МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ЧЕРНІГІВСЬКА ПОЛІТЕХНІКА

ОСНОВИ ПОБУДОВИ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з дисципліни

“ Основи побудови систем штучного інтелекту.”

для студентів напряму підготовки
123- "Комп'ютерна інженерія"

затверджено
на засіданні кафедри
інформаційних та комп'ютерних систем
Протокол №_ від __.__.2022 р

Чернігів 2022

.

Основи побудови систем штучного інтелекту.. Методичні вказівки до лабораторних робіт з дисципліни “Основи побудови систем штучного інтелекту.” для студентів напрямку підготовки 123 “Комп'ютерна інженерія”. / Укл. доц. В.А. Бичко – Чернігів: ЧДТУ, 2017. – 87 с. Укр. мовою.

Укладач:	Бичко В. А, кандидат фізико-математичних наук, доцент
Відповідальний за випуск:	Базилевич В.М., завідувач кафедри інформаційних та комп'ютерних систем, кандидат технічних наук, доцент
Рецензент:	Бівойно П.Г, кандидат технічних наук, доцент

Зміст

Вступ

Дисципліна "Основи побудови систем штучного інтелекту." відноситься до розряду дисциплін по вибору професійно-орієнтованого напряму "Комп'ютерна інженерія".

Необхідною передумовою для освоєння даної дисципліни є знання студентами таких навчальних курсів, як "Програмування".

Вивчення дисципліни сприяє більш глибокому розумінню інженерних задач у сфері Основи побудови систем штучного інтелекту.і освоєнню сучасних методів їх застосування.

1 Лабораторна робота №1

МЕТОДИ ПОШУКУ У ПРОСТОРІ СТАНІВ

Мета. - Отримати теоретичні та практичні навички по роботі з методами пошкуку розв‘язків задач.

1.1 Теоретичні відомості

Розрізняють локальний та системний підходи до подання інтелектуальних задач.

Локальний або «задачний» підхід заснований на точці зору, що для кожної задачі, властивої творчій діяльності людини, можна знайти спосіб її вирішення на електронній обчислювальній машині (ЕОМ), який, будучи реалізований у вигляді програми, дає результат, або подібний результату, отриманому людиною, або навіть кращий.

Системний або заснований на знаннях підхід пов’язаний із уявленням про те, що розв’язання окремих творчих задач не вичерпує всієї проблематики ШІ. Природний інтелект людини здатний не лише розв’язувати творчі завдання, а за потреби навчається того чи іншого виду творчої діяльності. Тому і програми ШІ повинні бути орієнтовані не лише або не стільки на вирішення конкретних інтелектуальних задач, скільки на створення засобів, що дозволяють автоматично будувати програми вирішення інтелектуальних задач, коли в таких програмах виникне потреба.

У такому підході проблема створення інтелектуальних систем розглядається як частина загальної теорії програмування. При цьому підході для складання інтелектуальних програм використовуються звичайні програмні засоби, що дозволяють писати потрібні програми за описами задач професійною природною мовою. Усі метазасоби, що виникають при цьому на базі часткового аналізу природного інтелекту, розглядаються тут лише з точки зору створення інтелектуального програмного забезпечення, тобто комплексу засобів, що автоматизують діяльність самого програміста.

Оскільки поняттям знання про задачи можуть відповідати стани задачи, а правилам виведення - оператори переходу з одного стану в інший від задачи до підзадачі, то процедура пошуку рішення називається *пошуком у просторі станів.*

Пошук рішень у просторі станів зводиться до визначення послідовності операторів, які відображають початкові стани в цільові. Причому якщо така послідовність не одна й задано критерій опти-

мальності, то пошук зводиться до визначення оптимальної послідовності операторів, які забезпечують оптимум заданого критерію оптимальності.

Методи пошуку рішень у просторі станів зручно розглянути, використовуючи дерево (граф) станів. На дереві станів (рис. 1.1) пошук рішення зводиться до визначення шляху (оптимального, якщо задано критерій опти¬мальності) від кореня дерева до цільової вершини A, тобто до вершини, яка відповідає цільовому стану. Вершини B і C є вершинами, що розкриваються (обчислюваними, проміжними). Вершина D термінальна, тобто завершальна. Ребра, що з'єднують вершини, означають приєднані процедури, які необхідно виконати, щоб перейти до наступного стану.

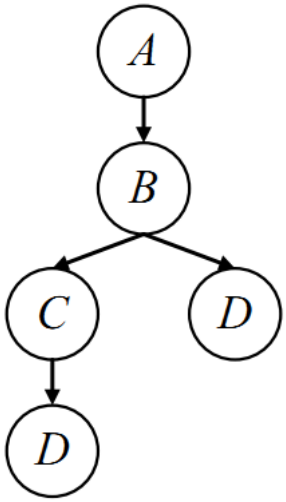

Рис. 1.1. Приклад дерева станів

Процес застосування приєднаних процедур називають породженням вершин або *перебором варіантів.* При породженні нової вершини обов'язково запам'ятовується покажчик на стару. У кінці перебору сукупність цих покажчиків утворює шлях вирішення задачі, який записується разом із іменами виконаних приєднаних процедур.

Для знаходження рішення необхідно знову і знову вибирати, перевіряти й розкривати вузли доти, поки не буде знайдено розв'язок або не залишиться більше станів, які можна було б розгорнути. Порядок, у якому відбувається розгортання станів, визначається *стратегією пошуку.*

Результатом застосування будь-якого алгоритму вирішення задачі є або невдале завершення, або отримання рішення. Деякі алгоритми можуть входити до нескінченого циклу й не повертати ніякого результату. Продуктивність алгоритму оцінюється за допомогою чотирьох показників:

- *повнота* дає відповідь на запитання, чи гарантує алгоритм виявлення рішення, якщо воно є;
- *оптимальнсть* відповідає за те, чи забезпечує дана стратегія знаходження оптимального рішення (тобто такого, яке має найменшу вартість шляху серед всіх інших рішень);

- *затрачений час,* за який алгоритм знаходить рішення;
- *необхідні ресурси,* тобто який обсяг пам’яті необхідний для здійснення пошуку.

Методи пошуку в одному просторі призначені для використання за таких умов:

- області невеликої розмірності,
- повнота моделі,
- точні та повні дані.

Стратегії пошуку в одному просторі можна класифікувати так:

1)Неінформований пошук («сліпі» методи):

- в ширину;
- в глибину (з обмеженням глибини, з ітеративним поглибленням).

2)Інформований пошук (евристичний пошук).

Пошук може здійснюватись у різних напрямках.

Прямий пошук йде від вихідного стану і, як правило, використовується тоді, коли цільовий стан задано неявно.

Зворотний пошук йде від цільового стану і використовується тоді, коли початковий стан задано неявно, а цільовий - явно.

Двонаправлений метод пошуку дозволяє одночасно проводити два пошуки в прямому і у зворотному напрямку, зупиняючись після того, як два процеси пошуку зустрінуться на середині. У такому разі передбачається перевірка в одному або в обох процесах пошуку кожного вузла перед його розгортанням для визначення того, чи не знаходиться він на периферії іншого дерева пошуку. У разі позитивного результату перевірки рішення знайдено і пошук припиняється.

Переваги цього методу:

- незначний час пошуку, оскільки перевірка приналежності вузла до іншого дерева пошуку може бути виконана за сталий час за допомогою хеш-таблиці;
- повнота методу;
- оптимальність.

Серед недоліків слід відзначити значну витрату пам’яті, оскільки необхідно зберігати принаймні одне з дерев пошуку, для того, щоб можна було виконати перевірку приналежності до іншого дерева.

На рис. 1.2 показано схематичне зображення двонаправленого пошуку в тому стані, коли він має успішно закінчитися після того, як одна з гілок, яка виходить із початкового вузла, зустрінеться з гілкою, що виходить із цільового вузла.

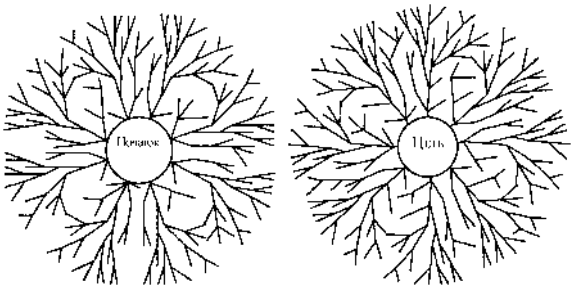

Рис. 1.2. Приклад двонаправленого пошуку

1.1.1 Методи «сліпого» пошуку

Методи *неінформованого* або «сліпого» пошуку (повного перебору) означають, що в даних стратегіях не використовується додаткова інформація про стани, крім тієї, яка подана у задачі. Такі

стратегії можуть лише виробляти наступників і відрізняти цільовий стан від нецільового. Потребують великої витрати часу.

Пошук *у ширину* є досить простою стратегією, в якій спочатку розгортається кореневий вузол, потім - усі його наступники (рис. 2.3). Після цього розгортаються наступники цих наступників і т. д. Тобто перш ніж відбувається розгортання будь- яких вузлів на наступному рівні, розгортаються всі вузли на даній конкретній глибині в дереві пошуку. Перевага - повнота. Недоліки: неоптимальний, значні витрати часу та пам’яті, оскільки необхідно зберігати всі проміжні значення.

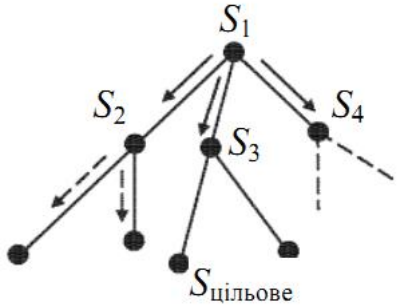

Рис. 1.3. Дерево пошуку в ширину

Пошук *у глибину* завжди розгортає найглибший вузол у дереві пошуку (рис. 1.4). Пошук безпосередньо переходить на найглибший рівень дерева пошуку, на якому вузли не мають наступників.

У міру того, як ці вузли розгортаються, вони видаляються з периферії, тому надалі пошук «відновлюється» з наступного поверхневого вузла, який все ще має недосліджених наступників.

Перевага - незначні потреби в пам’яті (зберігання лише єдиного шляху від кореня до листового вузла/

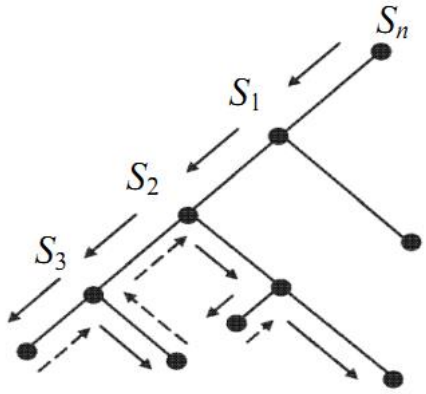

Рис. 1.4. Дерево пошуку

Недолік: може бути зроблений неправильний вибір і перехід у тупикову ситуацію, пов'язану з проходженням вниз по дуже довгому (чи навіть нескінченному) шляху, при тому, що інший варіант міг би привести до рішення, яке знаходиться недалеко від кореня дерева пошуку.

Пошук *із обмеженням глибини* передбачає застосування під час пошуку заздалегідь певної межі глибини I. Це дозволяє вирішити проблему необмежених дерев (тобто нескінченного шляху). Вузли на глибині *I* розглядаються як такі, що не мають наступників. Однак при неправильному виборі $I < d$, коли поверхнева ціль виходить за межі глибини, рішення не буде знайдено. Така ситуація цілком ймовірна, якщо значення *d* невідомо. Крім того, пошук із обмеженням глибини буде неоптимальним при виборі значення $I > d$.

Пошук у глибину *з ітеративним поглибленням* передбачає поступове збільшення глибини пошуку (яка спочатку дорівнює 1, потім 2, 3 і т. д.) доти, поки не буде знайдено ціль. Така подія відбувається після того, як межа глибини досягає значення d, глибини поверхневого цільового вузла.

У пошуку з ітеративним поглибленням поєднуються переваги пошуку в глибину та пошуку в ширину. Основні переваги методу:

незначні вимоги до пам'яті, як у пошуку в глибину;

повнота, як і в пошуку в ширину, якщо коефіцієнт розгалуження кінцевий;

оптимальність, якщо вартість шляху становить неспадну функцію глибини вузла.

Дана стратегія може здатися занадто витратною, оскільки одні й ті ж стани формуються декілька разів. Але, як виявилося, такі повторні операції не є надто дорогими. Причина цього полягає в тому, що в дереві пошуку з одним і тим же (або майже одним і тим же) коефіцієнтом розгалуження на кожному рівні більшість вузлів перебуває на нижньому рівні, тому не має великого значення те, що вузли на верхніх рівнях формуються багаторазово. У пошуку з ітеративним поглибленням вузли на нижньому рівні (з глибиною *d)* формуються один раз. Вузли, які перебувають на

рівні, що передує нижньому, формуються двічі і т. д., до дочірніх вузлів кореневого вузла, які формуються *d* разів.

1.2 Завданння до виконання роботи

1. Ознайомитися з теоретичними відомостями, розглянути та засвоїти описані тут методи пошуку у просторі станів.
2. За допомогою будь-якого середовища розробника реалізувати програмний модуль (ПМ), що виконує пошук цільової вершини у просторі станів. При цьому вхідний граф довільної конфігурації повинен бути заданим у формі матриці суміжності.
3. Протестувати роботу створеної Вами програми на 3-х різних початкових графах Провести аналіз роботи.
4. Зафіксувати результати роботи ПМ, опис та код ПМ у звіті.

1.3 Варіанти завдань до виконання роботи

№ варианту	Завдання
1	Реалізувати програму, що здійснює сліпий зворотній пошук початкової вершини в глибину на орієнтованому графі, (що містить 20 вершин) вглибину.Тестові графи створити самостійно.
2	Реалізувати програму, що здійснює сліпий прямий пошук цільової вершини в глибину на орієнтованому графі, (що містить 15 вершин) вглибину.Тестові графи створити самостійно.
3	Реалізувати програму, що здійснює сліпий двонаправлений пошук оптимального шляху між початковою та кінцевою вершинами в глибину на неорієнтованому графі, (що містить 30 вершин) вглибину..Тестові графи створити самостійно.
4	Реалізувати програму, що здійснює сліпий зворотній пошук початкової вершини в шириину на орієнтованому графі, (що містить 20 вершин) вглибину.Тестові графи створити самостійно.
5	Реалізувати програму, що здійснює сліпий прямий пошук цільової вершини в шириину на орієнтованому графі, (що містить 15 вершин) вглибину.Тестові графи створити самостійно.
6	Реалізувати програму, що здійснює сліпий двонаправлений пошук оптимального шляху між початковою та кінцевою вершинами в шириину на неорієнтованому графі, (що містить 30 вершин) вглибину..Тестові графи створити самостійно.
7	Реалізувати програму, що здійснює спрямований (евристичний) зворотній пошук початкової вершини в глибину на орієнтованому графі, (що містить 20 вершин) вглибину.Тестові графи створити

	самостійно.
8	Реалізувати програму, що здійснює спрямований (евристичний) прямий пошук цільової вершини в глибину на орієнтованому графі, (що містить 15 вершин) вглибину.Тестові графи створити самостійно.
9	Реалізувати програму, що здійснює спрямований (евристичний) двонаправлений пошук оптимального шляху між початковою та кінцевою вершинами в глибину на неорієнтованому графі, (що містить 30 вершин) вглибину..Тестові графи створити самостійно.
10	Реалізувати програму, що здійснює спрямований (евристичний) зворотній пошук початкової вершини в ширину на орієнтованому графі, (що містить 20 вершин) вглибину.Тестові графи створити самостійно.
11	Реалізувати програму, що здійснює спрямований (евристичний) прямий пошук цільової вершини в ширину на орієнтованому графі, (що містить 15 вершин) вглибину.Тестові графи створити самостійно.
12	Реалізувати програму, що здійснює спрямований (евристичний) двонаправлений пошук оптимального шляху між початковою та кінцевою вершинами в ширину на неорієнтованому графі, (що містить 30 вершин) вглибину..Тестові графи створити самостійно.
13	Реалізувати програму, що здійснює сліпий зворотній пошук початкової вершини в глибину на неорієнтованому графі, (що містить 20 вершин) вглибину.Тестові графи створити самостійно.
14	Реалізувати програму, що здійснює сліпий прямий пошук цільової вершини в глибину на неорієнтованому графі, (що містить 15 вершин) вглибину.Тестові графи створити самостійно.
15	Реалізувати програму, що здійснює сліпий зворотній пошук початкової вершини в ширину на неорієнтованому графі, (що містить 20 вершин) вглибину.Тестові графи створити самостійно.
16	Реалізувати програму, що здійснює сліпий прямий пошук цільової вершини в ширину на неорієнтованому графі, (що містить 15 вершин) вглибину.Тестові графи створити самостійно.

1.4 Завдання для самостійної роботи

1. Складіть дерево пошуку шляху від вершини А до вершини З (рис. 1.5) за стратегією в глибину.

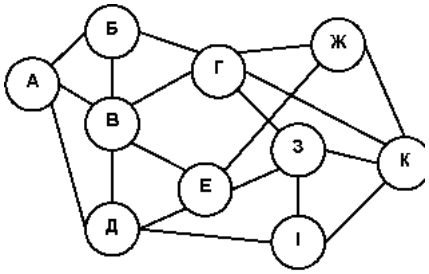

Рис. 1.9 Тестовий граф

Побудуйте дерево пошуку шляху від вершини А до вершини Е (див. рис. 1.5) за стратегією в ширину.

Побудуйте дерево пошуку шляху від вершини А до вершини К (див. рис. 1.5) за стратегією двонаправленого пошуку.

Намалюйте дерево пошуку за стратегією «в глибину» для задачі про комівояжера (див. рис. 1.5).

Знайдіть мінімальний шлях від вершини А до вершини К (див. рис. 1.5).

1.1.2 Контрольні запитання та завдання для самоперевірки

- Чим відрізняються системний та локальний підходи до подання інтелектуальних задач?
- Як реалізується пошук у просторі станів?
- Опишіть стратегію пошуку в глибину та в ширину.
- Які існують модифікації методу пошуку в глибину?
- У чому полягає відмінність евристичного пошуку від неін- формованого?
- Проаналізуйте прямий, зворотний та двонаправлений напрямки пошук.
- Як проводиться пошук рішення задачі на графі редукції?

Опишіть стратегії «сліпого» та евристичного пошуку для дерева типу «І-АБО

1.5 Вимоги до звіту

Звіт повинен містити:

- Титульну сторінку з даними про виконавця і перевіряючого, а також номер варіанта, тему і мету роботи.

-Лістинг програми.

-Результати застосування створеного ПМ до тестового графа.

-Висновки за результатами лабораторної роботи.

Звіт повинен бути оформлений відповідно до вимог СОКР.

2 Лабораторна робота №2

ПОШУК ОПТИМАЛНОГО РІШЕННЯ ЗА ДОПОМОГОЮ ГЕНЕТИЧНОГО АЛГОРИТМУ

Мета роботи: отримати навички розв'язання практичних задач за допомогою генетичних алгоритмів

2.1 Теоретичні відомості

Генетичні алгоритми (ГА) (Holland, 1969-1990) спрощено моделюють процеси природної еволюції і засновані на стохастических принципах.

Генетичні алгоритми зводяться до виконання наступних етапів:

1. Ініціалізувати популяцію.
2. Обчислити значення критерію якості для кожної особини популяції.
3. Виконати процес відтворення для кожної особини популяції.
4. Виконати схрещування і мутацію для кожної особини популяції.
5. Перевірити умову завершення. Якщо її не виконано, то повернутися до п. 2,.

Реалізація ГА зводиться до операцій з рядками: копіювання рядків, заміни фрагментів рядків і інверсії бітів.

Приклад.

Знайти

$$\max_{o \le x \le 255} f(x), \quad \text{де} \quad f(x) = \sin(\frac{\pi x}{256}),\ x \in Z\ . \ (1)$$

Функція залежить від однієї цілочисельної змінної. Особини популяції доцільно представити у вигляді бінарного рядка довжиною 1 байт.

Таблиця 7.1. Значення при ініціалізації

0	00000000
1	00000001
2	00000001
3	00000001
...	
255	11111111

Число особин однієї популяції в реальних задачах зазвичай складає 10–100. У даній задачі виберемо 8.

1. *Ініціалізація* — за допомогою генератора випадкових чисел у кожній з 8 позицій кожного рядка встановимо або 0 або 1.

Результати ініціалізації наведено в табл. 7.2.

