

# Материалы подготовки к кандидатскому минимуму по специальности 05.13.17

Антон Ба́хтин

19 октября 2012 г.

*Ценность книги определяется не тем, сколько человек ее прочтет. У величайших книг мало читателей, потому что их чтение требует усилия. Но именно из-за этого усилия и рождается эстетический эффект. Литературный фаст-фуд никогда не подарит тебе ничего подобного.*

Виктор Пелевин

## Содержание

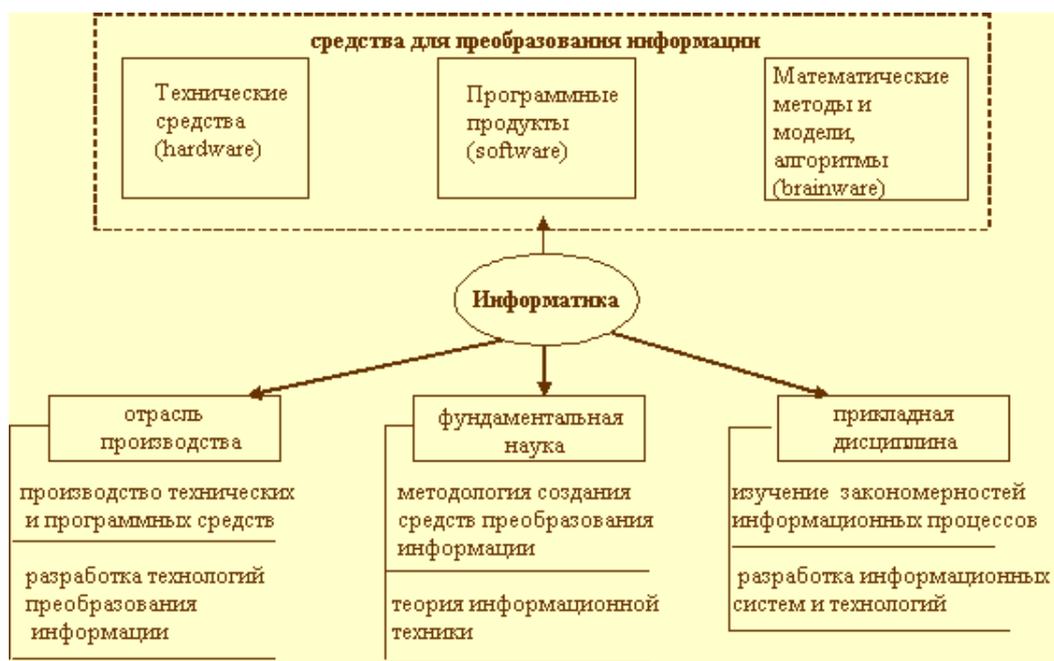
### 1 Билет 1

#### 1.1 Вопрос 1. Информатика как наука, изучающая информацию и её свойства в естественных, искусственных и гибридных системах. Место информации в системе наук.

Некоторые ученые понимают информатику широко — как науку об информации вообще. Эта наука изучает процессы и законы передачи распространения обработки преобразования кодирования запоминания отображения потребления информации и т.д. Некоторые ученые под информатикой имеют в виду комплексную научную и инженерную дисциплину, которая изучает все аспекты разработки, проектирования, создания, оценки функционирования основанных на ЭВМ систем переработки информации, их применения и воздействия на различные области социальной практики.

**Объект и предмет информатики** Поскольку информатика — многоаспектная научная область, объект и предмет ее неоднозначны. Объектом изучения информатики является, прежде всего, информация. Предметом информатики как науки являются информационные технологии в их взаимодействии со средой. Информатика — это теория и практика проектирования, встраивания новых информационных технологий в социальные среды и

их использования. Предметной областью информатики является информационная среда. Информационная среда включает в себя информационные процессы и информационные системы, а также все факторы, воздействующие на них на протяжении всего их существования — начиная от проектирования и до окончания использования. Можно выделить основные направления развития информатики: теоретическая информатика, прикладная и техническая. Основные цели теоретической информатики — развитие общей теории создания, переработки и хранения информации; изучение ее структуры и свойств; разработка теоретических проблем организации систем обработки информации; выяснение закономерностей, в соответствии, с которыми происходит создание семантической информации, ее преобразование, передача и использование в различных сферах деятельности человека; разработка современных методов расчета на ЭВМ; открытие общих законов, лежащих в основе переработки информации; изучение сложных взаимосвязей в системе «человек-ЭВМ», а также взаимного влияния социальных факторов и развития информационных технологий. Прикладная информатика охватывает возможности формализации и математизации областей ее применения, создание баз знаний, моделей информатики, имитации; разработку наиболее рациональных методов осуществления информационных процессов автоматизации исследований, методов и средств вычислительной техники, автоматизации производства; разработку теоретических основ проектирования и организации информационно-поисковых и информационно-логических систем; определение способов наиболее оптимальной организации связи как внутри науки, так и между наукой и производством с широким применением современных вычислительных технических средств, а также изучение закономерностей научно — информационной деятельности. Техническая информатика — отрасль народного хозяйства, основанная на «индустрии» информатики; она включает в себя разработку структуры, принципов конструкции автоматизированных систем обработки информации; создание вычислительной техники новых поколений (в том числе персональных компьютеров), математического обеспечения; эксплуатация средств обработки создание гибких технологических систем, роботов, а также другие проблемы связанные с адаптации человека к новой информационной технологии. К информатике относят и область искусственного интеллекта, содержащую как теоретические, так и технические аспекты. Информатика — область человеческой деятельности, связанная с процессами преобразования информации с помощью компьютеров и других средств вычислительной техники.



**Место информатики в системе наук.** Информатика — слуга других наук, техники и производства. Она снабжает их методами исследования. Но пока — это методы, в основном заимствованные ею из других областей: математической статистики, вычислительной математики, теория игр, графов, исследование операций, математического программирования и другие.

## 1.2 Вопрос 2

## 1.3 Вопрос 3. Понятие конечного автомата, основные определения. Обобщение понятия конечного автомата.

Все знают, что такое конечный автомат.

**Обобщение понятия конечного автомата.** Бесконечные множества состояний, входных и выходных сигналов.

Пример — однородный автомат

- $k$ -мерная бесконечная целочисленная решетка;
- $A, Q, B$  — множества отображений решетки в конечные множества  $A', Q', B'$ ;
- состояние в момент  $t + 1$  и выходной символ ячейки в момент  $t$  определяются состоянием и входными символами ячеек в некоторой окрестности в момент  $t$  — законы одинаковы для всех ячеек
- Однородные структуры — однородные автоматы без выходов

Недетерминированность состояний и выходных сигналов ( $\varphi : Q \times A \rightarrow 2^{Q \times B}$ ) Распознает класс последовательностей — входная последовательность символов распознается,

если среди выходных последовательностей есть такая, в которой последний символ из специального множества

Машина Тьюринга (см. 5.3)

## 2 Билет 2

### 2.1 Вопрос 1. Информационные потребности индивидуальных и коллективных пользователей. Информационные коммуникативные процессы. Социальные аспекты информатизации и компьютеризации общества.

См. 3.1.

### 2.2 Вопрос 2

### 2.3 Вопрос 3

**Определение 1.** Пусть дана система функций  $F$ , замыканием  $F$  называется множество  $[F]$ , состоящее из всех функций, выражаемых формулой над  $F$ .

**Утверждение 1.** Классы  $T_0, T_1, M, S, L$  — замкнуты.

**Лемма 1** (Лемма о нелинейной функции.). Из любой нелинейной функции, подставляя вместо некоторых переменных константы, и может быть отрицание переменных, и может быть навешивая отрицание на результат, можно получить конъюнкцию двух переменных.

*Доказательство.* Рассмотрим минимальный нетривиальный моном в полиноме Жегалкина, убить все переменные вне монома нулями, убить все переменные внутри монома кроме 2-х единицами, подставить отрицания где надо, радоваться.  $\square$

**Лемма 2.** Если  $f \notin L$ , то  $\& \in [\{f, 0, 1, \neg\}]$ .

**Лемма 3** (Лемма о несамодвойственной функции.). Пусть  $f(x_1, \dots, x_n) \notin S$ , тогда, подставляя в неё вместо переменной  $x_i$  переменную  $x_i$  или её отрицание  $\bar{x}_i$ , можно получить константу.

**Лемма 4** (Лемма о немонотонной функции.). Из любой немонотонной функции, подставляя вместо некоторых ее переменных константы, можно получить отрицание.

**Теорема 1** (Теорема о функциональной полноте.). Система  $F$  полная тогда и только тогда, когда  $F \not\subseteq T_0, F \not\subseteq T_1, F \not\subseteq L, F \not\subseteq S, F \not\subseteq M$ , т.е.  $F$  вылезает за пределы пяти, рассмотренных нами замкнутых систем.

*Доказательство.*  $\Rightarrow$  Если  $F$  в одном из этих классов, то и её замыкание в нём, а значит она не полна.

$\Leftarrow$  Т.к.  $F \not\subseteq T_0$ , то  $\exists f_0 \in F : f_0 \notin T_0$ . Аналогично  $\exists f_1 \notin T_1, f_S \notin S, f_M \notin M, f_L \notin L$ , где все  $f$  принадлежат системе (некоторые из них могут и совпадать).

1. получение констант. Возьмём функцию  $f_0 \notin T_0$ , т.е.  $f_0(0, \dots, 0) = 1$ , рассмотрим  $\varphi(x) = f_0(x, \dots, x)$ , тогда  $\varphi(0) = 1$ . Если  $\varphi(1) = 1$ , то это константа 1, а функция  $\psi(x) = f_1(\varphi(x), \dots, \varphi(x))$  - константа 0. Если  $\varphi(1) = 0$ , то это отрицание, и тогда, по лемме о несамодвойственной функции, при помощи отрицания можно получить обе константы, т.е.  $\{0, 1\} \subset \{f_M, \neg\} \subset [F]$ . Мы получили обе константы.
2. получение отрицания. По лемме о немонотонной функции, при помощи констант можно получить отрицание, т.е.  $\{\neg\} \subset \{f_M, 0, 1\} \subset [F]$ .
3. получение конъюнкции. По лемме о нелинейной функции, при помощи констант и отрицания можно получить конъюнкция, т.е.  $\{xy\} \subset \{f_L, \neg\} \subset [F]$ .

Мы получили, что содержит в себе полную систему, следовательно система полная.  $\square$

**Следствие 1.** Из любой полной системы можно выделить полную подсистему, состоящую из пяти функций.

**Теорема 2.** Пусть  $A$  - замкнутый класс и  $A \neq P_2$ , тогда  $A$  содержится в одном из пяти классов  $T_0, T_1, M, S, L$ .

**Лемма 5.** Для любых двух классов (из пяти нами рассмотренных ранее) существует функция принадлежащая одному и не принадлежащая другому.

**Определение 2.** Множество  $F$  называется предполным классом, если выполнены следующие условия:

1.  $F \neq P_2$ ;
2.  $F = [F]$ ;
3.  $f \notin F \Rightarrow \{f, F\} = P_2$ .

**Теорема 3.** В множестве всех функций алгебры логики существует равно пять предполных классов:  $T_0, T_1, M, S, L$ .

*Доказательство.* По предыдущей лемме все эти классы предполные.

Пусть есть некоторый другой предполный класс  $A$ . Тогда он либо вложен в какой-то из 5-ти классов, либо имеет функции не входящие в них. В последнем случае  $A$  - полный. В первом, если, например,  $A \subset T_0$ , то  $\exists f \in T_0 : f \notin A$  и  $\{f, A\} \subset T_0 \neq P_2$ .  $\square$

**Определение 3.** Функция  $f(x_1, x_2, \dots, x_n)$  удовлетворяет условию  $0^\infty$ , если существует  $i : 1 \leq i \leq n$ , такое что:

$$f(x_1, x_2, \dots, x_i, \dots, x_n) \geq x_i$$

**Определение 4.** Функция  $f(x_1, x_2, \dots, x_n)$  удовлетворяет условию  $1^\infty$ , если существует  $i : 1 \leq i \leq n$ , такое что:

$$f(x_1, x_2, \dots, x_i, \dots, x_n) \leq x_i$$

**Определение 5.** Функция  $f(x_1, x_2, \dots, x_n)$  удовлетворяет условию  $0^m, m \geq 2$ , если любые  $m$  наборов, на которых функция равна нулю, имеют общую нулевую компоненту.

**Определение 6.** Функция  $f(x_1, x_2, \dots, x_n)$  удовлетворяет условию  $1^m, m \geq 2$ , если любые  $m$  наборов, на которых функция равна единице, имеют общую единичную компоненту.

Другие замкнутые классы:

- $D$  — класс всех дизъюнкций;
- $K$  — класс всех конъюнкция;
- $O^\infty$  — класс всех функций, удовлетворяющих условию  $0^\infty$ ;
- $I^\infty$  — класс всех функций, удовлетворяющих условию  $1^\infty$ ;
- $O^m$  — класс всех функций, удовлетворяющих условию  $0^m$ ;
- $I^m$  — класс всех функций, удовлетворяющих условию  $1^m$ ;
- $U$  — класс функций, существенно зависящих не более чем от одной переменной;
- $C$  — класс функций, не имеющих существенных переменных.

**Теорема 4.** Множество замкнутых классов булевых функций исчерпывается следующим списком классов.

- Классы, содержащие константы 0 и 1:

$$P_2, L, M, D, K, U, MU, C.$$

- Классы, содержащие 0 и не содержащие 1:

$$L_0, M_0, T_0, D_0, K_0, U_0, C_0, I^m, MI^m, m = 2, 3, \dots, \infty$$

- Классы, не содержащие 0 и не содержащие 1:

$$L_{01}, M_{01}, S_{01}, T_{01}, D_{01}, K_{01}, U_{01}, S, SL, SM, SU, \\ O_0^m, MO_0^m, I_1^m, MI_1^m, m = 2, 3, \dots, \infty$$

*Доказательство.* В монографии Угольниковой. □

## 3 Билет 3

### 3.1 Вопрос 1. Понятие информационного продукта и информационной услуги. Классификация информационных продуктов и услуг. Жизненный цикл информационного продукта.

**Определение 7.** *Информационный продукт — документированная информация, подготовленная в соответствии с потребностями пользователей и представленная в форме товара.*

**Определение 8.** *Информационная услуга — услуга, ориентированные на удовлетворение информационных потребностей пользователей путем предоставления информационных продуктов.*

Рынок информационных продуктов и услуг (информационный рынок) - система экономических, правовых и организационных отношений по торговле продуктами интеллектуального труда на коммерческой основе.

К информационным продуктам относятся документы, данные, справки, аналитические обзоры, базы и банки данных, компьютерные игры, фильмы, книги и другие виды информационных продуктов. Информационными продуктами являются программные продукты, базы и банки данных, информационные системы, информационные технологии, лицензии, патенты, товарные знаки, ноу-хау, инженерно-технические услуги, различного рода информация и прочие виды информационных ресурсов.

Отметим основные особенности информационного продукта, которые кардинально отличают информацию от других товаров.

Во-первых, информация не исчезает при потреблении, а может быть использована многократно. Информационный продукт сохраняет содержащуюся в нем информацию, независимо от того, сколько раз она была использована.

Во-вторых, информационный продукт со временем подвергается своеобразному «моральному износу». Хотя информация и не изнашивается при употреблении, но она может терять свою ценность по мере того, как предоставляемое ею знание перестает быть актуальным.

В различных областях науки и техники темпы обесценения знания неодинаковы, процесс «старения» информации может длиться от пяти до пятнадцати лет.

В-третьих, разным потребителям информационных товаров и услуг удобны разные способы предоставления информации, ведь потребление информационного продукта требует усилий. В этом состоит свойство адресности информации.

В — четвертых, производство информации, в отличие от производства материальных товаров, требует значительных затрат по сравнению с затратами на тиражирование.

**Классификация информационных продуктов и услуг.** Выделяются 5 секторов рынка информационных продуктов и услуг. 1-й сектор — деловая информация, состоит

из следующих частей:

- биржевая и финансовая информация — котировки ценных бумаг, валютные курсы, учетные ставки, рынок товаров и капиталов, инвестиции, цены. Поставщиками являются специальные службы биржевой и финансовой информации, брокерские компании, банки;
- статистическая информация — ряды динамики, прогнозные модели и оценки по экономической, социальной, демографической областям. Поставщиками являются государственные службы, компании, консалтинговые фирмы;
- коммерческая информация по компаниям, фирмам, корпорациям, направлениям работы и их продукции, ценам; о финансовом состоянии, связях, сделках, руководителях, деловых новостях в области экономики и бизнеса. Поставщиками являются специальные информационные службы.

2-й сектор — информация для специалистов, содержит следующие части:

- профессиональная информация — специальные данные и информация для юристов, врачей, фармацевтов, преподавателей, инженеров, геологов, метеорологов и т.д.;
- научно-техническая информация — документальная, библиографическая, реферативная, справочная информация в области естественных, технических, общественных наук, по отраслям производства и сферам человеческой деятельности;
- доступ к первоисточникам — организация доступа к источникам информации через библиотеки и специальные службы, возможности приобретения первоисточников, их получения по межбиблиотечному абонементу в различных формах.

3-й сектор — потребительская информация, состоит из следующих частей:

- новости и литература — информация служб новостей и агентств прессы, электронные журналы, справочники, энциклопедии;
- потребительская информация — расписания транспорта, резервирование билетов и мест в гостиницах, заказ товаров и услуг, банковские операции и т.п.;
- развлекательная информация — игры, телетекст, видеотекст.

4-й сектор — услуги образования, включает все формы и ступени образования: дошкольное, школьное, специальное, среднепрофессиональное, высшее, повышение квалификации и переподготовку. Информационная продукция может быть представлена в компьютерном или некомпьютерном виде: учебники, методические разработки, практикумы, развивающие компьютерные игры, компьютерные обучающие и контролирующие системы, методики обучения и пр.

5-й сектор — обеспечивающие информационные системы и средства, состоит из следующих частей:

- программные продукты — программные комплексы с разной ориентацией — от профессионала до неопытного пользователя компьютера: системное программное обеспечение, программы общей ориентации, прикладное программное обеспечение по реализации функций в конкретной области принадлежности, по решению задач типовыми математическими методами и др.;
- технические средства — компьютеры, телекоммуникационное оборудование, оргтехника, сопутствующие материалы и комплектующие;
- разработка и сопровождение информационных систем и технологий — обследование организации в целях выявления информационных потоков, разработка концептуальных информационных моделей, разработка структуры программного комплекса, создание и сопровождение баз данных;
- консультирование по различным аспектам информационной индустрии — какую приобретать информационную технику, какое программное обеспечение необходимо для реализации профессиональной деятельности, нужна ли информационная система и какая, на базе какой информационной технологии лучше организовать свою деятельность и т.д.;
- подготовка источников информации — создание баз данных по заданной теме, области, явлению и т.п.

**Жизненный цикл информационного продукта.** Жизненный цикл программного обеспечения (ПО) — период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.

Стандарт ГОСТ 34.601-90 предусматривает следующие стадии и этапы создания автоматизированной системы:

1. Формирование требований к АС
  - (a) Обследование объекта и обоснование необходимости создания АС
  - (b) Формирование требований пользователя к АС
  - (c) Оформление отчета о выполнении работ и заявки на разработку АС
2. Разработка концепции АС
  - (a) Изучение объекта
  - (b) Проведение необходимых научно-исследовательских работ
  - (c) Разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователей
  - (d) Оформление отчета о проделанной работе

3. Техническое задание — 1. Разработка и утверждение технического задания на создание АС
4. Эскизный проект
  - (a) Разработка предварительных проектных решений по системе и ее частям
  - (b) Разработка документации на АС и ее части
5. Технический проект
  - (a) Разработка проектных решений по системе и ее частям
  - (b) Разработка документации на АС и ее части
  - (c) Разработка и оформление документации на поставку комплектующих изделий
  - (d) Разработка заданий на проектирование в смежных частях проекта
6. Рабочая документация
  - (a) Разработка рабочей документации на АС и ее части
  - (b) Разработка и адаптация программ
7. Ввод в действие
  - (a) Подготовка объекта автоматизации
  - (b) Подготовка персонала
  - (c) Комплектация АС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями)
  - (d) Строительно-монтажные работы
  - (e) Пусконаладочные работы
  - (f) Проведение предварительных испытаний
  - (g) Проведение опытной эксплуатации
  - (h) Проведение приемочных испытаний
8. Сопровождение АС.
  - (a) Выполнение работ в соответствии с гарантийными обязательствами
  - (b) Послегарантийное обслуживание

## 3.2 Вопрос 2

## 3.3 Вопрос 3. Теорема Кузнецова, Спупецкого-Яблонского и Саломаа в $k$ -значной логике. Континуальность множества замкнутых классов $k$ -значной логики. Классы Янова и Мучник.

