

組み込みプログラミング・ノウハウ入門(第34回)

動作モデルからの テスト・ケース生成(2)

——Prologを用いてステート・マシンを検証する

藤倉 俊幸

前回の最後に、Prolog を利用して動作モデルからイベント系列を網羅的に生成して、テスト・ケースとすることを述べた。今回は、この方法を具体的に説明する。

Prolog は 1970 年頃にフランス南部のマルセイユで生まれた。パイアベースと同郷のプログラミング言語で、“Programming in Logic”を短縮して Prolog と呼ばれる。Logic(論理)でのプログラミングあるいは論理に基づくプログラミングという意味である。1980 年の後半ごろには、日本語の本が数多く存在していたが、残念ながら今は古本屋で探すのも難しい。

テスト・ケースを作るためには、LTSA などで作成した動作モデルやタスク・モデルを、Prolog のプログラムに変換する手間が必要になるが、awk や perl などのスクリプトを一度作れば、簡単に変換することができる。また、LTSA でなくともステート・マシンを要求分析や設計で利用していれば、それを変換すれば良い。

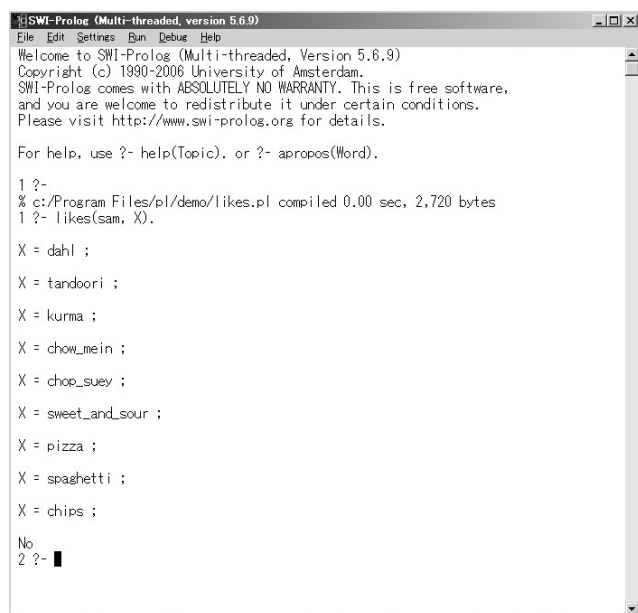


図1 SWI-Prolog 動作確認

注1：ダール、インドのレンズ豆の煮込み。

1 簡単な Prolog 講座

● フリーな実装系 SWI-Prolog

今回は、ライセンスとして GPL を採用したフリーな Prolog で、以前に本連載の第 10 回の時相論理-2 でも紹介した SWI-Prolog を使用する。http://www.swi-prolog.org/ から、ダウンロードしてダブルクリックするだけで Windows にインストールできるところが良い。当時は、バージョン 5.0 だったが、現在はバージョン 5.6 になっている。以前は、日本語のディレクトリ名が使用できなかったが 5.6 では使える。

インターネットで SWI-Prolog を検索すると、日本語の解説ページも当時より増えている。SWI-Prolog をそのまま使用するのではなく、Emacs を通して利用する方法も解説されている。Emacs を利用すれば、Prolog の出力を直接編集したり、Lisp によって変換したりできるので、より高度な開発環境を構築できる。GUI ベースの Eclipse も良いが、クールに Emacs を使用するのもお勧めである。ただし、Emacs はしばらく使用しないと使い方をすっかり忘れてしまうという欠点(?)がある。

● デモ・プログラムの実行

まず、簡単な動作確認手順として付属のデモ・プログラムを実行してみる。インストールが終了すると、Windows のスタートメニューから SWI-Prolog を起動できる。起動して、File → Consult...メニューを選択するとプログラムを選択するダイアログ・ウィンドウが開くので、SWI-Prolog をインストールしたディレクトリ下(C:/Program Files/pl/ など)の demo ディレクトリにある likes.pl というファイルを選択する。そして、図1のように、

```
likes(sam, X).
```

と打ち込む。このとき、ピリオド(.)を忘れてはいけない。そして、リターンを入力すると、

```
X = dahl 注1
```

と表示される。続いてセミコロン(;)を入力すると、

```
X = dahl ;
```

```
X = tandoori
```

と表示が更新される。セミコロンの入力を何度か繰り返すと、

```
?- likes(sam, What), write(What), nl, fail.
dahl
tandoori
kurma
chow_mein
chop_suey
sweet_and_sour
pizza
spaghetti
chips
```