Таблиця 7.2. Значення при ініціалізації

Особи	*x*	*fx*	*fnorm*
10111101	189	0.733	0.144
11011000	216	0.471	0.093
01100011	99	0.937	0.184
11101100	236	0.243	0.048
10101110	174	0.845	0.166
01001010	74	0.788	0.155
00100011	35	0.416	0.082
00110101	53	0.650	0.128

2. Обчислення значення критерію якості . В даномиу випадку це нормоване значення заданої функції

$$f_{norm}(x_i) = \frac{f(x_i)}{\sum_{i=1}^{N} f(x_i)},\ i = \overline{1, N}\ .$$

1. Формування нової популяції з тим же числом особин. При формуванні нової популяції використовується принцип рулетки (рис. 7.1).

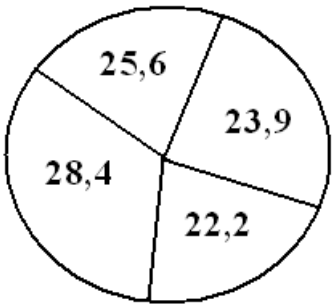

Рисунок. 7.1. Співвідношення ймовірності за критерієм якості

Результати застосування принципу рулетки показано в табл. 7.3.

Таблиця 7.3. Показники при формуванні покоління за принципом рулетки

N	Рядок	Значення	Крит. якості	% співвідн.
1	10111101	189	0,733	22,2
2	1100011	99	0,937	28,4
3	10101110	174	0,845	25,6
4	1001010	74	0,788	23,9
Разом:			3,303	100

Ймовірність влучення в кожний із сегментів пропорційна його величині.

Генеруються $N = 8$ випадкових значень з діапазону [0,1]:

$$r_i \in [0,1], i = \overline{1, N}.$$

Якщо

$$r_i \in \left[\sum_{j=1}^{i-1} f_{norm}(x_j), \sum_{j=1}^{i} f_{norm}(x_j) \right],$$

тоді

$$x \in G_{new}.$$

Наприклад, якщо $r_i \in$ [0, 0.144], то в нову популяцію включається x_1. Якщо $r_i \in$[0.144, (0.144+0.093)=0.237], то x_2 включається в нову популяцію.

Таким чином максимальна імовірність включення в нову популяцію особин з максимальним значенням критерію якості.

Візьмемо набір з 8 випадкових чисел:

0.293, 0.971, 0.160, 0.469, 0.664, 0.568, 0.371, 0.109.

Індекси особин першої популяції, що ввійдуть у наступне покоління: 3, 8, 2, 5, 6, 5, 3, 1.

Після відтворення популяція матиме вигляд:

Таблиця 7.4. Значення особин-батьків, вибраних для схрещування

01100011
00110101
11011000
10101110
01001010
10101110
01100011
10111101

4. Схрещування — основна риса генетичного алгоритму полягає в обміні частин двох батьківських особин.

(а) Вибирається ймовірність (приблизно 0.65–0.80) того, що між двома батьками відбудеться схрещування (виберемо p_c = 0.75).

(б) Популяція випадковим чином розбивається на пари. Для будь-якої пари генерується випадкове число:

$$r_k \in [0,1], \qquad k = \overline{0, \frac{N}{2}}.$$

Якщо

$$r_k < p_c < 0.75,$$

то пари піддаються схрещуванню.

(в) Для кожної з пар, що підлягають схрещуванню, випадковим чином задаються два числа (або одне число для одноточкового схрещування), що визначають границі рядка для обміну (табл. 7.3).

Таблиця 7.5. Значення при схрещуванні

Батьківська популяція	Нове покоління	x	$f(x)$
011-000-11	1110111	119	0.999
011-101-01	100001	33	0.394
1-1011-000	10101000	168	0.882
1-0101-110	11011110	222	0.405
01-00101-0	010001010	138	0.998
10-10111-0	1101110	110	0.976
01100011	01100011	99	0.937
10111101	10111101	189	0.733
Оптимальне значення		10000000	128

(г) Мутація — інвертування випадково обраних бітів (зазвичай з постійною імовірністю для кожного біта популяції, приблизно рівною 0.001–0.01).

Таким чином будь-який біт інвертується з ймовірністю 0.1%–1%. Оскільки в наведеному прикладі число бітів у популяції складає 64, то при ймовірності мутації p_m=0.001 або 0.01 швидше за все жоден біт не змінить значення.

Тепер двом особинам нового покоління відповідає значення критерію якості >0.99.

5. Перехід до нової ітерації.

2.2 Порядок виконання роботи

1. Ознайомитися з геометричним методом вирішення завдань лінійного програмування.

2. Письмово розрахуйте одну ітерацію із зазначенням всіх параметрів і проміжних результатів для задачи, що відповідає вашому варіанту.

3. Реалізувати генетичний алгоритм, пристосований для розв‘язання задачи що відповідає вашому варіанту. Доповніть створену програму графічними засобами контролю коректності роботи алгоритму.

4. За допомогою створеної програми розв’язати задачу згідно з номером вашого варіанту (розділ 7.3).

5. Результати роботи оформити звітом, який має містити: Варіанти завдань.

2.3 Варіанти завдань до виконання роботи

Враховуючи , що n – кількість особин популяції, pc – ймовірність схрещування, pm – ймовірність мутації Реалізуйте генетичний алгоритм для розв’язання задачі максимізації функції:

Варіант	Функція критерія якості	n	pc	pm
1	$f(x)=-(x-1)^2/256$, $x \in [0, 255]$	8	0.61	0.010
2	$f(x)=-(x^2-3x+2)/256$, $x \in [0, 255]$	10	0.62	0.025
3	$f(x)=-(x^2-4x+15)^2/256$, $x \in [0, 255]$	12	0.75	0.001
4	$f(x)=-(x^2-4x+3)/256$, $x \in [0, 255]$	14	0.68	0.005
5	$f(x)=-(x^2-6x+19)/256$, $x \in [0, 255]$	16	0.72	0.010
6	$f(x)=-(2x^2-5x+13)^2/256$, $x \in [0, 255]$	18	0.64	0.025
7	$f(x)=-(6x^2-5x-1)^2/256$, $x \in [0, 255]$	20	0.60	0.001
8	$f(x)=-(4x^2-4x+2)^2/256$, $x \in [0, 255]$	22	0.75	0.001
9	$f(x)=-(3x^2-15)^2/256$, $x \in [0, 255]$	24	0.62	0.005
10	$f(x)=-(17x^2-14x+15)^2/256$, $x \in [0, 255]$	26	0.75	0.010
11	$f(x)=-(x^2-5x+13)^2/256$, $x \in [0, 255]$	28	0.62	0.015
12	$f(x)=-(6x^2-4x-1)^2/256$, $x \in [0, 255]$	30	0.75	0.025
13	$f(x)=-(2x^2-4x+2)^2/256$, $x \in [0, 255]$	32	0.62	0.001
14	$f(x)=-(3x^2-12)^2/256$, $x \in [0, 255]$	34	0.75	0.001
15	$f(x)=-(x^2-4x+2)/256$, $x \in [0, 255]$	36	0.62	0.005
16	$f(x)=-(x^2-4x+14)^2/256$, $x \in [0, 255]$	38	0.75	0.010

2.4 Вимоги до звіту

Звіт повинен містити:

- Титульну сторінку з даними про виконавця і перевіряючого.
- Порядковий номер, номер варіанта, тему і мету роботи.
-Постановку задачі та опис послідовності дій при виконанні генетичного алгоритму.
- Письмовий варіант розрахунку однієї ітерації із зазначенням всіх параметрів і проміжних результатів для задачи, що відповідає вашому варіанту.
- Лістинг і інтерфейс програми з результатами її роботи результатів для задачи, що відповідає вашому варіанту.

- Висновки про виконання роботи.

Звіт повинен бути оформлений відповідно до вимог ДСТУ

3 Лабораторна робота №3

Знайомство з інструментальними засобами для створення експертних систем.

Мета - ознайомлення з оболонкою "GURU" для створення експертних систем з використанням діалогового режиму роботи і коректування бази знань існуючої експертної системи.

3.1 Теоретичні відомості

Під експертною системою розуміється система, що об'єднує можливості комп'ютера зі знаннями і досвідом експерта так, що система може запропонувати розумну пораду або здійснити розумне рішення поставленої задачі. Додатковою можливістю системи є здатність пояснити хід своїх міркувань у зрозумілій для запитувача формі.

При створенні своїх користувальницьких експертних систем на якій-небудь мові високого рівня програміст стикається з тим, що розробка інтерфейсу програми, реалізація її системних функцій вимагають більших витрат часу, ніж створення самого набору правил експертної системи (ЕС). Для того щоб розвантажити розробника ЕС від такої роботи, існують спеціальні інструментальні засоби (оболонки) експертних систем. Такі інструментальні засоби є в ЕС MYCIN, GURU, LEONARDO, DENDRAL та ін.

Цей лабораторний практикум пов'язаний з освоєнням оболонки "GURU".

1.1.1 Режими роботи

"GURU" має три режими роботи:

- Діалоговий: у ході діалогу типу "запит-відповідь" за допомогою розвиненої системи меню, не вдаючись до написання власних програм, користувач створює експертну систему;

- Природна мова: користувач на запит системи вводить фрази природною мовою і отримує результати. Наприклад, система запитує "Ваш запит?". Написавши в командному рядку фразу "Знайти всіх працюючих 1967 народження", користувач отримує від системи розумний відповідь;

- Командний: як в мовах високого рівня (МВР), пишеться програма, компілюється і працює відповідно до ваших вимог.

Звичайно застосовуються змішані режими.

3.1.1 Характеристики ЕС

Основними характеристиками є: інтерфейс користувача, машина логічних висновків і збережені експертизи.

Інтерфейс користувача описує відносини між користувачем і системою. Користувач ставить задачу, а машина должка її виконати або пояснити, чому не можна її виконати.

Машина логічних висновків - це програмне забезпечення (ПЗ), яке можна використовувати в рішенні задач шляхів аргументації.

Збережені експертизи - це набір правил, що відображають знання. У кожному правилі є посилка (IF) і висновок (THEN).

Якщо машина логічних висновків визнає посилку вірної, ТО і висновок буде вірним.

1.1.2 Оперативна допомога

Перебуваючи в будь-якому меню, можна отримати підказку по діям, допустимим в цьому меню. Для цього викликається допомога одночасним натисканням <Ctrl-L>.

1.1.3 Правила "GURU"

Система "GURU" базується на правилах. Правило складається з посилки (IF) і висновку (THEN). Посилка може включати:

- □ різні типи і види змінних, підтримуваних "GURU";
- □ -логічні оператори (EQ, NE, GT, GE, LT, LE, IN, AND, OR, XOR, NOT);
- □ числові оператори (+, -, /, *, **);
- □ числові функції (SIN, COS і т.д.);
- □ символьні функції.

Висновок може включати команди:

- □ присвоєння значення різним змінним;
- □ дозвол проконсультуватися з іншим набором правил;
- □ різні команди "GURU" і т.д.

Правила зберігаються в звичайному текстовому файлі.

приклад:

```
RULESET: EASYCALC
GOAL: INTRATE
RULE: R1
IF: MONTHPAY <50
THEN: PERIOD = 120
RULE: R2
IF: PERIOD> 90
THEN: INTRATE = 12.5
```

Тут EASYCALC - ім'я набору правил (RULESET вказувати не обов'язково);

INTRATE - ім'я змінної цілі;

R1, R2 - імена правил;

PERIOD, INTRATE, MONTHPAY - змінні.

3.1.2 Прямой висновок

Одне з важливих питань для ЕС - яке правило розглядати наступним. Цим процесом керує машина логічних висновків (МЛВ).

При виборі правила потрібно користуватися двома основними стратегіями управління: прямим і зворотним висновками.

Даний метод діє від посилки до дії до тих пір, поки змінної не буде присвоєно значення. Машина логічних висновків починає переглядати набір правил спочатку і проводить перегляд його до тих пір, поки змінній не буде присвоєно значення. Спочатку шукається перше правило, в якому й визначено справжне значення посилки. Це правило буде виконано, і отриманий результат можна буде використовувати для тестування інших правил. Далі система шукає наступне правило з певним і справжнім значенням посилки. Це продовжується до тих пір, поки не буде виконано правило для змінної цілі.

приклад:

RULESET: EASYCALC
GOAL: INTRATE
RULE: R1
IF: RERIOD> 90
THEN: INTRATE = 12.5
RULE: R2
IF: MOMTHPAY <50
THEN: PERIOD = 120
RULE: R3
IF: MOHTHPAY> 50
THEN: PERIOD = 60
RULE: R4
IF: PERIOD <90
THEN: INTRATE = 11.0

Нехай спочатку змінної MONTHPAY присвоєно значення 42. МЛВ шукає в наборі правил то правило, де визначено і справжне значення посилки (це R2). Тоді змінної PERIOD присвоюється значення 120. Слідом за тим, починаючи знову з першого правила, шукається правило, в якому визначено й справжнє значення посилки (це R1). Змінній цілі присвоюється значення 12.5. Мета досягнута, система закінчила роботу.

Спробуйте пояснити, що вийде, якщо MONTHPAY = 70.

3.1.3 Зворотний висновок

Зворотній висновок - найбільш часто використовуваний метод управління. При цьому МЛВ починає з цілі і, переглядаючи набір правил, знаходить перше правило, за допомогою якого можна досягти цілі. Якщо посилка цього правила визначена і вірна, то система виконує відповідні дії. Якщо посилка невизначена, то МЛВ тимчасово змінює мету - встановлює в якості цілі змінну, яка дозволить визначити істинність першої знайденої посилки і шукає перше правило, визначних і вірне для нової поставленої цілі.

Скористаємося прикладом з попереднього прикладу. У цій ЕС мета (GOAL) -знайти INTRATE. Шукаємо перше правило, в якому обчислювалася б змінна цілі (це R1). Але його не можна виконати, поки невідома PERIOD. Шукаємо правило, де знаходиться PERIOD (це R2). Припустимо, що MONTHPAY задано і одно 42. Тоді виконується R2 і потім R1. Мета досягнута.

Але тепер припустимо, що MONTHPAY = 70. Тоді ланцюжок R2 -R1 не приводить до знаходження цілі (PERIOD НЕ визначна як І, отже, не визначна в цьому ланцюжку і INTRATE).

Починаємо спочатку і шукаємо наступне правило, де знаходиться мета INTRATE (це R4). Тепер необхідно визначити PERIOD (нову змінну цілі). PERIOD знаходиться в правилі R3. Т.к. MONTHPAY = 70, то РЗ - вірно, тоді PERIOD = 60. Далі перевіряється R4. Воно вірно. Отже, INTRATE = 11.0.

3.1.4 Робочі змінні

Робоча змінна (РЗ) - це звичайна змінна, аналогічна змінним в ЯВУ.

Спочатку все РЗ мають значення UNKNOWN. Їм можна привласнити значення будь-якого типу.

A = 12.5 - приклад числової змінної;

B = "це строкова змінна" - приклад строкової змінної;

C = TRUE,

D - FALSE - логічні змінні.

1.1.8 Попередньо певні змінні

Існує два типи попередньо визначених змінних (ПВЗ): середовища і утиліти. Середовище "GURU" визначається змінними середовища. Вони визначають різні Функціональні характеристики середовища "GURU". Ім'я цієї змінної завжди починається з букви Е. Наприклад:

E.HELP = TRUE

Завдання ПВЗ змушує "GURU" автоматично реагувати на помилку в команді. Наприклад:

E.LSTR = 80

Максимальна довжина символьного рядка дорівнює 80.

Змінні типу утиліти служать для різних допоміжних цілей. Вони починаються зі знака #. наприклад:

#GOAL = INTRATE
COAL визначає мету ЕС.

3.1.5 Вирази зі змінними

Числові:
2 + 4
DEPTH = 5
6 + DEPTH
2 ** 2
SQRT (4) 60/5
5.67 * PI
Рядкові:
NAME = "Іванов" + "Іван" - зчеплення NAME стає рівною "Іванов Іван".
Логічні:
A = B
A <> B
A <= B
A> = B
IN - вводиться для позначення рівносильности одного елемента іншому (перевірки того, чи відповідають один одному права і ліва частини виразу); допустимо треба знайти службовця, чиє прізвище Мінев або Манев або Монєва, тоді вводимо логічне вираз:

NAME IN [M $ HEB].
Складові логічні вирази, наприклад:
15> 9 AND 20 <100 - справжнийє вираз;
9> 15 AND 20 <100 - помилковий вираз.

3.1.6 Пояснення аргументації

Важливою характеристикою "GURU" є можливість пояснити весь хід дій при консультації з набором правил. Це робиться за допомогою команд HOW і WHY. Як це робити, буде пояснено в подальшому.

3.1.7 Синтаксис правил "GURU"

Запишемо приклад згідно синтаксичним правилам "GURU" і докладно пояснимо.

Припустимо, ми проводимо технічний огляд автомобіля і хочемо знати, поїде він чи ні. Ми перевіряємо акумулятор і стартер і на основі результатів огляду приймаємо рішення. Ось наші правила в цій експертній системі:

Правило 1.

Якщо акумулятор сів або несправний стартер, то двигун не заведеться.

Правило 2.

Якщо акумулятор заряджений, справний стартер, то двигун заведеться.

Правило 3.

Якщо двигун заведеться, то автомобіль поїде.

Правило 4.

Якщо двигун не заведеться, то автомобіль не поїде.

А от як запишуться ці правила експертної системи з урахуванням синтаксису "GURU".

```
RULESET: CAR
GOAL: MOVE

INITIAL: CLEAR
E.LSTR = 80
MOVE = UNKNOWN
AKKUM = UNKNOWN
STARTER = UNKNOWN
MOTOR = UNKNOWN
OUTPUT "Система діагностики автомобіля"

VARIABLE: MOVE
LABEL: Чи буде рухатися автомобіль?
VARIABLE: AKKUM
LABEL: чи заряджена батарея?
FIND: input akkum using "u" with "Заряджений у Вас акумулятор (Y / N)?"
VARIABLE: STARTER
LABEL: Справний чи стартер?
FIND: input starter using "u" with "Справний у Вас стартер (Y / N)?"
VARIABLE: MOTOR
LABEL: Чи працює мотор?

DO:
CLEAR
OUTPUT "Автомобіль", MOVE
RULE: R1
IF: AKKUM <> "Y" OR STARTER <> "Y"
THEN: MOTOR = 0
REASON: Якщо акумулятор сів або стартер не працює, то мотор не заведеться.
RULE: R2
IF: AKKUM = "Y" AND STARTER = "Y"
THEN: MOTOR = 1
```

REASOH: ЯКЩО акумулятор заряджений і стартер працює, то мотор заведеться

RULE: R3

IF: MOTOP = 1

THEN: MOVE = "ПОЇДЕ"

REASON: Якщо мотор працює, то автомобіль поїде

RULE: R4

IF: MOTOR = 0

THEM: MOVE = "НЕ ПОЇДЕ"

Пояснюючи все більш докладно.

RULESET: CAR - це ім'я набору правил (необов'язково вказувати);

INITIAL: - це розділ ініціалізації. Сюди входять ті команди, які повинні виконатися до консультації з наборів правил.

CLEAR - команда для очищення екрана.

E.LSTR = 80 - ця змінна встановлює максимальну довжину символьних рядків.

MOVE = UNKNOWN,

MOTOR = UNKNOWN - ініціалізація змінних, при якій їм присвоюється значення UNKNOWN.

ODTPUT "Система діагностики автомобіля" - виводить на екран символьний рядок.

VARIABLE: MOVE - визначається змінна, використовувана в наборі правил. Всі змінні повинні бути визначені.

LABEL: - Пояснення на "природній мові, навіщо потрібна дана змінна.

FIND: - якщо в посилці зустрічається змінна з невизначеним значенням (UNKNOWN), яка не присутня в укладенні будь-якого правила, то виводяться ті команди, які знаходяться після FIND. Тут знаходиться команда вводу: input akkum using "u" with "Заряджений у Вас акумулятор?" Ця команда чекає введення з екрану в змінну akkum символів "Y" або "N". У цій команді вводу:

using "u" - шаблон вводу;

with "..." - виводяться на екран у вигляді запиту для підказки.

RULE: R1 - ім'я правила;

IF: - посилка правила;

THEN: - укладення правила;

REASOH -пояснення на природній мові, що робить правило.

DO - розділ завершення. Виконуються команди, які необхідні для виконання консультації з ЕС.

OUTPUT "Автомобіль", MOVE - виводиться рядок "Автомобіль" і слідом за нею змінна MOVE.