**Утверждение 2.** Пусть булевы функции  $f(x_1, \dots, x_n)$  и  $g(x_1, \dots, x_n)$  существенно зависят от всех их переменных. Тогда функция  $h(x_1, \dots, x_{n-1}, y_1, \dots, y_m) = f(x_1, \dots, x_{n-1}, g(y_1, \dots, y_m))$  также существенно зависит от всех переменных.

**Пример 1.** Это не верно в  $P_k, k \geq 3$ , например, для  $\varphi(x_1, x_2) \in P_3$ , т.ч.  $\varphi(x_1, x_2) := j_2(x_1)j_2(x_2)$ .

**Утверждение 3.** Система

$$\{0, 1, \dots, k-1, j_0(x), \dots, j_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\}$$

полна.

**Утверждение 4.** Система  $\{\max(x_1, x_2), x + 1 \pmod{k}\}$  полна.

**Утверждение 5** (Функция Вебба). Система  $\{\max(x_1, x_2) + 1\}$  полна.

**Теорема 5.** Существует алгоритм проверки замкнутости класса

**Определение 9.** Функция  $f(x_1, \dots, x_n)$  сохраняет множество  $A$ , если для любых функций  $h_{i_1}(y_1, \dots, y_p), \dots, h_{i_n}(y_1, \dots, y_p)$  из  $A$ :

$$f(h_{i_1}(y_1, \dots, y_p), \dots, h_{i_n}(y_1, \dots, y_p)) \in A.$$

**Лемма 6.** Класс функций, сохраняющих  $A$ , замкнут

**Лемма 7.** Если класс  $A$  таков, что  $[A]_{y_1 \dots y_p} = A$  и  $\{g_i(y_1, \dots, y_p) = y_i\} \subset A$ , то для класса  $B$ , сохраняющего  $A$ , имеет место равенство:

$$B_{y_1 \dots y_p} = A$$

**Теорема 6** (Кузнецова). Можно построить систему замкнутых классов, ни один из которых не содержится целиком в другом, и такую, что подсистема функций полна в  $P_k \Leftrightarrow$  она не содержится целиком ни в одном из классов.

*Доказательство.* (Яблонский, Введение в дискретную математику, стр. 54) Построение системы предполных классов. Возьмём  $A_1, \dots, A_l$  — все собственные подмножества  $P_k$  функций от двух переменных  $x_1, x_2$ , удовлетворяющие условиям:

1.  $A_i$  содержит  $g_1(x_1, x_2) = x_1$  и  $g_2(x_1, x_2) = x_2$ ;

$$2. [A_i]_{x_1 x_2} = A_i$$

Строим их перебором.

Пусть  $B'_i$  класс сохранения  $A_i$ . В силу лемм,  $B'_i$  - замкнут и  $(B'_i)_{x_1 x_2} = A_i$ . Отсюда следует, что все классы  $B'_i$  различны и не являются полными. Остаётся только удалить классы, вложенные в другие, и получим искомую систему  $B_1, \dots, B_r$

$\Rightarrow$  Очевидно из построения  $B_i$ .

$\Leftarrow$  Пусть  $B \subset P_k$  - система функций, целиком не содержащаяся ни в одном из классов  $B'_i$ . Будем считать, что она замкнута. Пусть  $B' = [X \cup \{g_1, g_2\}]$ . Очевидно, что они полны или неполны одновременно.

Рассмотрим  $A' = B'_{x_1 x_2}$  и докажем, что оно содержит все функции от двух переменных. В самом деле, если это не так, то, очевидно, что

$$A' = A_i \quad \text{и} \quad B' \subset B'_i \subset B_j.$$

Так как  $B \subset B'$ , то  $B \subset B_j$ , что противоречиво. Значит  $A'$  содержит все функции от двух переменных. В частности, оно содержит функцию Вебба.  $\square$

**Определение 10.** Функция  $f(x_1, \dots, x_n) \in P_k(n)$  — существенная функция, если

1. число переменных, от которых она существенно зависит не меньше 2-х;
2.  $f$  принимает  $k$  значений.

**Теорема 7** (Слупецкого-Яблонского). Пусть  $F$  содержит все функции от одной переменной. Тогда  $F$  полна  $\Leftrightarrow F$  содержит существенную функцию. (Усиление Яблонского: полнота системы была равносильна тому, что она содержит хотя бы одну существенную функцию, достаточно, чтобы эта система содержала все функции от одной переменной, принимающие  $\leq k - 1$  значений.)

*Доказательство.*  $\Rightarrow$  (случай Слупецкого)

Предположим, что  $F$  содержит все функции от одной переменной и ни одной существенной. Тогда покажем, что  $F$  неполная. В самом деле, любая формула над  $F$  имеет вид  $f_{i_1}(f_{i_2}(\dots f_{i_l}(g(x_{l_1}, x_{l_2}, \dots)) \dots))$ , где  $f_{i_j} \in P_k(1)$ , а  $g$  — самая внешняя в этой формуле среди всех функций более чем одной переменной. Но  $g$  не является существенной функцией по предположению, значит, она принимает не более  $k - 1$  значения. Но тогда и вся формула не может принимать больше  $k - 1$  значения, следовательно, любая функция над  $F$  принимает не более  $k - 1$  значения. Значит,  $F$  — неполная.

$\Leftarrow$  Докажем для случая Яблонского, а значит и для случая Слупецкого. Пусть  $k \geq 3$ . Нам понадобится

**Лемма 8** (о трёх наборах). Пусть  $f \in P_k(n)$  существенно зависит не менее, чем от 2-х

переменных и принимает не менее 3-х значений, тогда существуют три набора:

$$\begin{aligned} A &= (\alpha, \alpha_2, \dots, \alpha_n), \\ B &= (\beta, \alpha_2, \dots, \alpha_n), \\ C &= (\alpha, \delta_2, \dots, \delta_n), \end{aligned}$$

такие, что  $f(A) \neq f(B) \neq f(C)$ .

*Доказательство.* Пусть  $f$  существенно зависит от  $x_1$ , тогда существуют наборы  $A$  и  $B$ , такие что  $f(A) = \varepsilon_1, f(B) = \varepsilon_2, \varepsilon_1 \neq \varepsilon_2$ . Далее рассмотрим два случая: когда  $f$  принимает на  $M = \{(x, \alpha_2, \dots, \alpha_n) | \forall x\}$  два значения и когда больше 2-х.  $\square$

**Определение 11.** Квадрат — это четыре набора вида

$$\begin{aligned} &(\alpha_1, \dots, \alpha_{i-1}, \alpha, \alpha_{i+1}, \dots, \alpha_{j-1}, \beta, \alpha_{j+1}, \dots, \alpha_n), \\ &(\alpha_1, \dots, \alpha_{i-1}, \gamma, \alpha_{i+1}, \dots, \alpha_{j-1}, \beta, \alpha_{j+1}, \dots, \alpha_n), \\ &(\alpha_1, \dots, \alpha_{i-1}, \alpha, \alpha_{i+1}, \dots, \alpha_{j-1}, \delta, \alpha_{j+1}, \dots, \alpha_n), \\ &(\alpha_1, \dots, \alpha_{i-1}, \gamma, \alpha_{i+1}, \dots, \alpha_{j-1}, \delta, \alpha_{j+1}, \dots, \alpha_n), \end{aligned}$$

где  $\alpha \neq \gamma, \beta \neq \delta$ .

**Лемма 9** (о квадрате). Пусть  $f \in P_k(n)$  существенно зависит не менее, чем от 2-х переменных и принимает не менее 3-х значений, тогда существуют квадрат, на котором одно значение принимается ровно 1 раз.

*Доказательство.* Построим наборы из леммы о трёх наборах:

$$\begin{aligned} A &= (\alpha_1, \alpha_2, \dots, \alpha_n), \\ B &= (\delta_1, \alpha_2, \dots, \alpha_n), \\ C &= (\alpha_1, \delta_2, \dots, \delta_n), \end{aligned}$$

и рассмотрим последовательность квадратов:

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \dots & \alpha_n \\ \delta_1 & \alpha_2 & \alpha_3 & \alpha_4 & \dots & \alpha_n \\ \alpha_1 & \delta_2 & \alpha_3 & \alpha_4 & \dots & \alpha_n \\ \delta_1 & \delta_2 & \alpha_3 & \alpha_4 & \dots & \alpha_n \\ \alpha_1 & \delta_2 & \delta_3 & \alpha_4 & \dots & \alpha_n \\ \delta_1 & \delta_2 & \delta_3 & \alpha_4 & \dots & \alpha_n \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \delta_1 & \delta_2 & \delta_3 & \delta_4 & \dots & \alpha_n \end{pmatrix}$$

Квадраты образуются строками 1-4, 2-6, 4-8 и т.д. Заметим, что значения функции на первых 2-х и последнем наборе различно (т.е. они взяты из леммы о 3-х наборах), значит в какой-то момент будет квадрат, в котором некоторое значение принимается только одним из 4-х наборов.  $\square$

Приступим к доказательству основной теоремы. Сначала построим все функции, принимающие ровно 2 значения, а затем по индукции из всех функций, принимающих не более  $l$  значений, получим все функции, принимающие  $l + 1$  значение. По условию, у нас есть существенная функция  $f$ . Поскольку  $k \geq 3$ , она удовлетворяет условиям леммы о квадрате. Рассмотрим наборы, образующие квадрат, на котором какое-то значение, скажем,  $\varepsilon_1$ , принимается ровно 1 раз, т.е.  $\varepsilon_1 \notin \{\varepsilon_2, \varepsilon_3, \varepsilon_4\}$ :

$$\begin{aligned} f(\alpha \ \beta \ \alpha_3 \dots \alpha_n) &= \varepsilon_1 \\ f(\gamma \ \beta \ \alpha_3 \dots \alpha_n) &= \varepsilon_2 \\ f(\alpha \ \delta \ \alpha_3 \dots \alpha_n) &= \varepsilon_3 \\ f(\gamma \ \delta \ \alpha_3 \dots \alpha_n) &= \varepsilon_4 \end{aligned}$$

Введем обозначение  $\varphi(x_1, x_2) := f(x_1, x_2, \alpha_3, \dots, \alpha_n)$ . Теперь рассмотрим функции:

$$\psi := \begin{cases} 0 & \text{if } x = \varepsilon_1 \\ 1 & \text{иначе} \end{cases} \quad \lambda_1 := \begin{cases} \alpha & \text{if } x = 0 \\ \gamma & \text{иначе} \end{cases} \quad \lambda_2 := \begin{cases} \beta & \text{if } x = 0 \\ \delta & \text{иначе} \end{cases}$$

$$\omega(x_1, x_2) := \psi[\varphi(\lambda_1(x_1), \lambda_2(x_2))]$$

продолжение в лекциях лупанова  $\square$

**Теорема 8** (Саломая). Пусть система  $F$  функций из  $P_k$ , где  $k \geq 5$ , содержит все равнозначные функции одной переменной (группу всех перестановок  $\sigma_k$ ). Тогда для полноты системы  $F$  необходимо и достаточно, чтобы  $F$  содержала существенную функцию  $f(x_1, \dots, x_n)$  принимающую все  $k$  значений.

*Доказательство.* (Яблонский, Дискретная математика и математические вопросы кибернетики, стр. 55)

$\Rightarrow$  Доказывается аналогично теореме Слупецкого-Яблонского.

$\Leftarrow$  Доказательство состоит из следующих пунктов:

1. построение функции  $\varphi(x)$ , зависящей от одной переменной, принимающей  $l$  значений, где  $2 \leq l \leq k - 1$ ;
2. построение функции  $\psi(x)$ , зависящей от одной переменной, принимающей два значения 0 и 1;
3. построение функций  $x_1 \vee_{01} x_2$ ,  $x_1 \wedge_{01} x_2$ , являющихся на множестве  $\{0, 1\} \times \{0, 1\}$  соответственно максимумом и минимумом;

4. построение функций  $j_0(x), \dots, j_{k-1}(x)$ , т.е. функций-индикаторов;
5. построение всех функций из  $P_k$  (индукция по числу значений).

□

**Теорема 9.** Для всякого  $k, k \geq 3$ ,  $P_k$  содержит континуум различных замкнутых классов.

*Доказательство.* Так как функций в  $P_k$  счётно, то подмножеств — континуум, а значит замкнутых классов не более континуума.

С другой стороны, рассмотрим функции  $f_2, f_3, \dots$  из описываемого ниже примера Мучника. Легко видеть, что замыкания произвольных подмножеств этой системы различны, и таких подмножеств — континуум. □

**Теорема 10** (Пример Янова). Для всякого  $k, k \geq 3$ , в  $P_k$  существует замкнутый класс, не имеющий базиса.

*Доказательство.* (Яблонский, Дискретная математика и математические вопросы кибернетики, стр. 60) Рассмотрим последовательность функций:

$$f_0 = 0, \quad f(x_1, \dots, x_i) = \begin{cases} 1 & \text{если } x_1 = \dots = x_i = 2, \\ 0 & \text{иначе} \end{cases}$$

Обозначим за  $A_k$  множество всех функций, получающихся из  $\{f_0, f_1, \dots\}$  путём переименования (без отождествления!) переменных. Очевидно, что  $A_k$  замкнут. Пусть у него есть базис. Рассмотрим функцию базиса, получаемую из  $f_{n_0}$  перестановкой переменных с минимальным  $n_0$ . Возможны два случая:

1. В базисе есть другая функция. Она получается из  $f_{n_1}$ , причём  $n_1 > n_0$ . Но тогда  $f_{n_0}$  может быть получена из  $f_{n_1}$  отождествлением переменных, что противоречит определению базиса.
2. Это единственная функция базиса, но тогда нельзя получить  $f_n$ , с  $n > n_0$ , т.к.  $f_{n_0}(\dots, f_{n_0}(\dots)) := 0$ .

Значит базиса нет. □

**Теорема 11** (Пример Мучника). Для всякого  $k, k \geq 3$ , в  $P_k$  существует замкнутый класс со счётным базисом.

*Доказательство.* (Яблонский, Дискретная математика и математические вопросы кибернетики, стр. 61) Рассмотрим последовательность функций ( $i = 2, 3, \dots; j = 1, 2, \dots, i$ ):

$$f(x_1, \dots, x_i) = \begin{cases} 1 & \text{если } x_1 = \dots = x_{j-1} = x_{j+1} = \dots = x_i = 2, x_j = 1 \\ 0 & \text{иначе} \end{cases}$$

Обозначим за  $A_k$  замыкание системы функций из  $\{f_2, f_3, \dots\}$ . Докажем, что эта система есть базис  $A_k$ . Достаточно доказать, что никакая из функций не может быть выражена через остальные. Пусть можно ( $m > 2$ ):

$$f_m(x_1, \dots, x_m) = f_r(\mathcal{B}_1[f_2, \dots, f_{m-1}, f_{m+1}, \dots], \dots, \mathcal{B}_r(\dots))$$

Возможны три случая:

1. Среди формул  $\mathcal{B}_1, \dots, \mathcal{B}_r$  есть по крайней мере два символа, отличных от переменных, например,  $\mathcal{B}_{i_1}$  и  $\mathcal{B}_{i_2}$ . Тогда значение формулы тождественный ноль, что невозможно.
2. Среди формул  $\mathcal{B}_1, \dots, \mathcal{B}_r$  есть ровно один символ, отличный от переменных, например,  $\mathcal{B}_{i_1}$ . Тогда, т.к.  $r \geq 2$ , то  $\exists q$ , такой что  $\mathcal{B}_p = x_q$ . Возьмём набор  $x_1 = \dots = x_{q-1} = x_{q+1} = \dots = x_i = 2, x_q = 1$ , тогда слева будет единица, а справа — ноль.
3. Среди формул  $\mathcal{B}_1, \dots, \mathcal{B}_r$  все символы — переменные. Тогда  $r > m$  и какая-то переменная, например,  $x_q$  встречается справа более одного раза, тогда на наборе  $x_1 = \dots = x_{q-1} = x_{q+1} = \dots = x_i = 2, x_q = 1$  слева будет единица, а справа — ноль.

□

## 4 Билет 4

### 4.1 Вопрос 1. Экономика информационных сетей. Методы управления производством и распределением информационных продуктов. Методы анализа и оценки качества информационных продуктов и услуг.

**Экономика информационных сетей.** Интернет-технологии — новый образ экономики (сетевая экономика) Тенденции развития сетевой экономики:

- индивидуальный подход к квалифицированному покупателю;
- появление глобальной конкуренции;
- изменение структуры существующих предприятий и компаний.

**Методы управления производством и распределением информационных продуктов.** Стратегический характер информации как ресурса экономического и социального развития обуславливает высокую степень государственного регулирования, значительный уровень концентрации и монополизации информационного производства.

Многие продукты информационной деятельности по своему статусу являются общественными благами (фундаментальные научные исследования, государственное управление, национальные сети коммуникаций и т. д.) Как правило, государство берет на себя регулирование процесса производства и распределения информационных продуктов, без которых общество не может нормально развиваться.

Развитие рыночных отношений в информационной деятельности поставило вопрос о защите информации как объекта интеллектуальной собственности и имущественных прав на нее. В Российской Федерации принят ряд указов, постановлений, законов, таких, как: «Об информации, информатизации и защите информации».

В законе определены цели и основные направления государственной политики в сфере информатизации. Информатизация определяется как важное новое стратегическое направление деятельности государства. Указано, что государство должно заниматься формированием и реализацией единой государственной научно-технической и промышленной политики в сфере информатизации.

Закон создает условия для включения России в международный информационный обмен, предотвращает бесхозяйственное отношение к информационным ресурсам и информатизации, обеспечивает информационную безопасность и права юридических и физических лиц на информацию. В нем определяются комплексное решение проблемы организации информационных ресурсов, правовые положения по их использованию и предлагается рассматривать информационные ресурсы в двух аспектах:

- как материальный продукт, который можно покупать и продавать;
- как интеллектуальный продукт, на который распространяется право интеллектуальной собственности, авторское право.

С технологической и экономической точек зрения производство, обмен, распределение и потребление информации имеют целый ряд специфических особенностей.

Сегодня во всех странах независимо от уровня экономического и социального развития происходит структурная перестройка, связанная с ростом информационного сектора экономики и соответственно влекущая за собой значительные социальные, политические и культурные изменения в обществе.

Ярким проявлением этих тенденций является значительное увеличение числа занятых информационной деятельностью, т.е. деятельностью, связанной с производством, обработкой, хранением и распространением информации.

В свою очередь рынок информационных товаров и услуг является сегодня самым динамично развивающимся.

Классическим примером положительных информационных экстерналией является финансирование отдельными фирмами научных исследований и разработок, результаты которых часто становятся достоянием широкого круга заинтересованных лиц. Патенты или лицензии могут являться средством устранения положительных внешних эффектов информационного производства.

**Методы анализа и оценки качества информационных продуктов и услуг.** Цена на товар (например, книгу) определяется спросом.

Из-за специфики информации как товара и социальной значимости информационных продуктов при оценке эффективности производства и возможностей использования информации все чаще применяются моральные и эстетические критерии наряду с денежными.

## 4.2 Вопрос 2

Основные функции СУБД:

- управление данными во внешней памяти (на дисках);
- управление данными в оперативной памяти с использованием дискового кэша;
- журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
- поддержка языков БД (язык определения данных, язык манипулирования данными).

Обычно современная СУБД содержит следующие компоненты:

- ядро, которое отвечает за управление данными во внешней и оперативной памяти, и журнализацию,
- процессор языка базы данных, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода,
- подсистему поддержки времени исполнения, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД
- а также сервисные программы (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

## 4.3 Вопрос 3. Описание всех предполных классов $k$ -значной логики. Функциональная система двухполюсных сетей. Элементы общей теории функциональных систем.

**Описание всех предполных классов  $k$ -значной логики.**

*(Яблонский. Дискретная математика и . . . . Стр. 160 )*

**Определение 12.** *Сеть — граф с выделенными вершинами — полюсами.*

Например, дерево — сеть с одним полюсом.

**Определение 13.**  $(k, l)$ -полюсник — сеть с  $k$  входными и  $l$  выходными полюсами. Двух-полюсная сеть, это  $(1, 1)$ -полюсник.

Очевидны определения цепи, сильно связной сети (через каждое ребро — цепь).

**Определение 14.** Пусть  $S$  — произвольная частично ориентированная сеть, каждому ребру и приписана его пропускная способность  $c(u) \geq 0$ . **Потоком в сети  $S$**  называется пара  $(f, \omega)$ , где  $\omega$  — некоторая ориентация всех неориентированных рёбер сети, а  $f(u)$  — заданная на множестве всех рёбер функция с неотрицательными значениями, не превосходящими пропускных способностей рёбер, и такая, что в каждой внутренней вершине  $\alpha$  выполняется закон Кирхгофа (сумма входящих = сумме исходящих).

