

Лабораторна Робота № 4

Тема: Рекурсія

Надо сказать, что пример с родственными отношениями - самая распространенная иллюстрация применения систем искусственного интеллекта. Этот пример приводится в учебниках так часто и в таком примитивном виде, что скорее убеждает студентов в абсолютной бесполезности интеллектуальных систем. В результате студенты оказываются даже неспособны самостоятельно придумать аналогичный пример, не содержащий отношений «отец, сын» и т. п.

Профессор Д.В.Смолин (Россия).

Завдання роботи знаходяться в тексті.

До першої частини – завдання 1.1, 1.2, 1.3, 1.4.

До другої частини – завдання 2.1, 2.2, 2.3, 2.4, 2.5.

До третьої частини – завдання 3, 4.1, 4.2, 5, 6.

Для захисту лабораторної роботи необхідно підготувати в зошиті короткий звіт, в якому коротко відобразити виконання кожного завдання і отримані результати.

При подготовке данной работы возникли проблемы лингвистического порядка. В тексте встречаются понятия “родитель” и ”отец”, причем их точное употребление существенно для понимания того, о чем вообще идет речь. Например:

father (X, Y) :- parent(X, Y), man (X).

отец (X, Y) :- родитель(X, Y), мужчина (X).

(«X является отцом Y», если выполняются условия, что X – родитель Y,
и X – мужчина.)

Перевести это с английского языка на русский не проблематично. Но попробуйте перевести это на украинский, в котором и отец = батько, и родители = батьки!

А если родитель – женщина, то вообще выходит маразм. По-русски так: если Лиз – родитель и Лиз – женщина, то она – мать; а по-украински мы вынуждены писать: “якщо батько є жінкою, то він є матір’ю”, и т.п., и все это вносит большую путаницу.

Чтобы эту проблему как-то решить, я ничего лучше не придумал, чем оставить слово «родитель» без перевода и сделать вид, что так и надо. Кто-то сможет предложить лучший выход?

Короткі теоретичні відомості

Рекурсія (лат. Recursion) — метод визначення класу чи об'єкту через попереднє задання одного чи декількох (звичайно простих) його базових випадків чи методів, а потім заданням на їхній основі правила побудови класу, який визначається.

Іншими словами, рекурсія — часткове визначення об'єкта через себе, визначення об'єкта з використанням раніше визначених. Рекурсія використовується, коли можна виділити самоподібність задачі.

Термін «рекурсія» використовується в різних спеціальних галузях знань — від лінгвістики до логіки, але найширше застосування знаходить у математиці та інформатиці. У математиці та інформатиці рекурсія пов'язана з методом визначення функцій: рекурсивно задана функція у своєму визначенні містить себе, зокрема, рекурсивною є функція, задана рекурентною формулою. Таким чином, можна одним виразом дати нескінченний набір способів обчислення функції, визначити безліч об'єктів через саму себе з використанням раніше заданих окремих визначень. З рекурсією тісно пов'язана математична індукція: вона є природним способом доведення властивостей функцій на натуральних числах, рекурсивно заданих через свої менші значення.

Визначення у логіці, що використовує рекурсію, називається індуктивним (див., наприклад, Натуральні числа).

Рекурсія в програмуванні

Докладніше: Процедура рекурсивна

У програмуванні рекурсія — виклик функції чи процедури з неї самої (звичайно з іншими значеннями вхідних параметрів) безпосередньо чи через інші функції (наприклад, функція А викликає функцію В, а функція В — функцію А). Кількість вкладених викликів функції чи процедури називається глибиною рекурсії.

Міць рекурсивного визначення об'єкта в тім, що таке кінцеве визначення здатне описувати нескінченно велике число об'єктів. За допомогою ж рекурсивної програми можливо описати нескінченне обчислення, причому без явних повторень частин програми.

Існує спеціальний тип рекурсії, називаний «хвостовою рекурсією».

Інтерпретатори і компілятори функціональних мов програмування, що підтримують оптимізацію коду (вихідного та/або такого, що виконується), реалізують хвостову рекурсію в обмеженому обсязі пам'яті за допомогою ітерацій.

Варто уникати надлишкової глибини рекурсії, бо це може викликати переповнення стека викликів.