Система працює таким чином: послідовним перебором правил, починаючи з першого, знаходиться правило R3, що містить в укладенні

змінну цілі - MOVE. МЛВ визначає, що вона UNKNOWN, отже необхідно знайти умову для її знаходження. У правилі R3 це умова задається змінною MOTOR. MOTOR - змінна з невідомим значенням, що міститься в посилці правила R3, яка стає новою змінною цілі. Ця змінна при послідовному переборі правил, починаючи з першого, вперше зустрічається в укладенні правила R1. В посилці цього правила - дві змінних. Вони теж невідомі, причому їх не можна виявити в укладенні будь-якого правила, але вони описані в FIND. Отже, вводиться запит на введення цих змінних. Коли AKKUM і STARTER введені, то визначається MOTOR. Потім перевіряємо MOTOR і визначаємо MOVE. Мета досягнута.

3.2 Порядок виконання роботи

1. Запустіть "GURU" з командного рядка, для цього введіть GURU.EXE.

Система виводить на екран: ІМ'Я НОВОГО СЕАНСУ ... Введіть ім'я вашого сеансу роботи. Це ім'я буде у подальшій використовуватися для завантаження сеансу вашої роботи з "GURU".

2. Виберіть в основному меню рядок "експертні системи". У новому меню оберіть рядок "Створення експертної системи".

3. Ви будете редагувати наявний набір правил. Виберіть рядок "Існуюча база знань" (БЗ).

3.Виберіте існуючу БЗ "Animal", яка зберігається у файлі з розширенням * .rss і натисніть "Enter". Ознайомтеся зі структурою даної БЗ. Проведіть консультацію. Побудуйте дерево цієї БЗ. Доповніть дерево гілками правил, що дозволяють визначити живу істоту, згідно вашого варіанту (см. Додаток 3).

4. Виберіть рядок меню "Правила". Виберіть рядок меню "Створення". Введіть ім'я правила і натисніть «Enter». Ви потрапляєте в середовище створення правила. Перехід від одного до іншого правила - за допомогою клавіш PgDn і PgUp.

У полі IF введіть посилку. У полі THEN введіть висновок. У полі ВИСНОВОК введіть пояснення. За допомогою клавіші «ESC» вийдіть із середовища створення правила.

5. Виберіть рядок Маню "Редагування". Перегляньте введене правило. Виправте, помилки, якщо вони є.

6. Аналогічно введіть всі правила. Виберіть "Перегляд". Перегляньте всі введені правила. Виберіть "Попереднє меню" і ще раз "Попереднє меню".

7. Виберіть рядок меню "Змінні". Виберіть рядок "Створення". Ви потрапите в середу для створення змінних. Перехід між вікнами - за допомогою клавіш PgDown і PgUp.

Введіть у вікно "LABEL" пояснення.

Введіть у вікно "FIND" інформацію (якщо необхідно). Натисніть «ESC».

8. Виберіть рядок меню "Редагування". Перегляньте створену змінну. Введіть, якщо потрібно, виправлення.

9. Аналогічно створіть всі необхідні змінні. Виберіть рядок меню "Перегляд". Перегляньте всі створені змінні.

10. Виберіть двічі рядок "Попереднє меню".

11. Виберіть рядок меню "Завершення". На екрані з'являється текстовий редактор бази знань. Тут вводяться команди, які виконуються після того, як виконаний сеанс роботи з експертною системою (див. Розділ 8 цього опису лабораторної роботи). Найміть «ESC».

Виберіть рядок меню "Вихід".

Виберіть рядок меню "Збереження". Ваш набір правил зберігається у файлі з розширенням * .rss.

Виберіть "Компілювання". Відкомпілюйте ваш набір правил. Перегляньте результати компіляції. Виправте базу знань у відповідності з цими результатами і знову відкомпілюйте (якщо це буде потрібно). Для цього поверніться в попереднє меню і повторіть всі операції по пунктах 4-15, виключаючи ті, які Вам не будуть потрібні. Виберіть рядок меню "Кінець".

16. Виберіть рядок меню "Попереднє меню". Виберіть рядок: "Консультація з експертна системою". Перевірте працездатність вашої експертної системи. Якщо вона не працює, то виправте набір правил та виправте помилки.

3.3 Варіанти завдань

ВАРІАНТ №1

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «курки».

ВАРІАНТ №2

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «крокодила».

ВАРІАНТ №3

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «домашньої мухи».

ВАРІАНТ №4

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «зозулі».

ВАРІАНТ №5

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «баракуди».

ВАРІАНТ №6

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «мурени».

ВАРІАНТ №7

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «бджоли».

ВАРІАНТ №8

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «їжака».

ВАРІАНТ №9

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «лелеки».

ВАРІАНТ №10

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «тарантула».

ВАРІАНТ №211

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «рака».

ВАРІАНТ №12

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «анаконди».

ВАРІАНТ №13

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «зебри».

ВАРІАНТ №14

Доповніть Експертну систему "Animal" правилами і змінними, необхідними для визначення «криветки».

3.4 Вимоги до звіту

Звіт про роботу повинен містити:
1. Короткі теоретичні відомості з теми роботи
2. Дерево доповненої ЕС з виділеними гілками ваших доповнень.
3. Доповнений варіант експертної системи;
4. Висновки по роботі.

3.5 Додаток 1 Опис змінних середовища

Ім'я	Тип Опис	Значення за замовчуванням
E.BELL	Логічний Лунає дзвінок, якщо вводиться недійсне значення	TRUE
E.DECI	Числовий Встановлює кількість цифр праворуч від десяткової точки	2
E.HRES	Числовий Встановлює ступінь відповіді на команду HOW від 0 (немає відповіді) до 6 (найбільш докладну відповідь)	4
E.LLOG	Числовий Задає довжину логічного шаблону за замовчуванням (максимум - 5)	5
E.LSTR	(максимум - 255)	15
E.LNUM	Числовий Задає довжину для числового шаблону (максимум - 14)	14
E.OCOH	Логічний Виведення даних на дисплей	TRUE
E.OPRN	Логічний Виведення всіх вихідних даних на принтер	FALSE
E.PDEP	Числовий Довжина друкованої сторінки	60
E.WHN	Символьний Вказує, коли з'являються команди FIND при знаходженні невідомої змінної:	N - ніколи; L - тільки як останній засіб; F - перш ніж буде зроблена спроба оцінити значення невідомих змінних L

3.6 Додаток 2 Вирази та Функції GURU

Ім'я	Призначення
арифцілічн	
+	Додавання;
-	Віднімання;
*	Множення;
/	Ділення;
**	Зведення в ступінь;
MOD	ділення по модулю.
порівняння:	
EQ =	рівно;
NE, <>	не дорівнює;
GT,>	більше ніж;
LT, <	менше ніж;
GE, <=	менше або дорівнює;
LE,> =	більше або дорівнює.
логічні:	
NOT	ні;
AND, &	і;
OR	або;
XOR	що виключає "або";
=	Привласнення;
()	Індекси масиву.
строкові:	
+	Зчеплення рядків;
'	Лапка;
$	Символ відповідності символу;
*	Символ відповідності рядка.
Числові Функцій:	
ABS	абсолютне значення;
ARCSIN	арксинус;
EXP	е в ступені;
INIT	инициализирует масив;
LEN	визначає довжину рядка;
LN	обчислює натуральний логарифм;
LOG	обчислює логарифм з основою 10;
MAX	найбільше з двох чисел;
MENU	створює меню;

MIN	менше з двох чисел;
RAND	випадкове число;
SIN	синус;
SQRT	квадратний корінь.
Символьні Функції:	
CHR	перетворить код ASCII в його символьний еквівалент;
VAL	перетворить символ в його код ASCII;
INIT	инициализирует масив;
SUBSTR	виділяє підрядок з рядка;
TIME	повертає поточний час;
TOSTR	перетворить числа в символи;
TONUM	перетворить рядок в число;
TRIM	відсікає кінцеві прогалини;
TYPI	тип змінної.
Логічні Функції:	
ALPHASTR	чи вся рядок складається з букв;
INIT	инициализирует масив.

4 Лабораторна робота №4

Створення та налагодження експертних систем.

Мета - створення експертних систем з використанням діалогового режиму роботи і коректування бази знань.

4.1 Теоретичні відомості

4.1.1 Запит під час консультації

Під час консультації може створитися враження, що дії, виконувані машиною логічних висновків, не мають відношення до проблеми. Це можливо тому, що користувач не знає, як відбувається внутрішній процес аргументації. Якщо користувач дійсно не розуміє, чому від нього вимагають тієї чи іншої інформації, він може відреагувати, використовуючи Y (CTRL-Y). В цьому випадку він побачить на екрані дисплея поточне оброблюване правило. Після натискання ENTER це пояснення зникає, і він може ввести відповідь.

4.1.2 Запит після консультації

Після консультації з набором правил, користувач може попросити систему пояснити, які правила і змінні використовувалися. Для цього застосовуються дві команди:

HOW - видає змінні, які використовувалися;

WHY - пояснює правила, які використовувалися.

Пояснимо, яку інформацію дає їх використання.

HOW - видає значення змінної цілі, правило або правила, за допомогою яких було визначено мету.

HOW "ім'я змінної" - видає значення або значення з вказаною змінної.

HOW "число" - видає значення або значення змінної з порядковим номером, заданими цієї змінної в наборі правил.

WHY - відтворює на екрані дисплея пояснення (REASON) і змінні, які були потрібні для правила, що виконується останнім. Змінні відображаються з порядковий номером, який можна потім використати в команді HOW .

WHY "ім'я правила" - відтворює на екрані дисплея пояснення (REASON) і змінні, необхідні для даного правила.

WHY "число" - відтворює на екрані дисплея пояснення і змінні, необхідні для REASON правила з зазначеним порядковим номером в наборі правил.

Для того щоб пояснити процес аргументації, необхідно використовувати HOW і WHY спільно. Як це зробити, пояснюється в розділі "Порядок виконання роботи 1".

4.2 Контрольні питання

1. Що таке прямий і зворотний висновок? Назвіть основні відмінності між ними?
2. З яких частин складається правило?
3. Типи змінних та їх особливості.
4. Скільки інтерфейсів має "GURU"?
5. Що роблять команди INPUT і OUTPUT? Їх особливості.

4.3 Порядок виконання роботи

1. Запустіть "GURU" з командного рядка,
2. Виберіть в основному меню рядок "експертні системи". У новому меню оберіть рядок "Створення експертної системи".
3. Виберіть рядок "Нова база знань". З'являється підказка і запитує ім'я нового набору правил - бази знань (БЗ). Введіть ім'я нового набору правил (БЗ). Далі з'являється меню поточного набору правил "Будівник бази знань". Ви будете вводити весь набір правил, опис змінних, розділи ініціалізації і завершення заново
4. Виберіть рядок меню '"Визначення". Виберіть рядок меню "Мета". У рядку підказки ввдіть змінну цілі і натисніть «Enter». Виберіть рядок меню "Попереднє меню".
5. Виберіть рядок меню "Правила". Виберіть рядок меню "Створення". Введіть ім'я правила і натисніть «Enter». Ви потрапляєте в середовище створення правила.
6. Введіть всі правила для ЕС згідно вашого варіанту (см. Додаток 2). Виберіть "Перегляд". Перегляньте всі введені правила. Виберіть "Попереднє меню" і ще раз "Попереднє меню".
7. Виберіть рядок меню "Змінні". Створіть всі необхідні змінні. Виберіть рядок меню "Перегляд". Перегляньте всі створені змінні.
9. Виберіть двічі рядок "Попереднє меню".
10. Виберіть "ініціалізація". На екрані з'явиться текстовий редактор будівника бази знань. Тут вводяться команди, які відпрацьовуються до того, як буде запущена. Натисніть «ESC» ..
11. Виберіть рядок меню "Завершення". На екрані з'являється текстовий редактор бази знань. Тут вводяться команди, які виконуються

після того, як виконаний сеанс роботи з експертною системою (див. Розділ 8 цього опису лабораторної роботи). Найміть «ESC».

12. Виберіть рядок меню "Збереження". Ваш набір правил зберігається у файлі з розширенням * .rss.

13. Виберіть "Компілювання". Відкомпілюйте ваш набір правил. Перегляньте результати компіляції. Виправте базу знань у відповідності з цими результатами і знову відкомпілюйте.

14. Виберіть рядок меню "Попереднє меню". Виберіть рядок "Консультація з експертною системою". Перевірте працездатність вашої експертної системи. Якщо вона не працює, то виправте набір правил.

15. Щоб перевірити, як виконувалася ваша ЕС, необхідно використання HOW і WHY. Коли Ви закінчите консультацію, виберіть рядок меню "Пояснення логічних висновків". Виберіть рядок меню "Пояснення цілі". За замовчуванням - це команда HOW без параметрів. На екрані ви побачите, яке правило використовувалося для визначення значення змінної ланцюга. Після цього використовуйте WHY для того, щоб зрозуміти, чому використовувалося це правило. Якщо HOW показала, що використовувалося правило, наприклад, R10, то введіть WHY R10 для того, щоб розібратися, чому використовувалося це правило і які змінні це правило застосовувало. Скористайтеся HOW знову для того, щоб з'ясувати, які правила використовувалися для отримання значень, представлених у таблиці змінних з останньої команди WHY. Якщо з таблиці, призначеної для правила R10, стає відомо, що використовувалася змінна 8, введіть HOW 8 для того, щоб переглянути, яке правило дало змінної 8 її поточне значення. Продовжуючи цей процес, можна зрозуміти, як ЕС прийшла до зробленого нею висновку.

4.4 Вимоги до звіту

Звіт про роботу повинен містити:

1. Короткі теоретичні відомості з теми роботи
2. Дерево доповненої ЕС з виділеними гілками ваших доповнень.
3. Доповнений варіант експертної системи;
4. Висновки по роботі.

4.5 Варіанти завдань

Варіант 1

GOAL: ways

/ * Цей набір правил дозволить вам отримати ряд порад * /

```
/ * вирішення проблем вашого  авто * /
/ * причин їх появи. Звичайно, це маленький і не повний набір * /
/ * бо він  призначений для показу найтривіальніших засобів guru * /
/ *. Так що извините, будь ласка, за некоректною * /
/ * поради. * /
/ * на запитання системи слід вводяться відповідні * /
/ * значення булевскому змінною (так-TRUE, ні-FALSE) * /

INITIAL:
Clear
release variable / * прибираємо непотрібні нам змінні * /
e.lstr = 250 / * максимальна довжина рядка * /
output "добрий день, містер (місіс)."
output
output "ви, на своєму автомобілі, зупинилися перепочить "
output ". а коли набралися сил, то виявили, що
output "ваша машина не заводиться. ми постараємося дати вам поради"
output "щодо усунення причин появи несправностей."
output "але для цього ви повинні надати певну інформацію."
output "отже, поїхали"
output
fires = true
input fires logic with "чи є іскра в блоці зажигания?"

DO:
clear
Output "ось що я вам скажу, люб'язно."
Output
Output reasons
Output
Output "а ось що вам слід зробити в цій ситуації."
Output
Output ways

RULE: Rl
IF: fires
THEN: output
                input petrol logic with " чи надходить бензин в карбюратор?"
REASON: Якщо є іскра, то потрібен ще і бензин.
Comment чи: надходить бензин в карбюратор.

RULE: R2
```

IF: not fires
THEN: output
input acommulate logic with "чи окислені клеми акумулятора?"
REASON: якщо нема іскри, то швидше за все окислені клем акумулятора.

RULE: R3
IF: not petrol
THEN: output "Чи є бензин у вашому автомобілі?"
input ptrltank logic
REASON: Якщо бензин в карбюратор не надходить, то швидше за все, він просто скінчиться.
RULE: R4
IF: not ptrltank
THEN: rasons = "Закінчівся бензин у вашогму автомобілі."
ways = "заправте машину пальним."
REASON: якщо нема бензину, то треба заправитись.
COMMENT: порожній бак.

RULE: R5
IF: ptrltank
THEN: reasons = "засмічена трубка до бензонасоса."
wavs = "від'єднайте трубку та продуйте. "
ways = ways + "Потім спробуйте знову завести."
REASON: якщо бензин є, а в карбюраторі його немає, то треба прочистити трубку, яка поставляє паливо до карбюратор

RULE: R6
IF: acommulate
THEN: reasons = "поганий контакт вланцйюгі запалювання "
ways = "Слід зачистити клеми наждачною шкіркою і спробувати"
ways = ways + "завести знову."
REASON: якщо окислені контакти, то їх треба зачистити.

RULE: R7
IF: not acommulate
THEN: output "акумулятор виробив свій ресурс?"
input lowenergy logic
REASON: якщо ні іскри і контакт справні, то швидше за все ваш акумулятор став непридатним.

RULE: R8
IF: lowenergy
THEN: reasons = "ваш акумулятор став непридатний."

ways = "якщо є можливість, то замініть свій акумулятор."
ways = ways + "Або завод свої апарат 'ручкой' "
REASON: якщо сів акумулятор, то авто треба заводити 'ручкою'
COMMENT: сів акумулятор.

VAR: fires
LABEL: наявність іскри в блоці зажигания

VAR: WAYS
FIND: reasons = "причина появи цих неполадок невідома."
ways = "Постукайте ногою по колесу та толкайте ваше авто до автосервису."
LABEL: способи усунення проблем

END:

У варіанті 1 пропонується передбачити зміни, які враховують ситуації:
а) автомобіль заводиться, але не їде;
б) автомобіль заводиться, їде, але не туди, куди його направляє водій.

Варіант 2

GOAL: computer

/ * ЦЕЙ НАБІР ПРАВИЛ ДОЗВОЛИТЬ ВАМ ОТРИМАТИ РЯД РАД, * /
/ * ЯК ДІЯТИ, ЯКЩО РАПТОМ ВАШ КОМП'ЮТЕР ПРИ ВКЛЮЧЕННІ * /
/ * ЙОГО В МЕРЕЖА ВЕДЕ СЕБЕ НЕ ТАК, ЯК ЗВИЧАЙНО. І ОСЬ, в залежність для * /
/ * НОСТІ ВІД ЗОВНІШНЬОГО ПРОЯВИ ці дивацтва ВАМ БУДЕ * /
/ * ДАН РАДА, Ви повинні:. ЗВИЧАЙНО, ЦЕ малень- * /
/ * КІІ І ДАЛЕКО НЕПОВНИЙ НАБІР, І, МОЖЛИВО, ПОРАДИ ВАМ поки- * /
/ * Жутся недоречним, АЛЕ Ж ВІН І ПРИЗНАЧЕНИЙ ДЛЯ ПОКАЗУ * /
/ * Найтривіальніших ЗАСОБІВ GURU. * /
/ * НА ЗАПИТАННЯ СИСТЕМИ СЛІД вводяться відповідні * /
/ * ЗНАЧЕННЯ булевскому ЗМІННОЮ (ДА-TRUE, НЕТ-FALSE) * /

```
INITIAL:
clear
release variable / * ПРИБИРАЄМО НЕПОТРІБНІ НАМ ЗМІННІ * /
e.lstr = 250 / * Максимальна довжина рядка * /
output "ДЕНЬ ДОБРИЙ, МІСТЕР (МІСІС)."
output
output "увімкненого комп'ютера ВИ НЕ ОТРИМУЄТЕ ЗВИЧАЙНОГО"
output "РЕЗУЛЬТАТУ. ми намагаємось дати ВАМ РАДА В ЦЬОМУ НЕ-"
output "простій справі. АЛЕ ДЛЯ ЦЬОГО ВИ ПОВИННІ НАДАТИ"
output "МЕНІ ВРЮ ІНФОРМАЦІЮ. ОТЖЕ, НАЧНЕМ"
power = true
output "засвічується індикатор ХАРЧУВАННЯ на вашому комп'ютері?"
input power logic

DO:
clear
output "ОСЬ ЩО Я ВАМ СКАЖУ, люб'язно."
output
output computer

RULE: R1
IF: power
THEN: output
output "НУ ХОЧ ХАРЧУВАННЯ У ПОРЯДКУ, І ТО ДОБРЕ!"
output "виправив і ПРАВИЛЬНО ЧИ ПІД'ЄДНАНО ДІСПЛЕЙ?"
input displ logic
REASON: ЯКЩО В ПОРЯДКУ ХАРЧУВАННЯ, ТО У ПОРЯДКУ ЧИ ДІСПЛЕЙ.
COMMENT: ДЛЯ, ТОГО ЩОБ ЩОСЬ ПОБАЧИТИ, НЕОБХІДНО пристроїв відображення інформації.

RULE: R2
IF: not power
THEN: output
computer = "ПРОДІАГНОСТУЙТЕ НАПРУГУ В МЕРЕЖІ. УВІМКНІТЬ пита-"
computer = computer + "НІЕ або полагодити БЛОК ХАРЧУВАННЯ ТА"
computer = computer + "ПОПРОБУЙТЕ ЩЕ РАЗ."
REASON: ЯКЩО НІ ХАРЧУВАННЯ, ТО НЕОБХІДНО ЙОГО ЗАБЕЗПЕЧИТИ
```

COMMENT: БЕЗ НОРМАЛЬНОГО ХАРЧУВАННЯ НІХТО ВАМ ПРАЦЮВАТИ НЕ БУДЕ.