**Определение 15.** Величина потока — значение потока на полюсе.

**Определение 16.** **Сечение сети** — множество рёбер, при удалении которых сеть становится несвязной, причём полюсы оказываются в разных компонентах. При этом появляется левая и правая части. **Сечение простое**, если оно минимальное. **Ребро сечения прямое**, если оно неориентированно или ориентированно слева-направо. **Ребро сечения обратное** в обратном случае. **Пропускная способность простого сечения** — сумма пропускных способностей его прямых рёбер.

**Теорема 12.** Максимальная величина потока через сеть  $R_{max}$  равна минимальной из пропускных способностей ее  $c_{min}$  простых сечений.

*Доказательство.* Полное доказательство — Яблонский. Дискретная математика и ....  
Стр. 164.

План:

- докажем, что  $R_{max} \leq c_{min}$  (для этого достаточно доказать, что  $R_{max} \leq c(W)$ );
- докажем, что можно создать поток  $R_{max} = c_{min}$  индукцией по количеству рёбер;
  - заметим, что в сетях из предположения индукции можно создать поток любой меньшей величины;
  - уменьшим пропускные способности рёбер так, что в изменённой сети пропускная способность каждого простого сечения не меньше  $c_{min}$  и каждое ребро либо прямое в некотором простом сечении, либо имеет нулевую пропускную способность;
  - удалим нулевые рёбра;
  - если все цепи сети состоят из не более чем двух рёбер, то всё доказано;
  - иначе, есть внутреннее ребро; возьмём сеть с ним, все рёбра сети поделим новой вершиной посередине, а потом склеим все вершины в одно ребро — полученные две сети подходят под предположение индукции.

□

**Определение 17.** Будем говорить, что множество элементов сети (ребёр и вершин) **блокирует** все цепи, если оно содержит хотя бы по одному элементу из каждой цепи. **Барьер** сети  $S$  — множество рёбер, блокирующих все пути из  $\alpha_s$  в  $\beta_s$ . Барьер простой, если из него нельзя ничего выкинуть.

**Утверждение 6.** Простой барьер  $W$  можно дополнить до простого сечения так, что все рёбра из  $W$  будут в этом сечении прямыми, а все добавленные рёбра — обратными.

**Теорема 13.** Максимальная суммарная величина  $R_{\max}$  потока через многополюсную сеть  $S$  равна минимальной из пропускных способностей её барьеров, блокирующих все пути от входов к выходам.

## 5 Билет 5

**5.1 Вопрос 1. Основные секторы информационной сферы: информация, электронные коммуникации, тематическая классификация. Сектор деловой информации. Сектор информации для специалистов. Научно-техническая информация. Другие виды профессионально ориентированной информации. Социально значимая (правовая, социальная, политическая, экологическая, образовательная и др.) информация.**

**Информационная сфера** — совокупность субъектов информационного взаимодействия или воздействия; собственно информации, предназначенной для использования субъектами информационной сферы; информационной инфраструктуры, обеспечивающей возможность осуществления обмена информацией между субъектами; общественных отношений, складывающихся в связи с формированием, передачей, распространением и хранением информации, обменом информацией внутри общества.

Право на доступ к информации — фундамент обеспечения и соблюдения прав человека.

**Про сектора — см. 3.1.**

**Социально значимая информация.** Социальная информация — любая информация, циркулирующая в обществе, которая обеспечивает выполнение им функций именно как социальной системы. При этом для общества можно выделить некоторую информацию, имеющую для его членов наибольшее значение. Такая информация называется **социально значимой**.

Социально значимая информация — это информация, включающая в себя помимо всего следующие сведения:

- о состоянии экономической сферы;
- об интересующих значительное количество людей событиях общественной жизни внутри страны и за рубежом;
- о деятельности политических партий и движений, лидеров общества и государства;
- о рынке труда и капитала и т.д.

## 5.2 Вопрос 2. Объектно-ориентированный подход к проектированию и разработке программ: сущность объектно-ориентированного подхода, объектный тип данных, переменные объектного типа, инкапсуляция, наследование, полиморфизм, классы и объекты. Логическое программирование. Компонентное программирование.

**Определение 18.** *Объектно-ориентированное программирование — подход к проектированию и программированию, при котором основными концепциями являются объекты и классы.*

### Абстракция

Абстрагирование — это способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые. Соответственно, абстракция — это набор всех таких характеристик.

### Инкапсуляция

Инкапсуляция — это свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе и скрыть детали реализации от пользователя.

### Наследование

Наследование — это свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью. Класс, от которого производится наследование, называется базовым, родительским или суперклассом. Новый класс — потомком, наследником или производным классом.

### Полиморфизм

Полиморфизм — это свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.

### Класс

Класс является описываемой на языке терминологии (пространства имён) исходного

кода моделью ещё не существующей сущности (объекта). Фактически он описывает устройство объекта, являясь своего рода чертежом. Говорят, что объект — это экземпляр класса. При этом в некоторых исполняющих системах класс также может представляться некоторым объектом при выполнении программы посредством динамической идентификации типа данных. Обычно классы разрабатывают таким образом, чтобы их объекты соответствовали объектам предметной области.

## Объект

Сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса или копирования прототипа (например, после запуска результатов компиляции и связывания исходного кода на выполнение).

**Определение 19.** *Логическое программирование — подход к программированию, основанный на выводе новых фактов из данных фактов согласно заданным логическим правилам. Логическое программирование основано на теории математической логики.*

**Компонентно-ориентированное программирование.** Компонентно-ориентированное программирование включает в себя набор ограничений, налагаемых на механизм объектно-ориентированного программирования.

- Компонент — независимый модуль программного кода, предназначенный для повторного использования и развертывания.
- Может содержать «множественные классы».
- Как правило, независим от конкретного языка.

## 5.3 Вопрос 3. Полиномиальный алгоритмы на графах. Полиномиальный алгоритм линейного программирования. Потoki в сетях и теория паросочетаний. Теорема NP-полноты. Основные NP-полные задачи. Метод ветвей и границ. Градиентные и другие приближения для NP-полных задач.

**Задача о кратчайшем пути.** Задан ориентированный граф  $G = (V, E)$ ,  $w_e \geq 0$ ,  $e \in E$  — вес дуги. Найти кратчайшие расстояния от заданной вершины  $v$  до остальных вершин.

### Алгоритм Дейкстры

1. Положить  $W = v$ ;  $\rho(v) = 0$ ;  $p(v) = \emptyset$ .
2. Для всех  $y \in V \setminus \{v\}$ :
  - (a)  $\rho(y) := w_{vy}$ ;
  - (b)  $p(y) := v$ ;

3. До тех пор пока  $W \neq V$  выполнять

(a) Найти такую вершину  $x \in V \setminus W$ , что  $\rho(x) = \min\{\rho(y) | y \in V \setminus W\}$ ;

(b) Положить  $W := W \cup \{x\}$ ;

(c) Для всех  $y \in V \setminus W$ :

$$z := \rho(y);$$

$$\rho(y) := \min\{\rho(y), \rho(x) + w_{xy}\};$$

если  $\rho(y) < z$ ,  $p(y) := x$ .

**Утверждение 7.** Алгоритм Дейкстры находит кратчайшие пути из вершины  $v$  до каждой из остальных вершин за время  $O(|V|^2)$ .

**Задача.** Задан ориентированный граф  $G = (V, E)$ ,  $w_e \geq 0$ ,  $e \in E$  — вес дуги. Найти кратчайшее расстояние для каждой пары вершин.

**Определение 20.** Для квадратной матрицы  $(d_{ij})$  операцией треугольника относительно  $j$  называется пересчет  $d_{ik} = \min\{d_{ik}, d_{ij} + d_{jk}\}$  по всем  $i, k \neq j$ .

**Утверждение 8** (Алгоритм Флойда — Уоршелла). Пусть  $d_{ij} = w_{v_i v_j}$  при  $i \neq j$  и  $d_{ii} = 0$ . Тогда если выполнить над матрицей  $(d_{ij})$  операцию треугольника последовательно для  $j = 1, 2, \dots, |V|$ , то в полученной матрице каждый элемент  $d_{ik}$  равен длине кратчайшего пути из  $i$  в  $k$ .

*Доказательство.* Идея: заметим, что через  $k$  итерация  $d_{ij}$  равна цене минимального пути с пересадками в вершинах  $1, \dots, k$ . □

**Задача.** Построение остовного дерева минимальной стоимости (ОДМС) для неориентированного графа.

**Утверждение 9.** Пусть  $U \subset V$ ,  $(u, v)$  — ребро наименьшей стоимости, связывающее  $U$  и  $V \setminus U$ , тогда существует ОДМС, содержащее ребро  $(u, v)$

*Доказательство.* Если нет, то добавим это ребро, получится цикл, удалим параллельное ребро, стоимость уменьшится. □

**Алгоритм Прима.** Построение начинается с дерева, включающего в себя одну (произвольную) вершину. В течение работы алгоритма дерево разрастается, пока не охватит все вершины исходного графа. На каждом шаге алгоритма к текущему дереву присоединяется самое лёгкое из рёбер, соединяющих вершину из построенного дерева, и вершину не из дерева. Время  $O(n^2)$ .

**Определение 21.** Задача линейного программирования состоит в определении максимума целевой функции

$$f(x) = c^T x$$

при условиях

$$Ax \leq b$$

## Алгоритм Кармаркара.

**Определение 22.** Симплексом называется  $\{(x_1, \dots, x_n) \in \mathbb{R}^n \mid \forall i x_i \geq 0, \sum_{i=1}^n x_i = 1\}$ . Грань симплекса — множество точек симплекса, т.ч.  $x_i = 0$ .

Идея: найти минимум на симплексе, а потом свести общий случай к данному через фиктивные переменные.

**Про потоки смотреть билет 4.** Стоит заметить, что можно рассматривать задачу о потоках как задачу линейного программирования.

**Определение 23.** Паросочетание — подмножество ребер, таких, что никакие два не инцидентны одной вершине. Полное паросочетание — участвуют все вершины графа.

**Задача.** Надо найти максимальное паросочетание в двудольном графе.

Можно рассматривать это как задачу линейного программирования. А можно решать через явный алгоритм Куна.

**Определение 24.** Чередующейся цепью (в двудольном графе, относительно некоторого паросочетания) назовём цепь, в которой рёбра поочередно принадлежат/не принадлежат паросочетанию.

**Определение 25.** Увеличивающей цепью (в двудольном графе, относительно некоторого паросочетания) назовём чередующуюся цепь, у которой начальная и конечная вершины не принадлежат паросочетанию.

**Теорема 14** (Теорема Бержа). Паросочетание является максимальным тогда и только тогда, когда не существует увеличивающих относительно него цепей.

На этой теореме основан алгоритм Куна поиска максимальных паросочетаний. Для нахождения увеличивающих цепей алгоритм просматривает все вершины графа по очереди, запуская из каждой обход в глубину, пытающийся найти увеличивающую цепь, начинающуюся в этой вершине. Итак, алгоритм Куна можно представить как серию из  $n$  запусков обхода в глубину/ширину на всём графе. Следовательно, всего этот алгоритм выполняется за время  $O(nt)$ , что в худшем случае есть  $O(n^3)$ .

**Определение 26.** Машина Тьюринга задаётся семёркой  $(Q, T, I, \delta, b, q_0, q_f)$ , где

- $Q$  — состояния;
- $T$  — символы на лентах;
- $I$  — входные символы;  $I$  лежит в  $T$ ;
- $b$  — пустой символ; лежит в  $T \setminus I$ ;
- $q_0$  — начальное состояние;
- $q_f$  — заключительное состояние;

- $\delta$  — функция переходов.

**Определение 27.** Машина допускает слово, если оно, записанное на первой ленте, приводит ее в заключительное состояние. Машина распознает язык — допускает все слова из языка, и никакие другие.

**Определение 28.** Недетерминированная машина Тьюринга (НМТ) - есть несколько вариантов поведения. Множество распознаваемых языков эквивалентно множеству, распознаваемому обычной машиной Тьюринга.

**Определение 29.** Временная сложность  $NMT = T(n)$ , если для любого слова длины  $n$  найдется последовательность не более чем из  $T(n)$  шагов, которая приводит в заключительное состояние.

**Определение 30.** Емкостная сложность  $NMT = S(n)$ , если для любого слова длины  $n$  найдется последовательность шагов, приводящая в заключительное состояние, в которой число просмотренных головкой клеток на каждой ленте не превосходит  $S(n)$ .

**Определение 31.**  $P$  - класс всех языков, допускаемых ДМТ с полиномиальной временной сложностью.

**Определение 32.**  $NP$  - класс всех языков, допускаемых НМТ с полиномиальной временной сложностью.

### Основные NP-полные задачи

- Выполнимость булевых формул — существует ли набор переменных, что она равна 1
- Неориентированные графы
  - Существование полного  $k$ -узельного подграфа
  - Существование узельного покрытия размера  $k$  —  $k$  узлов, инцидентных всем ребрам
  - Существование гамильтонова цикла — цикл, содержащий все вершины
  - Является ли  $k$ -раскрашиваемым — можно ли приписать вершинам числа  $1, \dots, k$ , чтобы никаким смежным вершинам не был приписан один и тот же цвет
- Ориентированные графы
  - Существование  $k$ -элементного множества узлов, разрезающих циклы —  $k$  узлов, таких, что каждый цикл содержит такой узел
  - Существование  $k$ -элементного множества ребер, разрезающих циклы —  $k$  ребер, таких, что каждый цикл содержит такое ребро
  - Существование ориентированного гамильтонова цикла

- Множества

- Существование покрывающих  $k$  подмножеств
- Существование попарно непересекающихся покрывающих  $k$  подмножеств

**Определение 33.** *Метод ветвей и границ является вариацией полного перебора с отсе-вом подмножеств допустимых решений, заведомо не содержащих оптимальных реше-ний.*

**Пример 2.** *Поиск минимума функции  $f(x)$  на множестве допустимых значений пере-менной  $x$ . Функция и переменная могут быть произвольной природы. Для метода ветвей и границ необходимы две процедуры: ветвление и нахождение оценок (границ).*

*Процедура ветвления состоит в разбиении множества допустимых значений пере-менной на подобласти (подмножества) меньших размеров. Процедуру можно рекурсивно применять к подобластям. Полученные подобласти образуют дерево, называемое дере-вом поиска или деревом ветвей и границ. Узлами этого дерева являются построенные подобласти (подмножества множества значений переменной  $x$ ).*

*Процедура нахождения оценок заключается в поиске верхних и нижних границ для решения задачи на подобласти допустимых значений переменной  $x$ .*

*В основе метода ветвей и границ лежит следующая идея: если нижняя граница зна-чений функции на подобласти  $A$  дерева поиска больше, чем верхняя граница на какой-либо ранее просмотренной подобласти  $B$ , то  $A$  может быть исключена из дальнейшего рассмотрения (правило отсева).*

**Определение 34.** *Градиентные алгоритмы (локального поиска) начинают с про-извольного решения и пытаются улучшить его следуя по антиградиенту, пока не до-стигнут локально-оптимального решения.*

**Пример 3** (Задача коммивояжера). *. Возьмём произвольный путь в качестве начального решения. Далее на каждом шаге будем выбирать любую пару несмежных ребер пути и перекоммутировать их.*

**Пример 4** (Задача коммивояжера через остовное дерево). *. Построим минимальное остовное дерево и обойдём его в глубину. При этом не-листья будут посещены много-кратно, исправим это. Будем обходить дерево как раньше, но помечать вершины. Если следующая вершина помечена, по пропустим её, пока не до непомеченной.*

**Утверждение 10.** *Если матрица расстояний удовлетворяет неравенству треугольни-ка, то алгоритм через остовное дерево получает вариант не более чем в два раза хуже.*

## 6 Билет 6

### 6.1 Вопрос 1. Информационные ресурсы. Принципы оценки информации как ресурса общества и объекта интеллектуальной собственности. Проблемы правового регулирования научной интеллектуальной собственности.

**Информационный ресурс** — вся совокупность сведений, получаемых и накапливаемых в процессе развития науки и практической деятельности людей. Информационные ресурсы зафиксированы в различного рода документах (отчётах, патентах, технической документации и пр.), теориях, моделях и пр. В современном обществе ИР относятся к наиболее важному виду материальных ресурсов.

В индустриальном обществе, где большая часть усилий направлена на материальное производство, известно несколько основных видов ресурсов, ставших уже классическими экономическими категориями:

- материальные ресурсы — совокупность предметов труда, предназначенных для использования в процессе производства общественного продукта, например сырье, материалы, топливо, энергия, полуфабрикаты, детали и т.д.;
- природные ресурсы — объекты, процессы, условия природы, используемые обществом для удовлетворения материальных и духовных потребностей людей;
- трудовые ресурсы — люди, обладающие общеобразовательными и профессиональными знаниями для работы в обществе;
- финансовые ресурсы — денежные средства, находящиеся в распоряжении государственной или коммерческой структуры;
- энергетические ресурсы — носители энергии, например уголь, нефть, нефтепродукты, газ, гидроэнергия, электроэнергия и т.д.

В информационном обществе акцент внимания и значимости смещается с традиционных видов ресурсов на информационный ресурс, который, хотя всегда существовал, не рассматривался ни как экономическая, ни как иная категория; никто специально о нем не говорил и тем более не вводил никаких определений.

Одним из ключевых понятий при информатизации общества стало понятие «информационные ресурсы», толкование и обсуждение которого велось с того момента, когда начали говорить о переходе к информационному обществу. Этому вопросу посвящено довольно много публикаций, в которых отразились и разные мнения и определения, и разные научные школы, рассматривающие эти понятия.

С принятием Федерального закона «Об информации, информатизации и защите информации» большая часть неопределенности была снята. Руководствуясь не научной сто-

роной этого вопроса, а скорее прагматической позицией потребителя информации, целесообразно воспользоваться определением, которое приведено в этом законе. Тем более нельзя не учитывать тот факт, что юридическое толкование во всех случаях является для пользователя информации опорой при защите его прав.

Информационные ресурсы — отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, других информационных системах).

Надо понимать, что документы и массивы информации, о которых говорится в этом законе, не существуют сами по себе. В них в разных формах представлены знания, которыми обладали люди, создававшие их. Таким образом, информационные ресурсы — это знания, подготовленные людьми для социального использования в обществе и зафиксированные на материальном носителе.

Информационные ресурсы общества, если их понимать как знания, отчуждены от тех людей, которые их накапливали, обобщали, анализировали, создавали и т.п. Эти знания материализовались в виде документов, баз данных, баз знаний, алгоритмов, компьютерных программ, а также произведений искусства, литературы, науки.

В настоящее время не разработана методология количественной и качественной оценки информационных ресурсов, а также прогнозирования потребностей общества в них. Это снижает эффективность информации, накапливаемой в виде информационных ресурсов, и увеличивает продолжительность переходного периода от индустриального к информационному обществу. Кроме того, неизвестно, какой объем трудовых ресурсов должен быть задействован в сфере производства и распространения информационных ресурсов в информационном обществе. Несомненно, в будущем эти проблемы будут решены.

Информационные ресурсы страны, региона, организации должны рассматриваться как стратегические ресурсы, аналогичные по значимости запасам сырья, энергии, ископаемых и прочим ресурсам.

## **6.2 Вопрос 2. Базы данных. Основные понятия. Независимость программ и данных. Интегрированное использование данных. Непротиворечивость данных. Целостность и защита данных. Понятие концептуальной, логической, физической структуры БД.**

См. также 14.1

**Определение 35.** *База данных — представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ) (Гражданский кодекс РФ, ст. 1260).*

**Независимость программ и данных.** Информационные данные любого пользователя в БД должны быть независимы от всех других пользователей, т. е. не должны оказывать влияния на существующие внешние модели. Это первый уровень независимости данных. С другой стороны, внешние модели пользователей никак не связаны с типом физической памяти, в которой будут храниться данные, и с физическими методами доступа к этим данным. Это положение отражает второй уровень независимости данных.

**Непротиворечивость данных. Целостность и защита данных.** Непротиворечивость и целостность обеспечивается ограничениями, которые могут быть установлены на уровне поля, кортежа, отношения или базы целиком.

Защита данных обеспечивается шифрованием канала передачи данных, а также процедурами аутентификации и авторизации.

**Понятие концептуальной, логической, физической структуры БД.** Концептуальная модель отображает предметную область в виде взаимосвязанных объектов без указания способов их физического хранения. Концептуальная модель представляет интегрированные концептуальные требования всех пользователей к базе данных данной предметной области.

При этом усилия разработчика должны быть направлены в основном на структуризацию данных, принадлежащих будущим пользователям БД, и выявление взаимосвязей между ними.

Возможно, что отраженные в концептуальной модели взаимосвязи между объектами окажутся впоследствии нереализуемыми средствами выбранной СУБД. Это потребует изменения концептуальной модели. Версия концептуальной модели, которая может быть реализована конкретной СУБД, называется логической моделью.