(Вікіпедія)

ЧАСТИНА 1. (pp23-24)

На рис.1 зображені родинні відносини між членами деякої родини.

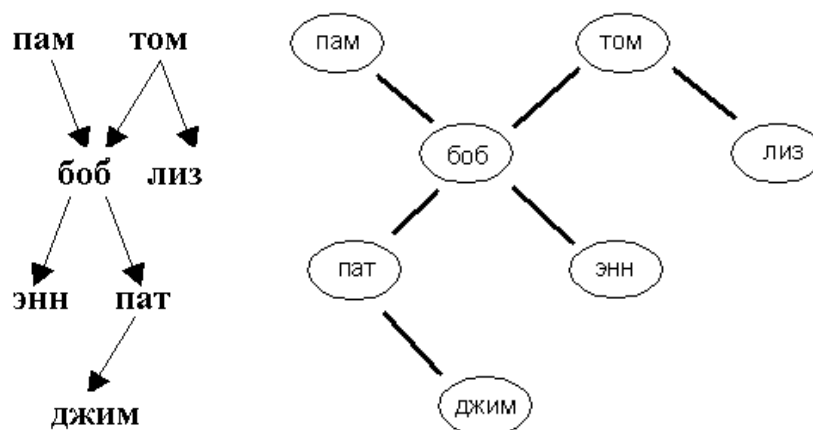


Рис.1. (одне і те ж в двох варіантах)

Щоб описати це дерево родинних відносин, необхідно описати кожную «стрілку» - відношення між двома об'єктами:

родитель (пам, боб). родитель (том, боб). родитель (боб, пат). родитель (пам, енн). родитель (том, лиз). родитель (пат, джим).

Рис. 2. Опис родини, зображеної на рис.1.

Кожне речення повідомляє про один факт наявності відносини «родитель».

ЗАВДАННЯ ДО ЧАСТИНИ 1:

Завдання 1.1. В програмі, наведеній на рис. 2, є помилка (зроблена навмисне). Знайдіть цю помилку.

Завдання 1.2. Запрограмуйте генеалогічне дерево рис.1 в Пролозі.

Вказівка: Не забудьте описати всі використані відносини в розділі `predicates`, вказуючи тип аргументів:

```
predicates
    parent(symbol, symbol).
    man(symbol).
clauses
    .....
```

А у розділі clauses пишіть тіло програми.

Завдання 1.3. Поставте системі наступні питання:

Чи є Боб родителем Пат?

Чи є Ліз родителем Пат?

Чи є Боб родителем Бена? (Це провокація – Бен на схемі взагалі не згадується. Що відповість машина?)

Хто є родителем Ліз?

Хто діти Боба? (Оскільки відношення “дитина” ми не визначали, то доведеться спитати: кому Боб доводиться родителем?)

Хто родитель Пат?

Чи є в Ліз дитина?

Хто є родителем родителя Пат? (Для того, щоб написати такий запит, потрібно спочатку визначити відношення «**родитель_родителя**». Як це написати? Спробуйте спочатку здогадатись самотійно. Якщо потрібна підказка, то подивіться зелений текст нижче - це просто шматок із книжки, але ідею одержати можна)

Завдання 1.4. Вважаючи, що відношення «родитель» визначене, знайдіть, якими будуть відповіді Пролог-системи на наступні питання:

(a) ? – родитель (джим, X).

(b) ? – родитель (X, джим).

(c) ? – родитель (пам, X), родитель (X, пат).

(d) ? – родитель (пам, X), родитель (X, Y), родитель (Y, джим).

Сформулюйте ці ж питання природною мовою.

Наше складене питання можна інтерпретувати й так:

“Знайти X и Y, що задовольняють наступним двом вимогам”:

родитель(Y, джим) і родитель(X, Y)

Якщо ми поміняємо порядок цих двох вимог, то логічний зміст залишиться незмінним:

родитель(X, Y) і родитель(Y, джим)

Це питання можна задати нашій Пролог-системі і в такій формі:

?- родитель(X, Y), родитель(Y, джим).

При цьому результат буде тим же.

У такий же спосіб можна запитати: “Хто онуки Тома?”

