

Министерство образования и науки Российской Федерации
Иркутский государственный технический университет
ФАКУЛЬТЕТ КИБЕРНЕТИКИ
КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ИНТЕЛЛЕКТНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Методические указания для практических занятий

Укрупненная группа направлений и специальностей 230000 «Информатика и вычислительная техника»

Направление подготовки: 230100 «Информатика и вычислительная техника»

Специальность: 230101 «Вычислительные машины, комплексы, системы и сети»

Иркутск
2012

Введение

Лабораторные работы (занятия) по дисциплине «Интеллектуальные информационные системы» предполагают практическое закрепление теоретических вопросов, рассматриваемых на лекциях. Используя учебную и научную литературу, студенты должны самостоятельно выполнить (программно реализовать) задания. Основной целью проведения лабораторных занятий является формирование у студентов практических навыков и эвристических знаний об подходах к выбору архитектуры программной системы, выбору удобного инструментария для решения задачи, поиска точек оптимизации переборных методик. Основным результатом лабораторных занятий студентов должно стать понимание особенностей задач искусственного интеллекта, их уровень абстракции относительно более часто встречающихся алгоритмически разрешимых задач, понятийного аппарата искусственного интеллекта.

Перечень лабораторных работ

1. Логическое программирование, факты и правила, формализация высказываний естественного языка.
2. Списки и структуры данных. Рекурсивная обработка информации. Интерпретация Пролог-программ.
3. Базы данных. Особенности Пролога как системы управления базами данных.
4. Перебор и повышение его эффективности. Задачи с удовлетворением ограничений.

Общие требования к форме отчетности по работам и заданиям

Контроль качества знаний студентов осуществляется с использованием различных форм диалогового метода:

- в форме прямого контакта преподавателя с одним студентом: вопрос преподавателя — ответ студента;
- в форме дискуссии с группой: вопрос преподавателя — выяснение позиции нескольких студентов по данному вопросу, либо чтение (повторение) лекционного и дополнительного материала;
- рассмотрение варианта дипломного проекта по теме курса.

Такое разнообразие возможностей использования диалогового контакта позволяет осуществить контроль качества знаний студентов в скрытой форме, не акцентируя внимание группы на его проверочной функции, и в то же время предоставляет преподавателю максимум необходимых данных о результатах обучения студентов. Преподаватель проводит опрос студентов и при подведении итогов практического занятия дает оценку наиболее активным студентам из группы. Важное место в диалоге занимает рассмотрение задач-аналогов, рассматриваемой проблеме, взятых из практического опыта преподавателя и студентов.

Общие рекомендации по проведению лабораторных занятий

Особенностью проведения лабораторных занятий является их творческий характер. Студенты не только демонстрируют глубокое освоение лекционного материала, но и активно реализуют эти знания в решении задач лабораторных работ, выполняемых самостоятельно или в группе по два человека. Основной формой лабораторных занятий является представление задачи в виде концептуальной логической модели в рамках формализмов дискретной математики, выбор метода и методики решения задачи, реализация методики в виде программной системы на языке программирования Пролог (или других языках программирования в зависимости от задания). Такое построение практических занятий позволяет

повысить качество подготовки студентов и объединить освоение теоретических знаний и овладение практическими навыками специалиста.

Структура организации лабораторных занятий студентов состоит из следующих элементов:

1. обсуждение лекционного материала с использованием конспектов лекций, учебных пособий и обязательной литературы, а также дополнительной литературы, список которой предлагается преподавателем;
2. проведение дискуссии в рамках решения проблемы формализации поставленной задачи в виде концептуальной логической модели, обсуждение современных методов решения задачи в контексте построенной формализации, обсуждение путей реализации задачи;
3. реализация программы студентами индивидуально и по группам в зависимости от сложности индивидуального задания;
4. диалоговый контакт преподавателя и студентов в форме «вопрос-ответ», направленный на оценку знаний студентов и восполнение обнаружившихся недостатков в понимании материала учебной дисциплины, а также специфики реализованного алгоритма и методики.

Далее рассматриваются индивидуальные указания по отдельной лабораторной работе.

Лабораторная работа 1. Факты и правила

Формализация высказываний естественного языка в виде Пролог–программы.

Задание. В работе¹ требуется формализовать высказывания в виде программы на языке Пролог. В программе требуется выполнить ряд запросов, объяснить выдаваемые системой результаты.

Цель работы. Приобрести навыки формализации высказываний на естественном языке в виде фактов, правил и запросов языка Пролог.

Время работы. На выполнение работы отводится два академических часа.

Индивидуальные задания.

1. Флэш — собака. Ровер — собака. Бутси — кошка. Стар — лошадь. Флэш черная. Бутси коричневая. Ревер рыжая. Стар белая. Домашнее животное — собака или кошка. Животное — домашнее животное или лошадь. У Тома есть собака не черного цвета. У Кейта есть лошадь или что-то черного цвета.

Запросы:

- Ровер рыжая?
 - Определить клички всех собак.
 - Определить владельцев чего-либо.
 - Определить владельцев животных небелого цвета.
2. Бутси — коричневая кошка. Корни — черная кошка. Мактэвити — рыжая кошка. Флэш, Ровер и Спот — собаки; Ровер

¹Задание на лабораторную работу N 1 разработано преподавателем кафедры Вычислительной техники Кибернетического факультета ИрГТУ, доцентом, к.т.н. С.С. Сосинской.