Логическая модель отражает логические связи между атрибутами объектов вне зависимости от их содержания и среды хранения и может быть реляционной, иерархической или сетевой. Таким образом, логическая модель отображает логические связи между информационными данными в данной концептуальной модели.

Различным пользователям в информационной модели соответствуют различные подмножества ее логической модели, которые называются внешними моделями пользователей. Таким образом, внешняя модель пользователя представляет собой отображение концептуальных требований этого пользователя в логической модели и соответствует тем представлениям, которые пользователь получает о предметной области на основе логической модели. Следовательно, насколько хорошо спроектирована внешняя модель, настолько полно и точно информационная модель отображает предметную область и настолько полно и точно работает автоматизированная система управления этой предметной областью.

Логическая модель отображается в физическую память, которая может быть построена на электронных, магнитных, оптических, биологических или других принципах.

Внутренняя модель предметной области определяет размещение данных, методы доступа и технику индексирования в данной логической модели и иначе называется физической моделью.

### 6.3 Вопрос 3

То же, что и 16.3

## 7 Билет 7

**7.1 Вопрос 1. Общие принципы моделирования окружающей среды, процессов мышления человека и человеко-машинного общения. Машинное представление знаний и данных. Методы хранения, поиска и обработки данных, методы естественно-языкового человеко-машинного общения. Предметная область и её модели. Понятия «план-содержание», «план-выражение». Объекты, характеристики и их значения. Единицы информации и информационные отношения. Машинное понимание.**

Моделирование — это процесс построения, изучения и применения моделей.

**1. Принцип информационной достаточности.** При полном отсутствии информации об исследуемой системе построение ее модели невозможно. При наличии полной информации ее моделирование лишено смысла. Существует некоторый критический уровень априорных сведений о системе (уровень информационной достаточности), при достижении которого может быть построена ее адекватная модель.

**2. Принцип осуществимости.** Создаваемая модель должна обеспечивать достижение поставленной цели исследования с вероятностью, существенно отличающейся от нуля, и за конечное время. Обычно задают некоторое пороговое значение  $P_0$  вероятности достижения цели моделирования  $P(t)$ , а также приемлемую границу  $t_0$  времени достижения этой цели. Модель считают осуществимой, если может быть выполнено условие  $P(t_0) \leq P_0$ .

**3. Принцип множественности моделей.** Данный принцип, несмотря на его порядковый номер, является ключевым. Речь идет о том, что создаваемая модель должна отражать в первую очередь те свойства реальной системы (или явления), которые влияют на выбранный показатель эффективности. Соответственно при использовании любой конкретной модели признаются лишь некоторые стороны реальности. Для более полного ее исследования необходим ряд моделей, позволяющих с разных сторон и с разной степенью детальности отражать рассматриваемый процесс.

**4. Принцип агрегирования.** В большинстве случаев сложную систему можно представить состоящей из агрегатов (подсистем), для адекватного математического описания которых оказываются пригодными некоторые стандартные математические схемы. Принцип агрегирования позволяет, кроме того, достаточно гибко перестраивать модель в зависимости от задач исследования.

**5. Принцип параметризации.** В ряде случаев моделируемая система имеет в своем составе некоторые относительно изолированные подсистемы, характеризующиеся определённым параметром, в том числе векторным. Такие подсистемы можно заменять в модели соответствующими числовыми величинами, а не описывать процесс их функционирования. При необходимости зависимость значений этих величин от ситуации может задаваться в виде таблицы, графика или аналитического выражения (формулы). Принцип параметризации позволяет сократить объем и продолжительность моделирования. Однако надо иметь в виду, что параметризация снижает адекватность модели.

**Методы хранения, поиска и обработки данных, методы естественно-языкового человеко-машинного общения.**

**Предметная область и ее модели.**

**Понятия «план-содержание», «план-выражение».** План-содержание — содержание, концептуальные свойства знака. План-выражение — необходимая составляющая знака (материальный носитель), то, что можно воспринять (звук, изображение, тактильный образ).

**Объекты, характеристики и их значения.** Выделение сущностей, имеющих определённые свойства и связанных с ними информационных блоков. Значение может иметь свою структуру (например, атрибут «согласование»). Единицы информации и информационные отношения.

**Машинное понимание.** Предположим, что на вход ИС поступает текст. Будем говорить, что ИС понимает текст, если она даёт ответы, правильные с точки зрения человека, на любые вопросы, относящиеся к тому, о чем говорится в тексте. Под *человеком* понимается конкретный человек-эксперт, которому поручено оценить способности системы к пониманию. Это вносит долю субъективизма, ибо разные люди могут по-разному понимать одни и те же тексты.

Классификация уровней понимания В существующих ИС можно выделить пять основных уровней понимания и два уровня метапонимания.

Первый уровень характеризуется схемой, показывающей, что любые ответы на вопросы система формирует только на основе прямого содержания, введенного из текста. Если, например, в систему введен текст: "В восемь утра, после завтрака, Петя ушел в школу.

В два часа он вернулся домой. После обеда он ушел гулять то на первом уровне понимания система обязана уметь отвечать правильно на вопросы типа: "Когда Петя ушел в школу?" или "Что сделал Петя после обеда?". В лингвистическом процессоре происходит морфологический, синтаксический и семантический анализ текста и вопросов, относящихся к нему. На выходе лингвистического процессора получается внутреннее представление текста и вопросов, с которыми может работать блок вывода. Используя специальные процедуры, этот блок формирует ответы. Другими словами, уже понимание на первом уровне требует от ИС определенных средств представления данных и вывода на этих данных.

Второй уровень: На втором уровне добавляются средства логического вывода, основанные на информации, содержащейся в тексте. Это разнообразные логики текста (временная, пространственная, каузальная и т. п., которые способны породить информацию, явно отсутствующую в тексте. Для нашего примера на втором уровне возможно формирование правильных ответов на вопросы типа: "Что было раньше: уход Пети в школу или его обед?" или "Гулял Петя после возвращения из школы?" Только построив временную структуру текста, ИС сможет ответить на подобные вопросы. Схема ИС, с помощью которой может быть реализован второй уровень понимания, имеет еще одну базу знаний. В ней хранятся закономерности, относящиеся к временной структуре событий, возможной их пространственной организации, каузальной зависимости и т. п., а логический блок обладает всеми необходимыми средствами для работы с псевдофизическими логиками.

Третий уровень: К средствам второго уровня добавляются правила пополнения текста знаниями системы о среде. Эти знания в ИС, как правило, носят логический характер и фиксируются в виде сценариев или процедур иного типа. На третьем уровне понимания ИС должна дать правильные ответы на вопросы типа: "Где был Петя в десять утра?" или "Откуда Петя вернулся в два часа дня?" Для этого надо знать, что означает процесс "пребывание в школе" и, в частности, что этот процесс является непрерывным и что субъект, участвующий в нем, все время находится "в школе". Схема ИС, в которой реализуется понимание третьего уровня, внешне не отличается от схемы второго уровня. Однако в логическом блоке должны быть предусмотрены средства не только для чисто дедуктивного вывода, но и для вывода по сценариям. Три перечисленных уровня понимания реализованы во всех практически работающих ИС. Первый уровень и частично второй входят в разнообразные системы общения на естественном языке.

Следующие два уровня понимания реализованы в существующих ИС лишь частично.

Четвертый уровень: Вместо текста в ней используется расширенный текст, который порождается лишь при наличии двух каналов получения информации. По одному в систему передается текст, по другому-дополнительная информация, отсутствующая в тексте. При человеческой коммуникации роль второго канала, как правило, играет зрение. Более одного канала коммуникации имеют интеллектуальные роботы, обладающие зрением. Зрительный канал коммуникации позволяет фиксировать состояние среды "здесь и сейчас" и вводить в текст наблюдаемую информацию. Система становится способной к пониманию текстов, в которые введены слова, прямо связанные с той ситуацией, в ко-

торой порождается текст. На более низких уровнях понимания нельзя понять, например, текст: "Посмотрите, что сделал Петя! Он не должен был брать это!" При наличии зрительного канала процесс понимания становится возможным. При наличии четвертого уровня понимания ИС способна отвечать на вопросы типа: "Почему Петя не должен был брать это?" или "Что сделал Петя?" Если вопрос, поступивший в систему, соответствует третьему уровню, то система выдает нужный ответ. Если для ответа необходимо привлечь дополнительную ("экзегетическую") информацию, то внутреннее представление текста и вопроса передается в блок, который осуществляет соотнесение текста с той реальной ситуацией его порождения, которая доступна ИС по зрительному или какому-нибудь иному каналу фиксации ситуации внешнего мира.

Пятый уровень: Для ответа на этом уровне ИС кроме текста использует информацию о конкретном субъекте, являющемся источником текста, и хранящуюся в памяти системы общую информацию, относящуюся к коммуникации (знания об организации общения, о целях участников общения, о нормах участия в общении). Теория, соответствующая пятому уровню, — это так называемая теория речевых актов. Было обращено внимание на то, что любая фраза не только обозначает некое явление действительности, но и объединяет в себе три действия: локуцию, иллокуцию и перлокуцию. Локуция — это говорение как таковое, т. е. те действия, которые говорящий произвел, чтобы высказать свою мысль. Иллокуция — это действие при помощи говорения: вопрос, побуждение (приказ или просьба) и утверждение. Перлокуция — это действие, которым говорящий пытается осуществить некоторое воздействие на слушающего: "лестить" "удивлять" "уговаривать" и т. д. Речевой акт можно определить как минимальную осмысленную (или целесообразную) единицу речевого поведения. Каждый речевой акт состоит из локутивного, иллокутивного и перлокутивного акта.

Для четвертого и пятого уровней понимания интересны результаты по невербальным (несловесным) компонентам общения и психологическим принципам, лежащим в основе общения. Кроме того, в правила пополнения текста входят правила вывода, опирающиеся на знания о данном конкретном субъекте общения, если такие знания у системы есть. Например, система может доверять данному субъекту, считая, что порождаемый им текст истинен. Но может не доверять ему и понимать текст, корректируя его в соответствии со своими знаниями о субъекте, породившем текст. Знания такого типа должны опираться на психологические теории общения, которые пока развиты недостаточно. Например, на вход системы поступает текст: "Нина обещала скоро прийти". Если о Нине у системы нет никакой информации, она может обратиться к базе знаний и использовать для оценки временного указателя "скоро" некоторую нормативную информацию. Из этой информации можно узнать, что с большой долей уверенности "скоро" не превышает полчаса. Но у системы может иметься специальная информация о той Нине, о которой идет речь во входном тексте. В этом случае система, получив нужную информацию из базы знаний, может приготовиться, например, к тому, что Нина скорее всего придет не ранее чем через час.

Первый метауровень: На этом уровне происходит изменение содержимого базы знаний. Она пополняется фактами, известными системе и содержащимися в тех текстах, которые в систему введены. Разные ИС отличаются друг от друга характером правил порождения фактов из знаний. Например, в системах, предназначенных для экспертизы в области фармакологии, эти правила опираются на методы индуктивного вывода и распознавания образов. Правила могут быть основаны на принципах вероятностей, размытых выводов и т. п. Но во всех случаях база знаний оказывается априорно неполной и в таких ИС возникают трудности с поиском ответов на запросы. В частности, в базах знаний становится необходимым немонотонный вывод.

Второй метауровень: На этом уровне происходит порождение метафорического знания. Правила порождения знаний метафорического уровня, используемые для этих целей, представляют собой специальные процедуры, опирающиеся на вывод по аналогии и ассоциации. Известные в настоящее время схемы вывода по аналогии используют, как правило, диаграмму Лейбница, которая отражает лишь частный случай рассуждений по аналогии. Еще более бедны схемы ассоциативных рассуждений.

## **7.2 Вопрос 2. Операционные системы. Функции операционной системы (ОС): управление задачами, управление данными, связь с оператором. Сообщения операционной системы. Команды и директивы оператора.**

**Определение 36.** *Операционная система набор программно-аппаратных средств, обеспечивающих управление компьютером.*

Классификация ОС:

- По назначению: Общего пользования / Специальные (например, для управления спутником)
- По количеству пользователей: Однопользовательские (MS DOS) / Многопользовательские (Windows, UNIX)
- По сетевой функциональности: Локальные / Сетевые
- По управлению задачами
  - Пакетные (перед началом работы — список задач на выполнение, вмешиваться в работу системы нельзя, пример — ввод с перфокарт)
  - С разделением времени (несколько задач, работающих одновременно, пример — Windows, UNIX)
  - ОС реального времени (особый приоритет отдается задачам, критичным по времени, пример — ОС спутников, фотоаппаратов)

Состав ОС:

- Загрузчик ОС
- Ядро ОС
- Планировщик процессов
- Подсистема ввода/вывода

Драйверы ОС — специальные программы, предназначенные для управления внешними устройствами. Являются частью подсистемы ввода/вывода ОС.

Резидентные программы (в ОС MSDOS) — программа, вернувшая управление ОС, но оставшееся в оперативной памяти компьютера.

Основные функции ОС:

- управление памятью;
- организация доступа к внешним устройствам;
- управление задачами;
- организация сетевого взаимодействия;
- разграничение прав доступа;
- организация интерфейса к пользователю;
- управление вводом-выводом;
- управление файловой системой;
- управление взаимодействием процессов;
- диспетчеризация процессов;
- защита и учет использования ресурсов;
- обработка командного языка.

Утилиты ОС — программы, входящие в поставку ОС, предназначенные для предоставления функций ОС пользователю.

Основные объекты ОС:

- Процесс
- Файл
- Событие
- Объект синхронизации (мьютекс, семафор)
- Канал данных

### 7.3 Вопрос 3. Способы задания автомата, канонические уравнения, диаграмма Мура. Автоматная функция. Детерминированная функция, понятие о.-д. функции.

Каноническое уравнение автомата  $V = (A, Q, B, \varphi, \psi, q_0)$ :

$$\begin{cases} q(1) = q_0 \\ q(t+1) = \varphi(q(t), a(t)) \\ b(t) = \psi(q(t), a(t)) \end{cases}$$

**Определение 37.** Функция  $f : A^* \rightarrow B^*$  детерминированная, если  $\forall n \in \mathbb{N}, a \in A^n, b \in A^* : f(ab)|_{1\dots n} = f(a)$ .

**Определение 38.** Пусть  $f : A^* \rightarrow B^*$  — детерминированная функция. Функции вида  $g(w) = f(aw)$ ,  $a, w \in A^*$  называются остаточными для  $f$ . Функция  $f$  называется ограниченно-детерминированной, если она порождает конечное число остаточных функций.

**Утверждение 11.** Конечные автоматы эквивалентны о.-д. функциям.

## 8 Билет 8

### 8.1 Вопрос 1. Когнитивные (интеллектуальные) системы. Декларативное и процедурное представление внешнего мира. Знание и компетенция, восприятие, мышление и двигательное возбуждение. База знаний и база данных.

Когнитивные (интеллектуальные) системы.

**Представление знаний** Знания 1) это совокупность хранимых в базах знаний или памяти человека фактов о некоторой предметной области, их взаимосвязей и правил, которые могут быть использованы для получения новых фактов или решения каких-либо задач, связанных с этой предметной областью. 2) совокупность сведений, образующих целостное описание, соответствующее определенному уровню осведомленности об описываемой ПО.

Свойства знаний:

- внутренняя интерпретируемость; Каждая информационная единица должна иметь уникальное имя, по которому ИС находит ее.
- структурированность; Информационные единицы должны обладать гибкой структурой. Одни информационные единицы должны быть вложимы в другие. Долж-

на быть возможность установления отношений типа «часть-целое», «род-вид», «элемент-класс» между информационными единицами.

- связанность. Должна быть предусмотрена возможность установления связей различного типа:
- отношения структуризации (иерархия инф. единиц);
- функциональные отношения (процедурная информация, позволяющая вычислять одни инф. ед. через другие)
- каузальные отношения (причинно-следственные связи);
- семантические отношения (ост. виды отношений);
- активность; В ИС актуализация действий вызывается знаниями, имеющимися в системе. Выполнение программ инициируется текущим состоянием базы знаний.
- семантическая метрика; Отражает силу ассоциативной связи между информационными единицами.

Основная особенность знаний — связанность всех понятий ПО в иерархическую сеть — иерархию понятий.

Виды знаний

- жесткие — позволяют получить однозначные рекомендации
- мягкие — допускают множественные решения
- декларативные — факты из предметной области
- процедурные — правила преобразования объектов предметной области

Факты — знания в форме утверждения, достоверность которого строго установлена.

Эвристические знания — знания, накапливаемые интеллектуальной системой в процессе ее функционирования, а также знания, заложенные в ней априорно, но не имеющие статуса абсолютной истинности в данной проблемной области. Обычно эвристические знания связаны с отражением в базе знаний неформального опыта решения задач.

Метазнания — знания о знаниях (свойства знаний, способы их использования и пр.)

**Знания декларативные** — Утверждения об объектах предметной области, их свойствах и отношениях между ними — факты из предметной области.

«Фреймы» - концептуальные структуры для декларативного представления знаний. Используются в продукционных, редуцированных и логических языках программирования (Пролог, Lisp). Достоинства: способ поиска решений универсален и не зависит ни от поставленной задачи, ни даже от ПО, что весьма важно при описании слабо изученных

и изменяющихся ПО. Недостаток: низкая вычислительная эффективность (по затратам времени и памяти), поскольку в процедурах поиска решения не учитывается специфика решаемой задачи и ПО, что делает эту форму непригодной для применения в системах реального времени.

**Знания процедурные (императивные)** Правила преобразования объектов предметной области (рецепты, алгоритмы, методики, инструкции, стратегии принятия решений), последовательность операций над данными. «Сценарии» - концептуальные структуры для процедурного представления знаний Используется в императивных языках программирования (например, Паскаль, С++). Достоинства: наиболее эффективна с вычислительной точки зрения (по затратам времени и памяти на решение задачи), поскольку в процедурах поиска решения глубоко учитывается специфика конкретной проблемной области (ПО), пригодна для применения в системах реального времени. Недостаток: сложность внесения изменений, что делает ее непригодной для применения в слабо изученных и изменяющихся ПО.

Комбинированные знания создаются, чтобы преодолеть недостатки и сохранить достоинства императивной и декларативной форм. Хорошо обоснованная, устойчивая и формализованная часть знания воплощается в эффективных процедурах, а слабо изученная и изменчивая составляющая знания представляется в декларативной форме. Недостаток: трудность их теоретизации Они используются в семантических сетях и сетях фреймов.

Метазнания:

- знания о получении знаний, т.е. приёмы и методы познания.
- знания экспертной системы о собственном функционировании и процессах построения логических выводов.

**Знание и компетенция, восприятие, мышление и двигательное возбуждение.**

**Определение 39.** *Восприятие — отражение окружающей ситуации и ее элементов при взаимодействии органов чувств человека или рецепторов искусственной системы с внешней средой. восприятие зрительной информации, восприятие тактильной информации и акустической информации (распознавание речи).*

**Определение 40.** *Компетенция — комплекс, связывающий воедино знания, умения и действия способность мобилизовать знания/умения в конкретной ситуации.*

**База знаний и база данных.** Знания — совокупность сведений, образующих целостное описание, соответствующее определённому уровню осведомлённости об описываемой проблеме.

Данные — факты и идеи, представленные в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе.

Отличие знаний от данных: данные описывают состояние объектов в текущий момент времени, а знания содержат также сведения о том, как оперировать этими данными. Знания используются в системах искусственного интеллекта, например, в экспертных системах.

Инженерия знаний научная дисциплина, занимающаяся исследованиями в области извлечения, представления, формализации, обработки, использования знаний.

База знаний совокупность фактов и правил, описывающая предметную область и позволяющая отвечать на вопросы из этой ПО, ответы на которые в явном виде не присутствуют в базе.

База знаний является основным компонентом интеллектуальных и экспертных систем.

Для представления знаний используются семантические сети, процедурные, фреймовые и др. модели.

Модели представления знаний делятся на формальные и неформальные.

В основе формальных моделей - лежит строгая математическая теория - вывод строг и корректен - они универсальны Их недостатки — закрытость, негибкость. Примеры: Логические модели: исчисление высказываний и предикатов.

В неформальных моделях - вывод определяется самим исследователем - они создаются для конкретных ПО Примеры: сетевые модели, продукционные модели, фреймовые модели.

## 8.2 Вопрос 2. Понятие разработки приложений. Состав системы программирования: язык программирования (ЯП), обработчик программ; библиотека программ и функций.

Вопрос-импровизации.

См также 3.1 про стадии разработки по ГОСТ.

## 8.3 Вопрос 3. Эквивалентность состояний автомата, сильная и слабая эквивалентность автоматов. Теорема о единственности приведённого автомата, эквивалентному данному.

