Студент: Красоткин Семён

Группа: М80-208Б-19

Вариант: 3

Лабораторная работа №2: Язык Пролог и его применение для решения логических задач

Введение

Программы, написанные на декларативных языках, состоят из набора фактов и правил. Фундамент работы в таких языков — сопоставление имеющихся фактов. В то же время также решаются логические задачи. Получается для решения логических задач весьма подходит язык Prolog, который в состоянии решить логическую задачу на основе записанных фактов.

Для решения логических задач на декларативных языках есть такие способы, как: метод генерации и проверки, метод подробного описания всех фактов. В моём задании удобно использовать метод отсечения и отрицания.

Задание

Вариант 3: “Как-то раз случай свел в купе известного астронома, поэта, прозаика и драматурга. Это были Алексеев, Борисов, Константинов и Дмитриев. Оказалось, что каждый из них взял с собой книгу, написанную одним из пассажиров этого купе. Алексеев и Борисов углубились в чтение, предварительно обменявшись

купленными книгами. Поэт читал пьесу. Прозаик, очень молодой человек, выпустивший свою первую книгу, говорил, что он никогда ничего не читает по астрономии. Борисов купил в дорогу одно из произведений Дмитриева. Никто из пассажиров не покупал и не читал книги, написанные им самим. Что читал каждый из них? Кто кем был?”

Принцип решения

Программе задаётся фамилии пассажиров из задачи и с помощью подстановки и комбинаций остальных переменных через описанные предикаты выводятся все варианты, удовлетворяющие правилам.

Пассажиры:
surname(alekseev).
surname(borisov).
surname(konstantinov).
surname(dmitriev).

Книги и профессии:
book(astronomy).
book(poetry).
book(prose).
book(piece).

Предикат, принимающий входной запрос в виде списка и выводящий ответ.
solve(Solve):-
  Solve = [passenger(X, XRead, XBuy, XWrite), passenger(Y, YRead, YBuy, YWrite),
        passenger(Z, ZRead, ZBuy, ZWrite), passenger(W, WRead, WBuy, WWrite)],

Проверка на уникальность:
  surname(X), surname(Y), surname(Z), surname(W), no_repetitions([X, Y, Z, W]),

Проверка единственности вхождения элемента в список:

```
no_repetitions([]):-!.
no_repetitions([Head|Tail]):-
   member(Head, Tail), !, fail;
   no_repetitions(Tail).
```

Проверка факта “человек не купил книгу, которую сам написал и не читал книгу, которую сам купил”.

```
check([]):-!.
check([passenger(_, XRead, XBuy, XWrite)|T]):-
  check(T),!,not(XRead = XWrite), not(XBuy = XWrite).
```

Проверка условий из задачи:

```
  % Поэт читает пьесу
  member(passenger(_, piece, _, poetry), Solve),

  % Прозаик читает не астрономию
  not(member(passenger(_, astronomy, _, prose), Solve)),

  % Прозаик не покупал астрономию
  not(member(passenger(_, _, astronomy, prose), Solve)),

  % Алексеев и Борисов обменялись книгами
  member(passenger(alekseev, AlekseevRead, AlekseevBuy, _), Solve),
  member(passenger(borisov, AlekseevBuy, AlekseevRead, _), Solve),

  % Борисов купил произведение Дмитриева
  member(passenger(dmitriev, _, _, DmitrievWrite), Solve),
  member(passenger(borisov, DmitrievWrite, _, _), Solve).
```

Результаты

?- solve(Solve).
Solve = [passenger(alekseev, prose, piece, astronomy), passenger(borisov, piece, prose, poetry), passenger(konstantinov, poetry, poetry, prose), passenger(dmitriev, astronomy, astronomy, piece)]

Выводы

Prolog позволяет решать занятные логические задачки подстановкой всех возможных значений, описанных в программе. Он их постепенно подставляет и проверяет могути ли они все выполняться в заданных условиях и выводит все возможные решения.

Удобно, когда за тебя решает задачи машина.