RULE: R3
IF: not displ
THEN: output
computer = "ПРОВЕРЬТЕ НАПРУГУ МЕРЕЖІ. активувати екран"
computer = computer + "або влаштувати ТАК, ЩОБ ВІН ПРАЦЮВАВ"
computer = computer + "І СКУШТУЙТЕ ЩЕ РАЗ."
REASON: НЕ ПРАЦЮЄ ДИСПЛЕЇ. НЕОБХІДНО, ЩОБ ВІН ПРАЦЮВАВ.
COMMENT: БЕЗ ПРАЦЮЮТЬ ДИСПЛЕЯ - НЕ ЖИТТЯ.

RULE: R4
IF: displ
THEN: output "ЧИ Є У ВАШОГО КОМП'ЮТЕРА ЖОРСТКИЙ ДИСК?"
input harddisk logic
REASON: ЯКЩО ХАРЧУВАННЯ КОМП'ЮТЕРА І ДИСПЛЕЇ ГАРАЗД, ТО ТРЕБА ЗНАТИ, ПІД'ЄДНАНО ЧИ ДО ВАШОЇ МАШИНІ ЖОРСТКИЙ ДИСК.
COMMENT: ПІД'ЄДНАНО ЧИ ЖОРСТКИЙ ДИСК.

RULE: R5
IF: harddisk
THEN: output "ВІДБУВАЄТЬСЯ звернення до диска (ИНДИКАТОР"
output "ГОРИТЬ)?"
input HDtest logic
REASON: ЯКЩО ВСТАНОВЛЕНО "ВІНЧЕСТЕР", ТО перевірити підключення.
COMMENT: ЯКЩО "ВИНТ" НА МІСЦІ, ТО Чи помічали ЙОГО СИСТЕМА.

RULE: R6
IF: not harddisk
THEN: output "ВСТАНОВЛЕНІ НА ВАШІЙ МАШИНІ дисковод гинув-"
output "КИХ ДИСКІВ (1 І БІЛЬШЕ)?"
input diskete logic
REASON: ЯКЩО НІ "ГВИНТА", ТО дисководи на МАШИНІ ВСТАНОВЛЕНІ?
COMMENT: ЧИ дисководи.

RULE: R7
IF: not HDtest
output "ВСТАНОВЛЕНІ чи інструмент дисководів"
output "ГНУЧКИХ ДИСКІВ (1 І БІЛЬШЕ)?
THEN: input diskete logic
REASON: ВСТАНОВЛЕНІ ЧИ дисководи на МАШИНІ?
COMMENT: ЧИ дисководи.

RULE: R8
IF: HDtest
THEN: output "Чи видає ПОВІДОМЛЕННЯ ПРО ПОМИЛКУ ЧИТАННЯ 'Bi-"
output "НЧЕСТЕРА '?"
input HDerror logic
REASON: БЕЗ ПОМИЛОК ЧИ ПРОХОДИТЬ ОПЕРАЦІЯ ТЕСТУВАННЯ.
COMMENT: НАЯВНІСТЬ ПОМИЛОК ТЕСТУВАННЯ ДИСКУ.

RULE: R9
IF: HDerror
THEN: output "ВСТАНОВЛЕНІ чи інструмент дисковод ги-"
output "БКІХ ДИСКІВ (1 І БІЛЬШЕ)?"
input diskete logic
REASON: ЯКЩО Є ПОМИЛКИ В РОБОТІ "ГВИНТА", ТО ЦІКАВО, ВСТАНОВЛЕНІ ЧИ дисководи на МАШИНІ? COMMENT: ЧИ дисководи.

RULE: R10
IF: diskete
THEN: computer = "ВАМ СЛІД вставити в дисковод СИСТЕМНУ"
computer = computer + "дискеті відповідає формату"
computer = computer + "І ПЕРЕЗАВАНТАЖИТИ КОМП'ЮТЕР."
REASON: ЯКЩО "ВИНТ" НЕ В ПОРЯДКУ, ТО СЛІД ВИКОРИСТОВУВАТИ ГНУЧКИЙ ДИСК ДЛЯ ЗАВАНТАЖЕННЯ СИСТЕМИ.
COMMENT: СЛІД ВИКОРИСТОВУВАТИ ГНУЧКИЙ ДИСК ДЛЯ ЗАВАНТАЖЕННЯ СИСТЕМИ.

RULE: R11
IF: not diskete
THEN: computer = "ВАМ НАЙКРАЩЕ ДАЛІ НЕ поглиблювати В"
computer = computer + "ЦЮ ПРОБЛЕМУ, А ВИКЛИКАТИ МАЙСТРА."
REASON: ЯКЩО НЕ ПРАЦЮЮТЬ (ВІДСУТНІ) дисків і, ТО НЕ ВАРТО НАМАГАТИСЯ ПЕРЕЗАВАНТАЖИТИ МАШИНУ.

COMMENT: НЕ ПРАЦЮЮТЬ дисків і, отже, не варто намагатися ПЕРЕЗАВАНТАЖИТИ МАШИНУ.

RULE: R12
IF: not HDerror
THEN: computer = "ЯКЩО НЕ З`ЯВЛЯЄТЬСЯ ЗАПРОШЕННЯ ОПЕРАЦІЙНОЇ"
computer = computer + "СИСТЕМИ, ТО ПРОДІАГНОСТУЙТЕ СОДЕРЩІМОЕ"
computer = computer + "ФАЙЛІВ autoexec.bat I config.sys,"
computer = computer + 'ЗАГРУЗІВШІСЬ С ДИСКЕТИ. І ЯКЩО ТАМ "
computer = computer + "ВСЕ ГАРАЗД, перезапишіть ОС."
REASON: ЯКЩО НІ ПОМИЛКИ ЧИТАННЯ "Вінчестер" ТО ВСЕ ЩЕ МОЖНА ВИПРАВИТИ, АБИ приводі дискет був справний.
COMMENT: НІ ПОМИЛКИ ЧИТАННЯ "ГВИНТА".

VAR: POWER
LABEL: спрацьовування світлової індикації увімкненого комп'ютера
END:

У варіанті 2 пропонується передбачити зміни, які враховують ситуації:
а) несправність клавіатури комп'ютера;
б) несправність маніпулятора типу "миша".

Варіант 3

GOAL: ways

/ * ЦЕЙ НАБІР ПРАВИЛ ДОЗВОЛИТЬ ВАМ ОТРИМАТИ РЯД РАД ПО * /
/ * Усунення деяких ДЕФЕКТІВ ВОДНИХ ФАРБ І повідомив * /
/ * ПРИЧИНИ ЇХ ПОЯВИ. ЗВИЧАЙНО, ЦЕ МАЛЕНЬКИЙ І НЕ ПОВНИЙ * /
/ * НАБІР, АЛЕ Ж ВІН І ПРИЗНАЧЕНИЙ ДЛЯ ПОКАЗУ САМИХ * /
/ * Тривіальних ЗАСОБІВ GURU. ТАК ЩО НЕ очікувалося * /
/ * ЧОГО-ЯКИХ СВЕРХІНТЕЛЛЕКТУАЛЬНОГО. * /
/ * НА ЗАПИТАННЯ СИСТЕМ СЛІД вводяться відповідні * /
/ * ЗНАЧЕННЯ булевскому ЗМІННОЮ. (ДА-TRUE, НЕТ-FALSE) * /
INITIAL:
clear
release variable / * ПРИБИРАЄМО НЕПОТРІБНІ НАМ ЗМІННІ * /

```
e.lstr = 250 / * Максимальна довжина рядка * /
output "ДЕНЬ ДОБРИЙ, МІСТЕР (МІСІС)."
output
output "ПІСЛЯ фарбування поверхні водної фарбою МОЖУТЬ"
output "проявити ДЕФЕКТИ."
output "Ми постараємося ДАТИ ВАМ РАДА щодо їх усунення та"
output "ПРИЧИН ПОЯВИ."
output "АЛЕ ДЛЯ ЦЬОГО ВИ ПОВИННІ НАДАТИ МЕНІ ВРЮ"
output "ІНФОРМАЦІЮ."
output "ОТЖЕ, НАЧНЕМ ..."
defects = true
output "З'ЯВИЛИСЯ ЧИ ДЕФЕКТИ НА ПОФАРБОВАНОЮ ПОВЕРХНІ?"
input defects logic

DO:
clear
output "ОСЬ ЩО Я ВАМ СКАЖУ, люб'язно."
output
output reasons
output
output "А ОСЬ ЩО ВАМ СЛІД ЗРОБИТИ В даній ситуації."
output
output ways

RULE: Rl
IF: defects
THEN: output
input circle logic with "ЦІ ДЕФЕКТИ - ПЛЯМИ?"
REASON: ЯКЩО Є ДЕФЕКТИ, ТО ШВИДШЕ ЗА ВСЕ ЦЕ ПЛЯМИ.
COMMENT: ЧИ Є ДЕФЕКТИ плями.

RULE: R2
IF: not defects
THEN: output
reasons = "НЕТ ДЕФЕКТІВ НА ПОВЕРХНІ, ЗНАЧИТЬ ВИ"
reasons = reasons + "ВСЕ ЗРОБИЛИ ПРАВИЛЬНО."
ways = "Піду КРАЩЕ відпочинемо. АДЖЕ У ВАС"
ways = ways + "ВСЕ ГАРАЗД."
REASON: ЯКЩО НІ ДЕФЕКТІВ, ТО НІЧОГО РОБИТИ НЕ ТРЕБА.
COMMENT: НІ ДЕФЕКТІВ - ВІДПОЧИВАЙ.

RULE: R3
IF: not circle
```

THEN: output
input lines logic with "ЦІ ДЕФЕКТИ - СМУГИ?"
REASON: ЯКЩО ДЕФЕКТИ НЕ ПЛЯМИ, ТО ШВИДШЕ ЗА ВСЕ ЦЕ УМОВАХ.
COMMENT: ЧИ Є ДЕФЕКТИ смуги.

RULE: R4
IF: circle
THEN: output
input fatcrcl logic with "ЦЕ жирні плями?"
REASON: ЯКЩО ДЕФЕКТИ ПЛЯМИ, ТО ШВИДШЕ ЗА ВСЕ ЦЕ жирні плями.
COMMENT: ЧИ Є ПЛЯМИ жирне.

RULE: R5
IF: fatcrcl
THEN: output
output "ЗАБАРВЛЕНА поверхня є"
output "штукатурка?"
input sfcshtkr logic
output
REASON: ЯКЩО ПЛЯМИ ЖИРНІ, ТО ВАЖЛИВО ЗНАТИ, ЯКА поверхні пофарбовані. COMMENT: на штукатурку жирні плями.

RULE: R6
IF: not sfcshtkr
THEN: output
output "ЗАБАРВЛЕНА поверхня є"
output "Залізобетон?"
input sfcsteel logic
REASON: якщо п'яти ЖИРНІ, ТО ВАЖЛИВО ЗНАТИ, ЯКА поверхні пофарбовані.
COMMENT: на залізобетонні жирні плями.

RULE: R7
IF: not sfcsteel
THEM: output
reasons = "ПРІЧІНА ПОЯВИ ЦИХ жирних плям"
reasons = reasons + "МЕНІ НЕВІДОМА."
ways = "ПОПРОБУКТЕ ЗВЕРНУТИСЯ до досвідченішого"
ways = ways + "СПЕЦІАЛІСТУ."
REASON: ЯКЩО жирні плями З'ЯВИЛИСЯ НА ЯКИЙСЬ ІНШИЙ ПОВЕРХНІ, ТО нічого путнього ЗАПРОПОНУВАТИ ВАМ Я НЕ МОЖУ.
COMMENT: жир на незарезервірованних ПОВЕРХНІ.

RULE: E8
IF: sfcsteel
THEN: reasons = "НА залізобетонний СЛІДИ Невисихаюче"
reasons = reasons + "МАСЕЛ ВІД СМАЗКИ ФОРМ."
ways = "очистити поверхню. ВІД ШАРУ БАРВИ РАЗОМ ЗІ"
ways = ways + "шпаклівка, обмити 5% -ним РОЗЧ."
ways = ways + "тринатрійфосфат або кальцію-ОЙ"
ways = ways + "СОДИ, НЕЙТРАЛІЗУВАТИ ПОВЕРХНОСТЬ 5% -вими"
ways = ways + "РАСТВ. СОЛЯНОЇ КИСЛОТИ І ЗНОВУ забарвити."
REASON: ЯКЩО жир на залізобетонну, ТО ЦЯ ПРОБЛЕМА цілком вирішувані.
COMMENT: СПОСІБ УСУНЕННЯ жирних плям на залізобетонні.

RULE: R9
IF: sfcshtkr
THEN: reasons = "НА штукатурки залишаються плями Невисихаюче"
reasons = reasons + "МІНЕРАЛЬНИХ І ТВАРИН ОЛІЙ."
ways = "вирубали штукатурка на ДІЛЯНЦІ ПЛЯМИ, ЗНОВУ"
ways = ways + "поштукатурити і пофарбувати, ПРОМИТИ повер"
ways = ways + "ХНОСТЬ ЛУЖНИЙ ВОДОЮ І ЗНОВУ забарвити."
REASON: ЯКЩО жирні плями на штукатурці, ТО ЦЯ ПРОБЛЕМА цілком вирішувані.
COMMENT: СПОСІБ УСУНЕННЯ жирних плям на штукатурці.

RULE: R10
IF: not fatcrcl
THEN: output
output "ЦЕ ЖОВТІ іржаві плями?"
input rsvcrcl logic
REASOH: ЯКЩО ДЕФЕКТИ НЕ жирні плями, ТО ШВИДШЕ ЗА ВСЕ ЦЕ іржаві плями.
COMMENT: ЧИ Є ПЛЯМИ іржаві.

RULE: R11
IF: not rsvcrcl
THEN: reasons = "ПРИЧИНА ПОЯВИ ЦИХ ПЯТЕН МЕНІ"
reasons = reasons + "НЕВІДОМА."
ways = "СКУШТУЙТЕ ЗВЕРНУТИСЯ до досвідченішого"
ways = ways + "СПЕЦІАЛІСТУ."
REASOH: ЯКЩО З'ЯВИЛИСЯ ЯКІСЬ ІНШІ ПЛЯМИ, ТО нічого путнього ЗАПРОПОНУВАТИ ВАМ Я НЕ МОЖУ.
COMMENT: ЯКІСЬ незарезервірованних ПЛЯМИ НА ПОВЕРХНІ.

RULE: R12
IF: rsvcrcl
THEH: reasons = "відбувається просочування смолисті речовини"
reasons = reasons + "ЧЕРЕЗ штукатурки і фарби."
ways = "ВИДАЛИТИ СТАРИЙ набіло, ПРОМИТИ ТЕПЛИМ 3% РОЗЧ."
ways = ways + "СОЛЯНОЇ КИСЛОТИ І, ЯКЩО ПЛЯМИ невелика"
ways = ways + ", ОТГРУНТОВАТЬ МЕДНОКУПОРОСНОЙ грунтовка"
ways = ways + "БЕЗ МЕЛА, А ПРИ ВЕЛИКИХ РОЗМІРАХ - щел"
ways = ways + "очним, спиртні або каніфольного ЛАКОМ."
REASON: ЯКЩО жирні плями на штукатурці, ТО ЦЯ ПРОБЛЕМА цілком вирішувані.
COMMENT: СПОСІБ УСУНЕННЯ жирних плям на штукатурці.

RULE: R13
IF: lines
THEN: reasons = "НЕДОСТАТНЬО перемішалося ПІГМЕНТИ В Колер,"
reasons = reasons + "ПЛОХОРАСТУШЕВАН КОЛЕР."
ways = "Промити ділянку і забарвити ІЗ"
ways = ways + "КРАСКОПУЛЬТА."
REASON: ЯКЩО СМУГИ НА ПОФАРБОВАНОЮ ПОВЕРХНІ, ТО ЦЯ ПРОБЛЕМА цілком вирішувані.
COMMENT: СПОСІБ УСУНЕННЯ СМУГ.

RULE: R14
IF: not lines
THEN: reasons = "ПРИЧИНА ПОЯВИ ІНШИХ ДЕФЕКТІВ МЕНІ"
reasons = reasons + "НЕВІДОМА."
ways = "СКУШТУЙТЕ ЗВЕРНУТИСЯ до досвідченішого"
ways = ways + "СПЕЦІАЛІСТУ."
REASON: ЯКЩО З'ЯВИЛИСЯ ЯКІСЬ ІНШІ ДЕФЕКТИ, ТО нічого путнього ЗАПРОПОНУВАТИ ВАМ Я НЕ МОЖУ.
COMMENT: ЯКІСЬ незарезервірованних ДЕФЕКТИ НА ПОВЕРХНІ.

VAR: defects
LABEL: НАЯВНІСТЬ ДЕФЕКТІВ НА ПОФАРБОВАНОЮ ПОВЕРХНІ.

END:

У варіанті 3 пропонується передбачити зміни, які враховують ситуації:

а) поява пухирів на пофарбованої поверхні;

б) відшаровування фарби після висихання.

варіант 4

GOAL: breaking

/ * ЦЕЙ НАБІР ПРАВИЛ ДОЗВОЛИТЬ ВАМ ОТРИМАТИ РЯД РАД ПО * /
/ * ОРГАНІЗАЦІЇ пограбування банку. ЗВИЧАЙНО, ЦЕ МАЛЕНЬКИЙ * /
/ * НАБІР І ВІН ПРИЗНАЧЕНИЙ ДЛЯ ПОКАЗУ САМИХ * /
/ * Тривіальних ЗАСОБІВ GURU. ТАК ЩО НЕ чекати чого-ЯКИХ * /
/ * ПОТУЖНОГО. НА ЗАПИТАННЯ СИСТЕМИ СЛІД ЗАПРОВАДЖУВАТИ * /
/ * Відповіднимизначеннями булевскому переменка * /
/ * (ДА-TRUE, НЕТ-FALSE) * /

```
INITIAL:
clear
release variable / * ПРИБИРАЄМО НЕПОТРІБНІ НАМ ЗМІННІ * /
e.lstr = 250 / * Максимальна довжина рядка * /
output "ДЕНЬ ДОБРИЙ, МІСТЕР (МІСІС)."
output
output "ВСЕ ХОЧУТЬ пограбувати банк, ДА НЕ ВСЕ МОЖУТЬ ..."
output "Ми постараємося ДАТИ ВАМ РАДА В цій непростій справі."
output "АЛЕ ДЛЯ ЦЬОГО ВИ ПОВИННІ НАДАТИ МЕНІ ВРЮ"
output "ІНФОРМАЦІЮ."
output "ОТЖЕ, ПОЇХАЛИ ..."
output
at 17,10 output "СКІЛЬКИ У ВАС ВІРНИХ ЛЮДЕЙ?"
input people num using "nn"
at 19,10 output "СКІЛЬКИ У ВАС 'СТВОЛОВ'?"
input guns num using "nn"
at 21,10 output "А СКІЛЬКИ ВИ ХОЧЕТЕ ВЗЯТИ?"
input nmoney num
output
clear

DO:
clear
```

output
output "ОСЬ ЩО Я ВАМ СКАЖУ, люб'язно."
output
output breaking

RULE: R1
IF: nmoney> = 10000000
THEN: output
output "А У ВАС ГУБА НЕ ДУРА!"
output "А СКІЛЬКИ ГРОШЕЙ зберігати в банку"
output "ГОТІВКОЮ?"
input emoney num
REASON: ЯКЩО ВИ ТАК БАГАТО ХОЧЕТЕ ВЗЯТИ, ТО СЛІД ЗНАТИ, чи бракує БАНК такої суми.
COMMENT: ВИ ТАК БАГАТО ХОЧЕТЕ ВЗЯТИ, ОТЖЕ. СЛІД ЗНАТИ, чи бракує БАНК такої суми.

RULE: R2
IF: nmoney <10000000
THEN: output "А СКІЛЬКИ ГРОШЕЙ зберігати в банку ГОТІВКОЮ?"
input emoney num
REASON: ЯКЩО ВИ ХОЧЕТЕ пограбувати банк, ТО СЛІД ЗНАТИ, чи має він такої суми.
COMMENT: ВИ ХОЧЕТЕ ВЗЯТИ ЧУЖІ ГРОШІ, ОТЖЕ, СЛІД ЗНАТИ, чи бракує БАНК такої суми.