(Кудрявцев и Ко. Введение в теорию автоматов. стр 44)

**Определение 41.** Состояния  $q$  и  $q'$  автоматов  $(A, Q, B, \varphi, \psi)$  и  $(A, Q', B, \varphi', \psi')$  неотличимы, если для всех  $\alpha \in A^*$  выполнено:  $\bar{\psi}(q, \alpha) = \bar{\psi}'(q', \alpha)$ . Обозначение:  $q \sim q'$ .

**Определение 42.** Автомат, где все состояния неотличимы - **автомат приведённого типа**. Если для любого состояния одного автомата существует неотличимое состояние в другом автомате (и наоборот), то автоматы называются **неотличимыми**. Множество неотличимых автоматов обозначается за  $K_V(A, B)$ . Тривиально определяются **изоморфные автоматы**.

**Теорема 15.** Пусть  $A, B$  — конечные множества и  $V \in K(A, B)$ , тогда в  $K_V(A, B)$  существует единственный с точностью до изоморфизма автомат приведённого типа.

**Теорема 16.** Для любых алфавитов  $A, Q, B$ , таких, что  $|A| \geq 2, |Q| \geq 3, |B| \geq 2$ , существует такой автомат **приведённого вида**  $V, V = (A, Q, B, \varphi, \psi)$ , что для любого слова  $\alpha \in A^*$ , существуют различные  $q, q' \in Q$ , т.ч.  $\bar{\psi}(q, \alpha) = \bar{\psi}(q', \alpha)$ .

Доказательство. См. ??.

□

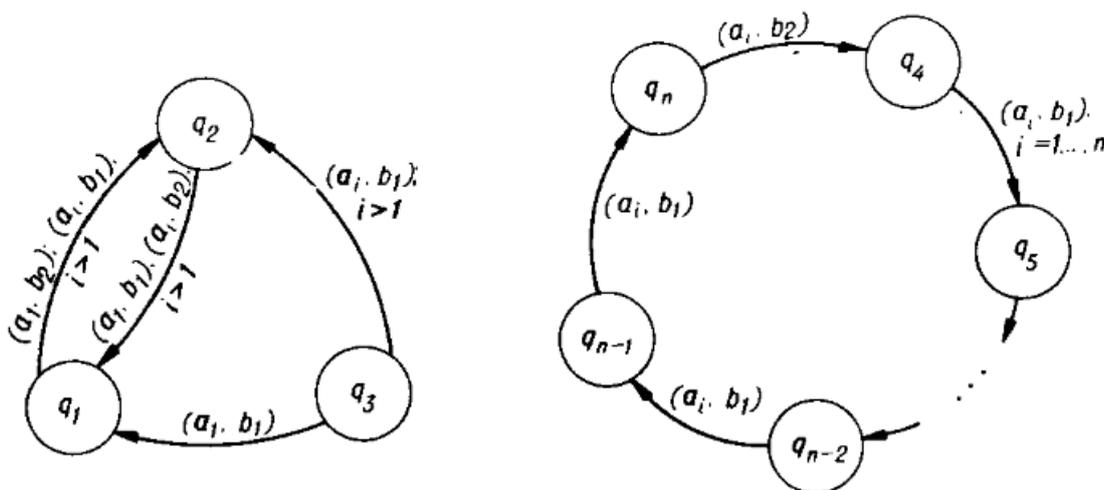


Рис. 1: Автомат со слабо различимыми состояниями.

**Определение 43.** Автомат  $V = (A, Q, B, \varphi, \psi)$  назовём слабо неотличимым от автомата  $V' = (A, Q', B, \varphi', \psi')$  неотличимы, если для любых состояний  $q \in Q$  и слова  $\alpha \in A^*$  существует состояние  $q' \in Q'$  такое, что:  $\bar{\psi}(q, \alpha) = \bar{\psi}'(q', \alpha)$ . Обозначение:  $V \leq V'$ .

**Определение 44.** Если  $V \leq V'$  и  $V' \leq V$ , то будем называть автоматы  $V$  и  $V'$  слабо неотличимыми.

## 9 Билет 9

### 9.1 Вопрос 1. Знаковые системы. Семиотический треугольник и его элементы. Понятия «экстенционал» и «интенционал».

1. Знак (Имя)
2. Денотат (объем понятия)  
Пример: денотат (президент США) = Обама.
3. Концепт(смысл, содержание понятия)  
Пример: концепт (президент США) = само понятие.

Экстенционал знака — множество всех его возможных денотатов.

Интенционал знака определяет содержание понятия и характеризует концепт.

Закон обратного соотношения объема понятия и содержания понятия (чем больше содержание, тем меньше объем). Пример: красных столов больше, чем просто столов.

## 9.2 Вопрос 2. Понятие о словарях данных, языках описания и манипулирования данными. Документальные и фактографические базы данных, базы знаний. Полнотекстовые БД. Частотный словарь, инверсный файл. Положительный и отрицательный словари

План БД определяется набором выражений (дефиниций), написанных на специальном языке, который называется **язык описания данных** (ЯОД) (data definition language).

Результатом компиляции выражений на ЯОД является набор таблиц, хранящийся в специальном файле, который называется **словарь данных** (data dictionary). В словаре данных хранятся метаданные, то есть данные о данных.

Язык манипулирования данными это командный язык, обеспечивающий выполнение основных операций по работе с данными:

- извлечение информации, хранящейся в БД;
- добавление новой информации в БД;
- уничтожение хранящейся в БД информации.

В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс. Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык SQL (Structured Query Language).

**Документальные и фактографические базы данных, базы знаний.** В фактографических БД регистрируются факты — конкретные значения данных (атрибутов) об объектах реального мира. Основная идея таких систем заключается в том, что все сведения об объектах (фамилии людей и названия предметов, числа, даты) сообщаются компьютеру в каком-то заранее обусловленном формате (например дата — в виде комбинации ДД.ММ.ГГГГ). Информация, с которой работает фактографическая информационная система, имеет внутреннюю структуру, что позволяет использовать ту или иную модель данных.

Существует и другой класс задач — хранение и анализ неструктурированных текстовых документов (статьи, книги, рефераты и т.д.) и графических объектов. В этом случае цель системы, как правило, — выдать в ответ на запрос пользователя список документов

или объектов, в какой-то мере удовлетворяющих сформулированным в запросе условиям. То есть система должна быть снабжена каким-либо формализованным аппаратом поиска. Для решения этих задач используются документальные базы данных, системы, основанные на других принципах.

База знаний в информатике и исследованиях искусственного интеллекта — это особого рода база данных, разработанная для оперирования знаниями (метаданными). База знаний содержит структурированную информацию, покрывающую некоторую область знаний, для использования кибернетическим устройством (или человеком) с конкретной целью. Современные базы знаний работают совместно с системами поиска информации, имеют классификационную структуру и формат представления знаний.

Полноценные базы знаний содержат в себе не только фактическую информацию, но и правила вывода, допускающие автоматические умозаключения о вновь вводимых фактах и, как следствие, осмысленную обработку информации.

**Полнотекстовые БД. Частотный словарь, инверсный файл. Положительный и отрицательный словари** Полнотекстовые БД определяются как БД, в которых хранятся записи полнотекстовых документов или их частей. Полнотекстовые БД значительно расширяют возможности доступа, предоставляя один и тот же текст в различных форматах (ascii text, .html, .sgml, .pdf), включая иллюстрации, графики и таблицы изображения.

Частотный словарь — набор слов данного языка (или подъязыка) вместе с информацией о частоте их встречаемости. Словарь может быть отсортирован по частоте, по алфавиту (тогда для каждого слова будет указана его частота), по группам слов (например, первая тысяча наиболее частотных слов, за ней вторая и т. п.), по типичности (слова, частотные для большинства текстов), и т. д. Используется, например, для построения образов документов в векторной и при выборе и для более эффективной выборки документов с использованием обратного индекса.

Инвертированный файл — индекс поисковой системы, в котором перечислены слова коллекции документов, а для каждого слова перечислены все места, в которых оно встретилось.

**Положительный и отрицательный словари ???**

### 9.3 Вопрос 3. Абстрактный автомат. Проверка состояний эквивалентности состояний автомата, теорема Мура о длине слова, проверяющего состояния автомата и её следствия.

(Кудрявцев и Ко. Введение в теорию автоматов. стр 57)

**Определение 45.** *Абстрактный автомат  $(A, Q, B, \varphi, \psi, q)$ :*

- $A$  — входной алфавит;

- $Q$  — множество состояний;
- $B$  — выходной алфавит;
- $\varphi: A \times Q \Leftarrow Q$  — функция переходов;
- $\psi: A \times Q \Leftarrow B$  — функция выходов;
- $q \in Q$  — начальное состояние.

**Определение 46.** Состояния  $q$  и  $q'$  автоматов  $(A, Q, B, \varphi, \psi)$  и  $(A, Q', B, \varphi', \psi')$  неотличимы, если для всех  $\alpha \in A^*$  выполнено:  $\bar{\psi}(q, \alpha) = \bar{\psi}'(q', \alpha)$ . Обозначение:  $q \sim q'$ .

**Определение 47.** Состояния  $q$  и  $q'$  автоматов  $(A, Q, B, \varphi, \psi)$  и  $(A, Q', B, \varphi', \psi')$  неотличимы множеством  $M \subset A^*$ , если для всех  $\alpha \in M$  выполнено:  $\bar{\psi}(q, \alpha) = \bar{\psi}'(q', \alpha)$ .

Множеством  $M$  состояния автомата разбиваются на классы  $R_V(M) = \{Q_1, \dots, Q_m\}$ .

**Лемма 10.** Пусть  $V = (A, Q, B, \varphi, \psi)$  — конечный автомат,  $M \subset A^*$ ,  $\Lambda \in M$ . Тогда  $R_V(M \cup AM)$  есть некоторое подразбиение разбиения  $R_V(M)$ . Если  $R_V(M \cup AM) = R_V(M)$ , то любые два состояния автомата  $V$ , попадающие в один и тот же класс разбиения  $R_V(M)$ , неотличимы.

**Лемма 11.** Пусть  $V = (A, Q, B, \varphi, \psi)$  — конечный автомат,  $M \subset A^*$ ,  $\Lambda \in M$ ,  $|R_V(M)| = m$ ,  $|Q| = n$ . Тогда любые два отличимые состояния  $q_1, q_2$  автомата  $V$  отличимы множеством  $M \cup AM \cup A^2M \cup \dots \cup A^{n-m}M$ .

**Теорема 17.** Состояния  $q_1, q_2$  конечного автомата  $V = (A, Q, B, \varphi, \psi)$  неотличимы тогда и только тогда, когда они неотличимы множеством  $A^{|Q|-1}$ .

**Следствие 2.** Состояния  $q_1, q'_1$  автоматов  $V = (A, Q, B, \varphi, \psi)$  и  $V' = (A, Q', B, \varphi', \psi')$  неотличимы тогда и только тогда, когда они неотличимы множеством  $A^{|Q|+|Q'|-1}$ .

Пример автоматов, на которых оценка достигается, на рис. ?? и рис. ??.

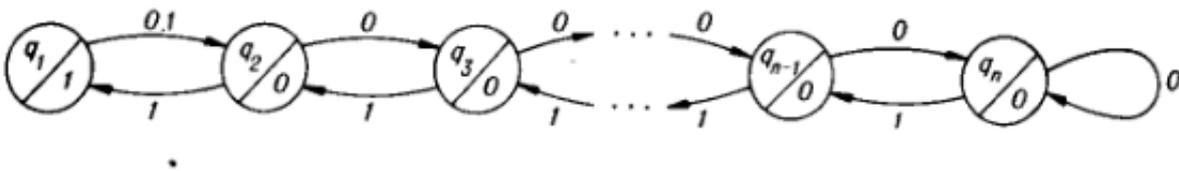


Рис. 2: Автомат 1

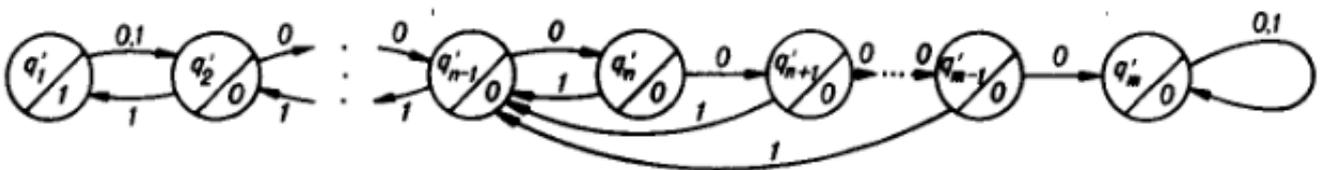


Рис. 3: Автомат 2

## 10 Билет 10

### 10.1 Вопрос 1. Классификационные системы: иерархическая классификация, фасетные классификации, алфавитно-предметные классификации. Тезаурусные методы представления данных. Системы, основанные на отношениях. Объектно-характеристические таблицы. Предикатно-октантные структуры.

**Определение 48.** *Классификация — выделение из множества объектов всех подклассов на основе значений выделенных атрибутов и установление между выделенными подклассами отношений порядка*

Атрибуты, участвующие в делении на классы — основания деления, классификационные атрибуты.

Соподчинение — отношение между классами, являющимися подклассами одного класса.

Методы классификации делятся на методы последовательной (иерархической) классификации и методы параллельной (многоаспектной, фасетной) классификации. Практические классификации строятся по комбинированному принципу.

**Иерархические (древовидные) классификации.** иерархическая система классификации строится на основе определения отношения подчинения между классификационными группировками. Логические правила классификаций:

- очередной шаг проводится только по одному основанию;
- получаемые подклассы не должны пересекаться;
- деление на подклассы должно быть соразмерным (классификация должна строиться таким образом, чтобы сумма подмножеств деления составляла делимое множество на любом иерархическом уровне).

Наиболее существенный и сложный вопрос при построении ИСК — выбор системы признаков, принимаемых в качестве основания деления, а также порядок их следования.

Если строится ИСК с независимыми признаками, то выбор последовательности признаков зависит, в основном, от вероятности обращения к ним.

Если же признаки зависимы и отражают реальные отношения соподчинения, то данная иерархическая структура и определяет состав и порядок следования классификационных признаков.

Преимущества ИСК: - большая информационная ёмкость; - традиционность и простота применения; - хорошая приспособленность для ручной обработки информации.

Недостатки ИСК: - структурная негибкость, обусловленная фиксированностью постоянных признаков и заранее установленным порядком их следования, не допускающим включения новых объектов и группировок без перестройки всей иерархической структуры или какой-либо её части; - невозможность объединения объектов в классификационные группировки по любому ранее не предусмотренному признаку или группе признаков (приходится собирать документы из нескольких классов).

Если иерархическая классификация соответствует естественному порядку классов (род — вид), то такая классификация будет служить эффективным средством информационного поиска.

**Фасетные (комбинативные) классификации.** ФСК применяет параллельно несколько независимых оснований деления, т.е. классификационное множество рассматривается одновременно в нескольких аспектах (отсюда другое название — многоаспектная система классификации).

Классификационный признак называется фасетом (например, фасет цвет содержит значения: красный, белый, зелёный, чёрный, жёлтый). Фасетная формула представляет из себя некоторую последовательность фасетов, не обязательно упорядоченную определённым образом и не обязательно содержащую ВСЕ множество фасет.

Примеры: УДК (универсальный десятичный классификатор)

Классификация фильмов:

- тип: документальный, игровой, анимация (мультипликация)
- жанр: боевик, комедия, романтика, фантастика
- продолжительность
- год
- страна

ФСК отличается большой гибкостью и удобством использования. Она дает возможность строить группировки по любому сочетанию выбранных признаков. Причём при построении классификационных группировок из разных фасет ненужные фасеты можно пропускать, что недопустимо для иерархической системы.

Достоинства: возможность объединения объектов в классификационные группировки по ранее не предусмотренному признаку или группе признаков \* возможность простой модификации всей системы классификации без изменения структуры существующих группировок.

Недостатки: \* сложность построения, т.к. необходимо учитывать все разнообразие классификационных признаков; \* большая длина классификационного кода; \* большая избыточность;

**Алфавитно-предметные классификации.** Системы, основной словарный состав которого состоит из упорядоченного по алфавиту множества слов и словосочетаний ЕЯ, обозначающих предметы какой-либо ПО

Виды:

- алфавитно-систематические (иерархия)
- словарные (списки)

Пример: Элементарные частицы — барионы — нуклоны — нейтроны

Достоинства: привычность; легко вводить новые термины.

Недостатки: не позволяет производить поиск по любым сочетаниям предметов; неудобно включать синонимы или полусинонимы; трудоемкий процесс выделения предметов.

Ни одна из традиционных систем не обеспечивает возможности поиска документов по любому, заранее не заданному сочетанию признаков.

**Тезаурусные методы представления знаний.** Тезаурус — особая разновидность словарей, в которых указаны семантические отношения (синонимы, антонимы, паронимы, гипонимы, гиперонимы и т. п.) между лексическими единицами.

Тезаурусы являются одним из действенных инструментов для описания отдельных предметных областей.

В отличие от толкового словаря, тезаурус позволяет выявить смысл не только с помощью определения, но и посредством соотнесения слова с другими понятиями и их группами, благодаря чему может использоваться в системах искусственного интеллекта.

**Системы, основанные на отношениях. Объектно-характеристические таблицы.**  
= таблицы реляционных структур данных = «отношения»

**Предикатно-актантные структуры.** предикат — функция  
актант — значения предметных переменных  
кванторы

## **10.2 Вопрос 2. Источники объекты и субъекты авторского права. Защита авторских и смежных прав. Правовая охрана и защита патентного права и прав на средства индивидуализации. Источники права о защите против недобросовестной конкуренции. Правовая охрана права на защиту против недобросовестной конкуренции.**

Согласно ст.1259 ГК РФ объектами авторских прав являются произведения науки, литературы и искусства независимо от достоинств и назначения произведения, а также от способа его выражения:

- литературные произведения;
- драматические и музыкально-драматические произведения, сценарные произведения;
- хореографические произведения и пантомимы;
- музыкальные произведения с текстом или без текста;
- аудиовизуальные произведения;
- произведения живописи, скульптуры, графики, дизайна, графические рассказы, комиксы и другие произведения изобразительного искусства;
- произведения декоративно-прикладного и сценографического искусства;
- произведения архитектуры, градостроительства и садово-паркового искусства, в том числе в виде проектов, чертежей, изображений и макетов;
- фотографические произведения и произведения, полученные способами, аналогичными фотографии;
- географические, геологические и другие карты, планы, эскизы и пластические произведения, относящиеся к географии, топографии и к другим наукам;
- другие произведения.

К объектам авторских прав также относятся программы для ЭВМ, которые охраняются как литературные произведения.

Закон устанавливает два общих критерия охраноспособности объектов авторского права, дает примерный перечень произведений, которые при условии соответствия их названным критериям могут являться объектами авторского права, а также определяет сферу действия авторского права.

Авторское право распространяется на произведения науки, литературы и искусства, во-первых, являющиеся результатом творческой деятельности и, во-вторых, существующие в какой-либо объективной форме.

При этом объектом авторского права может служить как произведение в целом, так и его часть, которая удовлетворяет критериям охраноспособности и может использоваться самостоятельно.

Творческой обычно считается умственная деятельность, завершающаяся созданием творчески самостоятельного результата науки, литературы или искусства. Закон предусматривает примерный перечень объективных форм произведений:

- письменная (рукопись, машинопись, нотная запись и т.п.);
- устная (публичное произнесение, исполнение и т.п.);

- звуко- или видеозапись (механическая, магнитная, цифровая, оптическая и т.п.);
- объемно-пространственная (скульптура, модель, макет, сооружение и т.п.).

Для признания произведения, выраженного в объективной форме, объектом авторского права не имеет значения также способ его выражения.

Авторское право распространяется на все обнародованные либо необнародованные произведения, находящиеся в какой-либо объективной форме на территории РФ, независимо от гражданства авторов и их правопреемников.

**Субъектами авторского права** являются создатели произведений науки, литературы и искусства, их правопреемники или работодатели и другие лица, приобретающие по закону или договору исключительные авторские права.

В качестве субъектов авторского права могут выступать как физические, так и юридические лица. К числу последних обычно относятся издатели газет, журналов и некоторых других составных произведений, изготовители аудиовизуальных произведений и работодатели авторов служебных произведений.

Автором произведения науки, литературы или искусства признается гражданин, творческим трудом которого оно создано. Лицо, указанное в качестве автора на оригинале или экземпляре произведения, считается его автором, если не доказано иное.

Авторское право на произведение возникает у его создателя в силу самого факта создания. Для возникновения и осуществления авторского права не требуется регистрации, иного специального оформления произведения или соблюдения каких-либо формальностей.

**Защита авторских и смежных прав.** Согласно ст.1255 ГК РФ автору произведения принадлежат следующие права:

- исключительное право на произведение;
- право авторства;
- право автора на имя;
- право на неприкосновенность произведения;
- право на обнародование произведения.

