Міністерство освіти і науки України

Тернопільський національний економічний університет

Факультет комп’ютерних інформаційних технологій

Кафедра КІ

Лабораторна робота №3 з дисципліни «Дослідження комп’ютерних систем штучного інтелекту»

Виконав:

Студент групи КСМм-51

Яцик С.М.

Перевірив :

Вербовий С.О.

Тернопіль-2013

**Тема:** Представлення знань на мові програмування Пролог.

**Мета:** Вивчити можливості засобів Пролог для представлення правил виводу.

SWI Prolog це сучасна кросплатформенна реалізація ПРОЛОГ системи. Після запуску вона звичайна працює в режимі інтерпретації. Аналогічно працює Turbo Prolog в пакетному режимі (в тексті програми відсутній розділ Goal). Програма у SWI Prolog не містить розділів domains, predicates, clauses та goal. Всі факти і правила подаються послідовно один за одним. Оголошувати тип аргументів предикату не потрібно.

Змінні в Пролог починаються з великої літери або зі знаку підкреслення. Змінна що позначається тільки знаком підкреслення \_ називається *анонімною* означає "будь-яку" або "деяку" величину.

Оператори порівняння і перевірки на рівність різних типів.

|  |  |
| --- | --- |
| Оператор | Пояснення |
| X=Y | Оператор = істинний якщо X та Y узгоджуються |
| X is E | Цей оператор істинний якщо Х відповідає значенню арифметичного виразу Е |
| E1 =:= E2 | Він приймає істинне значення якщо рівні значення математичних виразів Е1 та Е2. |
| E1 =\= E2 | Так можна перевірити на нерівність значення двох мат виразів |
|  | Більш строгі перевірки на рівність |
| T1 = = T2 | Приймає істинне значення якщо терми Т1 та Т2 ідентичні тобто мають повність однакову структуру включаючи однакові змінні. |
| T1\==T2 | Зворотне по смислу відношення перевірки на не ідентичність |

**Варіант 17**

Усунення несправностей багатофункціонального зовнішнього накопичувача Apacer DiskSteno CP-200. Скорочення Карта Пам’яті: КП

1) Якщо помилка виявлення КП то перевірте встановлення КП;

2) Якщо на встановленій КП файлова система не підтримується то використовуйте FAT12, FAT16, FAT32;

3) Якщо помилка зчитування КП то Встановіть іншу КП;

4) Якщо не вдається виявити диск то перевірте якість встановлення диску;

5) Якщо запис на встановлений диск неможлива то впевніться, що диск перезаписуваний;

6) Якщо помилка зчитування диску то диск пошкоджено.

***Лістинг програми для*** *інтерпретатора правил зворотного логічного виводу:*

% Автор: Яцик С.М.

% Дата: 29.10.2013

:- op( 800, fx, if).

:- op( 700, xfx, then).

:- op( 300, xfy, or).

:- op( 200, xfy, and).

:- dynamic fact/1. % можливість додавати нові факти в БД

fact(error(detection\_mc)).

fact(mc\_file\_system(not\_supported)).

fact(read\_mc(error)).

fact(not\_identify(disc)).

fact(recording\_on\_disc(not\_installed)).

fact(disk(reading\_error)).

if error(detection\_mc) then verify\_installation\_mc.

if mc\_file\_system(not\_supported) then use\_fat\_12\_16\_32.

if read\_mc(error) then set\_other\_mc.

if not\_identify(disc) then check\_the\_quality\_of\_the\_disc.

if recording\_on\_disc(not\_installed) then make\_sure\_that\_disc\_is\_rewritable.

if disk(reading\_error) then disc\_is\_damaged.

is\_true( P):-

fact( P).

is\_true( P):-

if Condition then P, %P підходяще правило, умова якого є істинною

is\_true( Condition).

is\_true( P1 and P2) :-

is\_true( P1) , is\_true( P2).

is\_true( P1 or P2) :-

is\_true( P1) ; is\_true( P2).