RULE: R3
IF: nmoney <emoney
THEN: output
output "І ТАК, ви зважилися на СПРАВА, А як йдуть"
output "СПРАВИ З ОХОРОНОЮ?"
output
output "Чи охороняється БАНК ДНЕМ?"
input dayctrl logic
output
output "Чи охороняється БАНК поліцейськими ВНОЧІ?"
input nitectrl logic
REASON: ЯКЩО В БАНКУ ДОСИТЬ ГРОШЕЙ, ТО НАЧИНАЕМ ЗБИРАТИ ІНФОРМАЦІЮ ПРО ОХОРОНУ.
COMMENT: БАНК МАЄ ДОСИТЬ ЗАСОБІВ ДЛЯ ЗАДОВОЛЕННЯ ВАШИХ ПОТРЕБ. ДІЗНАЄМОСЯ, охороняється чи ОН ДНЕМ І ВНОЧІ.

RULE: R4

IF: dayctrl

THEN: output

output "СКІЛЬКИ ЛЮДИНА КОШТУЄ НА ОХОРОНУ БАНКУ ДНЕМ?"

input daycol num using "nn"

output

output "проглядається ЧИ ПРИМІЩЕННЯ БАНКУ"

output "відеокамери?"

input kamera logic

REASON: ЯКЩО ДНЕМ БАНК ОХОРОНЯЄТЬСЯ, ТО ТРЕБА ДІЗНАТИСЯ КІЛЬКІСТЬ ОХОРОНИ ТА ОТРИМАТИ ІНФОРМАЦІЮ ПРО НАЯВНІСТЬ В ПРИМІЩЕННІ БАНКУ ВІДЕОКАМЕРИ.

COMMENT: ЯКЩО У ВХОДА ДНЕМ КОШТУЮТЬ ПОЛІЦЕЙСЬКІ, ТО ТРЕБА ДІЗНАТИСЯ СКІЛЬКИ ЇХ, і переглядати ЧИ приміщення відеокамери

RULE: R5

IF: nitectrl

THEN: output

output "СКІЛЬКИ ЛЮДИНА КОШТУЄ НА ОХОРОНУ БАНКУ ВНОЧІ?"

input nitecol num using "nn"

output

output "МОЖЕТЕ ВИ відключити сигналізацію,"

output "ЩОБ ВОНА НЕ спрацює?"

input signal logic

REASON: ЯКЩО ВНОЧІ БАНК ОХОРОНЯЄТЬСЯ, ТО НАДО ДІЗНАТИСЯ КІЛЬКІСТЬ ОХОРОННИКІВ І ЧИ МОЖНА ВІДКЛЮЧИТИ, сигналізація.

COMMENT: ЯКЩО ВНОЧІ БАНК ОХОРОНЯЮТЬ ПОЛІЦЕЙСЬКІ, ТО ТРЕБА ДІЗНАТИСЯ СКІЛЬКИ ЇХ І ЧИ МОЖНА відключити сигналізацію.

RULE: R6

IF: not nitectrl

THEN: output

output "МОЖЕТЕ ВИ відключити сигналізацію,"

output "ЩОБ ВОНА НЕ спрацює?"

input signal logic

REASON: ЯКЩО ВНОЧІ БАНК не охороняється, ТО ТРЕБА ДІЗНАТИСЯ, ЧИ МОЖЛИВО відключити сигналізацію.

COMMENT: ЯКЩО ВНОЧІ БАНК НЕ ОХОРОНЯЮТЬ ПОЛІЦЕЙСЬКІ, ТО ЧИ МОЖЛИВО відключити сигналізацію.

RULE: R7

IF: not nitectrl and signal

THEN: breaking = "ОПЕРАЦІЮ НАДО ПОЧИНАТИ В 2 ГОДИНИ 17 ХВИЛИН"

breaking = breaking + "за місцевим часом."

REASON: ЯКЩО БАНК ВНОЧІ ПІД ПОГАНИЙ ОХОРОНОЮ, ТО ПОГРАБУВАННЯ ПРОЙДЕ ЦІЛКОМ УСПІШНО. ГОЛОВНЕ ПОЧАТИ ЙОГО В 2 ГОДИНИ 17 ХВИЛИН. В НІЧ НА П'ЯТНИЦЮ. COMMENT: ТАКІ БАНКИ НАДО БРАТИ ВНОЧІ.

RULE: R8

IF: not nitectrl and not signal and dayctrl

THEN: breaking = "ОПЕРАЦІЮ НАДО ПОЧИНАТИ В 2 ГОДИНИ 17 ХВИЛИН"

breaking = breaking + "по місцевим часом. АЛЕ БУДЬТЕ"

breaking = breaking + "ОСТОРОЖНИ, ВАМ НАДО ВСЕ ЗРОБИТИ"

breaking = breaking + "ЗА 15 ХВИЛИН. ТРЕНУЙТЕСЯ!"

REASON: ЯКЩО БАНК ВНОЧІ ПІД ПОГАНИЙ ОХОРОНОЮ, ТО ПОГРАБУВАННЯ ПРОЙДЕ ЦІЛКОМ УСПІШНО. ГОЛОВНЕ ПОЧАТИ ЙОГО В 2 ГОДИНИ 17 ХВИЛИН. В НІЧ НА П'ЯТНИЦЮ. І врахуйте, У ВАС ВСЬОГО 15 ХВИЛИН.

COMMENT: ТАКІ БАНКИ НАДО БРАТИ ВНОЧІ.

RULE: R9

IF: (nitectrl or not signal and not nitectrl) and not dayctrl and not kamera

THEN: breaking = "ТАКОЙ БАНК НАДО БРАТИ В 16 ГОДИН 45 ХВИЛИН"

breaking = breaking + "УДВОХ, МАЮЧИ ПРИ СОБІ ПІСТОЛЕТИ."

REASON: ЯКЩО БАНК ВНОЧІ ПІД ОХОРОНОЮ або сигналізації, АЛЕ ДНІВ кинута напризволяще, ТО ПОГРАБУВАННЯ пройде ЦІЛКОМ УСПІШНО. ГОЛОВНЕ ПОЧАТИ ЙОГО В 16 ГОДИН 45 ХВИЛИН У П'ЯТНИЦЮ.

COMMENT: ТАКІ БАНКИ НАДО БРАТИ ДНЕМ привселюдно.

RULE: R10

IF: (nitectrl or not signal and not nitectrl) and not dayctrl and kamera

THEN: breaking = "ТАКОЙ БАНК НАДО БРАТИ В 16 ГОДИН 45 ХВИЛИН"

breaking = breaking + "УДВОХ, МАЮЧИ ПРИ СОБІ пістолета"

breaking = breaking + "І МАСКИ."

REASON: ЯКЩО БАНК ВНОЧІ ПІД ОХОРОНОЮ або сигналізації, АЛЕ ДНЕМ кинута напризволяще, ТО ПОГРАБУВАННЯ ПРОЙДЕ ЦІЛКОМ УСПІШНО. ГОЛОВНЕ ПОЧАТИ ЙОГО В 16 ГОДИН 45 ХВИЛИН У П'ЯТНИЦЮ.

COMMENT: ТАКІ БАНКИ НАДО БРАТИ ДНЕМ привселюдно.

RULE: R11
IF: dayctrl and nitectrl and (people <nitecol or guns <daycol)
THEN: breaking = "СИДІЛИ Б ВИ КРАЩЕ ДОМА І НЕ рипатися"
breaking = breaking + "даремно."

RULE: R12
IF: (people or guns) <l and not signal
THEH: breaking = "СИДІЛИ Б ВИ КРАЩЕ ДОНУ І НЕ рипатися"
breaking = breaking + "даремно при такому розкладі"
breaking = breaking + "ВИ ОБОВ'ЯЗКОВО потрапить. МІЙ"
breaking = breaking + "ВАМ РАДА: ШУКАЙТЕ ЛЮДЕЙ І"
breaking = breaking + "купує зброю."

RULE: R13
IF: not dayctrl and people> 1 and guns> l and nitectrl
THEN: breaking = "У ВАС НЕПОГАНІ ШАНСИ НА УСПІХ ПРИ РОБОТІ"
breaking = breaking + "ДНЕМ. І ЯКЩО ВИ захопили двох"
breaking = breaking + "САМНХ спритно РЕБЯТ, ТО ПРИ"
breaking = breaking + "Непогано розкладі ви зробили це"
breaking = breaking + "УСПІШНО. ТАК, І НЕ ЗАБУДЬТЕ МАСКИ,"
breaking = breaking + "ТАМ НАПЕВНЕ ВСТАНОВЛЕНО"
breaking = breaking + "ВІДЕОКАМЕРА."
COMMENT: ДНЕМ БАНК не охороняється.

RULE: R14
IF: daycol <min (people, guns) -2 and nitecol> daycol + l
THEN: breaking = "лучше ВСЬОГО БРАТИ ЦЕЙ БАНК ШТУРМОМ,"
breakine = breaking + "ТАК ЯК У ВАС ЦІЛКОМ ДОСТАТНЬО"
breaking = breaking + "ЛЮДЕЙ. АЛЕ ЧИ ВАРТО РОБИТИ ЦЕ?"
COMMENT: ДНЕМ КОНТРОЛЬ слабший, НІЖ ВНОЧІ.

VAR: GUNS
LABEL: КІЛЬКІСТЬ наявні у вашому розпорядженні ЕД. ОГНЕСТРЕЛЬНОГО ЗБРОЇ.

VAR: NMONEY
LABEL: планована сума пограбувань.

VAR: PEOPLE

LABEL: КІЛЬКІСТЬ ЧЛЕНІВ ВАШОЇ ТУСОВКИ.

END:

У варіанті 4 пропонується передбачити зміни, які враховують ситуації:

а) пограбування транспорту, що перевозить банківські цінності;

б) пограбування банку в умовах стихійного лиха.

варіант 5

GOAL: dinner

/ * ЦЕЙ НАБІР ПРАВИЛ ДОЗВОЛИТЬ ВАМ. ОТРИМАТИ РЯД РАД ПО * /
/ * Приготування обіду НА БАЖАНЕ ЧИСЛО ПЕРСОН В * /
/ * ЗАЛЕЖНОСТІ ВІД НАБОРУ І КІЛЬКОСТІ ПРОДУКТІВ, * /
/ * Наявні у вашому розпорядженні. Повідомити причину * / / * ПОЯВИ САМЕ ЦИХ РАД. * /
/ * ЗВИЧАЙНО, ЦЕ ДАЛЕКО НЕ ПОВНИЙ НАБІР, АЛЕ Ж ВІН І предна- * /
/ * Значив для ПОКАЗУ найтривіальніших ЗАСОБІВ GURU. ТАК ЩО * /
/ * НЕ ШУКАЙТЕ В НЬОМУ ЧОГО-ЯКИХ СВЕРХІНТЕЛЛЕКТУАЛЬНОГО. * /
/ * НА ЗАПИТАННЯ СИСТЕМИ СЛІД вводяться відповідні * /
/ * ЗНАЧЕННЯ булевскому ЗМІННОЮ (ДА-Y, НЕТ-N) * /

```
INITIAL:
clear
release variable / * ПРИБИРАЄМО НЕПОТРІБНІ НАМ ЗМІННІ * /
e.lstr = 250 / * Максимальна довжина рядка * /
output "ДЕНЬ ДОБРИЙ, МІСТЕР (МІСІС)."
output "понишпорив У СВОЄМУ ХОЛОДИЛЬНИКУ І виявив там"
output "деякі запаси, ВИ ВИРІШИЛИ ЩО-НЕБУДЬ сфабрикувати. НО"
output "ЩО САМЕ ВИ ЩЕ НЕ ПРИДУМАЛИ. ми намагаємось дати"
output "ВАМ РАДА, ЯК З наявних ПРОДУКТІВ ЗРОБИТИ ЩОСЬ"
output "ЇСТІВНЕ. АЛЕ ДЛЯ ЦЬОГО ВИ ПОВИННІ НАДАТИ МЕНІ"
output "ВСЮ ІНФОРМАЦІЮ."
output "ОТЖЕ, НАЧНЕМ ..."
meatexst = "Y"
output "СКАЖІТЬ, ВИ ЗНАЙШЛИ В ХОЛОДИЛЬНИКУ М'ЯСО (Y / N)?"
```

```
input meatexst str using "a"

DO:
clear
output "ОСЬ ЩО МЕНІ ЗДАЄТЬСЯ прийнятних у даній СИТУАЦІЇ."
output
output dinner

RULE: Rl
IF: meatexst <> "Y" or meatexst <> "y"
THEN: output "ИЗВИНИТЕ, А ви вегетаріанець (Y / N)?"
input vegitar str using "a"

RULE: R2
IF: (meatexst <> "Y" or meatexst <> "y") and (vegitar <> "y" or vegitar
<> "Y")
THEN: dinner = "НАЙКРАЩЕ ВІЗЬМІТЬ СВІЙ ТОВСТИЙ
БУМАЖНІК"
dinner = dinner + "і відправляється в РЕСТОРАН, ТОМУ ЩО БЕЗ"
dinner = dinner + "МЯСА НОРМАЛЬНОГО ОБЕДА НЕ ВИЙДЕ"

RULE: R3
IF: vegitar = "y" or vegitar = "Y"
THEN: output
output "НУ, РАЗ ви вегетаріанець, то, напевно, ОВОЧІВ"
output "У ВАС НАВАЛОМ (Y / N)?"
input vagitables str using "a"

RULE: R4
IF: meatexst = "Y" or meatexst = "y"
THEN: output
output "СКІЛЬКИ ГРАМ М'ЯСА ВИ МАЄТЕ?"
input howmeat num using "nnnn"
output
output "НА СКІЛЬКИ ПЕРСОН задуманих ВАШ ОБІД?"
input howfamily num using "n"

RULE: R5
IF: meatexst <> "Y" or meatexst <> "y"
THEN: output
output "ИЗВИНИТЕ, А ви вегетаріанець (Y / N)?"
input vegitar str using "a"

RULE: R6
```

IF: howmeat / howfamily> 300
THEN: dinner = "НАЙКРАЩЕ ВІЗЬМІТЬ ЦЕЙ ШМАТОК М'ЯСА,"
dinner = dinner + "ПОРЕЖТЕ ЙОГО УЗДОВЖ ВОЛОКОН НА"
dinner = dinner + "НЕТОЛСТИЕ ПЛАСТИ, відбийте, нашпигують"
dinner = dinner + "І НА 30 МІН. відкласти. потім бере"
dinner = dinner + "Сковорода й смажите на повільному"
dinner = dinner + "ОГНЕ. Як кажуть, СІЛЬ, АНАНАСИ,"
dinner = dinner + "ФІСТАШКИ - за смаком."

RULE: R7
IF: (howmeat / howfamily> 80) and (howmeat / howfamily <= 300)
THEN: enoughmeat = true

RULE: R8
IF: enoughmeat
THEN: output
output "М'ЯСО - ЦЕ ЗАПОРУКА УСПІХУ."
output
output "Чи має У ВАШОМУ РАПОРЯЖЕНІІ КАПУСТА (Y / N)?"
input kapusta str using "a"
output
input svekla str using "a" with "БУРЯК (Y / N)?".
output
input potetou str using "a" with "КАРТОПЛЯ (Y / N)?"

RULE: R9
IF: (kapusta = "y" or kapusta = "Y") and (svekla = "y" or svekla = "Y") and
(potetou = "y" or potetou = "Y")
THEN: dinner = "варіть УКРАЇНСЬКИЙ БОРЩ."

RULE: R10
IF: howmeat / howfamily <= 80
THEN: enoughmeat = false

RULE: R11
IF: not enoughmeat
THEN: dinner = "М'ЯСА ДУЖЕ МАЛО, ЩОБ ПРИГОТУВАТИ ПЕРШЕ АБО"
dinner = dinner + "ВТОРОЕ, НО, БУТИ Може, ЙОГО вистача НА"
dinner = dinner + "м'ясний салат."

RULE: R12
IF: vagitables = "Y" or vagitables = "y"

THEN: dinner = "звалити все В ОДНУ КУПУ, ПОТІМ дрібно покришити"

diriner = dinner + "(можна пропустити через м'ясорубку),"

dinner = dinner + "Перекласти масу В" dinner = dinner + "салатницю і подавати до столу. ЯКЩО ВАМ"

dirmer = dinner + ': ПОВЕЗЕТ, ТО всі будуть задоволені. ЯКЩО НІ - "

dinner = dinner + "НЕ вибачайте."

VAR: DINNER

FIND: dinner = "ПРИЧИНА ПОЯВИ ЦИХ НЕПОЛАДОК НАМ НЕВІДОМА."

LABEL: як вчинити, якщо потрібно приготувати ОБІД.

VAR: MEATEXST

LABEL: НАЯВНІСТЬ М'ЯСА В ВАШОМУ ХОЛОДИЛЬНИКУ.

VAR: ENOUGHMEAT

LABEL: ДОСТАТНЬО М'ЯСА ДЛЯ ОДНОГО З ПЕРШИХ СТРАВ.

VAR: HOWMEAT

LABEL: ПРИМІРНИЙ ВЕС шматків м'яса.

VAR: HOWFAMILY

LABEL: КІЛЬКІСТЬ ПЕРСОН, ЗАПРОШЕНИХ НА ОБІД.

VAR: KAPUSTA

LABEL: ЧИ В ДОМІ КАПУСТА.

VAR: SVEKLA

LABEL: ЧИ В ДОМІ СВЕКЛА.

VAR: POTETOU

LABEL: ЧИ В ДОМІ КАРТОПЛЯ.

VAR: VEGITAR

LABEL: Чи є Ви вегетаріанців.

END:

У варіанті 5 пропонується передбачити зміни, які враховують ситуації:

а) наявність в холодильнику сиру;

б) наявність в холодильнику пива.

варіант 6

```
GOAL: whattodo

/ * ЦЕЙ НАБІР ПРАВИЛ ДОЗВОЛИТЬ ВАМ ОТРИМАТИ РЯД РАД НА * /
/ * ТЕМУ "ЯК ВСТИГНУТИ НА ІСПИТ", ЗАЛЕЖНО ВІД запізненням І * /
/ * Важливість своєчасної ПРИХОДА. Очевидно, що це * /
/ * ДАЛЕКО НЕ ПОВНИЙ НАБІР, АЛЕ Ж ВІН І ПРИЗНАЧЕНИЙ ДЛЯ * /
/ * ПОКАЗУ найтривіальніших ЗАСОБІВ GURU. ТАК ЩО НЕ * /
/ * ШУКАЙТЕ В НЬОМУ СВЕРХІНТЕЛЛЕКТУАЛЬНОГО. * /
/ * НА ЗАПИТАННЯ СИСТЕМИ СЛІД вводяться відповідні * /
/ * ЗНАЧЕННЯ булевскому ЗМІННОЮ (ДА-Y, НІ-N) * /
/ * НУ, І ЗВИЧАЙНО, НА ПРОХАННЯ СИСТЕМИ ВВЕСТИ соответ- * /
/ * Ствующая ЧИСЛО. * /

INITIAL:
clear
release variable / * ПРИБИРАЄМО НЕПОТРІБНІ НАМ ЗМІННІ * /
e.lstr = 250 / * Максимальна довжина рядка * /
output "ДЕНЬ ДОБРИЙ, МІСТЕР (МІСІС)."
output
output "У ВАС СЬОГОДНІ ІСПИТ, А ВИ ПРОКИНУЛИСЬ ЗАНАДТО"
output "ПІЗНО ... ВАМ, ПРИРОДНО, НАДО ВСТИГНУТИ НА НЬОГО, АЛЕ"
output "ЯК? ми намагаємось дати ВАМ РАДА, ЯК, ВИХОДЯЧИ З"
output "Склалася ситуація, ВАМ СЛІД ВСТУПИТИ. АЛЕ ДЛЯ"
output "ЦЬОГО ВИ ПОВИННІ НАДАТИ МЕНІ ВРЮ ІНФОРМАЦІЮ."
output "ОТЖЕ, НАЧНЕМ ..."
output
lating = "Y"
output "СКАЖІТЬ, ВИ ДІЙСНО спізнюватися (Y / N)?"
input lating str using "a"

DO:
clear
output "ОСЬ ЩО МЕНІ ЗДАЄТЬСЯ прийнятних у даній СИТУАЦІЇ."
output
output whattodo
```

RULE: Rl
IF: mainexam and biglate
THEN: whattodo = "БЕРІТЬ ТАКСІ НА ВЕСЬ ШЛЯХ ДО ІНСТИТУТУ. В"
whattodo = whattodo + "ТАКОЙ СИТУАЦІЇ ГРОШІ ЗНАЧЕННЯ НЕ"
whattbdo = whattodo + "ІМЕЮТ."