Глава 71 ГК РФ регулирует права, смежные с авторскими. Смежными с авторскими правами (смежными правами) являются интеллектуальные права:

- на результаты исполнительской деятельности (исполнения),
- на фонограммы,
- на сообщение в эфир или по кабелю радио- и телепередач (вещание организаций эфирного и кабельного вещания),

- на содержание баз данных,
- на произведения науки, литературы и искусства, обнародованные после их перехода в общественное достояние.

**Правовая охрана и защита патентного права и прав на средства индивидуализации.** ГК РФ определяет условия патентоспособности изобретения (ст. 1350 ГК РФ). В качестве изобретения охраняется техническое решение в любой области, относящееся к продукту (в частности, устройству, веществу, штамму микроорганизма, культуре клеток растений или животных) или способу (процессу осуществления действий над материальным объектом с помощью материальных средств). Изобретению предоставляется правовая охрана, если оно является новым, имеет изобретательский уровень и промышленно применимо.

Не являются изобретениями:

- открытия;
- научные теории и математические методы;
- решения, касающиеся только внешнего вида изделий и направленные на удовлетворение эстетических потребностей;
- правила и методы игр, интеллектуальной или хозяйственной деятельности;
- программы для ЭВМ;
- решения, заключающиеся только в представлении информации.

Не предоставляется правовая охрана в качестве изобретения:

- сортам растений, породам животных и биологическим способам их получения, за исключением микробиологических способов и продуктов, полученных такими способами;
- топологиям интегральных микросхем.

Средство индивидуализации — обозначение, служащее для различения товаров, услуг, предприятий, организаций и других объектов в сфере хозяйственного оборота.

К средствам индивидуализации обычно относят:

- фирменное наименование, наименование некоммерческой организации,
- товарный знак, знак обслуживания,
- коммерческое обозначение,
- наименование места происхождения товара,

- доменное имя.

В России средства индивидуализации приравнены законом (4-ой частью Гражданского кодекса) к результатам интеллектуальной деятельности, на которые установлено исключительное право интеллектуальной собственности.

**Источники права о защите против недобросовестной конкуренции. Правовая охрана права на защиту против недобросовестной конкуренции.** Недобросовестная конкуренция — нарушение общепринятых правил и норм конкуренции. При этом нарушаются законы и неписаные правила.

Ст. 4 Федерального закона от 26.07.2006 №135-ФЗ «О защите конкуренции» трактует это понятие как: любые действия хозяйствующих субъектов (группы лиц), которые направлены на получение преимуществ при осуществлении предпринимательской деятельности, противоречат законодательству Российской Федерации, обычаям делового оборота, требованиям добропорядочности, разумности и справедливости и причинили или могут причинить убытки другим хозяйствующим субъектам-конкурентам либо нанесли или могут нанести вред их деловой репутации.

Формы недобросовестной конкуренции установлены ст. 14 указанного выше закона:

- распространение ложных, неточных или искажённых сведений, способных причинить убытки другому хозяйствующему субъекту либо нанести ущерб его деловой репутации;
- введение потребителей в заблуждение относительно характера, способа и места изготовления, потребительских свойств, качества товара;
- некорректное сравнение хозяйствующим субъектом производимых или реализуемых им товаров с товарами других хозяйствующих субъектов;
- продажа товара с незаконным использованием результатов интеллектуальной деятельности и приравненных к ним средств индивидуализации юридического лица, индивидуализации продукции, выполнения работ, услуг;
- получение, использование, разглашение научно-технической, производственной или торговой информации, в том числе коммерческой тайны, без согласия её владельца.

### **10.3 Вопрос 3. Проверка эквивалентности конечных автоматов. Теорем Мура о длине слова, отличающего конечные автоматы. Достижимость оценки длины слова, отличающего конечные автоматы.**

(Кудрявцев и Ко. Введение в теорию автоматов. стр 60)

**Определение 49.** Инициальный автомат  $V_q = (A, Q, B, \varphi, \psi, q)$ , у которого любые два состояния отличимы друг от друга, причём для любого  $q' \in Q$  существует  $\alpha \in A^*$ , такое что  $q' = \varphi(q, \alpha)$ , называется **автоматом приведённого типа**.

Неотличимость инициальных автоматов эквивалентна неотличимости соответствующих им автоматов приведённого типа. Поэтому будем рассматривать только последние.

**Определение 50.** Базисом достижимости инициального автомата приведённого типа  $V_q = (A, Q, B, \varphi, \psi, q)$  называется произвольное множество  $\{\alpha_1, \dots, \alpha_n\}$  слов в алфавите  $A$ , удовлетворяющее следующим условиям:

- $\{\varphi(q, \alpha_i) | i = 1, 2, \dots, n\} = Q$ ;
- если  $i \neq j$ ,  $i, j \in \{1, 2, \dots, n\}$ , то  $\varphi(q, \alpha_i) \neq \varphi(q, \alpha_j)$ ;
- если  $\alpha_i = \alpha'_i a$ ,  $a \in A$ , то  $\alpha'_i \in \{\alpha_1, \dots, \alpha_n\}$ ,  $i = 1, \dots, n$ .

**Лемма 12.** Каждый конечный инициальный автомат приведённого типа имеет хотя бы один базис достижимости.

*Доказательство.* В лоб. □

**Определение 51.** Базисом отличимости автомата приведённого типа  $V_q = (A, Q, B, \varphi, \psi)$  называется непустое множество  $\{\alpha_1, \dots, \alpha_n\}$  слов в алфавите  $A$ , удовлетворяющее следующим условиям:

- $\forall q, q' \in V \exists i \in \{1, 2, \dots, n\} : \bar{\psi}(q, \alpha_i) \neq \bar{\psi}(q', \alpha_i)$ ;
- $\alpha_i = a\alpha'_i \Rightarrow \alpha_i \in \{\alpha_1, \dots, \alpha_n\}$ ;
- множество  $\{\alpha_1, \dots, \alpha_n\}$  минимально.

**Лемма 13.** Каждый конечный инициальный автомат приведённого типа имеет базис отличимости, число слов в котором не более  $|Q|$ .

*Доказательство.* Через леммы предыдущего билета. □

**Теорема 18.** Пусть  $M_1$  - базис достижимости конечного инициального автомата  $V_{q_1}$  приведённого вида  $M_2$  — базис отличимости автомата  $V$ ,  $V_{q_1} = (A, Q, B, \varphi, \psi, q_1)$ ,  $|Q| \leq n$ . Инициальный автомат  $V'_{q'_1}$  приведённого вида  $V'_{q'_1} = (A, Q', B, \varphi', \psi', q'_1)$ , имеющий не более, чем  $n$  состояний, отличим от  $V_{q_1}$  тогда и только тогда, когда он отличим от  $V_{q_1}$  множеством  $M_1 A^{n-|Q|+1} M_2$ .

Внимание! Пример достижимости этой оценки не приведён. Возможно стоит рассказывать здесь предыдущий билет, для которого такой пример есть.

## 11 Билет 11

### 11.1 Вопрос 1. Семантические сети. Понятие сущности. Семантические отношения и их виды. Лингвистические, логические, теоретико-множественные, квантификационные отношения. Абстрактные и конкретные семантические сети. Фреймы — системно-структурное описание предметной области. Принципы фрейм-представлений. Понятие «СЛОТА».

В самом общем случае сетевая модель — это информационная модель предметной области. В сетевой модели представляются множество информационных единиц (объекты и их свойства, классы объектов и их свойств) и отношения между этими единицами.

Обычно сетевая модель представляется в виде графа, вершины которого соответствуют информационным единицам, а дуги — отношениям между ними.

В зависимости от типов отношений между информационными единицами различают сети:

- классификационные (отношения типа часть-целое, род, вид, индивид); В классификационных сетях используются отношения, позволяющие описывать структуру предметной области, что позволяет отражать в базах знаний разные иерархические отношения между информационными единицами.
- функциональные (преобразование информационных единиц); Функциональные сети часто называют вычислительными моделями, т.к. они позволяют описывать процедуры "вычислений" одних информационных единиц через другие.
- каузальные (причинно-следственные отношения); В каузальных сетях, называемых также сценариями, используются причинно-следственные отношения, а также отношения типов "средство — результат" "орудие — действие" и т.п.
- смешанные (использующие разнообразные типы отношений).

**Фреймы — системно-структурное описание предметной области. Принципы фрейм-представлений. Понятие «СЛОТА».** Фреймы — это фрагменты знания, предназначенные для представления стандартных ситуаций. Фрейм — это модель знаний, которая активизируется в определенной ситуации и служит для ее объяснения и предсказания.

Фреймы объединяются в сеть, называемую системой фреймов. Характерными для этого подхода являются:

- представление знаний в виде достаточно крупных, содержательно завершенных единиц, называемых фреймами;

- иерархическая структура фреймов, где иерархия основана на степени абстрактности фреймов;
- совмещение в фреймах декларативных и процедурных знаний.

В качестве значения слота может выступать имя другого фрейма; так образуют сети фреймов. Различают фреймы-образцы, или прототипы, хранящиеся в базе знаний, и фреймы-экземпляры, которые создаются для отображения реальных ситуаций на основе поступающих данных.

## 11.2 Вопрос 2. Понятие модели данных. Иерархическая, сетевая модели данных, сравнительный анализ, противоречия и парадоксы. Реляционная модель данных. Операции над отношениями: селекция, проекция, естественное соединение. Модель данных «сущность-связь».

Минимальным фрагментом данных является *элемент данных* (поле, элемент). Элемент данных не может подразделяться на меньшие типы данных, не теряя при этом смысла для пользователей. Сам по себе элемент данных ничего не представляет. Он приобретает смысл только тогда, когда он связан с другими элементами данных. База данных состоит из элементов данных и связей между ними. В базе много различных типов элементов данных, и поэтому необходима специальная схема, позволяющая изобразить связи между типами элементов данных. Эта схема иногда называется **моделью данных**.

В иерархической и сетевой модели есть записи, некоторые поля которых могут ссылаться на другие записи. Разница в том, что первая модель представляет дерево, а вторая — произвольный ориентированный граф.

**Реляционная модель данных. Операции над отношениями: селекция, проекция, естественное соединение.** Реляционная модель строится на понятии отношения и операций с ними (реляционная алгебра).

**Определение 52.**  *$n$ -арным отношением  $R$ , или отношением  $R$  степени  $n$ , называют подмножество декартова произведения множеств  $D_1, D_2, \dots, D_n (n \geq 1)$ , не обязательно различных. Исходные множества  $D_1, D_2, \dots, D_n$  называют в модели доменами.*

Реляционная алгебра представляет собой набор таких операций над отношениями, что результат каждой из операций также является отношением. Это свойство алгебры называется замкнутостью.

Операции реляционной алгебры:

- Переименование
- Объединение

- Пересечение
- Вычитание
- Декартово произведение
- Выборка (ограничение, селекций)

Отношение с тем же заголовком, что и у отношения  $A$ , и телом, состоящим из кортежей, значения атрибутов которых при подстановке в условие  $s$  дают значение ИСТИНА.  $s$  представляет собой логическое выражение, в которое могут входить атрибуты отношения  $A$  и/или скалярные выражения.

Синтаксис:  $A$  WHERE  $s$

- Проекция

Отношение с заголовком  $(X, Y, \dots, Z)$  и телом, содержащим множество кортежей вида  $(x, y, \dots, z)$ , таких, для которых в отношении  $A$  найдутся кортежи со значением атрибута  $X$  равным  $x$ , значением атрибута  $Y$  равным  $y$ , ..., значением атрибута  $Z$  равным  $z$ . При выполнении проекции выделяется «вертикальная» вырезка отношения-операнда с естественным уничтожением потенциально возникающих кортежей-дубликатов.

- Соединение

Операция соединения есть результат последовательного применения операций декартового произведения и выборки. Если в отношениях имеются атрибуты с одинаковыми наименованиями, то перед выполнением соединения такие атрибуты необходимо переименовать.

- Деление

**Модель данных «сущность-связь».** Будем считать, что любая предметная область представляет собой совокупность некоторых объектов реального мира, которые взаимодействуют между собой. В терминах ER-модели объекты реального мира называются сущностями (англ. «entity»). Для отражения в модели взаимодействия сущностей между собой используются связи (англ. «relationship»).

Однородные сущности, т.е. сущности, обладающие одинаковыми характеристиками, объединяются в множества сущностей (англ. entities set). Характеристики сущностей являются атрибутами (англ. attribute). Имеет место понятие множества значений (англ. values set), которые могут принимать эти характеристики. Формально атрибут определяется, как отображение из множества сущностей в множество значений. Элементы одного множества сущностей имеют одинаковые атрибуты. Каждое множество сущностей имеет уникальное имя.

Связи между элементами одних множеств сущностей, обладающие одинаковой семантикой (смыслом), объединяются в множество связей. Следует отметить, что степень связи,

т.е. количество сущностей, входящих в связь, не ограничено. Связи могут строиться и на элементах одного множества сущностей. Характеристики связей определяются атрибутами связей, которые являются отображениями из множества связей в множества значений. У множеств связей атрибутов может и не быть. Очевидно, что элементы одного множества связей имеют одинаковые атрибуты. Семантическое значение сущности в связи называют ее ролью. Естественно, что роли соответствующих сущностей в элементах одного множества связей совпадают. Понятие роли чисто вспомогательное, существующее для удобства пользователя, поэтому роли зачастую не определяются.

Таким образом, схема базы данных в ER-модели представляет собой множество множеств сущностей с определенным для каждого из них конечного непустого множества атрибутов и множество множеств связей, для каждого из которых определено конечное (может быть пустое) множество атрибутов, а также упорядоченный набор множеств сущностей, на которых строятся связи и для каждой из которых может быть определена роль.

### 11.3 Вопрос 3. Конечные автоматы как акцепторы. Регулярные множества ирегулярные выражения. Теорема Клини. Конечные автоматы как свержакцепторы. Теорема Мак-Нотона.

(Кудрявцев и Ко. Введение в теорию автоматов. стр 91)

**Определение 53.** Пусть  $V_q = (A, Q, B, \varphi, \psi, q)$  - инициальный конечный автомат,  $B' \subset B$ . Множество  $M = \{\alpha | \alpha \in A^*, \psi(q, \alpha) \in B'\}$  называется представимым в автомате  $V_q$  с помощью подмножества  $B'$  выходных символов. Говорят, что автомат  $V_q$  представляет  $M$  посредством  $B'$ . Подмножества  $A^* \setminus \Lambda$  называются событиями в  $A$ . Если существует конечный автомат  $V_q$  представляющий событие  $M$ , то  $M$ -представимо.

Определяются понятия произведения событий и итерации события  $M^+$ .

**Лемма 14.** Соотношение  $X = XC \cup D$  выполняется тогда и только тогда, когда  $X = DC^+ \cup D$ .

**Определение 54.** Событие регулярно, если его можно получить из событий вида  $\emptyset, \{a\}$ ,  $a \in A$ , применением конечного числа операций  $\cup, \times, ^+$ .

**Лемма 15.** Пусть  $R_{ij}; i = 0, 1, \dots, n, j = 0, 1, \dots, n$ , — регулярные события, события  $X_1, \dots, X_n$  удовлетворяют системе:

$$\begin{cases} X_1 = X_1 R_{11} \cup \dots \cup X_n R_{n1} \cup R_{01} \\ \dots \\ X_n = X_1 R_{1n} \cup \dots \cup X_n R_{nn} \cup R_{0n} \end{cases}$$

Тогда события  $X_1, \dots, X_n$  регулярны.

**Лемма 16.** Все события, представимые конечными автоматами, регулярны.

**Определение 55.** *Обобщённый источник — конечный ориентированный граф с выделенными начальной и конечной вершинами, и приписанной буквой или пустым словом к каждому ребру. Можно брать пути  $\pi$  и рассматривать соответствующие слова  $[\pi] = \alpha$ . Множество вершин, до которых можно добраться по слову от текущей вершины:  $\theta(u, \alpha)$ . Каждый обобщённый источник  $G$  с начальной вершиной  $u$  и конечной вершиной  $w$  определяет событие  $|G| = \{\alpha \mid \alpha \in A^*, \alpha \neq \Lambda, w \in \theta(v, \alpha)\}$ .*

**Лемма 17.** *Если  $R$  событие регулярно, то существует обобщённый источник  $G$ :  $|G| = R$ .*

**Лемма 18.** *Если  $G$  — обобщённый источник, то событие  $|G|$  представимо.*

**Теорема 19 (Клини).** *Событие  $E$  в алфавите  $A$  представимо тогда и только тогда, когда оно регулярно.*

**Определение 56.** *Предел  $\lim(\alpha)$  сверхслова  $\alpha, \alpha \in A^\infty, \alpha = a(1)a(2)\dots$ , есть множество таких символов  $a \in A$ , что для некоторой бесконечной последовательности  $i_1, i_2, \dots$  выполняется:  $a(i_{j_i}) = a$ .*

**Определение 57.** *Пусть  $V_q = (A, Q, B, \varphi, \psi, q)$  — конечный автомат. Для произвольного сверхслова  $\alpha$  в алфавите  $A, \alpha = a(1)a(2)\dots$ , обозначим через  $\bar{\psi}(q, \alpha)$  такое сверхслово  $\beta = b(1)b(2)\dots$ , что  $b(i) = \psi(q, a(1)\dots a(i)), i = 1, 2, \dots$ .*

**Определение 58.** *Пусть  $B_1, \dots, B_k$  — некоторые попарно различные подмножества  $B$ . Множество  $M = \{\alpha \mid \alpha \in A^\infty, \lim(\psi(q, \alpha)) \in \{B_1, \dots, B_k\}\}$  называется представимым в автомате  $V_q$  с помощью семейства  $\{B_1, \dots, B_k\}$ . Подмножества  $A^\infty$  называются сверхсобытиями.*

Операции на сверхсловах:

- произведение события  $M_1$  на сверхсобытие  $M_2$  в том же алфавите;
- сверхитерация события  $M$ ;

Общерегулярные сверхсобытия в алфавите  $A$ :

- если  $R$  — регулярное событие в алфавите  $A$ , то  $R^\infty$  есть общерегулярное сверхсобытие в алфавите  $A$ ;
- если  $R_1$  — регулярное событие в алфавите  $A, R_2$  - общерегулярное сверхсобытие, то  $R_1 R_2$  есть общерегулярное сверхсобытие в алфавите  $A$ ;
- если  $R_1, R_2$  — общерегулярные сверхсобытия в алфавите  $A$ , то  $R_1 \cup R_2$  есть общерегулярное сверхсобытие в алфавите  $A$ ;
- общерегулярность любого события устанавливается предыдущими операциями за конечное число шагов.

**Определение 59.** *Обозначим за  $U(\alpha)$  множество символов, входящих в  $\alpha$ .*

**Лемма 19.** Пусть  $V_q = (A, Q, B, \varphi, \psi, q)$  — конечный автомат,  $q_1 \in Q$ ,  $B' \subset B$ . Тогда множество  $M$  всех непустых слов  $\alpha$  в алфавите  $A$ , для которых  $\varphi(q_1, \alpha) = q_1$  и  $U(\bar{\psi}(q_1, \alpha)) = B'$  является регулярным.

**Лемма 20.** Если сверхсобытие представимо, то оно общерегулярно.

**Определение 60.** Сверхисточник — конечный ориентированный граф с выделенными начальной  $v$  и множеством финальных вершин  $\{v_1, \dots, v_n\}$ , и приписанной буквой или пустым словом к каждому ребру. Начало может быть концом. Можно брать бесконечные сверхпути  $\pi$  и рассматривать соответствующие сверхслова  $[\pi] = \alpha$ . Вершина  $w$  называется предельной вершиной сверхпути  $\pi$  ( $w \in \lim(\pi)$ ), если  $w = w_i$  для бесконечного множества индексов  $i$ . Каждый сверхисточник  $G$  определяет сверхсобытие  $R$ , состоящее из всех таких сверхслов  $\alpha$ ,  $\alpha \in A^\infty$ , что для некоторого сверхпути  $\pi$  с началом в  $v$  выполняется  $[\pi] = \alpha$  и хотя бы одна из финальных вершин сверхисточника  $G$  принадлежит множеству  $\lim(\pi)$ . Сверхсобытие  $R$  будем обозначать за  $\|G\|$

**Лемма 21.** Если сверхсобытие  $R$  в алфавите  $A$  общерегулярно, то существует такой сверхисточник  $G$ , что  $R = \|G\|$ .

**Лемма 22.** Если  $G$  — сверхисточник, то сверхсобытие  $\|G\|$  в алфавите  $A$  представимо.

**Теорема 20** (Мак-Нотана). Сверхсобытие является представимым тогда и только тогда, когда оно общерегулярно.

## 12 Билет 12

### 12.1 Вопрос 1. Продукционные системы представления знаний. Канонические системы Поста. Редукционные системы. Синтез плана решения задач с автоматическим построением редукционной модели.

