の親である」という1つの事実を定義しています。

このプログラムをPrologに与えると、Prologはparent関係についての質問に答えることができるようになります。上の6行分のプログラムをエディタで入力し、Prologに読み込んでみましょう。質問はPrologインタプリタのプロンプトから以下のように行います。

```
?- parent(sazae,tara).
```

<sup>1</sup>長谷川町子作「サザエさん」