— рыжая, а Спот — белая. Все животные, которыми владеют Том и Кейт, имеют родословные. Том владеет всеми черными и коричневыми животными. Кейт владеет всеми собаками небелого цвета, которые не являются собственностью Тома.

Алан владеет Мактэвити, если Кейт не владеет Бутси и если Спот не имеет родословной. Флэш — пятнистая собака.

Запросы:

- Какие животные не имеют хозяев?
- Найдите всех собак и укажите их цвет.
- Укажите всех животных, которыми владеет Том.
- Перечислите всех собак Кейта.

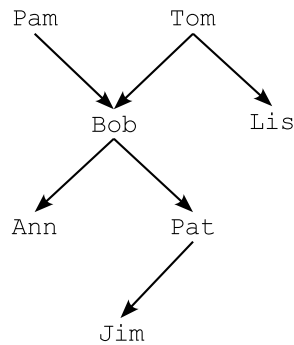
3. Определить следующие отношения: СЫН, ДОЧЬ, ОТЕЦ, МАТЬ, МУЖЧИНА и ЖЕНЩИНА. Описать факты для некоторых из них. **Запросы:**

- Определить всех сыновей определенной матери.
- Определить всех детей определенной пары родителей.
- Определить родителей определенного человека.
- Является ли определенный человек женщиной?

4. Мэри любит персики. Мэри любит кукурузу. Мэри любит яблоки. Бет любит то, что любит Мэри, если это — фрукт и если он красный. Бет любит то, что любит Мэри, если это кукуруза. Персики — фрукт. Яблоки — фрукт. Цвет персиков желтый. Цвет апельсинов оранжевый. Цвет яблок красный. Цвет яблок желтый. **Запросы:**

- Что любит Бет?
- Любит ли Мэри кукурузу?
- Какие фрукты известны?
- Какого цвета фрукты, которые любят Бет и Мэри?

5. Задано дерево родственных связей.



Кроме того, определить отношения ПОЛ, РЕБЕНОК, РОДИТЕЛЬ_РОДИТЕЛЯ, ПРЕДОК и МАТЬ.

Запросы:

- Кто родитель Pat?
- Есть ли у Lis ребенок?
- Кто потомки Pat?
- Является ли Pam матерью Bob?

6. Медведь большой. Слон большой. Кот маленький. Медведь коричневый. Кот черный. Слон серый.

Любой черный или коричневый предмет является темным. **Запросы:**

- Кто одновременно большой и темный?
- Есть ли коричневые маленькие слоны?
- Есть ли большие и темные медведи?
- Есть ли черный кот?

7. Мэри, Сьюзи и Джейн работают в дневную смену. Сэм, Джейн, Боб и Патриция работают в вечернюю смену. Знают друг друга те, кто работает в одну смену. **Запросы:**

Компания	Из	В	Вид транспорта
Амтрак	Нью-Йорк	Бостон	Ж/д
Транзит	Нью-Йорк	Принстон	Ж/д
Амтрак	Бостон	Портленд	Ж/д
Грейхаунд	Бостон	Портленд	Автобус
Амтрак	Нью-Йорк	Вашингтон	Ж/д
Пиплз	Нью-Йорк	Вашингтон	Самолет
Пиплз	Бирлингтон	Нью-Йорк	Самолет

Таблица 0.1: Расписание рейсов

- Знают ли друг друга Мэри и Джейн?
- Кто работает в дневную смену?
- Есть ли кто-то, кто работает в обе смены?
- Есть ли кто-то, кто не знает друг друга?

8. Можно совершить путешествия, перечисленные в табл. 0.1.

Любые две транспортные компании являются конкурентами, если они обслуживают один и тот же маршрут. Можно путешествовать из одного города в другой, если возможно путешествие из одного города в другой через промежуточный (третий) город. **Запросы:**

- Являются ли Амтрак и Пиплз конкурентами?
- Какие компании дают возможность путешествовать из Нью-Йорка в Вашингтон?
- Можно ли путешествовать из Бирлингтона в Портленд?
- Определить всех конкурентов.

9. Определить факты о принадлежности студента определенной студенческой группе. Считается, что два студента знают друг друга, если они учатся в одной группе. **Запросы:**

- Кого знает определенный студент?

- Определить состав определенной группы.
- В каких группах учатся люди с определенным именем?
- Знает ли один студент другого?

10. Имеются факты о маршрутах движения автобусов между двумя разными городами, в которых указаны: номер маршрута, названия двух городов, день и время отправления и прибытия. Известны также фамилии водителей, работающих на определенных маршрутах. Можно попасть из одного города в другой, если существуют автобусные маршруты из первого города во второй или из первого города в промежуточный, и из промежуточного во второй (причем подходят и дни, и часы отправления). **Запросы:**

- Можно ли проехать из одного города в другой?
- Указать автобусы, выходящие из определенного города в определенный день, и время отправления.
- Перечислить фамилии водителей определенного маршрута.
- Указать дни и часы отправления определенного маршрута.

11. Есть факты об отцах некоторых людей и о братьях некоторых людей. Определить отношение ДЯДЯ. **Запросы:**

- Определить братьев конкретного человека.
- Кто является отцом конкретного лица?
- Связаны ли два человека отношением ОТЕЦ.
- Определить, является ли один человек дядей другого.