?- родитель(тім, X), родитель(X,Y).

Система відповість так:

X = боб

Y = енн;

X = боб

Y = пат

Наступне питання могло б бути таким: “Чи є в Єнн і Пат загальний родитель?” Його теж можна виразити у два етапи:

- (1) Який X є родителем Єнн?
- (2) чи є (той же) X родителем Пат?

Відповідний запит до Пролог-системи буде тоді виглядати так:

?- родитель (X, енн), родитель (X, пат).

Відповідь:

X = боб

:

ЧАСТИНА 2.

Розширення програми-прикладу за допомогою правил

Нашу програма-приклад можна легко розширити *дуже цікавими способами*. Давайте спершу додамо інформацію про те, якої статі люди, що беруть участь у відношенні **родитель**. Це можна зробити, просто додавши в неї наступні факти:

жінка (пам).
 % (Пам – це побутово-скорочене ім'я Памела, між іншим.)
чоловік (том).
чоловік (боб).
жінка (ліз).
жінка (пат). % (Пат – Patricia – Наприклад, Патрісія Каас)
жінка (енн).
чоловік (джим).

Рис. 3.

Ми ввели тут два нових відношення - **чоловік і жінка**. Ці відносини - унарні (або одномісні). Бінарне відношення, таке як **родитель**, визначає відношення між *двома* об'єктами; унарні ж можна використати для оголошення наявності простих властивостей у одного окремо взятого об'єкта.

Перше з наведених вище речень читається так: Пам - жінка.

Тепер повернемося до генеалогічного дерева й запишемо кілька правил.

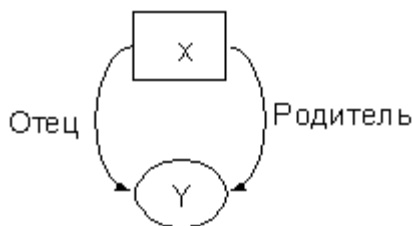


Рис.4

батько (X, Y) :- родитель(X, Y), чоловік (X). (Рис.4)

«X є батьком Y», якщо виконуються умови, що X - батько Y, і X - чоловік.

дочка (X, Y) :- родитель (Y,X), жінка (X). (Рис.5)

Ці відносини можна назвати «вертикальними».

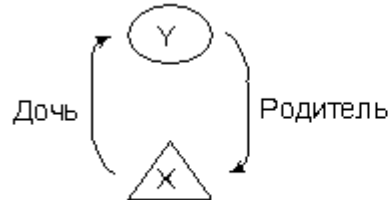


Рис.5

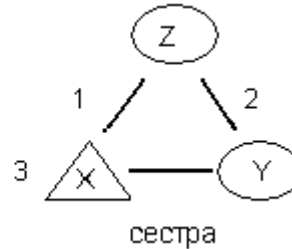


Рис.6

Відносини (чи, більш коректно, – відношення), які мають у структурі більше двох об'єктів і мають вигляд, подібний до відношення “сестра”, зображеного на рис.6, називають горизонтальними:

сестра(X,Y) :- родитель(Z, X), родитель(Z, Y), жінка(X).

Для коректності варто було б ввести сюди ще відношення **різні (X, Y)**, що є істинним тоді й тільки тоді, коли X и Y не рівні між собою. Інакше в результаті виконання програми вийде, що кожна жінка – сама собі сестра. Удосконалене правило для відношення **сестра** набуде тоді наступного вигляду:

сестра(X,Y) :-

родитель(Z, X), родитель(Z, Y), жінка(X), різні(X, Y).

(У Турбо Пролозі вимогу «різні» можна записати так: $X \neq Y$)

Досі ми розглядали відносини першого рангу. Далі розглянемо відносини другого рангу – вони дещо складніші:

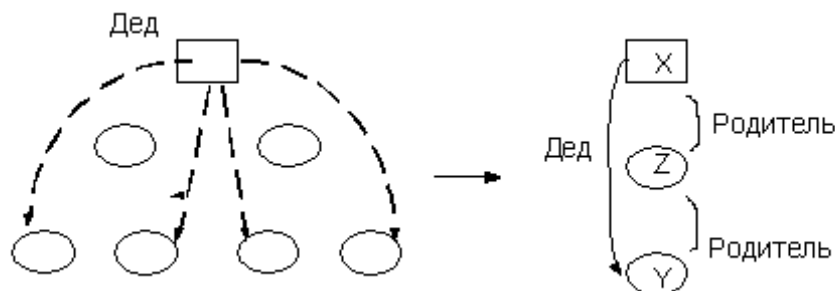


Рис.7. Відношення “дід”.