RULE: R2
IF: not mainexam
THEN: whattodo = "заспокоїли, НА НЕ ДУЖЕ ВАЖЛИВИЙ ІСПИТ"
whattodo = whattodo + "НЕ КОШТУЄ СИЛЬНО ПОСПІШАТИ. Повірте,"
whattodo = whattodo + "ВАМ простими ВАШЕ ЗАПІЗНЕННЯ АБО НАВІТЬ"
whattodo = whattodo + "ВІДСУТНІСТЬ. ТАК ЩО Неспеша ПО-"
whattodo = whattodo + "ЕЗЖАЙТЕ НА ГРОМАДСЬКОМУ ТРАНСПОРТІ."

RULE: R3
IF: not biglate and mainexam
THEN: whattodo = "НЕ хвилею, ДОСІ БУДЕ ДОБРЕ. ВАМ"
whattodo = whattodo + "ВАРТО ВЗЯТИ ТАКСІ НА ЧАСТИНА ШЛЯХИ,"
whattodo = whattodo + "НАПРИКЛАД, ДО ЯКОГО-НЕБУДЬ"
whattodo = whattodo + "ВУЗЛОВОГО ПУНКТУ (ДО МЕТРО, АВТ."
whattodo = whattodo + "ОСТАНQВКІ)."

RULE: R4
IF: onlyge4
THEN: mainexam = false

RULE: R5
IF: veroyatn> = 90
THEN: mainexam = false

RULE: R6
IF: (veroyatn <90) and not onlyge4
THEN: mainexam = true

ROLE: R7
IF: lating <> "Y" and lating <> "y"
THEN: whattodo = "ВСЕ ГАРАЗД. БАЖАЮ ВАМ НИ ПУХА."

RULE: R8
IF: howcommon <onwalk + bymetro + bybus + 15
THEN: biglate = true

RULE: R9
IF: howcommon> = onwalk + bymetro + bybus + 15
THEN: biglate = false

VAR: WHATTODO
FIND: whattodo = "Жалкую, Я НЕ ЗНАЮ, ЩО ВАМ ЗАПРОПОНУВАТИ ..."
LABEL: РАДА ЯК ДІЯТИ У ЦІЙ СИТУАЦІЇ.

VAR: MAINEXAM
LABEL: майбутній іспит - ВАЖЛИВИЙ.

VAR: BIGLATE
LABEL: ПОТОЧНЕ ЗАПІЗНЕННЯ - ЗНАЧНЕ.

VAR: LATING
LABEL: ВИ РЕАЛЬНО спізнюватися.

VAR: HOWCOMMON
FIND: output
output "СКІЛЬКИ ХВИЛИН ВАМ Добиратися до ІНСТИТУТУ"
output "СУСПІЛЬНОЇ ТРАНСПОРТОМ?"
input howcommon num using "nnn"
LABEL: ЧАС В ДОРОЗІ ДО ІНСТИТУТУ.

VAR: ONWALK
FIND: output
output "СКІЛЬКИ ХВИЛИН ВАМ доводиться йти ПІШКИ?"
input onwalk num using "nn"
LABEL: час пішого пересування.

VAR: BYMETRO
FIND: output
output "СКІЛЬКИ ХВИЛИН ВАМ доводиться проводити в"
output "МЕТРО?"
input byinetro num using "nnn"
LABEL: ЧАС ПРОЇЗДУ В метрополітені.

VAR: BYBUS

FIND: output
output "СКІЛЬКИ ХВИЛИН ВАМ доводиться проводити в"
output "АВТОБУСІ?"
input bybus num using "nnn"
LABEL: ЧАС ПРОЇЗДУ В автобусі, тролейбусі, трамваї.

VAR: ONLYGE4
FIND: output
output "НА майбутні іспити НЕ СТАВЛЯТЬ МЕНШЕ 4?"
input onlyge4 logic
LABEL: НА майбутнього іспиту НЕ НЕ СТАВЛЯТЬ МЕНШЕ 4.

VAR: VEROYATN
FIND: output
output "ЯКА ОБ'ЄКТИВНА ЙМОВІРНІСТЬ ОДЕРЖАННЯ ВАМИ"
output "бажаної оцінки?"
input veroyatn num using "nn"
LABEL: ОБ'ЄКТИВНА ЙМОВІРНІСТЬ ОДЕРЖАННЯ ВАМИ бажаної оцінки.

END:

У варіанті 6 пропонується передбачити зміни, які враховують ситуації:

а) своє погане самопочуття (хвороба);

б) можливість подзвонити екзаменатору в аудиторію, де проходить іспит (попередити).

варіант 7

GOAL: youhouse

/ * ЦЯ ЕКСПЕРТНА СИСТЕМКА ДАСТЬ ВАМ РЯД РАД НА ТЕМУ: * /
/ * ЧИ ВИСТАЧИТЬ ВАМ ЗАГОТОВЛЕНОЇ СТРОЙМАТЕРИАЛОВ ДЛЯ * /
/ * БУДІВЛІ ЗАДУМАНИЙ ВАМИ ВЛАСНОГО БУДИНОЧКА НА * /
/ * ВЛАСНОЇ ЗЕМЛІ. ЗВИЧАЙНО, ЦЕ ДАЛЕКО НЕ ПОВНИЙ * /
/ * НАБІР, АЛЕ Ж ВІН І ПРИЗНАЧЕНИЙ ДЛЯ ПОКАЗУ * /
/ * НАЙТРИВІАЛЬНІШИХ ЗАСОБІВ GURU. * /
/ * ТАК ЩО НЕ ШУКАЙТЕ В НЬОМУ ЧОГО-ЯКИХ СВЕРХІНТЕЛЛЕКТУАЛЬНОГО. * /
/ * НА ЗАПИТАННЯ СИСТЕМИ СЛІД вводяться відповідні * /
/ * ЗНАЧЕННЯ строкових змінних (ДА-Y, НЕТ-N) АБО ЧИСЛО. * /

```
INITIAL:
clear
release variable / * ПРИБИРАЄМО НЕПОТРІБНІ НАМ ЗМІННІ * /
e.lstr = 250 / * Максимальна довжина рядка * /
output "ДЕНЬ ДОБРИЙ, МІСТЕР (МІСІС)."
output "Настала пора обзавестися нерухомістю (ви зважилися"
output "ПОБУДУВАТИ СОБІ ДІМ). ми намагаємось дати ВАМ РАДА, С"
output "ДОПОМОГОЮ ЯКОГО ВИ ЗМОЖЕТЕ ВИЗНАЧИТИ, ЧИ ВИСТАЧИТЬ ВАМ *
output "СТРОЙМАТЕРИАЛОВ НА НЬОГО."
output "АЛЕ ДЛЯ ЦЬОГО ВИ ПОВИННІ НАДАТИ МЕНІ ВРЮ"
output "ІНФОРМАЦІЮ."
output "ОТЖЕ, НАЧНЕМ ..."
sqrland = 0
output "СКІЛЬКИ СОТОК ЗЕМЛІ ЗНАХОДИТЬСЯ В ВАШОМУ володіннях?"
input sqrland num using "nn"

DO:
clear
output "ОСЬ ЩО МЕНІ ЗДАЄТЬСЯ прийнятних у даній СИТУАЦІЇ."
output
output youhouse

RULE: Rl
IF: sqrland <10
THEN: enoughland = false

RULE: R2
IF: sqrland> = 10
THEN: enoughland = true

RULE: R3
IF: enoughland
THEN: output "СКІЛЬКИ ПОВЕРХІВ В ДОМІ БУДЕ?"
input howflors num using "nn"
output "КАКОВА ПЛОЩА ОДНОГО ПОВЕРХУ (КВ. М.)?"
input howsqr num using "rin"

RULE: R4
IF: howsqr> sqrland * 5
THEN: rightsqr = false
```

RULE: R5
IF: howsqr <= sqrland * 5
THEN: rightsqr = true

RULE: R6
IF: not rightsqr
THEN: youhouse = "ДЛЯ ВАШОЇ ДІЛЯНКИ ТАКА ПЛОЩА ПОВЕРХУ НЕ" youhouse = youhousa + "ПІДХОДИТЬ."

RULE: R7
IF: howflors <= 2
THEN: rightflors = true

RULE: R8
IF: howflors> 2
THEN: rightflors = false

RULE: R9
IF: rightflors and rightsqr
THEN: output
output "СКІЛЬКИ ТИС. ШТ. ЦЕГЛИ ВИ ПРИГОТУВАЛИ ДЛЯ"
output "БУДІВНИЦТВА?"
input howbloks num using "nn"

RULE: R10
IF: howbloks <howflors * howsqr * 3
THEN: enoughbloks = false

RULE: R11
IF: howbloks> = howflors * howsqr * 3
THEN: enoughbloks = true

RULE: R12
IF: not enoughbloks
THEM: youhouse = "НА ДІМ ВАМ НЕ ВИСТАЧАЄ ЦЕГЛИ."

RULE: R13
IF: enoughbloks
THEN: output "СКІЛЬКИ КВ. М. ШИФЕРУ АБО ІНШОГО"
output "покрівельний матеріал ВИ МАЄТЕ?"
input howshifers num using "nn"

RULE: R14

IF: howshifers> = howsqr * l .2
THEH: enoughshifers = true

RULE: R15
IF: howshifers <howsqr * 1.2
THEN: enoughshifers = false

RULE: R16
IF: not enoughshifers
THEN: youhouse = "ВАМ НЕ ВИСТАЧАЄ ПОКРІВЕЛЬНИЙ МАТЕРІАЛ ДЛЯ" youhouse = youhouse + "ТОГО, ЩОБ НАКРИТИ СВІЙ" youhouse = youhouse + "МАЙБУТНІЙ ДОМ. ВАМ ПОТРІБНО ЩЕ" youhouse = youhouse + "малості ПОДЗАКУПІТЬСЯ АБО" youhouse = youhouse + "ПЕРЕГЛЯНУТИ ПРОЕКТ."

RULE: R17
IF: enoughshifers and enoughbloks and enoughland
THEN: youhouse = "ВИ МАЄТЕ МАЙЖЕ ВСІ НЕОБХІДНА ДЛЯ"
youhouse = youhouse + "здійснення своїх планів."
youhouse = youhouse + "І ЯКЩО ВИ докласти достатньо"
youhouse = youhouse + "ЗУСИЛЬ, ТО через рік-другий" youhouse = youhouse + "БУДЕ У ВАС СВОЇ БУДИНОЧОК."

RULE: R18
IF: not rightflors
THEN: youhouse = "ТАКНЕ ДОМА БУДУВАТИ НЕ ТРЕБА. НЕ дражнили"
youhouse = youhouse + "ВАШІХ СУСІДІВ. АДЖЕ ВОНИ ПРОСТІ" yotshouse = youhouse + "СМЕРТНИЕ. підпалити МОЖУТЬ ..."

VAR: SQRLAND
LABEL: ПЛОЩА ВАШОГО ДІЛЯНКИ

VAR: EHOUGHLAND
LABEL: ДОСТАТНЬО ПЛОЩІ ВАШОГО ДІЛЯНКИ

VAR: HOWSQR
LABEL: ПЛОЩА ОДНОГО ПОВЕРХУ У БУДИНКУ

VAR: HOWFLORS
LABEL: КІЛЬКІСТЬ ПОВЕРХІВ В ДОМІ

VAR: RIGHTSQR
LABEL: ВІДПОВІДНІСТЬ ПЛОЩІ ПОВЕРХУ ПЛОЩІ БУДИНКУ

VAR: RIGHTFLORS
LABEL: МОЖЛИВІСТЬ будівля будинку З ТАКОЮ поверхового

VAR: HOWSHIFERS
LABEL: КІЛЬКІСТЬ покрівельний матеріал (КВ. М.)

VAR: EHOUGHSHIFERS
LABEL: ДОСТАТНЬО покрівельний матеріал

END:

У варіанті 7 пропонується передбачити зміни, які враховують ситуації:

а) можливість замовити будівництво будинку будівельної організації;

6) необхідність побудови гаража.

варіант 8

GOAL: ways

/ * ЦЕЙ НАБІР ПРАВИЛ ДОЗВОЛИТЬ ВАМ ОТРИМАТИ РЯД РАД ПО * /
/ * Усунення деяких ДЕФЕКТІВ пістолетів- * /
/ * Краскопультам І повідомити причину їх появи. * /
/ * ЗВИЧАЙНО, ЦЕ МАЛЕНЬКИЙ І ДАЛЕКО НЕ ПОВНИЙ * /
/ * НАБІР, АЛЕ Ж ВІН І ПРИЗНАЧЕНИЙ ДЛЯ * /
/ * ПОКАЗУ найтривіальніших ЗАСОБІВ GURU. ТАК ЩО * /
/ * НЕ ШУКАЙТЕ В НЬОМУ СВЕРХІНТЕЛЛЕКТУАЛЬНОГО. * /
/ * НА ЗАПИТАННЯ СИСТЕМИ СЛІД вводяться відповідні * /
/ * ЗНАЧЕННЯ булевскому ЗМІННОЮ (ДА-Y, НЕТ-N) * /

INITIAL:
clear
release variable / * ПРИБИРАЄМО НЕПОТРІБНІ НАМ ЗМІННІ * /
e.lstr = 250 / * Максимальна довжина рядка * /
output "ДЕНЬ ДОБРИЙ, МІСТЕР (МІСІС)."
output "Ви зважилися забарвити ЯКУСЬ поверхню за допомогою"
output "Пістолет-краскопультам. А ВАШ пістолетики ЩОСЬ"
output "НЕНОРМАЛЬНО ПРАЦЮЄ. ми намагаємось дати ВАМ РАДА,"
output "ЯК ВИПРАВИТИ ДЕЯКІ НЕСПРАВНОСТІ І СОВЩІМ ВАМ"

output "ПРИЧИНІ Их ПОЯВИ."
output "АЛЕ ДЛЯ ЦЬОГО ВИ ПОВИННІ НАДАТИ МЕНІ ВРЮ"
output "ІНФОРМАЦІЮ."
output "ОТЖЕ, ПОЇХАЛИ ..."
gunsdef = true
output "ПРИ РОБОТІ ПІСТОЛЕТА З'ЯВИЛИСЯ ЯКІСЬ ДЕФЕКТИ (Y / N)?"
input gunsdef str using "a"

DO:
clear
output
output "ОСЬ ЩО Я ВАМ СКАЖУ:"
output
output reasons
output
output "'А ОСЬ ЩО ВАМ СЛІД ЗРОБИТИ В ЦІЙ СИТУАЦІЇ."
output ways

RULE: Rl
IF: gunsdef = "Y" or gunsdef = "y"
THEN: output "щось негаразд із струменем БАРВИ (Y / N)?"
input drops str using "a"
REASON: ЯКЩО Є ДЕФЕКТИ, ТО ШВИДШЕ ЗА ВСЕ ВОНИ ПОВ'ЯЗАНІ ЗІ
Цівок БАРВИ.
COMMENT: Чи пов'язані ДЕФЕКТИ із струменем.

RULE: R2
IF: gunsdef <> "Y" or gunsdef <> "y"
THEN: output
reasons = "НЕТ ДЕФЕКТІВ, ЗНАЧИТЬ нічого хвилюватися ЗРЯ." ways = "Піду КРАЩЕ відпочину, АДЖЕ У ВАС ВСЕ В"
ways = ways + "ПОРЯДКУ."
REASON: ЯКЩО НІ ДЕФЕКТІВ, ТО НІЧОГО РОБИТИ НЕ ТРЕБА COMMENT: НІ ДЕФЕКТІВ - ВІДПОЧИВАЙ

RULE: R3
IF: notkrask = "Y" or notkrask = "y"
THEN: reasons = "засмічуючи СЕТКА РЕЗЕРВУАРА ДЛЯ ФАРБИ АБО"
reasons = reasons + "НЕДОСТАТНІЙ РІВЕНЬ БАРВИ В"
reasons = reasons + " ЗЛИВУ. "
ways = "У ПЕРШОМУ ВИПАДКУ ТРЕБА прочистити сітку, а ВО"

ways = ways + "ДРУГОМУ ДОДАТИ ФАРБУ В КРАСКОНАГНЕТА"
ways = ways + " Тельнов БАЧОК. "
REASON: ЯКЩО ЗОВСІМ НЕ НАДХОДИТЬ ФАРБА В краскопультам,
ТО ЦЯ ПРОБЛЕМА цілком вирішувані.
COMMENT: способи усунення проблем, ПОВ'ЯЗАНИХ З ненадходження БАРВИ В ПІСТОЛЕТ.

RULE: R4
IF: drops = "Y" or drops = "y"
THEN: output
output "СЛАБКИЙ розпил струменями?"
input lowfluct str using "a"
REASON: ЯКЩО дефект пов'язані із струменем, ТО ШВИДШЕ ЗА ВСЕ ЦЕ
ЇЇ СЛАБКИЙ розпили.
COMMENT: СЛАБКИЙ ЧИ розпил струменя.

RULE: R5
IF: (drops = "Y" or drops = "y") and lowfluct <> "Y" and lowfluct <> "y"
THEN: output
output "ЗАНАДТО розкиданих СТРУМІНЬ, СИЛЬНА"
output "туманообразование?"
Input highfluct str using "a"
REASON: ЯКЩО ДЕФЕКТИ зв'язаність із струменем І СТРУМІНЬ НЕ СЛАБКА,
ТО ШВИДШЕ ЗА ВСЕ ЗАНАДТО СИЛЬНИЙ розпил ційної.
COMMENT: СИЛЬНИЙ ЧИ розпил струменя.

RULE: R6
IF: (drops = "Y" or drops = "y") and (harddrops = "Y" or harddrops = "y")
THEN: reasons = "ТАКИЙ ЕФЕКТ МОЖЛИВИЙ ТІЛЬКИ В ТРЬОХ"
reasons = reasons + "ВИПАДКАХ: СЛАБКА ТИСК НА"
reasons = reasons + " ФАРБУ, ЗАНАДТО густою фарбою АБО "
reasons = reasons + "НЕПЛОТНО притиснуті повітропроводів."
ways = "ВІДПОВІДНО НЕОБХІДНО: ВІДРЕГУЛЮВАТИ"
ways = ways + "ДАВЛЕНЙЕ, розбавивши ФАРБУ розчинники,"
ways = ways + "притиснути І ЗАКРІПИТИ повітропроводів."
REASON: ЯКЩО СТРУМІНЬ БАРВИ ПОДАЄТЬСЯ різкого поштовху, ТО ЦЯ ПРОБЛЕМА цілком вирішувані.
COMMENT: способи усунення проблем, ПОВ'ЯЗАНИХ З різким поштовхом струменями БАРВИ В пістолета.

RULE: R7

IF: (drops = "Y" or drops = "y") and (lowfluct = "Y" or lowfluct = "y")

THEM: reasons = "ТАКИЙ ЕФЕКТ МОЖЛИВИЙ ТІЛЬКИ В ДВОХ ВИПАДКАХ:"

reasons = reasons + "МАЛО ПОВІТРЯ або забруднений ОДОБУТОК"

reascns = reasons + "СОПЛА."

ways = "ВІДПОВІДНО НЕОБХІДНО: ПЕРЕВІРИТИ І влаштую"

ways = ways + "НИТКА МІСЦЯ УТЕЧКИ ПОВІТРЯ, прочистив соплі."

REASON: ЯКЩО СТРУМІНЬ БАРВИ МАЄ СЛАБКИЙ розпил, ТО ЦЯ ПРОБЛЕМА ЦІЛКОМ ДОЗВОЛИЛА.

COMMENT: способи усунення проблем, пов'язано зі слабкою розпорошити струменями краскопультам.

RULE: R8

IF: (drops = "Y" or drops = "y") and (highfluct = "Y" or highfluct = "y")

THEN: reasons = "ТАКОН ЕФЕКТ МОЖЛИВИЙ ТІЛЬКИ В ТРЬОХ ВИПАДКАХ:"

reasons = reasons + "забруднив кінчику голки, спрацював"

reasons = reasons + "сальникове ущільнення ГОЛКИ"

reasons = reasons + "ЗАЙВА ПОДАЧА ПОВІТРЯ АБО Недосів"

reasons = reasons + "ТАТОЧНАЯ ПОДАЧА БАРВИ."

ways = "ВІДПОВІДНО НЕОБХІДНО: ОЧИСТИТИ кінчик"

ways = ways + "ГОЛКИ, ПІДТЯГТИ Гайки Сальникова"

ways = ways + "УЩІЛЬНЕННЯ, ВІДРЕГУЛЮВАТИ ПОДАЧУ ПОВІТРЯ"

ways = ways + "І ФАРБИ."

REASON: ЯКЩО СТРУМІНЬ БАРВИ МАЄ СИЛЬНИЙ розпил, ТО ЦЯ ПРОБЛЕМА цілком вирішувані.

COMMENT: способи усунення проблем, ПОВ'ЯЗАНИХ З СИЛЬНИМ розпорошити струменями краскопультам.