12. Определить отношения РОДИТЕЛЬ, ЖЕНЩИНА как набор фактов, правило РАЗЛИЧНЫ, СЕСТРА (определяемое через РОДИТЕЛЬ, ЖЕНЩИНА и РАЗЛИЧНЫ) и ТЕТЯ (определяемое через РОДИТЕЛЬ и СЕСТРА). **Запросы:**

- Кто является родителями определенного человека?
- Определить всех детей определенных родителей.
- Определить, есть ли сестры у определенного человека.
- Определить, есть ли тетя у определенного человека.

Методические указания к выполнению лабораторной работы

Процесс построения некоторого формального представления высказываний естественного языка называется *формализацией*. Что это такое? Ответ на этот вопрос столь сложен, сколь сложен ответ на вопрос: Что такое модель? В научных кругах под формализацией понимается словосочетание “дружеский шарж”, т.е. формальное представление некоторого естественного объекта (например, высказывания) — это дружеский шарж.

Продemonстрируем на примерах, почему формализация — это именно **шарж**. Пусть дано высказывание: “Лена любит кататься на велосипеде и на горных лыжах”. Какая логическая связка будет соответствовать союзу “и”?... На самом деле это будет связка “V”, потому, что с формально-логической точки зрения высказывание обозначает: “Лена катается на велосипеде **или** горных лыжах”. Второй пример: “Я пойду домой, а моя жена на работу”. Здесь союз “а” по смыслу соответствует логической связке “&”. Таким образом, формализация естественного текста не может быть сделана “в лоб”, необходимо понять, что было сказано.

При выполнении лабораторной работы следует придерживаться следующих общих правил:

1. Прочитать весь текст высказывания и определить, что будет **объектами**, а что **свойствами**, связывающими эти объекты. Например, пусть даны следующие высказывания: “Аня любит Колю. Коля любит Лену. А Лена смотрит в светлое будущее.” Тогда, объектами будут: Аня, Коля, Лена и “светлое будущее”, а свойствами — отношения “любит” и “смотреть в”, которые связывают два объекта (“Кто” “любит” “Что”², “Кто” “смотрит

²См. замечательный интенсивный курс перевода с английского языка Милошевича.

в” “Что”).

2. Свойства объектов могут быть заданы перечислением, либо через другие известные свойства. В нашем примере свойство “любит” задается перечислением:

```
be_in_love(ann, niko).  
be_in_love(niko, helen).
```

Но высказывание, вроде любовного треугольника, можно задать через `be_in_love/2`:

```
love_triangle(X, Y, Z) :-  
    % любовный треугольник  
    be_in_love(X, Y),      % первого рода, когда  
    be_in_love(Z, Y).      % двое любят одного.  
love_triangle(X, Y, Z) :-  
    % любовный треугольник  
    be_in_love(X, Y),      % второго рода – без-  
    be_in_love(Y, Z).      % ответная любовь.
```

Признаком хорошей формализации (дружественности шаржа) является, как и везде в программировании, хорошая гибкость и интерпретируемость программы: более сложные отношения формулируются через более простые; свойства в достаточной мере абстрактны.

Вопросы для самопроверки

1. Какие структурные единицы формируют программу на языке Пролог?
2. Перечислите простые структуры данных Пролога.
3. Что такое “терм”, в чем отличие переменной от символа?

4. Приведите пример унификации двух структур, представляющих логические выражения.
5. Какова будет унифицирующая подстановка Θ двух следующих термов: $X = \text{fib}(Y+1)$ и $Y = \text{fib}(C+5+D)$ ³.

³ $Y = \text{fib}((C+5)+D)$.

Лабораторная работа 2. Списки и структуры данных

Задание. В работе требуется реализовать **по крайней мере два отношения** из индивидуального задания в виде правил и фактов на языке Пролог. К программе требуется выполнить ряд запросов, объяснить выдаваемые системой результаты: дать процедурную и декларативную интерпретацию определенных отношений.

Цель работы. Приобрести навыки рекурсивной обработки рекурсивных структур данных; научиться интерпретировать (переводить на естественный язык) Пролог-программы.

Время работы. На выполнение работы отводится два академических часа.

Индивидуальные задания.

1. Сформировать новый список из всех четных элементов списка.
2. Определить, является ли один список подписанием другого.
3. Удалить все вхождения заданного элемента из списка.
4. Написать программу пословного (подстрочного) перевода предложения, представленного в виде списка слов, с английского на французский (или любой другой) язык.
5. Сформировать новый список, в котором каждый элемент исходного списка входит в новый список два раза подряд.
6. Слить два упорядоченных списка, сохранив упорядоченность.
7. Определить, являются ли два заданных элемента соседними в списке.

8. Определить последний элемент списка.
9. Подсчитать количество элементов списка.
10. Упорядочить список методом пузырька.
11. Упорядочить список методом вставки.
12. Упорядочить список методом быстрой сортировки.
13. Заменить элемент списка на заданное значение.
14. Определить все перестановки элементов списка.
15. Найти сумму элементов списка, стоящих на нечетных местах в списке.
16. Инвертировать список (составить элементы в обратном порядке).
17. Добавить элемент в конец списка.
18. Удалить два последних элемента списка.
19. Найти максимальный элемент списка.
20. В заданном списке выделить подсписок, содержащий N элементов с начала списка.
21. В заданном списке выделить подсписок, начиная с N -го элемента списка и кончая K -ым элементом этого же списка.
22. Определить сумму отрицательных элементов списка, стоящих на четных местах.
23. Удалить из списка максимальный элемент.
24. Выполнить циклический сдвиг списка на заданное число элементов. Например, результат сдвига списка $[a, b, c, d]$ на два элемента есть список $[c, d, a, b]$.