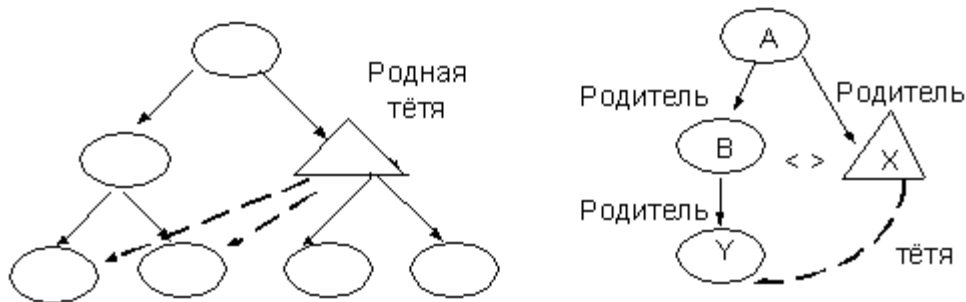


Рис.8. Відношення “тітка”.

дід (X, B) :- батько (X, Z), батько (Z, Y), чоловік (X). (Рис.7).

тітка (X, Y) :- батько (A, B), батько (A, X), батько (B, Y),
жінка (X), B <> X. (Рис.8).

ЗАВДАННЯ ДО ЧАСТИНИ 2:

Завдання 2.1. Створіть генеалогічне дерево будь-якої родини (не менше 7 чоловік) і опишіть його в Пролозі.

Спочатку опишіть базові відносини (чоловік, жінка, батько), щоб на їхній основі будувати подальші відносини - правила.

Завдання 2.2. Тепер ваше завдання - написати кілька простих відношень у вигляді правил.

Наприклад:

Визначте відношення *онук*, використовуючи відношення *батько*.

Визначте відношення *тітка*(X, Y) через відношення *батько* й *сестра*.

Складіть правила, що визначають поняття:

1 варіант - мати, дитина, син, дід, внучка, рідна тітка, племінник.

2 варіант - батько, дитина, дочка, бабуся, онук, рідний дядько, племінниця.

Для полегшення роботи можна спочатку зобразити потрібні вам відносини (тітка, тощо) у вигляді діаграм по типу наведених вище. В разі необхідності коригуйте склад родини :).

Завдання 2.3. Введіть у програму одне складне відношення (наприклад, двоюрідна бабуся).

Завдання 2.4. Запрограмуйте наступні твердження в правила на Пролозі:

Усякий, в кого є дитина, - щасливий (введіть одноаргументне відношення «щасливий»).

Усякий, в кого є двоюрідна бабуся - щасливий.

Усякий X, що має сина, у якого є сестра, має двох дітей (введіть нове відношення **має_двох_дітей**).

Завдання 2.5. Запустіть програму на виконання (Alt + R), і напишіть запити по вашим відносинам.

Наприклад:

1. «Хто батько боба?» - батько (X, боб).
2. «Чи є батьки у боба?» - батько (_, боб).
3. «Чи є пам батьком боба?» - батько (пам, боб).
4. «Чи є в представленій родині двоюрідні бабусі?»
 - двоюрідна_бабуся (X,Y).
 - двоюрідна_бабуся (_,_).
5. «Хто є онуком Тома»? (*Щоб задати таке питання, спочатку визначте відношення «онуки»*).
6. «Хто зі членів родини щасливий?»

Відповіді на запити:

1. X = Пам
X = Тім
2. Yes
3. Yes
4. X = Ліз, Y = Джим.
5.
6. (здогадаєтесь самі)

ЧАСТИНА 3. Рекурсивне визначення правил (пр. 33-34)

Давайте додамо до нашої програми про родинні зв'язки ще одне відношення - **предок**. Визначимо його через відношення **родитель**. Все відношення можна виразити за допомогою двох правил. Перше правило буде визначати безпосередніх (найближчих) предків (тобто батьків), а друге - віддалених. Будемо говорити, що деякий X є віддаленим предком деякого Z, якщо між X і Z існує ланцюжок людей, пов'язаних відносинами “**родитель**”.