Продукционная модель — модель, основанная на правилах. Продукционная система — набор продукций, организованный по определенному принципу.

База знаний состоит из набора правил типа: Если (условие), то (действие). Машина вывода — программа, перебирающая правила из базы. Данные — это исходные факты, на основании которых запускается машина вывода

Вывод бывает прямой (от данных к поиску цели) и обратный (от цели для ее подтверждения — к данным).

Пример. База знаний:

- если «отдых — летом» и «человек — активный», то «ехать в горы»;
- если «любит солнце», то «отдых летом».

Данные: «человек активный» и «любит "солнце»ю

В общем случае продукционную модель можно представить в следующем виде:

$i = \langle S; L; A \rightarrow B; Q \rangle$ , где:

- $S$  — описание класса ситуаций;
- $L$  — условие, при котором продукция активизируется;
- $A \rightarrow B$  — ядро продукции;
- $Q$  — постусловие продукционного правила.

Здесь  $i$  - имя продукции, с помощью которого данная продукция выделяется из всего множества продукций. В качестве имени может выступать некоторая лексема, отражающая суть данной продукции (например, «покупка книги»), или порядковый номер продукций в их множестве, хранящимся в памяти системы.

Элемент  $S$  характеризует сферу применения продукции. Такие сферы легко выделяются в когнитивных структурах человека. Наши знания как бы «разложены по полочкам». На одной полочке хранятся знания о том, как надо готовить пищу, на другой как добраться до работы, и т.п. Разделение знаний на отдельные сферы позволяет экономить время на поиск нужных знаний. Такое же разделение на сферы в базе знаний ИИ целесообразно и при использовании для представления знаний продукционных моделей.

Основным элементом продукции является ее ядро:  $A \rightarrow B$ . Интерпретация ядра продукции может быть различной и зависит от того, что стоит слева и справа от знака секвенции  $\rightarrow$ . Обычное прочтение ядра продукции выглядит так: ЕСЛИ  $A$ , ТО  $B$ , более сложные конструкции ядра допускают в правой части альтернативный выбор, например, ЕСЛИ  $A$ , ТО  $B_1$ , ИНАЧЕ  $B_2$ . Секвенция может истолковываться в обычном логическом смысле как знак логического следования  $B$  из истинного  $A$  (если  $A$  не является истинным выражением, то о  $B$  ничего сказать нельзя). Возможны и другие интерпретации ядра продукции, например  $A$  описывает некоторое условие, необходимое для того, чтобы можно было совершить действие  $B$ .

Элемент  $L$  есть условие применимости ядра продукции. Обычно  $L$  представляет собой логическое выражение (как правило предикат). Когда  $L$  принимает значение *истина*, ядро продукции активизируется. Если  $L$  *ложно*, то ядро продукции не может быть использовано.

Элемент  $Q$  описывает постусловия продукции. Они актуализируются только в том случае, если ядро продукции реализовалось. Постусловия продукции описывают действия и процедуры, которые необходимо выполнить после реализации. Выполнение  $Q$  может происходить сразу после реализации ядра продукции.

Продукционная модель чаще всего применяется в промышленных экспертных системах. Преимущества: наглядность, высокая модульность, лёгкость внесения дополнений и изменений, простота механизма логического вывода.

Если попытаться охарактеризовать эту область, то о ней можно сказать, что производственные системы эффективны в тех случаях, когда процесс решения проблемы может быть представлен как поведение, опирающееся на ограниченный контекст. При необходимости учета сложных смысловых связей для решения проблемы производственные системы быстро теряют свою эффективность из-за быстрого роста их объема и сложности управления ими.

В ряде ИС используется комбинация сетевых и производственных моделей (производственная система над семантической сетью). Декларативные знания описываются в сетевом компоненте, а процедурные знания — в производственном.

**Канонические системы Поста.** Каноническая система Поста задаётся четвёркой  $A, P, \mathcal{A}, p$ , где  $A$  - алфавит исчисления,  $P$  (не имеющий общих букв с  $A$ ) - алфавит переменных,  $\mathcal{A}$  - список слов в (аксиом исчисления),  $p$  - список правил вывода вида:

$$\begin{array}{cccccc} G_{1,1} & p_{1,1} & \dots & G_{1,n_1} & p_{1,n_1} & G_{1,n_1+1} \\ G_{2,1} & p_{2,1} & \dots & G_{2,n_2} & p_{2,n_2} & G_{2,n_2+1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ G_{m,1} & p_{m,1} & \dots & G_{m,n_m} & p_{m,n_m} & G_{m,n_m+1} \\ \hline G_1 & p_1 & \dots & G_n & p_n & G_{n+1} \end{array},$$

где  $G_{ij}$  суть обозначения слов в  $A$ ,  $p_{i,j}$  - обозначения букв из  $P$ . Слово  $Q$  получается из  $Q_1, \dots, Q_m$  применением правила, если для каждой входящей в него буквы из  $P$  можно подобрать слово в  $A$  (значение этой переменной), подставляя к-рое вместо всех вхождений рассматриваемой переменной, мы получим после такого замещения всех переменных слова  $Q_1, \dots, Q_m$  - над чертой и  $Q$  - под чертой.

**Представление неформальных знаний. Редукционные системы. Синтез плана решения задач с автоматическим построением редукционной модели.** Решение задач методом редукции часто приводит к хорошим результатам потому, что часто решение задач имеет иерархическую структуру.

Цель состоит в том, чтобы представить сложную задачу как совокупность более простых относительно независимо решаемых задач.

Редукции также называются переписыванием. Основные свойства систем переписывания можно сформулировать не прибегая к конкретной реализации их в виде операций над терминами. Для этого часто используется понятие «Абстрактной Системы Редукций» или «ARS». ARS состоит из некоторого множества  $A$  и набора бинарных отношений  $(\rightarrow_\alpha)_{\alpha \in I}$ , на нём, которые называются редукциями. Говорят, что  $a$  редуцируется, или переписывается в  $b$  в один шаг относительно данной ARS, если пара  $(a, b)$  принадлежит некоторому  $\rightarrow_\alpha$ .

Важнейшими свойствами редукционных систем являются:

- Конфлюентность — если  $a$  может за некоторое число шагов редуцироваться как в  $b$

так и в  $c$ , то существует элемент  $d$ , в который могут редуцироваться оба  $b$  и  $c$ .

- Остановочность — любая цепочка одношаговых редукций всегда конечна, то есть достигается элемент, который не может больше быть редуцирован.
- Локальная (или слабая) конфлюентность — то же, что и конфлюентность, но при условии, что  $a$  переписывается в  $b$  и  $c$  ровно за один шаг.
- Слабая остановочность — для каждого элемента существует обрывающаяся цепочка его последовательных редукций.
- Каноничность или свойство Чёрча-Россера — конфлюентность плюс остановочность.

## 12.2 Вопрос 2. Языковые средства информационных технологий. Входные и внутренние языки. Структура входных языков. Языковые средства для ввода и обновления информации, для поиска, обобщения и выдачи информации. Языковые средства общения с БД.

Языковые средства информационных технологий — наборы символов, соглашений и правил, которые используются для организации взаимодействия человека со средствами, реализующими информационный процесс управленческой деятельности и описания таких процессов.

Далее, про SQL.

## 12.3 Вопрос 3. Автоматы с магазинной памятью. Языки, допускаемые магазинным автоматом. Эквивалентность множеств автоматов с магазинной памятью и контекстно-свободных грамматик. Детерминированные автоматы с магазинной памятью.

(Джон Хопкрофт и Ко, Введение в теорию автоматов, языков и вычислений)

**Определение 61.** Автомат с магазинной памятью  $M = (Q, A, \Gamma, \pi, q_0, Z_0, F)$ , где:

- $Q$  — конечное множество состояний автомата;
- $A$  — конечный допустимый входной алфавит, из которого формируются строки, считываемые автоматом;
- $\Gamma$  — конечный алфавит памяти (магазина);
- $\pi$  — конечное подмножество  $Q \times (A \cup \{\varepsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$  — функция перехода;

- $q_0 \in Q$  — единственно допустимое начальное состояние автомата;
- $Z_0 \in \Gamma$  — начальный магазинный символ (маркер дна);
- $F \subset Q$  — множество конечных состояний, причём допустимо  $F = \emptyset$ , и  $F = Q$ .

Слово допускается, если при его применении слово переходит в заключительное состояние.

Память работает как стек, то есть для чтения доступен последний записанный в неё элемент. Функция перехода по комбинации текущего состояния, входного символа (или пустого слова, если читать входную последовательность не требуется) и символа на вершине магазина выбирает следующее состояние и символы для записи в магазин (или пустое слово). В случае, когда в правой части автоматного правила присутствует  $\varepsilon$ , в магазин ничего не добавляется, а элемент с вершины стирается. Если магазин пуст, то срабатывают правила с  $\varepsilon$  в левой части.

**Определение 62.** Конфигурацией автомата является тройка  $(q, w, \gamma)$ , где

- $q \in Q$  — текущее состояние автомата;
- $w \in A^*$  — суффикс входной цепочки, начинающийся с позиции указателя;
- $\gamma \in \Gamma$  — содержимое стека.

**Определение 63.** Контекстно-свободная грамматика задаётся следующим набором:

- $\Sigma$  — набор терминальных символов;
- $N$  — набор нетерминальных символов;
- $P$  — набор правил вида «левая часть»  $\rightarrow$  «правая часть», где левая часть есть нетерминал, а правая часть произвольный набор терминалов и нетерминалов;
- $S$  — стартовый символ.

Язык распознаётся грамматикой, если применением правил можно получить все слова языка и только их.

**Теорема 21.** Язык допускается МПА тогда и только тогда, когда он может быть представлен с помощью контекстно-свободной грамматики.

*Доказательство.* (Джон Хопкрофт и Ко, Введение в теорию автоматов, языков и вычислений, стр. 251) От КС к Автоматам. Входной алфавит — алфавит терминалов грамматики. Состояние — одно. Алфавит стека — объединение множеств терминалов и нетерминалов. Изначально в стеке лежит  $S$ . Алгоритм:

- если в стеке нетерминал, то он расширяется и кладётся в стек, входные символы не читаются;

- если в стеке терминал, то он сравнивается с входным символом: если совпал, то выкидываем входной символ и символ стека и продолжаем, если не совпал (или если входные символы закончились) - обрываем цепочку;
- если закончился стек и вход пуст — успех.

Доказательство эквивалентности полученных языков по индукции по числу операций вывода.

От Автомата к КС. Грамматика  $G = (V, \Sigma, R, S)$ :

- Множество  $V$  состоит из  $S$  и всех символов вила  $[pXq]$ , где  $p$  и  $q$  — состояния из  $Q$ , а  $X$  — магазинный символ из  $\Gamma$ .
- Продукции:
  - продукция  $S \rightarrow [q_0Z_0p]$  для всех состояний  $p$ .
  - пусть  $\pi(q, a, X)$  содержит пару  $(r, Y_1Y_2 \dots Y_k)$ , где  $a$  есть либо символ из  $\Sigma$ , либо  $\varepsilon$ , а  $k$  — некоторое неотрицательно число; при  $k = 0$  пара имеет вид  $(r, \varepsilon)$ . Тогда для всех списков состояний  $r_1, r_2, \dots, r_k$  в грамматике  $G$  есть продукция

$$[qXr_k] \rightarrow a[rY_1r_1][r_1Y_2r_2] \dots [r_{k-1}Y_kr_k]$$

□

**Определение 64.** Если в определении потребовать, что для каждой конфигурация входит в функцию перехода не более одного раза, то получим детерминированный автомат с магазинной памятью.

**Утверждение 12.** Регулярные языки распознаются ДМПА.

**Утверждение 13.** Если алфавит  $A$  содержит более одного символа, то язык  $L = \{a_1 \dots a_n a_n \dots a_1 \mid a_i \in A\}$  распознаётся некоторым НМПА, но не распознаётся никаким ДМПА.

## 13 Билет 13

### 13.1 Вопрос 1. Обработка данных. Структура данных.

**Определение 65.** Структура данных — программная единица, позволяющая хранить и обрабатывать множество однотипных и/или логически связанных данных.

Структуры данных бывают физическими и абстрактными.

Физические структуры данных разделяются на статические (массив) и динамические (список, каждый элемент содержит данные и связку).

Абстрактные структуры данных — интерфейс для операций с хранимыми объектами, скрывающий детали реализации от пользователя. Они могут быть реализованы с помощью различных физических структур, как массивов, так и списков. Пример: стек, очередь FIFO, дек, словарь

Абстрактные структуры данных предназначены для удобного хранения и доступа к информации. Они предоставляют удобный интерфейс для типичных операций с хранимыми объектами, скрывая детали реализации от пользователя. Пример: Языки программирования высокого уровня (Паскаль, Си..) предоставляют удобный интерфейс для чисел: операции  $+$ ,  $*$ ,  $=$  .. и т.п, но при этом скрывают саму реализацию этих операций, машинные команды.

Конкретные реализации АД называются структурами данных. В программировании абстрактные типы данных обычно представляются в виде интерфейсов, которые скрывают соответствующие реализации типов. Программисты работают с абстрактными типами данных исключительно через их интерфейсы, поскольку реализация может в будущем измениться. Такой подход соответствует принципу инкапсуляции в объектно-ориентированном программировании. Сильной стороной этой методики является именно сокрытие реализации. Раз вонне опубликован только интерфейс, то пока структура данных поддерживает этот интерфейс, все программы, работающие с заданной структурой абстрактным типом данных, будут продолжать работать. Разработчики структур данных стараются, не меняя внешнего интерфейса и семантики функций, постепенно дорабатывать реализации, улучшая алгоритмы по скорости, надежности и используемой памяти.

Примеры АД: список, стек, очередь, ассоциативный массив, очередь с приоритетом.

### **13.2 Вопрос 2. Языковые средства документальных (в том числе полнотекстовых) ИПС: три уровня грамматики информационно-поисковых языков (теоретико-множественный, линейный, сетевой). Информационно-поисковой язык. Язык процедурно ориентированный. Непроцедурный язык концептуального уровня. Язык диалога. Естественный язык. Словарный комплекс АИС. Классификаторы. Кодификаторы. Тезаурусы: состав и структура. Языки описания данных и словарь данных. Языки запросов SQL и QBE.**

**Языки запросов SQL и QBE.** Эксплуатационным преимуществом поиска QBE (Query By Example) является то, что для формирования запроса не требуется использовать специализированный язык запросов, синтаксис которого может быть сложен и недоступен конечному пользователю. Пользователю выводится окно, в котором указаны все поля данных, встречающиеся в каждой записи данных; введение информации в конкретное поисковое поле ограничит поиск совпадением (полным или частичным, в зависимости от

договорённости реализации) по данному полю. Проверка условий осуществляется только по заполненным условиям на поля, а поля, условия на которые указаны не будут, могут соответствовать чему угодно. Многие практические реализации QBE допускают также не только конъюнктивное соединение условий в заполненных полях, но и другие варианты соединения условий (например, дизъюнкцию, отрицание, существование или несуществование связанных записей и другие).

### 13.3 Вопрос 3. Структурные автоматы, операции суперпозиции и композиции. Схемы в базисе из булевых функций и «задержки». Оператор замыкания. Проблема полноты и выразимости.

(Кудрявцев и Ко. Введение в теорию автоматов. стр 151)

С каждым инициальным автоматом  $V$  можно связать о.-д. функцию  $F_V = \bar{\psi}(q_0, \alpha)$ .

Можно считать, что есть некоторый *основной* алфавит  $T$ , которыми закодированы  $A, Q, B$ . Будет считать, что  $T = \{0, 1\}$ ,  $A = T^n$ ,  $Q = T^r$ ,  $B = T^m$ , для некоторых  $n, r, m$ . Заметим, что слова из  $\alpha \in (T^s)^*$  можно рассматривать как набор из  $s$  слов  $(\alpha_1, \dots, \alpha_s)$  из  $T^*$ , причём все  $\alpha_i$  одинаковой длины. С этой точки зрения любой инициальный автомат реализует словарную вектор-функцию  $F_V(x_1, \dots, x_n) = (f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n))$ , такую, что  $F_V(\alpha_1, \dots, \alpha_n) = (\beta_1, \dots, \beta_m)$  и все  $\alpha_i, \beta_j$  — одинакового длины.

Будем рассматривать функции с  $m = 1$ , т.е. функции вида  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ . Тогда автомат задаётся системой булевых функций:

$$\begin{cases} \varphi_i(q_1, \dots, q_r, a_1, \dots, a_n) = q'_i, & i = 1, \dots, r, \\ \psi(q_1, \dots, q_r, a_1, \dots, a_n) = b. \end{cases}$$

Если переписать это всё через  $t$ , то получится каноническая система о.-д. функции.

Пусть у нас есть бесконечный алфавит переменных, и о.д. функции записанные в указанных выше обозначениях. Тогда можно рассматривать следующие операции на функциях (и на соответствующих автоматах).

- Добавление фиктивной переменной.
- Изъятие фиктивной переменной. Здесь стоит заметить, что даже если  $x_i$  - фиктивная, то  $\psi$  может существенно зависеть от  $x_i$ :  $\psi(q_1, \dots, q_r, x_1, \dots, 0 \dots x_n) \neq \psi(q_1, \dots, q_r, x_1, \dots, 1 \dots x_n)$  для некоторых  $q_j$  и  $x_j$ . Но в таком случае данное состояние не достигается, и можно переопределить функцию  $\psi$  константой на таких наборах так, чтобы она существенно не зависела от  $x_i$ . Аналогично поступим с функциями  $\varphi_j$ .
- Отождествление переменных.
- Переименование переменных (без отождествления).

- Подстановка. Пусть  $f(x_1, \dots, x_i, \dots, x_n)$  и  $h(x'_1, \dots, x'_m)$  — о.д. функции, причём множества их переменных не пересекаются. Тогда можно выполнить подстановку:

$$g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n, x'_1, \dots, x'_m) = f(x_1, \dots, h(x'_1, \dots, x'_m), \dots, x_n)$$

- Операция обратной связи. Пусть функция  $f(x_1, x_2, \dots, x_n)$  зависит от  $x_1$  со сдвигом. Тогда аналогично операции изъятия фиктивного переменная можно считать, что  $\psi$  не зависит существенно от  $x_1$ . Тогда переменную  $x_1$  можно изъять из  $\psi$  и полученную функцию от  $x_2, \dots$  подставить в  $\phi_i$  вместо  $x_1$

Первые 5 операций называются операциями суперпозиции.

**Определение 66.** Пусть  $M \subset P$  некоторое множество о.-д. функций. Обозначим за  $\Sigma(M)$  множество функций, получаемых из  $M$  с помощью операций суперпозиции за конечное число шагов, а за  $K(M)$  — с помощью операций суперпозиции и обратной связи.

$f$  — суперпозиция над  $M \Leftrightarrow f \in \Sigma(M)$

$f$  — композиция над  $M \Leftrightarrow f \in K(M)$

Автомат можно рассматривать как схемку с  $n$  ножками-входами и  $m$  ножками-выходами. Операции над ними аналогичны операциям для о.-д. функций.

**Определение 67.** Пара  $(S, V)$ , где  $S$  — схема над  $E_M$ ,  $V$  — инициальный автомат, вычисляющий функцию, сопоставленную схеме  $S$ , называется структурным автоматом над  $E_M$ .

**Определение 68.** Оператор  $O$ , определённый на множестве всех подмножеств  $P$ , называется оператором замыкания, если он удовлетворяет следующим свойствам:

- $O(M) \supset M$ ;
- $O(O(M)) = O(M)$ ;
- если  $M_1 \supset M_2$ , то  $O(M_1) \supset O(M_2)$ .

Для любого оператора замыкания  $I$  естественно определяются понятия  $I$ -замкнутого множества и  $I$ -полного подмножества.

**Теорема 22.**  $K(\{G_0(x), f_{\vee}(x_1, x_2), f_{\&}(x_1, x_2), f_{-}(x_1, x_2)\}) = P$

*Доказательство.* Вначале докажем, что любая истинностная о.д. функция (без памяти) вычисляется суперпозицией  $\{f_{\vee}(x_1, x_2), f_{\&}(x_1, x_2), f_{-}(x_1, x_2)\}$  — это просто.

Общий случай: (Кудрявцев и Ко. Введение в теорию автоматов. стр 169)

□

## 14 Билет 14

### 14.1 Вопрос 1. Системы управления базы данных. Архитектура СУБД. Основные конструкции структур данных. Функции СУБД. Категории пользователей.

**Определение 69.** *База данных:*

- *структурированный организованный набор данных, описывающий характеристики каких-либо физических или виртуальных систем;*
- *структурированная организованная совокупность блоков информационных элементов, представленных на машиночитаемых носителях, характеризующих актуальное состояние некоторой предметной области и предназначенных и пригодных для оперативного решения пользовательских задач с использованием средств вычислительной техники.*

Обращение к базам данных осуществляется с помощью системы управления базами данных (СУБД).