***Лістинг програми для*** *інтерпретатора правил прямого логічного виводу:*

% Автор: Яцик С.М.

% Дата: 29.10.2013

:- op( 800, fx, if).

:- op( 700, xfx, then).

:- op( 300, xfy, or).

:- op( 200, xfy, and).

:- dynamic fact/1. % можливість додавати нові факти в БД

fact(error(detection\_mc)).

fact(mc\_file\_system(not\_supported)).

fact(read\_mc(error)).

fact(not\_identify(disc)).

fact(recording\_on\_disc(not\_installed)).

fact(disk(reading\_error)).

if error(detection\_mc) then verify\_installation\_mc.

if mc\_file\_system(not\_supported) then use\_fat\_12\_16\_32.

if read\_mc(error) then set\_other\_mc.

if not\_identify(disc) then check\_the\_quality\_of\_the\_disc.

if recording\_on\_disc(not\_installed) then make\_sure\_that\_disc\_is\_rewritable.

if disk(reading\_error) then disc\_is\_damaged.

forward :-

new\_derived\_fact( P), % Новий виведеней факт

!,

write( 'Derived: '), write(P), nl,

assert( fact( P)), % Додати новий факт в БД

forward % Продовжити роботу

;

write( 'No more facts'). % Процес породження правил завершений

new\_derived\_fact( Conc1):-

if Cond then Conc1, % Деяке правило

not( fact( Conc1)), % Висновок правила що не є фактом БД

composed\_fact( Cond). % Довести істинність умови

composed\_fact( Cond):-

fact( Cond). % Простий факт в БД

composed\_fact( Cond1 and Cond2):- % Обидві умови істинні - логічне І

composed\_fact( Cond1),

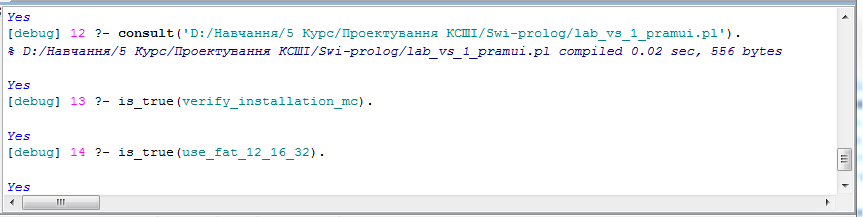
composed\_fact( Cond2).

composed\_fact( Cond1 or Cond2):- % Одна з умов істинна - логічне АБО

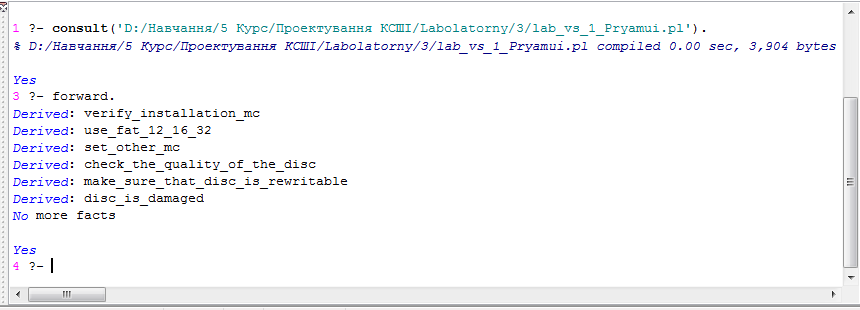
composed\_fact( Cond1);

composed\_fact( Cond2).

Скріншот роботи програми інтерпретатора правил зворотного логічного виводу



Скріншот роботи програми інтерпретатора правил прямого логічного виводу



Висновок: на лаболаторні роботі було вивчено і представлено можливості засобів Пролог для представлення правил виводу (прямого і зворотнього), а також розроблені факти і праила згідно варіантів лаболаторної роботи.