Для всіх X і Z,

X - предок Z, якщо існує Y, такий, що

- (1) X - родитель Y і
- (2) Y - предок Z.

Речення Прологу, що має той же зміст, записується так:

предок(X, Z) :- родитель(X, Y), предок(Y, Z).

Тепер ми побудували повну програму для відношення **предок**. Вона містить два правила: одне для найближчих предків й інше для віддалених предків. Тут наводяться обидва правила разом:

предок(X, Z) :- родитель(X, Z).

предок(X, Z) :- родитель(X, Y), предок(Y, Z).

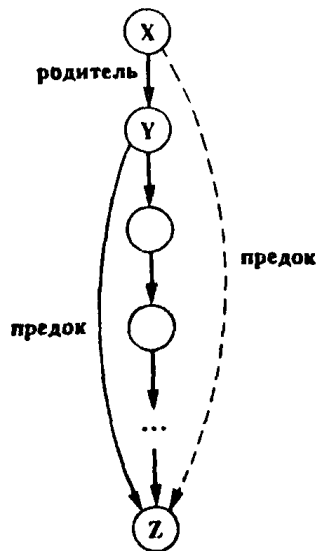


Рис. 1.7. Рекурсивне формулювання відношення **предок**.

(38)

Повертаючись до нашої програми, можна тепер задати системі питання:

“Хто предок Пам?”

?- предок(X, пам).

а також і питання

“Хто нащадки Пам?” Тобто:

“Хто та людина, чийм предком є Пам ?”

?- предок(пам, X).

X = боб;

X = єнн;

X = пат;

X = джим

Програма, наведена на рис. 1.8, визначає кілька відношень - **родитель**, **чоловік**, **жінка**, **предок** і т.д. Відношення **предок**, наприклад, визначене за допомогою двох речень. Будемо говорити, що ці два речення входять до складу відносини **предок**.

(36)

родитель(пам, боб).

% Пам - родитель Боба ПАМ-БАТЬКО БОБА

родитель(том, боб).

```

родитель( том, лиз).
родятель( боб, энн).
родитель( боб, пат).
родитель( пат, джим).
женщина( пам).           % Пам - женщина
мужчина( том).           % Том - мужчина
мужчина( боб).
женщина( лиз).
женщина( энн).
женщина( пат).
мужчина( джим).
отпрыск( Y, X) :-        % Y - отпрыск X, если
    родитель( X, Y).      % X - родитель Y
мать( X, Y) :-           % X - мать Y, если
    родитель( X, Y),      % X - родитель Y и
    женщина( X).         % X - женщина
родительродителя( X, Z) :- % X - родитель родителя Z, если
    родитель( X, Y),      % X - родитель Y и
    родитель( Y, Z).      % Y - родитель Z
сестра( X, Y) :-         % X - сестра Y
    родитель(Z, X), родитель( Z, Y) % X и Y имеют общего родителя
женщина( X, Y),         % X - женщина и
различны( X, Y).        % X отличается от Y
предок( X, Z) :-        % Правило пр1: X - предок Z
    родитель( X, Z).
прсдок( X, Z) :-        % Правило пр2: X - предок Z
    родитель( X, Y), предок( Y, Z).

```

Рис. 1.8. Програма про родинні відносини.

ЗАВДАННЯ 3. Розгляньте інший варіант відношення **предок**:

```

предок( X, Z) :-родитель( X, Z).
предок( X, Z) :-родитель( Y, Z), предок( X, Y).

```

Чи є вірним таке визначення? Чи зможете Ви змінити діаграму на рис. 1.8 таким чином, щоб вона відповідала новому визначенню?