25. Определить предикат `code(X, List)`, где X — целое неотрицательное число ($0 \leq X \leq 9$), а `List` — это последовательность единичек, число которых равно X . Например, `code(3, L)` дает `L=[1, 1, 1]`, а `code(0, L)` дает `L=[]`.
26. Определить двухаргументный предикат `translate(C1, C2)` для перевода списка цифр в список соответствующих слов. Например, истинным будет следующее высказывание `translate([3, 4, 8], ['три', 'четыре', 'восемь'])`.

Методические указания к выполнению лабораторной работы

Сначала необходимо понять между какими объектами задается отношение, его аргументность (все то же самое, как и в предыдущей лабораторной работе). Далее:

1. Сначала определить частный случай (самый простой) отношения (в терминах доказательства теорем методом математической индукции $P(0, \dots)$ — база индукции).
2. Затем в предположении, что условие базы индукции ложно, определить для всех остальных случаев $P(n, \dots) \rightarrow P(n+1, \dots)$.

Пример 2.1 *Задача — Определить последний элемент списка.*

Отношение задается между списками и элементами списка. Аргументность отношения — 2.

Частный случай — одноэлементный список (пустой список не имеет последнего элемента). “Последний элемент одноэлементного списка и есть этот единственный элемент”.

`last([X], X).`

Пусть список содержит не меньше одного элемента (отрицание предыдущего утверждения), тогда последний элемент списка — это последний элемент его хвоста.

```
last([_ | T], X) :- last(T, X).
```

или более “строго”

```
last([_ | T], X) :-  
    T = [_|_],  
    last(T, X).
```

Декларативная интерпретация: “Последний элемент одно-элементного списка и есть требуемый элемент (решение). Если список содержит более одного элемента, то последний элемент этого списка — последний элемент хвоста”.

Процедурная интерпретация: “Чтобы найти последний элемент списка нужно: Если список содержит ишь один элемент, то «возвратить» в качестве второго элемента отношения (результата) `last/2` первый элемент списка. Иначе, если, в списке содержится более одного элемента, то (а) найти последний элемент хвоста и (б) «возвратить» его как результат”.

Вопросы для самопроверки

1. Что является причиной неэффективности системы программирования Пролог?
2. Семантика предиката отсечение?
3. Каково будет истинностное значение запроса `| ?- var(X).?`
4. Какова процедурная интерпретация правила `A :- Q, W, E, !, R, T, Y.?`
5. Возымеет ли действие отсечение в следующем запросе `fail, !.?`

Лабораторная работа 3. Базы данных

Задание. Необходимо разработать **интерактивную** программу ведения базы данных. База данных должна содержать **по крайней мере одно** отношение “один–ко–многим” ($1:N$) или “многие–ко–многим” ($N:M$). Кроме того, необходимо реализовать выполнение **запроса**, демонстрирующего эту связь, а также корректно реализовать функцию удаления записей.

Цель работы. Ознакомиться с вариантом реализации принципа реляционных баз данных в виде Пролог-программы. Научиться пользоваться предикатами с побочными действиями, а также управлять процессом поиска решения Пролог-системой.

Время работы. На выполнение работы отводится четыре академических часа.

Методические указания к выполнению лабораторной работы. Для непривычного к логическому программированию ума программиста довольно трудно реализовать меню на Прологе, однако, это сделать легче, чем в каком-либо другом языке программирования. Внимательно разберите следующий пример:

```
menu_do(1) :-
    write('Приступим к добавлению записи...'),
    .....
menu_do(0).

main :-
    repeat,
    write('Меню программы:'), nl,
    write('1 - добавление ...'), nl,
    .....
    write('0 - выход из программы'), nl,
    write(' > '), read_int(I),
```

```
menu_do(I),  
I = 0, !.
```

Индивидуальные задания. Разработайте на Прологе программу управления базой данных в следующих предметных областях¹:

1. Институт (деканаты, кафедры, учебный отдел).

- Студенты: паспортные данные, адрес, дата зачисления, номер приказа, факультет, группа, является ли старостой, кафедра (специализация), изучаемые (изученные) предметы, оценки, задолженности, стипендия.
- Учебные курсы: название, факультет(ы), групп(ы), кафедра, семестр(ы), форма отчётности, число часов.
- Преподаватели: паспортные данные, адрес, телефон, фотография, кафедра, должность, учёная степень, начальник (зав. кафедрой), предмет(ы), число ставок, зарплата.

2. Библиотека института.

- Книги: авторы, название, раздел УДК, раздел (техническая, общественно-политическая и т.п.), место и год издания, издательство, количество страниц, иллюстрированность, цена, дата покупки, номер сопроводительного документа (чек, счёт/накладная), вид издания (книги, учебники, брошюры, периодические издания), инвентарный номер (есть только для книг и некоторых учебников), длительность использования читателями (год, две недели, день), электронная версия книги или ее реферата (отсканированный текст).
- Читатели: номер читательского билета, ФИО, год рождения, адрес, дата записи, вид (студент, аспирант, преподаватель, сотрудник), курс, номер группы, названия взятых книг и даты их выдачи.

¹Взято с <http://crec.mipt.ru/study/materials/db/DBvariants.doc>

3. Отдел кадров и бухгалтерия некоторой компании.