図2 連続実行する場合

最後は No と表示されて図 1 の状態になる．ここまで動けば，正常にインストールできたと考えてよいだろう．

次に，このデモ・プログラムの意味について考えてみる．likes(a, b) というのは「a が b を好きだ」という意味を表す述語である．ただし，実は Prolog 自身は意味については関知していない．単に文字列のマッチングを取っているだけである．プログラムを書いた人がそのような意味を与えているのである．

述語とは，Yes か No が決まる文のようなものだとして，とりあえず考えればよい．実行のようすからすると，sam という人が好きな食べ物を話題にしているようなので，たとえば，

```
likes(sam, a 和食).
```

と入力すると，No と返事が返るので，この人は和食を好まないことがわかる(和食を a 和食としたのは，小文字から始めるようにするため．環境によっては日本語が通らないことがあるので，その場合は aWashoku などとする)．

```
likes(sam, tandoori).
```

とすると，Yes が返るので，タンドリーが好みであることがわかる．このように，Yes か No が決まる文を述語で表す．Prolog の実行中に与える述語は，Yes/No で答えてくれるので，疑問文だと思えばよい．

疑問文の中には，WH-疑問文というものがある．一つずつ，好き嫌いを聞くのがめんどろなときには，一体何が好きなのかとまとめて質問したくなる．そのような場合，

```
likes(sam, What).
```

と入力すれば良い．こうすると図 1 と同じ結果になる．図 1 の最後の No は chips が嫌いということではなく，「好きなものはもうない」という意味である．Prolog は，述語の引き数として大文字で始まる名前を使用すると，それを変数と解釈してその変数に当てはまるものをプログラムの中から探し出して教えてくれるのである．先頭が大文字かどうかだけが重要なので，What でも X でも結果は同じである．

変数を利用すると，一つずつ質問しなくても該当するものを Prolog が答えてくれるが，答えが一つ表示されるたびにセミコロンを入力しなければならない．これも，めんどろな場合は，

```
likes(sam, What), write(What), nl, fail.
```

とする．こうすると，該当するものがなくなるまでセミコロン

リスト1 likes.pl

```
likes(sam, Food) :- indian(Food), mild(Food).
likes(sam, Food) :- chinese(Food).
likes(sam, Food) :- italian(Food).

likes(sam, chips).

indian(curry).
indian(dahl).
indian(tandoori).
indian(kurma).

mild(dahl).
mild(tandoori).
mild(kurma).

chinese(chow_mein).
chinese(chop_suey).
chinese(sweet_and_sour).

italian(pizza).
italian(spaghetti).
```

ルール部

事実部

の入力なしで実行してくれる．write(What) は当てはまったものの出力，nl は改行，fail は故意に失敗させて最初からやり直すために使用する．fail すると，Prolog は likes に戻って次の該当するものを探しに行く．該当するものがなくなると likes 自身が fail するので No を出力して実行が終了する(図 2)．

● デモ・プログラムの中身

では，Prolog がどのような処理をするか likes.pl ファイルを見ながらたどってみる．likes.pl は，コメントなどを除くとリスト 1 のように，ルール部と事実部から構成されている．事実部は，Yes と答えるべき事実を列挙する部分である．ここにはないものを質問された場合，Prolog は No と答える．たとえば，

```
indian(a 和食).
```

とすると，No と答える．変数を使用した場合は，Yes になるものをすべて答えて最後に No を返す．

```
indian(X).
```

と入力すると，likes.pl に記述してある curry, dahl, tandoori, kurma^{注2}を答える．

事実部だけだと，検索して答えるだけなので，単なるデータベースでありおもしろくも何ともない．しかし，Prolog には，ルールを記述する部分があり，知識を表現することができる．Yes になるものとは，単に事実として登録されたものだけではなく，ルールによって導かれたもの，証明されたもの，推論されたものも含まれる．No になるものは，それでもだめだったものである．たとえば，

```
likes(sam, Food) :- indian(Food), mild(Food).
```

これは，「マイルドなインド料理であれば，sam は好きである」というルールを表現している．indian(Food) は Food がインド料理であれば Yes となる述語，mild(Food) は Food がマイルドであれば Yes となる述語である．カンマ(,) は論理積(AND)の意味で使用される．また，「:-」という記号は，「ならば(<-)」

注2：クルマ・南インドのカレーでラムやココナッツ・ミルクなどを入れて煮込む．ちなみに，北のカレーはコルマ(korma)と言う．