ЗАВДАННЯ 4. Розгляньте наступну програму про родинні відносини:

```

/* Program 5 */
domains
    person = symbol
predicates
    male(person)
    female(person)
    father(person,person)

```

mother(person,person)
parent(person,person)
sister(person,person)
brother(person,person)
uncle(person,person)
grandfather(person,person)

clauses

% - 20 -

male(alan).
male(charles).
male(bob).
male(ivan).
female(beverly).
female(fay).
female(merilyn).
female(sally).
mother(beverly,marilyn).
mother(sally,alan).

father(bob,alan).
father(charles,beverly).
father(bob,fay).
father(alan,marilyn).

parent(Y,X) if mother(Y,X).
parent(Y,X) if father(Y,X).

brother(Y,X) if /*The brother of X is Y if */
male(Y) and /*Y is a male and */
parent(P,X) and /*the parent of X is P and */
parent(P,Y) and /*the parent of Y is P and */
X <> Y . /* X and Y are not the same*/

sister(Y,X) if /*The sister of X is Y if */
female(Y) and /*Y is female and */
parent(P,X) and /*the parent of X is P and */
parent(P,Y) and /*the parent of Y is P and */
X <> Y . /*X and Y are not same */

uncle(U,X) if /*The uncle of X is U if */
mother(P,X) and /*the mother of X is P and */
brother(U,P) . /*the brother of P is U */
uncle(U,X) if /*The uncle of X is U if */
father(P,X) and /*the father of X is P and */
brother(U,P) . /*the brother of P is U */

grandfather(G,X) if /*The grandfather of X is G*/
father(G,P) and /*if the father of P is G */
mother(P,X) . /*and the mother of X is P */
grandfather(G,X) if /*the grandfather of X is G*/
father(P,X) and /*if the father of X is P */

father(G,P) . /*the father of P is G */

ЗАВДАННЯ 4.1.

Запустіть цю програму й задайте системі наступні питання:

- а. Чи є Алан братом Івана?
- б. Хто дідусь Мерілін?
- в. Хто є чиею-небудь сестрою?
- г. Які відносини між Мерілін і Беверлі?

ЗАВДАННЯ 4.2.

Відносини "дядько" й "дідусь" у цій програмі описані двома реченнями кожне. Необхідно переписати дядька й дідуся, використовуючи по одному реченню для кожного з них. Які зміни необхідно для цього внести в програму?

ЗАВДАННЯ 5.

САМОСТІЙНО ПРИДУМАЙТЕ ПРИКЛАД ПРЕДМЕТНОЇ ОБЛАСТІ З ВІДНОШЕННЯМИ, ЯКІ МОЖНА ОПИСАТИ ЗА ДОПОМОГОЮ РЕКУРСІЇ, АЛЕ БЕЗ РОДИННИХ ВІДНОСИН (дивись епіграф...)

ЗАВДАННЯ 6.

Ознайомтеся з поняттям «хвостової» та «оптимальної» рекурсії. Наведіть відповідні приклади програм і дослідіть хід їх виконання.

Контрольні питання

1. Що таке «унарні відношення» та «бінарні відношення»?
2. Дати визначення відношенням першого і другого рангу.
3. Що таке «вертикальні» та «горизонтальні» відношення?
4. Що в Пролозі називається процедурою?
5. Що називається головою правила? Тілом правила?
6. Як в TURBO-PROLOG записати умову, що деякі об'єкти – «різні»?
7. Що таке рекурсивне визначення правил?
8. Як можна визначити, що у пред'явленій Вам програмі присутня рекурсія?
9. Яка помилка є в програмі на рис.3?
10. В чому різниця між запитами:
 - двоюрідна_бабуся (X,Y).
 - двоюрідна_бабуся (_,_).
11. Що таке «хвостова рекурсія», «оптимальна і неоптимальна рекурсія»?

12. В чому полягає «оптимальність» рекурсії? Яке практичне значення може мати ця оптимальність?
13. Розгляньте уважно програму з завдання 4. Ця програма має довгу і складну історію... В книжці, з якої її було взято, все було написано шиворіт-навиворіт. Наприклад: твердження, що Саллі – мати Алана, там виглядало не *mother (sally, alan)*, а навпаки: *mother(alan, sally)* . І так далі, все в такому ж дусі. Тепер цю програму вже переписано вашими попередниками більш по-людськи. Спробуйте знайти в ній помилки, і, якщо знайдете, скажіть мені.