- Сотрудники: ФИО, паспортные данные, фотография, дом. и моб. телефоны, отдел, комната, раб. телефоны (в т.ч. местный), подчинённые сотрудники, должность, тип(ы) работы, задание(я), проект(ы), размер зарплаты, форма зарплаты (почасовая, фиксированная).
- Отделы: название, комната, телефон(ы), начальник, размер финансирования, число сотрудников.
- Проекты: название, дата начала, дата окончания, размер финансирования, тип финансирования (периодический, разовый), задачи и их исполнители, структура затрат и статьи расходов.

4. Отдел поставок некоторого предприятия:

- Поставщики: название компании, ФИО контактного лица, расчётный счёт в банке, телефон, факс, поставляемое оборудование (материалы), даты поставок (по договорам и реальные), метод и стоимость доставки.
- Сырьё: тип, марка, минимальный запас на складе, время задержки, цена, продукты, при производстве которых используется, потребляемые объёмы (необходимый, реальный, на единицу продукции).

5. Технологический отдел некоторого предприятия:

- Производственные процессы: продукты, объёмы их производства, необходимые материалы, количества разных видов материалов на единицу продукции, отходы производства; используемое оборудование и его тип, даты ввода оборудования в строй, сроки амортизации, производительность оборудования; человеческие ресурсы (сколько всего и сколько по производству единицы продукции — сколько необходимо и сколько реально).

- Материалы: тип (категория), марка, является ли сырьем (или производится на предприятии), потребляемые объемы (в т.ч. на единицу конечной продукции), в рамках каких технологических процессов используется, цена.

6. Отдел продаж некоторой фирмы.

- Клиенты: название компании, ФИО контактного лица, адрес выставления счёта, адрес доставки, телефон, факс.
- Заказы: тип заказа (покупка, гарантийный ремонт, негарантийный ремонт), общая стоимость, скидка, товар(ы), их изготовители, модели (марки), серийные номера, описание неисправностей, необходимые ресурсы, клиент, дата получения заказа, срок завершения, дата выставления счёта и его оплаты, метод оплаты, дата поставки, метод и стоимость доставки.
- Ресурсы: ФИО, отдел(ы) и телефон(ы) исполнителя(ей), число рабочих часов для выполнения заказа, ставка заработной платы, ответственный за выполнение заказа, необходимое оборудование и расходные материалы, их количество и стоимость, а также наличие материалов на складе.

7. Магазин (внутренний учет).

- Клиенты: юридическое или физическое лицо, ФИО, адрес, телефон, адрес выставления счёта, вид и номер карточки, факс.
- Продажи: наименования, модели (марки) и серийные номера товаров, поставка из магазина или со склада, количество и общая стоимость товаров, размер скидки, тип скидки, форма оплаты (на-личными, оплата счёта, по карточке), необходимость доставки, стоимость и тип доставки, адрес доставки.
- Товары: категория, модель, название производителя, адрес производителя, цена, количество в магазине и на складе.

8. Электронный магазин (информация для клиентов).

- Товары: категория, модель, производитель, цены (в т.ч. средняя и минимальная), есть ли в наличии, описание, характеристики, внешний вид; магазины, где можно купить товар, их телефоны и адреса; аксессуары, их цены и где их купить.
- Магазины: название, компания-владелец, её юридический адрес и home-site, контактные телефоны, адрес, схема проезда, эмблема; товары и цены на них; рекламная информация: некоторые товары с фотографиями, описаниями и ценами, основные отделы (категории товаров).

9. Пункт проката видеозаписей (внутренний учет).

- Видеокассеты: идентификационный номер видеокассеты, тип видеокассет, дата его создания, компания-поставщик, число штук данного типа (общее, в магазине, выдано в настоящее время, выдано всего, выдано в среднем за месяц), общая длительность записей; записи видеокассет: название, длительность, категория, год выпуска и производитель (оригинала).
- Клиенты: ФИО, паспортные данные, адрес, телефон; заказы, т.е. взятые видеокассеты (сейчас и в прошлом): номер, дата выдачи, дата возвращения, общая стоимость заказа.

10. Пункт проката видеозаписей (информация для клиентов).

- Видеокассеты: краткое описание, внешний вид (этикетка), марка (пустой) видеокассеты, цена за единицу прокатного времени (например: 1 день, 3 дня, неделя), есть ли в наличии, общая длительность записей; записи на видеокассете: название, длительность, жанр (категория), тема, год и страна выпуска (оригинала), кинокомпания, описание, актеры, режиссер.

- Заказы: идентификационные номера и названия выданных видеокассет, дата выдачи, дата воз-вращения (продления), общая стоимость заказа, возвращены ли кассеты заказа.

11. Кинотеатры (информация для зрителей).

- Фильмы: название, описание, жанр (категория), длительность, популярность (рейтинг, число проданных билетов в России и в мире), показывается ли сейчас (сегодня, на текущей неделе), в каких кинотеатрах показывается, цены на билеты (в т.ч. средние).
- Кинотеатры: название, адрес, схема проезда, описание, число мест (в разных залах, если их несколько), акустическая система, широкоэкранность, фильмы и цены на них: детские и взрослые билеты в зависимости от сеанса (дневной, вечерний и т.п.) и от категории мест (передние, задние и т.п.); сеансы показа фильмов (дата и время начала).

12. Ресторан (информация для посетителей).

- Меню: дневное или вечернее, список блюд по категориям.
- Блюда: цена, название, вид кухни, категории (первое, второе и т.п.; мясное, рыбное, салат и т.п.), является ли вегетарианским, компоненты блюда, время приготовления, есть ли в наличии.
- Компоненты блюд: тип (гарнир, соус, мясо и т.п.), калорийность, цена, рецепт, время приготовления, есть ли в наличии, ингредиенты (продукты) и их расходы на порцию.