RULE: R9

IF: not (drops = "Y" or drops = "y") and (badklap = "Y" or badklap = "y")

THEN: reasons = "ТАКИЙ ЕФЕКТ МОЖЛИВИЙ ТІЛЬКИ В ДВОХ"

reasons = reasons + "ВИПАДКАХ: спрацювати ИГЛА СОПЛА,"

reasons = reasons + "ИГЛА НЕ закриває одобуток СОПЛА."

ways = "ВІДПОВІДНО НЕОБХІДНО: притертися АБО"

ways = ways + "СМЕНІТЬ ІГЛУ, ВИСУНУТИ ІГЛУ ШЛЯХОМ"

ways = ways + "відгвинчуванням регулювальні ГАЙОК."

REASON: ЯКЩО НЕПОЛАДКИ СКЛАДАЮТЬСЯ В протечкой БАРВИ ПРИ закриття клапанів, ТО ЦЯ ПРОБЛЕМА цілком вирішувані.

COMMENT: способи усунення проблем, пов'язані з протіканням БАРВИ ПРИ закриття клапанів.

VAR: GUNSDEF
LABEL: НАЯВНІСТЬ ДЕФЕКТІВ пістолета-краскопультам.

VAR: NOTKRASK
FIND: output "ФАРБА ЗОВСІМ НЕ ПОДАЄТЬСЯ В ПІСТОЛЕТ (Y / N)?"
input notkrask str using "a"
LABEL: ПОДАЧА БАРВИ В ПІСТОЛЕТ.

VAR: HARDDROPS
FIND: output "СПОСТЕРІГАЮТЬСЯ різкого поштовху І ПУЛЬСАЦІЯ"
output "струменя (Y / N)?"
input harddrops str using "a"
LABEL: СТРУМІНЬ пульсує І БУВАЮТЬ різкого поштовху.

VAR: BADKLAP
FIND: output "КРАСКА ПРОХОДИТЬ ПРИ закриття клапанів (Y / N)?"
input badklap str using "a"
LABEL: негідника КЛАПАН пістолет.

VAR: WAYS
FIND: reasons = "ПРИЧИНА ПОЯВИ ЦИХ НЕПОЛАДОК МЕНІ"
reasons = reasons + "НЕІЗВЕСТНА."
ways = "СКУШТУЙТЕ ЗВЕРНУТИСЯ В МАЙСТЕРНЮ."
LABEL: способи усунення проблем.

END:

У варіанті 8 пропонується передбачити зміни, які враховують ситуацію:

наявність дефекту в пістолеті, не пов'язаного із струменем (не було натиснуто кнопка приведення пістолета в дію або фарба випливає з пістолета через якісь щілини, але не через сопло).

варіант 9

GOAL: ways

/ * ЦЕЙ НАБІР ПРАВИЛ ДОЗВОЛИТЬ ВАМ ОТРИМАТИ РЯД РАД ПО * /
/ * Усунення деяких ПОШКОДЖЕНЬ ЗВИЧАЙНОГО * /

```
/ * Паяльник і повідомили МОЖЛИВІ ПРИЧИНИ ЇХ ПОЯВИ. * /
/ * ЗВИЧАЙНО, ЦЕ МАЛЕНЬКИЙ, НЕПОВНИЙ НАБІР, АЛЕ ВІН І
* /
/ * ПРИЗНАЧЕНИЙ ДЛЯ ПОКАЗУ найтривіальніших ЗАСОБІВ
GURU. * /
/ * ТАК ЩО БУЛО Б З ВАШОГО БОКУ НАЇВНО ЧЕКАТИ ЧОГО-
ЯКИХ * /
/ * СВЕРХІНТЕЛЕКТУАЛЬНОГО. * /
/ * НА ЗАПИТАННЯ СИСТЕМИ СЛІД вводяться відповідні * /
/ * ЗНАЧЕННЯ булевскому ЗМІННОЮ (ДА-TRUE, НЕТ-FALSE). * /

INITIAL:
clear
release variable / * ПРИБИРАЄМО НЕПОТРІБНІ НАМ ЗМІННІ * /
e.lstr = 250 / * Максимальна довжина рядка * /
output "ДЕНЬ ДОБРИЙ, МІСТЕР (МІСІС)."
output
output "ВИ ВКЛЮЧИЛИ В МЕРЕЖА ПАЯЛЬНИК І ВИЯВИЛИ, ЩО
ВІН"
output "невиправний. ми постараємося надати Вам пораду"
output "УСУНЕННЯ І причина появи несправностей."
output "АЛЕ ДЛЯ ЦЬОГО ВИ ДОЛИНИ НАДАТИ МЕНІ"
output "ВСЮ ІНФОРМАЦІЮ."
output "ОТЖЕ, НАЧНЕМ ..."
output
temperature = true
output "ПАЯЛЬНИК НЕДОСТАТНЬО розігріли?"
input temperature logic

DO:
clear
output "ОСЬ ЩО Я ВАМ СКАЖУ, люб'язно."
output
output reasons
output
output "А ОСЬ ЩО ВАМ СЛІД ЗРОБИТИ В ЦІЙ СИТУАЦІЇ."
output
output ways

RULE: Rl
IF: temperature
THEN: output
output "ПІСЛЯ ВКЛЮЧЕННЯ ПАЯЛЬНИК нагріли?"
input lowtemp logic
```

REASON: ЯКЩО НЕДОСТАТНЬО нагріли, ТО ДО ЯКОЇ ТЕМПЕРАТУРИ. COMMENT: нагрілися ПАЯЛЬНИК ХОЧ НА малості.

RULE: R2
IF: not temperature
THEN: output
output "ПІСЛЯ ВКЛЮЧЕННЯ ПАЯЛЬНИК перегрівся?"
input hightemp logic
REASON: ЯКЩО перегрівся, ТО ШВИДШЕ ЗА ВСЕ ЦЕ МЕЖВИТКОВОЕ ЗАМИКАННЯ.
COMMENT: перегрівся ЧИ ПАЯЛЬНИК.

RULE: R3
IF: not lowtemp or hightemp
THEN: output
output "ГАРАЗД ЧИ НАПРУГУ В МЕРЕЖІ?"
input power logic
REASON: ЯКЩО ПАЯЛЬНИК НЕ нагріли ХОЧ НА малості АБО перегрівся, ТО ШВИДШЕ ЗА ВСЕ несправних МЕРЕЖА.
COMMENT: НАПРУГУ В МЕРЕЖІ.

ROLE: R4
IF: lowtemp
THEN: output
output "НЕ чутні ЧИ потріскування В паяльником"
output "АБО Соеда. шнуром?"
input noice logic
REASON: ЯКЩО ПАЯЛЬНИК НЕДОСТАТНЬО розігріли, ТО ШВИДШЕ ЗА ВСЕ ЦЕ ОЗНАЧАЄ розрив у ланцюгу ХАРЧУВАННЯ АБО нагрівальний елемент.
COMMENT: чуєте тріск.

RULE: R5
IF: not hightemp and not temperature
THEN: reasons = "ПО ВАШИМ відповідях ПАЯЛЬНИК І НЕ перегрівся"
reasons = reasons + "І НЕ НЕДОГРЕЛСЯ ОДНОЧАСНО."
ways = "Піду КРАЩЕ відпочину, поспи І Т. Д .."
REASON: ЯКЩО такий розклад, ТО ТРЕБА поспати.
COMMENT: НІ ДЕФЕКТІВ - ВІДПОЧИВАЙ.

RULE: R6
IF: not power
THEN: output

reasons = "ПРИЧИНА ПОЯВИ ЦИХ НЕПОЛАДОК ЛЕЖИТЬ ПОЗА"
reasons = reasons + "МЕЖ паяльника. ВОНА КРИЄТЬСЯ В"
reasons = reasons + "ВЕЗОТВЕТСТВЕННОЙ РОБОТІ"
reasons = reasons + Горелектросеті, НЕСПРАВНОСТІ ПРОБОК "
reasons = reasons + "АБО протизаконне діяння ЯКИХОСЬ" reasons = reasons + "ХУЛІГАНІВ, ЯКІ ВАШИХ СУСІДІВ."
ways = "ЯКЩО ВИ в змозі усунути всі перераховані"
ways = ways + "ВИЩЕ ПРИЧИНИ, то зробити це."
REASON: ЯКЩО НІ напруги В МЕРЕЖІ, то варто почекати, ПОКИ ВОНО З'ЯВИТЬСЯ, ЯКИХ саморучно ВИПРАВИТИ ПРИЧИНИ ЙОГО ВІДСУТНОСТІ.
COMMENT: ВІДСУТНІСТЬ напруги В МЕРЕЖІ.

RULE: R7
IF: power and not lowtemp
THEN: output
output "ГАРАЗД шнур живлення (НІ ЧИ пориви,"
output "Горіло МІСЦЬ)?"
input cabel logic
REASON: ЯКЩО мережевої напруги ГАРАЗД, ТО ШВИДШЕ ЗА ВСЕ ЦЕ ОЗНАЧАЄ РОЗРИВІ ШНУРА ХАРЧУВАННЯ паяльника.
COMMENT: В ПОРЯДКУ шнур живлення.

RULE: R8
IF: power and hightemp
THEN: reasons = "ПРИЧИНА ПОЯВИ ЦИХ НЕПОЛАДОК ШВИДШЕ" reasons = reasons + "ВСЬОГО КРИЄТЬСЯ В ТОМУ, ЩО СТАЛОСЯ" reasons = reasons + "МЕЖВИТКОВОЕ ЗАМИКАННЯ У" reasons = reasons + "нагрівальними елементами паяльника." ways = "Розтин ПАЯЛЬНИК І перемотати нагрівальними" ways = ways + "ЕЛЕМЕНТ, АБО ВІДДАТИ В РЕМОНТ АБО ПРОСТО" ways = ways + "ВИКИНУТИ."
REASON: ЯКЩО мережевої напруги ГАРАЗД, ТО ШВИДШЕ ЗА ВСЕ СТАЛОСЯ МЕЖВИТКОВОЕ ЗАМИКАННЯ У нагрівальних елементів паяльника.
COMMENT: коротке замикання.

RULE: R9
IF: noice
THEN: cabel = false
REASON: ЯКЩО чути тріск, ТО НЕ В ПОРЯДКУ ШНУР ХАРЧУВАННЯ. COMMENT: СПОСІБ УСУНЕННЯ НЕСПРАВНОСТІ, супроводжувані тріском.

RULE: R10

IF: not noice

THEN: reasons = "ПАЯЛЬНИК ПІД'ЄДНАНО до невідповідності"

reasons = reasons + "джерела напруги."

ways = "підключив прилад до джерел," ways = ways + "які відповідають умовам експлуатації,"

ways = ways + "ВКАЗАНИМ НА КОРПУСІ паяльника."

REASON: ЯКЩО ПАЯЛЬНИК НЕ досягається необхідний ТЕМПЕРАТУРИ БЕЗ БУДЬ-ЯКИХ ДОДАТКОВИХ ВІДХИЛЕНЬ, ТО ШВИДШЕ ЗА ВСЕ ЦЕ ВІДБУВАЄТЬСЯ через невідповідність ДЖЕРЕЛА ХАРЧУВАННЯ.

COMMENT: невідповідність ДЖЕРЕЛО ХАРЧУВАННЯ.

RULE: R11

IF: not cabel

THEN: reasons = "Знос ШНУР ХАРЧУВАННЯ (переломи ПРОВОД)" reasons = reasons + "або штепсельна вилка."

ways = "ЗАМІНИТИ кабелі живлення чи ШТЕПСЕЛЬНУ ВИЛКУ."

COMMENT: зносилося ШНУР ХАРЧУВАННЯ.

RULE: R12

IF: cabel

THEN: reasons = "ПРИЧИНА ПОЯВИ ЦИХ НЕПОЛАДОК НАМ" reasons = reasons + "НЕІЗВЕСТНА."

ways = "СКУШТУЙТЕ ЗВЕРНУТИСЯ до досвідченішого"

ways = ways + " СПЕЦІАЛІСТУ. "

REASON: якщо такий ЗОВНІШНЄ ОПИС НЕСПРАВНОСТІ, ТО нічого путнього ЗАПРОПОНУВАТИ ВАМ Я НЕ МОЖУ.

COMMENT: ЯКІСЬ НЕ зарезервували ПОЛОМКИ.

VAR: temperature

LABEL: ПАЯЛЬНИК розігріли НЕДОСТАТНЬО.

VAR: WAYS

FIND: reasons = "ПРИЧИНА ПОЯВИ ЦИХ НЕПОЛАДОК НАМ НЕВІДОМА"

ways = "СКУШТУЙТЕ ЗВЕРНУТИСЯ ДО досвідченішого фахівця."

LABEL: способи усунення проблем.

END:

У варіанті 9 пропонується передбачити зміни, які враховують ситуації:

а) поломку жала паяльника;

б) дуже тривале нагрівання паяльника.

варіант 10

```
GOAL: howrest

/ * ЕТА ЕКСПЕРТНА системка ДАСТЬ ВАМ РЯД РАД, ЯК ПРОВЕСТИ * /
/ * ДОВГООЧІКУВАНИЙ ЛІТНІЙ ВІДПУСТКУ ПРИ ВАШОМУ ФІНАНСОВОМУ * /
/ * ПОЛОЖЕННІ І щодо числа БАЖАЮЧИХ * /
/ * ВІДПОЧИТИ ЗА ВАШ РАХУНОК. ЯКЩО ЗАСОБІВ ДЛЯ ВІДПОЧИНКУ НА ВОДІ * /
/ * НЕДОСТАТНЬО, ТО МОЖЕ БУТИ, ВАШІ СИМПАТІЇ БУДУТЬ * /
/ * Віддатися відпочинку В ЛІСІ І ГОРАХ. ЗВИЧАЙНО, * /
/ * ЦЕ ДАЛЕКО НЕ ПОВНИЙ НАБІР, АЛЕ Ж ВІН І ПРИЗНАЧЕНИЙ * /
/ * ДЛЯ ПОКАЗУ найтривіальніших ЗАСОБІВ GURU. ТАК * /
/ * ЩО НЕ ШУКАЙТЕ В НЬОМУ СВЕРХІНТЕЛЕКТУАЛЬНОГО. * /
/ * НА ЗАПИТАННЯ СИСТЕМИ СЛІД вводяться відповідні * /
/ * ЗНАЧЕННЯ строкових змінних (ДА-Y, НЕТ-N) АБО ЧИСЛО. * /

INITIAL:
clear
release variable / * ПРИБИРАЄМО НЕПОТРІБНІ НАМ ЗМІННІ * /
e.lstr = 250 / * Максимальна довжина рядка * /
output "ДЕНЬ ДОБРИЙ, МІСТЕР (МІСІС)."
output "Настала пора ВАШОГО довгоочікувані літні відпустки."
output "ЯК ЙОГО ПРОВЕСТИ? ми намагаємось дати ВАМ РАДА, ЯК"
output "ПРИ Ваше фінансове становище ПРОВЕСТИ БАЖАНІ"
output "ДЕНЬКИ."
output "АЛЕ ДЛЯ ЦЬОГО ВИ ПОВИННІ НАДАТИ МЕНІ"
output "ВСЮ ІНФОРМАЦІЮ."
output "ОТЖЕ, НАЧНЕМ ..."
finance = 0
output "ЯКУ СУМУ ВИ ГОТОВІ БУЛИ Б ВИТРАТИТИ?"
input finance num

DO:
clear
output "ОСЬ ЩО МЕНІ ЗДАЄТЬСЯ прийнятних у даній СИТУАЦІЇ."
output
output howrest
```

RULE: Rl

IF: (finance / acount)> = 5000 and (f inance / acount) <9000

THEN: howrest = "ВИ МОЖЕТЕ ПРИДБАТИ КОЖНОМУ ПО путівку в" howrest = howrest + "САНАТОРІЇ НА ПОБЕРЕЖЬЕ БУДЬ-ЯКОЮ" howrest = howrest + "РІЧКИ, ОЗЕРА АБО ВОДОСХОВИЩА."

RULE: R2

IF: (finance / acount)> = 9000 and (finance / acount) <90000

THEN: howrest = "ВИ МОЖЕТЕ ПРИДБАТИ КОЖНОМУ ПО путівку в" howrest = howrest + "САНАТОРІЙ на узбережжі Чорного"

howrest = howrest + "МОРЯ. ТАКИЙ ВІДПОЧИНОК ВАМ ЗАПАМ'ЯТАЄТЬСЯ"

howrest = howrest + "НАДОВГО."

RULE: R3

IF: (finance / acount)> = 90000

THEN: howrest = "ВИ МОЖЕТЕ ПЕРЕКАЗАТИ ЧАСТИНА ВАШИХ КОШТІВ В" howrest = howrest + "ВКВ І ПРИДБАТИ КОЖНОМУ ПО"

howrest = howrest + "путівку на Канарські острови АБО" howrest = howrest + " ДВУХНЕДЕЛЬНИІ КРУЇЗ. ФІРМА "howrest = howrest +" ГАРАНТУЄ, ЩО ТАКИЙ ВІДПОЧИНОК "howrest = howrest +" ДОЗВОЛИТЬ ВАМ ЗАБУТИ АБСОЛЮТНО ВСЕ "howrest = howrest +" ВАШІ ПРОБЛЕМИ. "

RULE: R4

IF: (finance / acount) <5000 and (finance / acount)> = 3000

THEN: howrest = "ВАМ ВАРТО подумати про відпочинок В ЛІСІ АБО" howrest = howrest + "ГОРАХ, ДЛЯ ТАКОГО ВІДПОЧИНКУ У ВАС" howrest = howrest + "СРЕДСТВ ЦІЛКОМ ДОСТАТНЬО, А" howrest = howrest + "ВОЗМОЖНО варто відправити дикунами" howrest = howrest + "НА МОРЕ. "

RULE: R5

IF: finance> 3000

THEN: output

output "А СКІЛЬКИ ЛЮДИНА БУДУТЬ ВІДПОЧИВАТИ ЗА ВАШ РАХУНОК?"

input acount num using "nn"

output

RULE: R6

IF: (finance / acount) <3000

THEN: howrest = "ВАМ ВАРТО ПОДУМАТИ ЯКОЇ Про СОКРАЩЕНИЙ ЧИСЛА" howrest = howrest + "відпочиває за ВАШ РАХУНОК, ЯКИХ ОБ"
howrest = howrest + "ЗБІЛЬШЕННЯ СУМИ НА ВІДПОЧИНОК."

RULE: R7
IF: finance <= 3000 and finance> 1000
THEN: output "ПРОСТИТЕ ЗА нетактовне, ВИ"
output "Одружений (Замужем) (Y / H)?"
input wife str using "a"

RULE: R8
IF: (wife = "Y") or (wife = "y")
THEN: output "ВИ проведе відпустку РАЗОМ (Y / H)?"
input withwife str using "a"

RULE: R9
IF: (withwife = "Y") or (withwife = "y")
THEN: howrest = "НА ТАКУ СУМУ НА ДВОХ МОЖНА ВІДПОЧИТИ"
howrest = howrest + "ТІЛЬКИ СИДЯЧИ ВДОМА АБО КОЛУПАЮЧИ НА ДАЧІ."

RULE: R10
IF: finance <= 1000
THEN: howrest = "НА ТАКУ СУМУ МОЖНА ВІДПОЧИТИ ТІЛЬКИ"
howrest = howrest + " сидячи вдома АБО колупаючи НА ДАЧІ. "

RULE: R11
IF: wife <> "Y" or wife <> "y" or withwife <> "Y" or withwife <> "y"
THEN: output "ВИ ХОТІЛИ Б вибратися на природу (Y / N)?"
input townout str using "a"

RULE. R12
IF: townout = "Y" or townout = "y"
output "ВАМ ПОДОБАЄТЬСЯ ВІДПОЧИНОК В ЛІСІ (Y / N)?"
input forest str using "a"

RULE: R13
IF: (townout = "Y" or townout = "y") and forest <> "Y"
and forest <> "y"
THEN: output "ВАМ ПОДОБАЄТЬСЯ ВІДПОЧИНОК В ГОРАХ (Y / N)?"

input mountain str using "a"

RULE: R14

IF: (townout = "Y" or townout = "y ") and mountain <>" Y "and mountain <>" y "

THEN: howrest = "В ТАКОМУ фінансове становище ВАМ ВАРТО" howrest = howrest + "ОСТАТЬСЯ ДОМА І МОЖЛИВО разок -" howrest = howrest + "інші сходяться В КАФЕ 'МОРОЗИВО'."

howrest = howrest + "ІНОЙ ВІДПОЧИНОК ДЛЯ ВАС НЕДОСТУПНИЙ."

