

Лабораторна робота №1.3

Механізм виконання Prolog-програми

Мета роботи: практично ознайомитись з операціями зіставлення, конкретизації, звільнення змінних при виконанні Prolog-програми.

Завдання роботи: створити та запустити в середовищі Турбо-Прологу прості програми, вивчити процес їх виконання.

Короткі теоретичні відомості

В Турбо-Пролозі розрізняють два види цілей: зовнішня і внутрішня. Зовнішня ціль (зовнішня мета) Пролог-програми – це запит, сформульований після запуску програми (runtime) безпосередньо в діалоговому вікні. Зовнішня ціль забезпечує знаходження усіх можливих рішень.

Внутрішня мета – це формулювання задачі програми безпосередньо в самому коді, в розділі goal. Така ціль забезпечує знаходження першого успішного зіставлення.

Зіставлення – це процес, на вхід якого подаються два терми, і результатом якого є висновок, про те, чи відповідають ці терми один одному. Якщо виявилось, що терми співставимі, то кажуть, що зіставлення завершилось успіхом.

Конкретизація змінної – надання змінній будь-якого значення, відмінного від неконкретизованої змінної.

Відкат – повернення до спроби узгодження попередньої цілі в разі неуспіху.

Звільнення змінної – процес, що відбувається при відкаті. Результатом звільнення є повернення змінної до стану неконкретизованої.

Порядок виконання роботи.

Завдання 1. Наберіть (раджу *copypaste*) програму, що містить відомості про батьківство й стать батьків і дітей.

```
trace
predicates
father(symbol,symbol)
```

```

m(symbol)           % male (чоловік)
f(symbol)           % female (жінка)
clauses
  father(tom,ann).
  father(bob,tom).
  father(bob,liz).
  m(tom).
  m(bob).
  f(ann).
  f(liz).

```

Запустіть програму на виконання і у вікні діалогу введіть запит – хто діти Боба (кому Боб доводиться батьком):

`father(bob, X).`

Система ТурбоПролог повинна знайти дві відповіді – **tom** і **liz**.

Завдання 2. Простежте хід пошуку розв’язку програми завд. 1 в режимі трасування. Для цього на самому початку програми слід вставити оператор трасування – **trace** і знову запустити її. Для покрокового виконання програми служить функціональна клавіша F10.

Розгляньте протокол трасування й спробуйте зрозуміти значення ключових слів, що тут зустрічаються - CALL, RETURN, REDO.

З якими значеннями пов'язується змінна X?

Завдання 3. Дослідіть процес пошуку для запиту – чи є в Боба дочка?
`father(bob,X), f(X).`

Запишіть трасувальні повідомлення і намалюйте дерево пошуку. Відмітьте (перекресліть чи намалюйте шлях повернення) гілки, на яких відбувається відкат. Відзначте шлях, що приводить до відповіді.

Завдання 4. Дослідіть дерево пошуку для запиту - знайти батька із двома дітьми:

`father(Z,X), father(Z,Y), X<>Y.`

Запишіть трасувальні повідомлення і намалюйте дерево пошуку. Помітьте (наприклад, перекресліть) гілки, на яких відбувається відкат. Відзначте шляхи, що приводять до відповіді.

Завдання 5. Зупиніть виконання програми й перейдіть у режим редагування тексту.

Додайте наприкінці програми секцію goal. Ціль, розміщена тут, називається *внутрішньою*. Розмістіть тут ту ж ціль, що ми писали у вікні діалогу в завданні 8: кому Боб доводиться батьком?

```
goal  
father(bob, X).
```

Трасування поки скасуйте (закоментуйте оператор trace).

Запустіть програму. **ЩО ВИ ОДЕРЖАЛИ?**

... Схоже, що нічого. Програма виконується, але відповіді не видно.

Завдання 6. Спробуємо таки отримати відповідь із внутрішньою метою. Для того, щоб одержати на екрані відповідь, доведеться додати до мети оператор write:

```
goal  
father(bob, X), write(X).
```

Тепер ціль буде досягнута тільки після того, як будуть виконані обидві умови: по-перше, буде знайдений такий X, якому Боб доводиться батьком, а по-друге, цей X буде надрукований (предикат **write** повертає значення TRUE в разі успішного друку своїх аргументів).

Так ми хоч про щось дізнаємось...

Запустіть програму знову.

Одержали відповідь? Чудово. А тепер порівняйте її з відповіддю, отриманою в завданні 4, і зробіть висновок.

На відміну від завд. 4, тут Пролог виводить тільки одне ім'я. Це пов'язано з тим, що **для зовнішньої мети система знаходить всі рішення, а для внутрішньої мети – тільки одне рішення.**

Завдання 7. Для одержання всіх рішень із внутрішньою метою необхідно "обдурити" систему, задавши їй ціль, яка завжди "створює неуспіх", наприклад $2=3$. Спробуйте:

```
goal  
father(bob, X), write(X), nl, 2=3.
```

Завдання 8. Для штучного створення неуспіху в Пролозі існує і широко використовується спеціальний вбудований предикат відкату (предикат неуспіху) – **fail**. Вставте fail замість « $2=3$ » і запустіть програму.

За необхідності використайте предикат `nl` (скорочення від `new line`) для переходу на новий рядок.

Відокремити відповіді одну від одної можна і в такий спосіб: `write(X, ' ')` (лише один пробіл, трактується як `char`), або `write(X, " ")`.

Завдання 9. Повторіть виконання програми з такою внутрішньою метою:

`goal`

`father(X,Y),father(X,Z),Y<>Z,write(Y, ' ',Z),nl,fail`

Що означає ця мета? (Сформулюйте цей запит природною мовою.)

А що ви одержали у відповіді?

Завдання 10. Замалюйте й вивчіть дерево трасування для внутрішньої мети по пошуку батька із двома дітьми.

Завдання 11. Щоб позбутися надлишковості відповідей, спробуйте додати до мети умову `f(Y)`. Начебто допомагає (звичайно, якщо ви змогли правильно розмістити це в програмі). Але, на жаль, це не універсальний розв'язок: якщо в Боба обоє дітей – сини, то це не спрацює.

Спробуйте, чи спрацює умова типу `Y>Z`. (Пригадаємо, що `Y` і `Z` мають тип `symbol`.) У багатьох мовах така операція над символічними рядками припустима, а тут?

Завдання 12. Додайте в програму про батьківство факт `father(tom,liz)` і складіть наступні запити:

12.1. Знайти ім'я батька, що має двох дітей.

12.2. Знайти ім'я батька, що має двох різностатевих дітей.

12.3. Знайти ім'я батька, що має двох одностатевих дітей.

12.4. Знайти ім'я дитини, у якої є двоє батьків з різними іменами (вважаючи, що це помилковий запис)

Запитання і завдання для самоперевірки

1. Перевірте, чи узiali Ви з собою зошит з конспектом лекцій?
(Без конспекта здати цю лабораторну не вийде ☺)

2. Що називають зовнішньою і внутрішньою метою Пролог-програми? Яка між ними різниця?
3. Що називають анонімною змінною? Які її властивості? Для чого вона потрібна? Наведіть приклад її використання.
4. Що називають предикатом неуспіху? Для чого він потрібен? Наведіть приклад його використання.
5. Поясніть поняття: «зв'язування змінної», «звільнення змінної», «відкат».
6. Поясніть поняття «зіставлення». Розгляньте правила зіставлення.
7. Як отримати всі розв'язки при виконанні програми з внутрішньою метою?
8. Яке значення мають предикати «fail» та «nl»?