13. Аналитический отдел некоторой компании (поиск и анализ публикаций).

- Категории: название, тип (область исследований, область приложений и т.п.), родительская категория, дочерние ка-

тегории, связанные по смыслу категории (с пояснениями о связях), найденные публикации.

- Публикации: название, тип (газетная, книжная, web и т.п.), название, тип, адрес и телефон источника (газета, книга, сайт и т.п.), выходные данные (date-line), язык, реферат, ключевые слова, категории (с указанием степени уверенности отнесения к ним), текст и его тип (обычный, DOC, HTML, отсканированные картинки и т.п.), обзор.
- Задачи: тип задачи (классификация или поиск), сотрудник (создавший категорию или нашедший публикацию, ответственный за категорию или публикацию и т.п.), завершена ли работа над задачей.

14. Аналитический отдел некоторой компании (анализ рынка технологий, например, по публикациям, см. п.13).

- Организация: название, тип (промышленная, финансовая, торговая, исследовательская и т.п.), категория(и), организация-владелец (акционеры), страна, контактная информация; договорные отношения с другими организациями.
- Технология (продукт): название, категория(и), организация-разработчик и производитель(и), использующие организации.
- Человек: фамилия, имя, тип (начальник, менеджер, создатель технологии и т.п.), организация(и), в которой работает, контактная информация.

15. Компания по (разработке и) сопровождению программного обеспечения.

- Ошибка (bug): краткое и полное описание, срок поступления информации об ошибке, её источник (пользователь,

тестировщик) и его координаты, уровень ошибки (критическая, важная, незначительная и т.п.), категория функциональности (интерфейс, данные, расчетный алгоритм, другое, неизвестная категория), часть проекта, модуль (пакет), программист, ответственный за модуль, программист, ответственный за исправление ошибки, срок исправления (необходимый и реальный), исправлена ли, проверено ли исправление тестировщиком.

Вопросы для самопроверки

1. В чем основное отличие предикатов `is` и `=`?
2. Какая директива объявляет предикат динамическим?
3. В каком случае результат выполнения `retract(_)` будет успешным.
4. Какова декларативная интерпретация запроса `retract(_), fail.?`

Лабораторная работа 4. Перебор и повышение его эффективности

Задание. В работе требуется реализовать и, по возможности, усовершенствовать переборный алгоритм для **одной из задач** из индивидуального задания. В процессе усовершенствования программы анализируйте трудоемкость очередного этапа и полученное ускорение алгоритма. Отметим, что к категории переборных и оптимизационных относятся большинство олимпиадных задач, несколько из них приведены в качестве варианта задания.

Цель работы. Приобрести навыки поиска решений задач с удовлетворением ограничений при помощи полного и частичного перебора; приобрести навыки анализа алгоритма и сокращения пространства поиска.

Время работы. На выполнение работы отводится четыре академических часа.

Индивидуальные задания.

1. Реализовать программу поиска счастливых билетов из материалов лекций и провести дальнейшее усовершенствование переборного алгоритма.
2. Решение диофантова уравнения $4x + 5y = 0$ для значений переменных x и y из некоторого диапазона, значения переменных — целые числа.
3. Сгенерировать списки-палиндромы, состоящие из чисел из заданного диапазона.
4. Реализовать процедуру упорядочения списка методом перебора перестановок.

5. Решить задачу о выполнимости функции $x_1x_2\bar{x}_3 \vee \bar{x}_1x_2x_3$, подсчитать количество выполняющих наборов.
6. Решить задачу о восьми ферзях полным перебором.
7. Решить задачу о раскраске планарного графа (карты). Провести эксперименты со временем решения задачи в зависимости от количества доступных цветов для раскраски.
8. Известно, что пароль состоит из трех букв и цифр, в системе хранится хэш-номер как сумма ASCII-кодов пароля. Сгенерировать возможные пароли. Можно усложнить задачу: в системе хранится MD5-хэш паролей.
9. Задан набор из N слов, из которых требуется составить связный кроссворд. Слова в кроссворде должны располагаться либо вертикально, либо горизонтально, причем каждое слово, записанное по вертикали, должно пересекаться с каждым словом, записанным по горизонтали. Слова, записанные в одном направлении, отделяются друг от друга как минимум одним пустым рядом. Каждое слово в кроссворде должно встречаться в точности столько раз, сколько раз оно присутствует в наборе.
10. Заданы N различных точек плоскости и натуральное число M . Требуется найти максимальный по площади невырожденный M -угольник без самопересечений и самокасаний, вершинами которого являются некоторые из этих N точек.
11. Троллейбусы одного маршрута проходят через остановку каждые k ($1 \leq k \leq 500$) минут. Известны времена прихода пассажиров на эту остановку. Если пассажир приходит на остановку в момент прихода троллейбуса, то он успевает уехать на нем.

Напишите программу, которая бы определяла, во сколько должен пройти первый троллейбус (это время от 0 до $k - 1$),

чтобы: 1) Суммарное время ожидания троллейбуса для всех пассажиров было минимально. 2) Максимальное из времен ожидания троллейбуса было минимально.

