SWI Prolog

Стандартные предикаты управления логическим выводом

Стандартный предикат fail

Встроенный предикат fail является тождественно ложным предикатом, который включает механизм автоматического возврата. Присутствие этого предиката в правиле вызывает возврат на согласование первой цели в резольвенте.

Предикат fail используется для программирования повторяющихся действий (циклических программ). В общем случае итерационное правило имеет следующий вид:

<заголовок итерационного правила>:
—<предикаты>,fail.

Пример использования предиката fail. Текст программы

Рассмотрим пример программы, использующей предикат fail.

Пусть программа включает утверждения о столицах различных стран:

```
city('Deli', 'India', 'Asia').
city('Moscow', 'Russia', 'Europe' ).
city('Praha', 'Chehia', 'Europe' ).
city('London', 'Endland', 'Europe' ).
city('Rome', 'Italy', 'Europe' ).
city('Mexico', 'Mexica', 'America' ).
city('Pekin', 'China', 'Asia' ).
city('Tokio', 'Japan', 'Asia' ).
 show_cities(K):-city(X,_, K),write(X),nl,fail.
show_cities(_).
```

Пример использования предиката fail. Вычисление запроса.

```
?- show_cities('Asia').
Deli
Pekin
Tokio

yes
?-
```

Стандартный предикат cut (!)

Предикат отсечения (!) предназначен для запрещения поиска с возвратом.

Если существуют два или более взаимоисключающих правила для одного и того же отношения, т. е. при любом запросе успех возможен только в одном из правил, то при успешном согласовании некоторого из правил нет необходимости проверять остальные правила, согласование которых будет заведомо неуспешно.

Стандартный предикат cut (!)

Допустим, надо написать программу, определяющую значение функции Y по формуле:

$$Y = \begin{cases} f1(X), p(X) - истина; \\ f2(X), p(X) - ложь \end{cases}$$

В алгоритмических языках в этом случае используется оператор if —then —else, блок —схема которого показана следующем слайде.

Блок-схема

Соответствующее правило на языке Пролог без предиката отсечения

На языке Пролог в этом случае будет процедура из двух правил:

```
ps(X,Y): —p(X),Y is f1(X).
ps(X,Y): —not(p(X)),Y is f2(X).
```

Правило на языке Пролог с предикатом отсечения

Поскольку выше указанные правила взаимно исключают друг друга, используя предикат отсечения, процедуру можно представить в следующем виде:

```
ps(X,Y): —p(X),!,Y is f1(X).
ps(X,Y): —Y is f2(X).
```

Таким образом, во втором правиле проверка условия 'not(p(X))' не нужна.

Пример использования предиката отсечения

Рассмотрим другой пример. Пусть требуется вычислить значение функции Y в зависимости от условия:

$$Y = \begin{cases} 2 * X + 1, X \le 0 \\ X^2, 0 < X \le 2 \\ 0, X > 2 \end{cases}$$

Пример использования предиката отсечения. Текст программы.

```
ps(X,Y): —X=<0,!,Y is 2*X+1.
ps(X,Y): —X=<2,!,Y is X*X.
ps(X,Y):  —Y is 0.
```

Преимущества использования предиката отсечения

Использование предиката отсечения '!' сокращает запись правил в Прологе и ограничивает перебор вариантов, в этом случае процедура вычисления запроса выполняется эффективнее.

Другие предиката управления логическим выводом

Кроме предикатов управления логическим выводом fail и cut, в системе PDC Prolog существуют стандартные предикаты true, repeat и not.

not(p) — отрицание предиката p. Если p — истина, то not(p) — ложь, и наоборот.

Предикат repeat истинен всегда. При бэктрекинге этот предикат вызывает повторное выполнение всех следующих за ним целей.

Предикат true — тождественно истинный предикат.

Лабораторная работа 2.1. Создание базы данных “Сессия” и запросов к этой базе данных на языке Пролог с использованием стандартного предиката fail.

База данных “Сессия” содержит факты, которые описывают отношения двух типов:

1) lector(<фамилия>,<дисциплина>,<номер группы>,<дата экзамена>)
2) student(<фамилия>,<номер группы>,<номер зачетки>).

Лабораторная работа 3.1. Задание.

1)Создать с помощью фактов базу данных «Сессия», включающую предикаты lector и student.

2) Написать правила, обеспечивающие ответ на следующие вопросы:

А) Выдать на экран фамилии всех студентов, которые сдают экзамен определенного числа, дата задается следующим образом: ‘xx.xx.xxxx’.

Например, ’10.01.2001’.

Б) Выдать на экран фамилии всех студентов, которые сдают экзамен определенному лектору.

В) Выдать на экран названия дисциплин, по которым будет сдавать экзамен определенный студент.

Лабораторная работа 3.1. Задание.

Г) Выдать на экран фамилии преподавателей, которые принимают экзамены в данной группе.

Д) Выдать на экран фамилии всех студентов, которые учатся в данной группе.

Е) Выдать на экран названия дисциплин, по которым принимает экзамен данный преподаватель.

3) Отладить программу с помощью интерпретатора Arity Prolog.

4) Продемонстрировать работу программы с помощью вопросов.

5) Составить отчет по лабораторной работе.

Лабораторная работа 2.2. Использование стандартного предиката отсечения в правилах на языке Пролог.

В лабораторной работе 2 выбрать соответствующий вариант задания на вычисление y=f(x) в зависимости от условия, и переписать правила с использованием стандартного предиката отсечения «!».