RULE: R15

IF: forest = "Y" or forest = "y"

THEN: howrest = "ПРИ ВАШИХ ЗАСОБАХ ВАМ ВАРТО ВЗЯТИ" howrest = howrest + "НАПРОКАТ, НА 3 ТИЖНЯ, одномісний" howrest = howrest + "націлі, рюкзаки, спальний мішок," howrest = howrest + "заготовила На решту суми" howrest = howrest + "МАКАРОНИ , тушонка, СУХОФРУКТИ, "howrest = howrest +" СПІЧКІ І Т. П. І відправив у "howrest = howrest +" придивилася лісок. "

RULE: R16

IF: mountain = "Y" or mountain = "y"

THEN: howrest = "ПРИ ВАШИХ ЗАСОБАХ ВАМ ВАРТО ВЗЯТИ НА" howrest = howrest + "ІПРОКАТ, НА 2 ТИЖНЯ, одномісний" howrest = howrest + "націлі, рюкзаки, спальний мішок," howrest = howrest + "КУПИТИ КВИТОК туди-назад." howrest = howrest + "ЗАГОТОВНТЬ На решту суми" howrest = howrest + "МАКАРОН, тушонка, СУХОФРУКТІВ"

howrest = hewrest + "І Т.П. І відправила до вершин ..."

RULE: R17

IF: townout <> "y" or townout <> "Y"

THEN: howrest = "У ВАШИХХ ІНТЕРЕСАХ ВАМ ВАРТО ЗАЛИШИТИСЯ У БУДИНКУ" howrest = howrest + "І МОЖЛИВО разок-другий СХОДИТЬ" howrest = howrest + "В КАФЕ 'МОРОЖЕНОЕ'. БУТИ МОЖЕ, ЗА" howrest = howrest + "ВІДПУСТКУ ви встигнете зробити багато" howrest = howrest + " КОРИСНОГО ДЛЯ СЕБЕ І БЛИЗЬКИХ ВАМ "howrest = howrest +" ЛЮДЕЙ. "

VAR: finance

LABEL: СУМА, яку ви готові витратити НА ВІДПОЧИНОК.

END:

У варіанті 10 пропонується передбачити зміни, які враховують:

а) возможность подзаработать за некоторое время отпуска и потратить заработанное на отпуск;

б) возможность одолжить деньги на проведение отпуска.

5 Лабораторна робота №5
Створення пробної експертної системи.

Мета - самостійне програмування в повному обсязі найпростішої експертної системи.

Реалізація названої цілі лабораторної роботи передбачає грунтовну проробку наведеного в додатку І цієї лабораторної роботи прикладу реалізація експертної системи. Для полегшення такого пророоблення це додаток забезпечено докладним коментарем.

5.1 Теоретичні відомості

Основні команди GURU

BUILD <ім'я набору правил> - використовується для створення, модифікації і компіляції набору правил.

COMPILE <ім'я набору правил> - створює файл відкомпільованого набору правил.

CONSULT <ім'я експертної системи> - звернення до ЕС за консультацією.

WHY <вираз> - пояснює, чому "GURU" використовувала конкретне правило.

HOW <вираз> - пояснює, як "GURU" знайшла значення змінної.

TEXT <ім'я файлу> - запускається текстовий редактор "GURU".

CLEAR - очистити екран.

INPUT <змінна> <USING шаблон> <WITH вираз> - введення даних в змінну з використанням шаблону (USING) і з підказкою (WITH). шаблони:

a - для буквених символів (латинський шрифт);

c - для букви або числа;

d - для цифри, знака (+ чи -) або десяткового дробу;

e - перетворення в символи нижнього регістра;

n - символ шаблону, який сприймає тільки цифри в займаній ним позицій;

r - для символів ASCII;

u - перетворить в символи верхнього регістру. наприклад:

INPUT num USING "dddd" with "Введіть номер"

На екрані з'являється текст:

Введіть номер __

Розглянемо ще одну команду "GURU".

OUTPUT <ім'я змінної> <USING шаблон> - виводить на екран змінну або рядок.

Шаблони У цій команді аналогічні шаблонам для команди INPUT.

наприклад:

OUTPUT " Лабораторна робота N1 "

Виводить на екран:

Лабораторна робота N1

Інший приклад:

OUTPUT num

Виводить на екран значення змінної num. Наведемо також перелік наступних команд "GURU":

HELP - виводить довідкову інформацію;

RUN - виконує зовнішню програму;

DIR - переглядає директорію;

BYE - виходить із режиму;

RELEASE - звільняє пам'ять, видаляючи дані і програми "GURU";

PEPFORM - виконує процедуру;

WAIT - призупиняє обробку до натискання будь-якої клавіші.

1. Перелічіть

5.2 Підготовка до роботи

1. Ознайомтеся з матеріалом лекцій та даними методичними вказівками.

2. Дайте відповідь на контрольні питання.

3. Запустіть DOSBOX та пропишить конфігураційному файли для монтування папки з оболонкою VC та GORU та запуску оболлнки VC.

```
MOUNT c E:\DOS22\
c:
CD VC
VC
```

4. Запустsть оболонку VC, потім - утиліту cyrilik.com, потім - GORU. Ознайомтеся з готовим варіантом ЕС (додаток 1 до цієї лабораторної роботи).

5. Відповідно до заданих варіантом (додаток 2 до цієї лабораторної роботи) напишіть програму, що реалізовує невелику експертну систему. Практично будь-який з заданий варіанту може бути реалізований невеликим набором правил.

5.3 Порядок виконання роботи

1. Пред'явіть викладачеві текст програмної реалізації заданого варіанту ЕС, написаний в ході домашньої підготовки.

2. Отладьте ЕС.

3. За допомогою HOW і WHY перевірте правильність роботи системи.

4. Покажіть результати роботи ЗС викладачеві.

5.4 Приклад експертної системи

```
/ * INVESTORS.RSS * /
/ * * /

GOAL: advice

INITIAL:
e.tryp = "e"
e.lstr = 80 / * Максимальна довжина рядка 80 * /
e.deci = 0 / * нема цифр після десяткової коми * /
e.lnum = 8 / * Довжина числа * /
savperdep = 5000 / * Допустимий мінімум заощаджень * /
/ * У розрахунку на одного утриманця, при якому загальні сбере- * /
/ * ження сім'ї можна назвати 'хорошими' (true) * /
incperdep = 4000 / * Мінімально допустимий дохід на * /
/ * Одного утриманця * /
basincome = 15000 / * Мінімально допустимий дохід * /
  / * Глави сім'ї * /
/ * Загальний хороший дохід сім'ї визначається об'єднанням * /
/ * Incperdep і basincome * /
advice = unknown
goodsave = unknown
goodincome = unknown
income = unknown
savings = unknown
steady = unknown
needincome = unknown
dependents = unknown
newcash = unknown
clear
output "КАПІТАЛОВКЛАДЕННЯ"
input newcash num
```

```
DO: / * Цей розділ виконується після того, * /
/ * Як оброблені правила * /

output "НА ОСНОВІ ЦІЄЇ ІНФОРМАЦІЇ:"
test advice
case "АКЦІЇ":
output "ВАМ СЛІД ВКЛАСТИ ВРЮ СУМУ В АКЦІЇ."
break
case "ЗАОЩАДЖЕННЯ":
output "ВАМ СЛІД помістити всю СУМУ В ЗАОЩАДЖЕННЯ."
break
case "КОМПРОМІС":
tosave = min (newcash, (savperdep * dependents) - savings)
tostock = max (0, newcash - tosave)
output "ВАМ слід помістити в ЗАОЩАДЖЕННЯ"
tosave using "$ ff, fff, fff"
if tostock> 0 then
output "І ВКЛАСТИ", tostock using "$ ff, fff, fff", "В АКЦІЇ."
endif
break
endtest
e.deci = w
e.lnum = 14

RULE: R1
IF: goodincome and goodsave
THEN: advice = "АКЦІЇ"
NEEDS: goodincome goodsave
REASON: ВКЛАДАТИ В АКЦІЇ, якщо клієнт
НАДІЙНИЙ У фінансовому відношенні

RULE: R2
IF: not goodincome
THEN: advice = "ЗАОЩАДЖЕННЯ"
NEEDS: goodincome
REASON: НЕ вкладаються в акції, ЯКЩО ВАШ ДОХІД
НИНІ нестійкий

RULE: R3
IF: not goodsave and goodincome
THEN: advice = "КОМПРОМІС"
REASON: ЯКЩО ЗАОЩАДЖЕННЯ НЕВЕЛИКІ, ТОДІ ВОНИ ПОВИННІ
Бути збільшена до ТОГО, ЯК ЇХ ВКЛАДАТИ
```

RULE: R4
IF: not steady
THEN: goodincome = false
REASON: ДЛЯ ГАРНОГО ДОХОДУ НЕОБХІДНА ПОСТІЙНА РОБОТА

RULE: R5
IF: not (income> needincome)
THEN: goodincome = false
REASON: ДОХІД НЕ ЗАЛЕЖИТЬ ВІД ВАС І ВІД утриманців

RULE: R6
IF: not (income> needincome)
THEN: goodincome = true
REASON: ЩОБ ДОХІД БУВ ХОРОШИЙ, КЛІЄНТ ПОВИНЕН МАТИ
ПОСТІЙНУ РОБОТУ

RULE: R7
IF: known ("income") and known ("dependents")
THEN: needincome = baseincome + (dependents * incperdep)
REASON: НЕОБХІДНИЙ ДОХІД - ЦЕ ДОХІД, ЯКИЙ ВАМ
НЕОБХІДНИЙ ПЛЮС ЗАГАЛЬНИЙ ДОХІД ВСІХ ВАШИХ утриманців

RULE: R8
IF: savings> (saveperdep * dependents)
THEN: goodsave = true
REASON: ЗАОЩАДЖЕННЯ КЛІЄНТІВ повинні залежати від НИХ
САМИХ І ВІД утриманців

RULE: R9
IF: savings <= (saveperdep * dependents)
THEN: goodsave = false
REASON: ЗАОЩАДЖЕННЯ КЛІЄНТІВ повинні залежати від НИХ
САМИХ І ВІД утриманців

NEEDS: savings dependents
REASON: ЗАОЩАДЖЕННЯ КЛІЄНТІВ НЕ ЗАЛЕЖАТЬ ВІД НИХ САМИХ І ВІД утриманців.
/ * Визначення змінних * /
VAR: NEWCASH
LABEL: суми готівки ДЛЯ ВКЛАДУ

```
VAR: ADVICE
LABEL: ДАНИЙ РАДА
VAR: GOODINCOME
LABEL: поточнідоходи - ХОРОШИЙ
VaR: GOODSAVE
LABEL: поточні заощадження - ХОРОШІ
VAR: NEEDINCOME
LABEL: НЕОБХІДНА СУМА ДОХОДУ
VAR: INCOME
FIND: input income num with "ЯКОЮ ВАШ РІЧНИЙ ДОХІД СІМ'Ї?"
LABEL: поточнідоходи
VAR: SAVINGS
FIND: input savings num with "СКІЛЬКИ у вас ЗАОЩАДЖЕНЬ?"
LABEL: поточні заощадження
VAR: STEADY
FIND:
output "МОЖЕТЕ ВИ чекати стабільного ДОХІД НА"
output "СЛІД. РІК? (y / n)"
input steady str using "u"
steady = (steady = "Y")
LABEL: НАДІЙНИЙ ДОХІД
VAR: DEPENDENTS
FIND:
output "СКІЛЬКИ У ВАС утриманців? "
input dependents num using "dd"
LABEL: кількість утриманців
END:
```

Опишемо детально роботу набору правил. Він призначений для ілюстрація зворотного аргументації.

В INITIAL йде ініціалізація змінних. Розглянемо її окремі рядки.

e.tryp = 'e' - задає стратегію оцінки посилки (частини "if" правила), що містить невідомі змінні. Істинність посилки оцінюється відразу ж після того, як чергова невідома змінна стає відомою. Тестування посилки припиняється (незважаючи на те, що всі змінні в ній ще не визначені), якщо тільки вдається точно встановити її істинність або хибність.

e.lstr = 80 - максимальна довжина символьного рядка, яка може виводитися на екран.

e.lnum = максимальна довжина числа.

В VAR описуються користувача змінні (см. Списання лабораторної роботи 1).

Частина DO - закінчення роботи експертної системи. Конструкція test ... case ... endtest перевіряє змінну advice і залежно від її значення виконує ті чи інші дії.

Розглянемо, як може працювати ця система. Після запуску відбувається ініціалізація змінних. Консультація з ЕС йде методом зворотної аргументації. Система "GURU" в цьому випадку починає е кінця. Визначається мета ADVICE. Т.к. вона невідома, то проглядаються ті правила, які визначають ADVICE. У нашому випадку першим правилом, де визначається ADVICE буде R1. Тут невідомі GQODINCOME і GOODSAVE.

GQODINCOME спочатку визначається в R4. GOODSAVE - в R8. В R4 перевіряється змінна STEADY. Її значення запитується за допомогою оператора FIND в описі змінної. Якщо R4 не може бути виконано (посилка помилкова), то тестується R5, якщо R4 істинно (посилка вірна), змінної GOODINCOME присвоюється значення "false", "GURU" визначає, що посилка правила R1 невірна і переходить до оцінки посилки правила R2. В R2 посилка вірна, тому з цього правила визначається advice - мета системи. На цій висновок завершується. При тестуванні R5 нам доведеться визначити значення змінних INCOME і NEEDINCOME і т.д.

Аналогічно знаходиться змінна GOODSAVE (правила R8 і R9).

Перше питання, яке задасть ЕС: "Яку суму готівкових грошей Ви хотіли б вкласти?"

Введіть: 12000 <Enter>

Наступне питання: "Чи можете Ви очікувати стабільний дохід на наступний рік?"

Введіть: "Y"

"Який Ваш річний дохід сім'ї?"

Введіть: 35000 <Enter>

"Скільки у Вас утриманців?"

Введіть: 4

"Скільки у Вас заощаджень?"

Введіть: 10000 <Enter>

Експертна система виводять суму, яку слід відкласти в заощадження і суму, яку необхідно вкласти в акція.

Розберіть докладно роботу цієї ЕС для того, щоб створити свою.

Пріложеніе.2 Варіанти завдань

1.2 Варіанти завдань

варіант 1

Розробіть ЕС, яка повторює хід ваших думок при переході через дорогу. Спочатку ви визначаєте, який колір на світлофорі. Якщо червоний - чекаєте, якщо зелений - дивіться, чи немає якого-небудь "божевільного"

водія, якій міг би їхати на червоний колір. Якщо є, то чекайте, поки він проїде. Якщо ні, то переходите через дорогу.

варіант 2

Створіть ЕС, що визначає несправність магнітофона. Ви включаєте магнітофон, і він працює, але звучання погане. Ви перевіряєте чи, забруднена головка. Якщо так, то необхідно протерти головку. Якщо ні, то перевіряйте, чи правильно встановлена касета. Якщо ні, то поправляєте, а якщо правильно, то кажіть, що необхідно викликати майстра.

варіант 3

Створіть ЕС, що визначає приймати чи ні людини на роботу. Якщо людина не має вищої освіти, то відмовити. Якщо має, то скільки років пропрацював претендент за фахом. Якщо менше року, то відмовити, якщо більше - то прийняти. Якщо претендент має вчене звання, то запропонувати посаду наукового співробітника, якщо ні, то посаду інженера-конструктора.

варіант 4

Розробіть ЕС. яка дає поради при створенні та налагодженні програми .. Якщо ви написали програму на ЯВУ і при компіляції виявили синтаксичні помилки, то необхідно виправити програму. Якщо помилок немає, то необхідно запустити редактор зв'язків. Якщо редактор зв'язків видає помилки, то необхідно перевірити наявність всіх вихідних модулів. Якщо помилок при Лінкування немає, то програма готова до роботи.

варіант 5

Розробіть ЕС, яка визначає, чи буде сьогодні дощ. Спочатку ви визначаете, чи ясне небо. Якщо небо ясне, то дощу не буде. Якщо небо похмуре, то ви дивитеся, чи є на небі чорні грозові хмари. Якщо ні, то дощу не буде. Якщо є, то дивіться, в який бік вони рухаються. Якщо в вашу сторону, то дощ буде. Якщо ж ні, то не буде.

варіант 6

Розробіть ЕС, визначальну чи є у дитини труднощі при вивченні арифціліки. ЕС дитині пропонує по одному прикладу на додавання, віднімання, множення і ділення. Якщо він правильно відповідає на всі питання, то у нього немає проблем з арифцілікою. При неправильних відповідях система вказує на труднощі з виконанням конкретних операцій.

варіант 7

Розробіть ЕС, яка буде прогнозувати на біржі рівень цін. Якщо валютний курс долара падає, і процентні ставки ростуть то рівень цін на біржі падає. Якщо валютний курс долара зростає, і процентні ставки падають., то рівень цін на біржі падає. В інших випадках ситуація оцінюється як нестабільна.

варіант 8

Розробіть ЕС, яка буде прогнозувати, чи буде в результаті весняного паводку повінь чи ні. Якщо рівень води в річці в межах міста високий і йдуть сильні дощі, тепла погода і багато снігу в горах, то очікується

повінь. Якщо ж хоча б один з цих факторів не виконується, то повені не буде.

варіант 9

Необхідно визначити, даний об'єкт є танком або автомобілем. У танка є гармата і люк. У автомобіля є дверцята і колеса. У танка і автомобіля є кузов. Уточнюючи всі ці характеристики, ЕС повинна визначити об'єкт.

варіант 10

Ви вмикаєте телевізор, а він не працює. Ви хочете визначити, чому це сталося. Якщо запобіжник згорів, то його необхідно замінити. Якщо запобіжник цілий, то перевіряєте кабель живленнія. Якщо він розірваний в якомусь місці, то необхідно його замінити. Якщо кабель живлення цілий, і ви самі розбираєтеся в радіоелектроніці, то чиніие телевізор. Якщо не розбираєтеся, то викликаєте майстра.

варіант 11

Розробіть ЕС, яка повторює хід ваших думок при переході через залізницю. Спочатку ви визначаєте чи є свііглофор, якщр є то який колір на свііглофорі. Якщо червоний - чекаєте, якщо зелений - то переходите через дорогу. Якщо нема свііглофора дивіться, чи немає якого-небудь потяга якій міг би їхати. Якщо є, то чекайте, поки він проїде. Якщо ні, то переходите через дорогу.

варіант 12

Розробіть ЕС, що визначає несправність CD-плеєра. Ви включаєте його, і він працює, Але не звучить. Ви перевіряєте, чи забруднений диск. Якщо так, то необхідно його протерти . Якщо ні, то перевіряйте, чи правильно диск встановлений. Якщо ні, то поправляєте, а якщо правильно, то необхідно викликати майстра.

варіант 13

Створіть ЕС, визначальну приймати чи ні людини роботу водія. Якщо людина не має прав, то відмовити. Якщо має, то скільки років пропрацював претендент за фахом. Якщо менше року, то відмовити, якщо більше - то прийняти. Якщо претендент має категорію «Д», то запропонувати посаду водія автобуса, якщо ні, то посаду водія вантажівки.

5.5 Вимоги, до звіту

Звіт про роботу повинен містити

1) тема роботи та короткі теоретичні відомості.

2) граф міркувань вашого варіанту ЕС виконаний природньою мовою;

3} пояснення, чому в написаній вами ЕС обрані ті чи інші змінні

4) скан дерева ЕС вашого варіанту.

5) роздруківку rss-файлу варіант експертної системи;
6) висновки по роботі.

РЕКОМЕНДОВАНА ЛИТЕРАТУРА

1. Методи Системи штучного інтелекту : Навч. посіб. для студ. / В. М. Заяць, Р. М. Камінський; Нац. ун-т “Львів. політехніка”. – Л., 2004. – 173 с. – Бібліогр.: 21 назв.
2. Айзерман М.А., Браверманн Э.М., Розоноэр Л.И. Метод потенциальных функций в теории обучения машин.– М.: Наука, 2004. –384 с.
3. Лепский А.Е.. Броневич А.Г. Математические методы распознавания образов: Курс лекций. Таганрог:Изд-во ТТИ ЮФУ, 2009.– 155 с.
4. Потапов А.С., Распознавание образов и машинное восприятие.– С-Пб.: Политехника, 2007. 548 с.
5. Дэвид А. Форсайт, Джин Понс. [Computer Vision: A Modern Approach Компьютерное зрение. Современный подход]. — М. :«Вильямс», 2004. — 928 с. —ISBN 0-13-085198-1.
6. Журавлев Ю.И. Об алгебраическом подходе к решению задач распознавания или классификации. // Проблемы кибернетики.– М.: Наука, 2005. –Вып.33. с. 5-68

1.