12. Расшифровать ребус, полученный в результате замены одинаковых букв одинаковыми цифрами: БЛОК \times 7=СТЕНА.
13. Найти гамильтонов путь в графе.
14. Найти эйлеров путь в графе.
15. Расставить минимальное число белых коней, чтобы пробивались все свободные позиции.
16. Расставить минимальное число белых ладей, чтобы пробивались все свободные позиции.
17. Расставить минимальное число белых ферзей, чтобы пробивались все свободные позиции.
18. Расставить минимальное число белых слонов, чтобы пробивались все свободные позиции.
19. Расставить максимальное число белых коней, чтобы они не били друг друга.
20. Расставить максимальное число белых ладей, чтобы они не били друг друга.
21. Расставить максимальное число белых ферзей, чтобы они не били друг друга.
22. Расставить максимальное число белых слонов, чтобы они не били друг друга.
23. Найти все кратчайшие маршруты коня между двумя заданными позициями.

24. Найти все кратчайшие маршруты ладьи между двумя заданными позициями.
25. Найти все кратчайшие маршруты ферзя между двумя заданными позициями.
26. Составить из костяшек набора домино все магические квадраты размера 4×4 . Костяшки можно класть только горизонтально, костяшка занимает 2 позиции.
27. Составить из костяшек набора домино все возможные замкнутые цепочки прямоугольной формы.
28. Расставить на клеточном поле всеми возможными способами фишки таким образом, чтобы в каждой линии (горизонтальной, вертикальной, диагональной) располагалось четное число фишек.
29. Имеется n деталей и m станков. Каждая деталь характеризуется временем обработки. Станок обрабатывает любую деталь сразу, все станки одинаковы. Определить порядок обработки деталей на станках, когда все детали будут обработаны за минимальное время.
30. Разрезать прямоугольник размера $X \times Y$ на детали прямоугольной формы размера $X_1 \times Y_1$ и $X_2 \times Y_2$, чтобы отходы были минимальны.
31. Упаковать 7 деталей размера $X_i \times Y_i$ в прямоугольник минимальной площади. Построить все минимальные остовные деревья в графе.

Методическое указание к работе. Рассмотрим пример задачи:

Пример 4.1 *Разработать программу поиска списка счастливых билетов, состоящих из шести цифр. Подсчитать их количество.*

Рассмотрим формальную постановку задачи как задачи с удовлетворением ограничений. Вектор переменных \vec{V} — это набор переменных $[A, B, C, D, E, F]$, где каждая переменная представляет число. Области значений всех переменных в первоначальной постановке — числа из диапазона $[0, 1, 2, \dots, 9]$, поэтому достаточно было запрограммировать всего один генератор `gen/1`.

```
num(X) :- member(X, [0,1,2,3,4,5,6,7,8,9]).
gen([]).
gen([X|T]) :- num(X), gen(T).
```

```
p([A,B,C, D,E,F]) :-
    A + B + C == D + E + F.
```

```
lucky([A,B,C, D,E,F]) :-
    gen([A,B,C, D,E,F]),
    p([A,B,C, D,E,F]).
```

Программа при помощи предиката `gen/1` порождает идентификаторы билетов. Предикат `p/1` проверяет является ли билет счастливым. Процедура порождения списка счастливых билетов оформлена в виде предиката `lucky/1` и в комментариях не нуждается. Для запуска порождения списка надо выполнить команду:

```
| ?- lucky(L).
```

```
L = [0,0,0,0,0,0] ? ;
L = [0,0,1,0,0,1] ? ;
L = [0,0,1,0,1,0] ? ;
L = [0,0,1,1,0,0] ? ;
L = [0,0,2,0,0,2] ?
```

```
yes.
```

Для подсчета количества счастливых билетов создадим еще одно вспомогательное правило:

```
count(N) :- findall(Ticket, lucky(Ticket), Tickets),
            length(Tickets, N).
```

Данное правило позволяет подсчитывать количество счастливых билетов, но не выводить их полный список на экран.

Выполним запрос:

```
| ?- count(N).
```

```
N = 55252
```

```
(630 ms) yes.
```

Приведенные программы являются также примерами использования стандартных предикатов обработки списков `member/2` и `length/2`.

Ускорение решения задачи. Программа перебирает 10^6 вариантов, из которых, как мы только что увидели, только около $5.5 \cdot 10^4$ относятся к решению задачи. То есть примерно один из двадцати билетов — счастливый. Возникает вопрос: Можно ли усовершенствовать программу, чтобы уменьшить количество неправильных вариантов¹ и сэкономить время решения задачи на проверке этих неправильных вариантов?

Первым делом, давайте попробуем вычислить значение переменной `F is A+B+C-D-E`. Добавим к нашей программе следующий код:

```
lucky2([A,B,C, D,E,F]) :-
    gen([A,B,C, D,E]),
    F is A+B+C-D-E,
    num(F),
    p([A,B,C, D,E,F]).
```

¹Часто требуется как можно быстрее найти первое решение или самое короткое решение. В этом случае можно рассматривать и сокращение перебора и в области решений.

```

count2(N) :- findall(Ticket, lucky2(Ticket), Tickets),
              length(Tickets, N).

| ?- count2(N).

N = 55252

(133 ms) yes.

```

Получено такое же количество решений но за время, в пять раз меньшее. Вычисленное значение F может быть отрицательным и большим 9, что противоречит условиям задачи, поэтому в новую процедуру порождения билетов необходимо добавить дополнительную проверку `num(F)`, которая выполняется, если F в требуемом диапазоне. Теперь порождается в 10 раз меньше билетов, даже с учетом тех, где F находится вне диапазона. То есть каждый второй сгенерированный билет — счастливый. Если убрать уже ненужную повторную проверку $p/1$, то скорость исполнения программы увеличиться еще на 30% до 106 микросекунд, т.е. в 6.3 раза быстрее первоначальной.

```

| ?- count2(N).

N = 55252

(103 ms) yes.

```

Дополнительное ускорение. Теперь попробуем рассчитать **два** последних числа! Выражение $A+B+C-D$ изменяется в пределах $-9, -8, \dots, 0, 1, \dots, 26, 27$: от $0+0+0-9$ до $9+9+9-0$. Варианты, кода результат выражения — отрицательный заведомо не подходящие, так же как, если этот результат больше 18, $9+9+9-9$. Можно еще усовершенствовать алгоритм, но оставим это в качестве упражнения. Теперь надо разработать подпрограммы, которые будут для диапазона $0, 1, \dots, 18$. Будем решать просто отдельную

переборную задачу: Задано число $S \in 0, 1, \dots, 18$, найти два слагаемых E и F , дающих в сумме N . Дополним программу следующим кодом:

```
lucky3([A,B,C, D,E,F]) :-  
    gen([A,B,C, D]),  
    S is A+B+C-D,  
    S >= 0, S=<18,  
    gen2(S, E,F).
```

```
count3(N) :- findall(Ticket, lucky3(Ticket), Tickets),  
    length(Tickets, N).
```

```
gen2(0,0,0):-!. % Выделим отдельно наглядные  
gen2(18,9,9):-!. % тривиальные варианты.  
gen2(N,A,B):-N<10, !, igen(N,A), B is N - A.  
gen2(N,A,B):-D is N - 9, Z is 9 - D,  
    igen(Z, A1), A is A1 + D, B is N - A.
```

```
% igen(N, A) для A порождает последовательности  
% 0,1,2,...,N  
igen(N, A) :- N>=1, M is N - 1, igen(M, A).  
igen(N, N).
```

Запускаем запрос:

```
| ?- count3(N).
```

```
N = 55252
```

```
(47 ms) yes.
```

Теперь программе работает в 13.5 раз быстрее первоначальной, и в два раза быстрее предыдущей. т.е. примерно один из трех билетов не является счастливым. Конечно, программу можно совершенствовать дальше: перейти к порождению первых трех цифр, и,

отталкиваясь от полученной суммы трех первых цифр, по аналогии с последним примером порождать соответствующие последовательности. Однако, необходимо заметить, что программа² постепенно становится сложной, а текст все меньше и меньше воспринимаемым. Например, в процессе совершенствования программы возможные диапазоны изменения значений стали зависимыми от текущего состояния назначения переменных A, B, C, D . Многие задачи требуют каждым усовершенствованием увеличения программы в два раза для сокращения перебора ненужных вариантов также в два раза. Поэтому важно вовремя остановиться, либо искать какой-либо новый подход.

Вопросы для самопроверки

1. В чем суть алгоритма “Британского музея”? На сколько он эффективен?
2. Как организуется генератор данных на проверку?
3. Приведите общую схему алгоритма решения диофантова уравнения.
4. Какова общая постановка задач на удовлетворение ограничений?
5. Разработайте генератор чисел от 1 до 100 без использования списка значений.

² Авторы пособия не ставили целью **найти** эффективную и короткую программу для решения этой задачи. Наша задача — продемонстрировать ход мыслей.

Литература

- [1] *Искусственный интеллект*: В 3 кн.: Справочник/ Под ред. Э.В. Попова. — М: Радио и связь, 1990. — 464 с.: ил.
- [2] S. Russell, P. Norvig. *Artificial Intelligence: A Modern Approach* — Prentice Hall, 1995. — ок. 800 с.: ил. (к настоящему времени вышло второе издание)
- [3] Ж.–Л. Лорьер. *Системы искусственного интеллекта*: Пер. с франц. — М.: Мир, 1991. — 568 с.: ил.
- [4] *Математический энциклопедический словарь*. — М.: Изд-во Сов. энциклопедия, 1988.
- [5] С.Н.Васильев, А.К.Жерлов, Е.А.Федосов, Б.Е.Федунов. *Интеллектуальное управление динамическими системами* — М.: Физматлит, 2000. — 352 с.
- [6] И.Братко. *Программирование на языке ПРОЛОГ для искусственного интеллекта*: Пер. с англ. — М.: Мир, 1990. — 560 с., ил.
- [7] *The GNU Prolog web site*. URL:<http://www.gprolog.org/> (дата обращения – 28.11.2012).
- [8] *SWI-Prolog's home*. URL:<http://www.swi-prolog.org/> (дата обращения – 28.11.2012).
- [9] Н.Н. Непейвода. *Прикладная логика*: Учеб. пособие. — 2 изд. — Новосибирск, Изд-во. Новосиб. ун-та, 2000. — 521 с., ил.
- [10] Дж. Малпас. *Реляционный язык Пролог и его применение* — М.: Наука, 1990. — 464 С.
- [11] *DC-грамматика*. <http://ru.wikipedia.org/wiki/DC-грамматика>.

- [12] Р.Андерсон. *Доказательство правильности программ*: Пер. с англ. — М.:Мир, 1982. — 168 стр., ил.
- [13] Н.Н. Непейвода, И.Н. Скопин. *Основания программирования*, — Москва-Ижевск: Институт компьютерных исследований, 2003, 880 с., ил.