**Определение 70.** *Система управления базами данных — комплекс программных и лингвистических средств, предназначенный для создания, ведения и эксплуатации баз данных многими пользователями.*

Основные функции СУБД:

- управление данными во внешней памяти;
- управление данными в оперативной памяти;
- журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
- поддержка языков БД (язык определения данных, язык манипулирования данными).

Компоненты СУБД:

- ядро (отвечает за управление данными во внешней и оперативной памяти и журнализацию);
- процессор языка базы данных;
- подсистема поддержки времени исполнения (интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД);
- сервисные программы (внешние утилиты) (обеспечивают ряд дополнительных возможностей по обслуживанию информационной системы);

- словарь данных содержит информацию о структуре БД;
- индексы служат для быстрого поиска данных с конкретными значениями (атрибу- тами);
- данные.

Классификация СУБД.

По модели данных:

- Иерархические
- Сетевые
- Реляционные
- Объектно-реляционные
- Объектно-ориентированные

По организации хранения данных:

- локальные (все части локальной СУБД размещаются на одном компьютере)
- распределённые (части СУБД могут размещаться на двух и более компьютерах)

По способу доступа:

- *файл-серверные*

Файлы передаются на рабочие станции, где производится их обработка. На сервере происходит только хранение данных.

В файл-серверных СУБД файлы данных располагаются централизованно на файл-сервере. Ядро СУБД располагается на каждом клиентском компьютере. Доступ к данным осуществляется через локальную сеть. Синхронизация чтений и обновлений осуществляется посредством файловых блокировок. Преимуществом этой архитектуры является низкая нагрузка на ЦП сервера, а недостатком — высокая загрузка локальной сети. На данный момент файл-серверные СУБД считаются устаревшими.

Примеры: Microsoft Access, Paradox, dBase.

- *клиент-серверные*

Сервер обеспечивает не только хранение данных, но и основной объем обработки данных. Спецификой архитектуры клиент-сервер является использования языка запросов SQL. Такие СУБД состоят из клиентской части (которая входит в состав прикладной программы) и сервера. Клиент-серверные СУБД, в отличие от файл-серверных, обеспечивают разграничение доступа между пользователями и мало загружают сеть и клиентские машины. Сервер является внешней по отношению к клиенту программой, и по надобности его можно заменить другим. Недостаток клиент-серверных СУБД в самом факте существования сервера (что плохо для локальных

программ — в них удобнее встраиваемые СУБД) и больших вычислительных ресурсах, потребляемых сервером.

Примеры: Firebird, Interbase, IBM DB2, MS SQL Server, Sybase, Oracle, PostgreSQL, MySQL, ЛИНТЕР, MDBS.

- *встраиваемые*

Встраиваемая СУБД — библиотека, которая позволяет унифицированным образом хранить большие объёмы данных на локальной машине. Доступ к данным может происходить через SQL либо через особые функции СУБД. Встраиваемые СУБД быстрее обычных клиент-серверных и не требуют установки сервера, поэтому востребованы в локальном ПО, которое имеет дело с большими объёмами данных (например, геоинформационные системы).

Примеры: OpenEdge, SQLite, BerkeleyDB, один из вариантов Firebird, один из вариантов MySQL, Sav Zigzag, Microsoft SQL Server Compact, ЛИНТЕР.

Категории пользователей.

Пользователей СУБД можно разбить на три категории:

- *администратор сервера баз данных*

Он ведаёт установкой, конфигурированием сервера, регистрацией пользователей, групп, ролей и т.п. Администратор сервера имеет имя `ingres`. Прямо или косвенно он обладает всеми привилегиями, которые имеют или могут иметь другие пользователи.

- *администраторы базы данных*

К этой категории относится любой пользователь, создавший базу данных, и, следовательно, являющийся ее владельцем. Он может предоставлять другим пользователям доступ к базе и к содержащимся в ней объектам. Администратор базы отвечает за ее сохранение и восстановление. В принципе в организации может быть много администраторов баз данных. Чтобы пользователь мог создать базу и стать ее администратором, он должен получить (вероятно, от администратора сервера) привилегию `creatdb`.

- *прочие (конечные) пользователи*

Они оперируют данными, хранящимися в базах, в рамках выделенных им привилегий.

**14.2 Вопрос 2. Информационный поиск. Основные понятия и виды. Модели поиска. Стратегии поиска. Понятие об ассоциативном поиске. Подготовка запросов и отчётов. Оперативный поиск и регламентный режим поиска. Формирование отчётов.**

См. 1.16

**14.3 Вопрос 3. Мощность полных систем автоматов. Класс автоматов с безусловными переходами, полные системы в этом классе.**

(Кудрявцев и Ко. Введение в теорию автоматов. стр 174)

**Определение 71.** Оператор замыкания  $I$  называется алгебраическим, если для любого множества  $M \subset P$  и любого элемента  $f \in I(M)$  существует конечное подмножество  $\{f_1, \dots, f_s\} \subset M$  такое, что  $f \in I(\{f_1, \dots, f_s\})$ .

**Теорема 23.** Операторы замыкания  $\Sigma$  и  $K$  являются алгебраическими операторами замыкания  $P$ .

**Теорема 24.** Из любой  $K$ -полной системы  $F$  можно выделить конечную полную подсистему

*Доказательство.* Т.к.  $K$  — алгебраический, то из  $F$  можно выделить конечную подсистему, выражающую функции  $G_0(x)$ ,  $f_{\vee}(x_1, x_2)$ ,  $f_{\&}(x_1, x_2)$ ,  $f_{-}(x_1, x_2)$ . Тогда эта подсистема будет полной.  $\square$

**Лемма 23.** О.д. функция  $f$  веса  $n$ ,  $f : A^* \Leftarrow B^*$ , преобразует периодическое сверхслово  $\alpha$ ,  $\alpha \in A^\infty$ , с наименьшей длиной периода  $\tau$  в периодическое сверхслово с наименьшей длиной периода вида  $\theta\tau$ , где  $\theta$  — некоторый делитель числа  $\tau$ ,  $\theta \leq n$ .

*Доказательство.* Пусть  $V_{q_1} = (A, \{q_1, \dots, q_n\}, B, \varphi, \psi, q_1)$  — конечный автомат, для которого  $f(\gamma) = \bar{\psi}(q_1, \gamma)$ .

Рассмотрим произвольное периодическое сверхслово  $\alpha \in A^\infty$ ,  $\alpha = a(1)a(2)\dots$ . Пусть  $f^\infty(\alpha) = \beta$ ,  $\beta = b(1)b(2)\dots$ . Пусть наименьшая длина периода сверхслова  $\alpha$  равна  $\tau$ , длина предпериода  $\tau'$ . Тогда при всех  $i \leq \tau' + 1$  выполняется  $a(i) = a(i + \tau)$ . Обозначим  $\alpha_1 = a(1)\dots a(\tau')$ ,  $\alpha_2 = a(\tau' + 1)\dots a(\tau' + \tau)$ . В последовательность  $\varphi(q_1, \alpha_1\alpha_2), \varphi(q_1, \alpha_1\alpha_2\alpha_2)\dots \varphi(q_1, \alpha_1(\alpha_2)^{n+1})$  существует не более  $n$  различных элементов. Поэтому существуют  $i_1, i_2$ ,  $1 \leq i_1 \leq i_2 \leq n + 1$ , для которых  $\varphi(q_1, \alpha_1(\alpha_2)^{i_1}) = \varphi(q_1, \alpha_1(\alpha_2)^{i_2})$ . Пусть  $q = \varphi(q_1, \alpha_1(\alpha_2)^{i_1})$ ,  $\alpha_3 = \alpha_2^{i_2 - i_1}$ , тогда  $q = \varphi(q, \alpha_3)$ . Обозначим, далее,  $q(i) = \varphi(q_1, a(1)\dots a(i - 1))$ . Если  $i \geq \tau' + i_1\tau + 1$ , то слово  $a(1)\dots a(i - 1)$  можно

представить в виде  $\alpha_1(\alpha_2)^{i_1}(\alpha_3)^j\alpha'_3$ . Тогда имеем:

$$q(i) = \varphi(q_1, \alpha_1(\alpha_2)^{i_1}(\alpha_3)^j\alpha'_3) = \varphi(q, (\alpha_3)^j\alpha'_3) = \varphi(q, \alpha'_3).$$

А значит  $q(i + \tau(i_2 - i_1)) = q(i)$  и

$$b(i + \tau(i_2 - i_1)) = \psi(q(i + \tau(i_2 - i_1)), a(i + \tau(i_2 - i_1))) = \psi(q(i), a(i)) = b(i)$$

Следовательно  $\beta$  периодически в длиной периода  $\tau(i_2 - i_1)$  и длиной предпериода  $\tau' + i_1 * \tau$ . Значит минимальные период удовлетворяет условию леммы.  $\square$

**Теорема 25.** *Всякое  $\Sigma$ -полное множество о.-д. функций бесконечно.*

*Доказательство.* Рассмотрим  $M_n$  — множество всех о.-д. функций с весом не более  $n$  и  $H_n$  — множество всех периодичных последовательностей нулей и единиц, у которых длины периодов раскладываются в произведение простых множителей не больше  $n$ :  $p_1^{s_1} \dots$ .

Докажем, что функции из  $M_n$  сохраняют множество  $H_n$ . Пусть  $f \in M_n$ ,  $f = (f_1, \dots, f_l)$ ,  $\gamma_1, \dots, \gamma_l \in H_n$ . По лемме каждая  $f_i(\gamma_i)$  периодична с периодом  $\theta_i m_i$ , где  $\theta_i$  - делитель  $p_1^{s_{1i}} p_2^{s_{2i}} \dots$ , а  $m_i \leq n$ . Значит  $f$  периодична с периодом  $\prod(\theta_i m_i) = (\prod m_i) p_1^{s_{1i} + s_{2i} + \dots}$ , где все  $m_i, p_i$  не превосходят  $n$ .

Докажем, что функции из  $\Sigma(M_n)$  сохраняют множество  $H_n$ . Применим индукцию по числу суперпозиций. Достаточно проверить переход для случая подстановки. Пусть  $h(x_1, \dots, x_{k_1})$  и  $g(x_{k_1+1}, \dots, x_{k_1+k_2})$  сохраняют  $H_n$ , и пусть  $\gamma_1, \dots, \gamma_{k_1+k_2}$  — произвольные последовательности из  $H_k$ . Рассмотрим функцию

$$f(x_1, \dots, x_{k_1+k_2}) = h(x_1, \dots, x_{i-1}, g(x_{k_1+1}, \dots, x_{k_1+k_2}), x_{i+1}, \dots, x_{k_1}).$$

По предположению индукции

$$g(\gamma_{k_1+1}, \dots, \gamma_{k_1+k_2}) = \gamma'_i \in H_n,$$

а значит и

$$h(\gamma_1, \dots, \gamma'_i, \dots, \gamma_{k_1}) \in H_n.$$

Пусть имеется конечная система функций  $F = \{f_1, \dots, f_s\}$ . Очевидно, что  $F \subset M_{n_0}$  для некоторого  $n_0$ . А значит  $F$  сохраняет  $H_{n_0}$  и содержит не все о.-д. функции  $P$ .  $\square$

## 15 Билет 15

### 15.1 Вопрос 1. Классы структур данных. Иерархическая структура. Сетевые структуры. Реляционные структуры.

Модель данных — формальная теория представления и обработки данных в СУБД, включающая:

- аспект структуры (методы описания типов и логических структур данных);
- аспект манипуляции (методы манипулирования данными);
- аспект целостности (методы описания и поддержки целостности БД).

Аспект структуры определяет, что из себя логически представляет база данных, аспект манипуляции определяет способы перехода между состояниями базы данных (то есть способы модификации данных) и способы извлечения данных из базы данных, аспект целостности определяет средства описаний корректных состояний базы данных.

**Реляционные структуры.** Согласно реляционной модели, общая структура данных (отношение) может быть представлена в виде таблицы, в которой каждая строка значений (кортеж) соответствует логической записи, а заголовки столбцов являются названиями полей (элементов) в записях. Таким образом, данные и отношения между ними в реляционной модели представлены в виде набора таблиц, аналогичным по своей структуре таблицам модели сущность-связь.

Примеры реляционных БД: dBASE IV, FoxPro, Paradox.

**Иерархическая структура.** Иерархическая модель данных — представление базы данных в виде древовидной (иерархической) структуры, состоящей из объектов (данных) различных уровней. Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка (объект более близкий к корню) к потомку (объект более низкого уровня), при этом возможна ситуация, когда объект-предок не имеет потомков или имеет их несколько, тогда как у объекта-потомка обязательно только один предок. Объекты, имеющие общего предка, называются близнецами.

**Сетевые структуры.** Сетевая БД состоит из набора записей, соединенных друг с другом при помощи ссылок (links), которые могут быть видны пользователю как указатели (pointers). Ссылка соединяет ровно две записи. Записи организованы в виде произвольного графа (arbitrary graph).

## 15.2 Вопрос 2. Коммуникативные форматы обмена документами. Модель документа и ее использование. Карточный формат по ISO 2709. Процессы обмена документами в машиночитаемой форме, основные проблемы. Формат HTTP-2. Позиционные и помеченные электронные документы (ЭД). Функции модели ЭД. Проекты и стандарты, отражающие различные подходы к моделям ЭД. Модели ODA, SGML (основные понятия и представления).

ODA определяет формат документа, которые может содержать сырой текст, растровые изображения и векторную графику. Документ имеет как логическую структуру, так и разметку. Текст может быть разбит на логические куски: глава, сноски и другие элементы, родственные HTML. Формат для передачи ODA-файлов в бинарном виде называется ODIF (Open Document Interchange Format) и основан на SGML.

SGML (англ. Standard Generalized Markup Language) — метаязык, на котором можно определять язык разметки для документов. Изначально SGML был разработан для совместного использования машинно-читаемых документов в больших правительственных и аэрокосмических проектах. Он широко использовался в печатной и издательской сфере, но его сложность затруднила его широкое распространение для повседневного использования.

Основные части документа SGML:

- SGML-декларация — определяет, какие символы и ограничители могут появляться в приложении;
- Document Type Definition — определяет синтаксис конструкций разметки. DTD может включать дополнительные определения, такие, как символьные ссылки-мнемоники;
- Спецификация семантики, относится к разметке — также даёт ограничения синтаксиса, которые не могут быть выражены внутри DTD;
- Содержимое SGML-документа — по крайней мере, должен быть корневой элемент.

## 15.3 Вопрос 3. Вычислимые и гёделевы (главные) нумерации вычислимых функций. Теорема Райса о неразрешимости проблемы распознавания любого нетривиального свойства вычислимой функции по её описанию.

*Успенский, Семёнов, Теория алгоритмов. стр 122*

**Определение 72.** **Ансамбль** - множество конструктивных объектов одной природы. Например, ансамбль слов над заданным алфавитом.

**Определение 73.** **Вычислимая функция** — вычисляется каким-либо алгоритмом. Если функция не определена на входе, алгоритм не должен давать никакого ответа. Для произвольных подмножеств ансамблей  $A$  и  $B$  будем обозначать за  $\text{Com}(A, B)$  множество вычислимых функций.

**Определение 74.** **Породимое множество** — множество, порождаемое некоторым исчислением. **Исчисление** — конечный набор правил вывода, которые позволяют переходить от одних конструктивных объектов к другим

**Определение 75.** **Гёделева функция**  $V$  для  $X, Y$  с индексным множеством  $E$  - вычислимая функция из  $E \times X$  в  $Y$ , такая что для каждого породимого множества  $I$  и для каждого алгоритма  $U : I \times X \Rightarrow Y$  существует алгоритм  $D : I \Rightarrow E$  с областью определения  $I$ , для которого  $U(i, x) = V(D(i), x)$  для всех  $x, i$ .

**Определение 76.** **Нумерацией** множества  $M$  называется произвольное отображение  $\alpha$  произвольного множества  $E \subset \mathbb{N}$  на  $M$ . Если  $\alpha(e) = m$ , то  $m$  называется  $\alpha$ -номером, или номером, элемента  $m$ . **Нумерация разрешима**, если существует алгоритм, который применим к любой паре элементов  $E$  и даёт ответ на вопрос, являются ли они или нет номерами одного и того же элемента из  $M$ .

**Определение 77.** Можно расширить понятие нумерации и допускать рассматривать в качестве основания нумераций подмножества ансамблей. **Тотальная нумерация** - основанием является весь ансамбль.

**Пример 5.** Программа  $\rightarrow$  функция, задаваемая программой.

Существует соответствие между тотальными нумерациями семейств функций из  $X$  в  $Y$  с основаниями  $E$  и функциями из  $E \times X$  в  $Y$ .

**Определение 78.** Если данная функция и данная нумерация соответствуют друг другу, то будем называться их сопряжёнными.

**Определение 79.** Тотальная нумерация  $\alpha$  семейства функций из  $X$  в  $Y$  с основанием нумерации  $E$  называется **вычислимой**, если сопряжённая с ней функция является вычислимой.

**Определение 80.** Тотальная вычислимая нумерация семейства  $\text{Com}(X, Y)$  с основанием нумерации  $E$  называется **главной (гёделевой)**, если сопряжённая с ней функция является гёделевой универсальной функцией; как легко видеть это эквивалентно тому, что всякая вычислимая нумерация семейства  $\text{Com}(X, Y)$  сводится к ней.

**Теорема 26** (Теорема Райса). Задача определения того, обладает ли некоторый объект нетривиальным свойством, алгоритмически неразрешима. Свойство называется нетривиальным, если существуют объекты, обладающие им, и объекты, не обладающие им.

## 16 Билет 16

**16.1 Вопрос 1. Информационный поиск. Основные понятия и виды поиска. Информационно-поисковые языки. Понятие pertinентности, смысловой и формальной релевантности. Критерии выдачи. Модели поиска. Стратегия поиска. Функциональная эффективность поиска. Поисковые массивы, способы их организации. Понятия об ассоциативном поиске и условиях его реализации.**

**Определение 81.** *Информационный поиск — процесс нахождения и выдачи определённой заранее заданными признаками информации из массивов и записей любого вида и на любых носителях.*

Виды информационного поиска

- полнотекстовый поиск (поиск по всему содержимому документа);
- поиск по метаданным (поиск по атрибутам документа: название документа, дата создания, размер, автор);
- поиск по изображению (поиск по содержанию изображения).

**Определение 82.** *Информационно-поисковая система (ИПС) — пакет ПО, реализующий процессы создания, актуализации, хранения документов и поиска информации в информационных базах данных*

ИПС могут выдавать только такую информацию, которая была ранее введена в них. Этим они отличаются от информационно-логических систем (могут производить логическую переработку информации с целью получения новой информации).

Виды ИПС

- Тематические каталоги
- Специализированный каталоги
- Поисковые машины (Яндекс, Google)
- Средства метапоиска (сравнение результатов работы нескольких поисковых машин)
- документальные ИПС информация о содержании документа + информация о самом документе (автор, год ...)
- фактографические ИПС

**Информационно-поисковые языки** ИПЯ — формализованный искусственный язык, предназначенный для индексирования документов, информационных запросов и фактов в форме, удобной для автоматического поиска.

Состоит из

- алфавита (списка элементарных символов)
- правил образования (какие комбинации элементарных символов допускаются при построении слов и выражений)
- правил интерпретации (как надлежит понимать эти слова и выражения)

Выражения на ИПЯ допускают только одно истолкование, благодаря чему возможно сравнивать ПОД (поисковой образ документа) и ПОЗ (поисковой образ запроса) формально, не вникая в смысл.

**Понятия пертинентности, смысловой и формальной релевантности.** Степень соответствия документа запросу называют **релевантностью**.

Различают содержательную и формальную релевантности.

Содержательная релевантность — соответствие документа информационному запросу, определяемое неформальным путем.

Формальная релевантность — соответствие документа информационному запросу, определяемое алгоритмически на основании применяемого в информационно-поисковой системе критерия выдачи.

Пертинентность — соответствие полученной информации информационной потребности пользователя.

**Критерий выдачи** Критерий выдачи — совокупность признаков, по которым:

- определяется степень соответствия поискового образа документа поисковому предписанию;
- принимается решение о выдаче или невыдаче того или иного документа в ответ на информационный запрос.

Запрос — это формализованный способ выражения информационных потребностей пользователя.

Процесс поиска представляет сопоставление поисковых образов документов (ПОД) с поисковым образом запроса (ПОЗ)

Поисковый образ документа получается в результате процесса индексирования, которое выполняется квалифицированными специалистами и состоит из двух этапов: - выявление смысла документа, - описание смысла на специальном информационно-поисковом языке (ИПЯ).

Запрос к ИПС описывается также на этом языке. Поиск документа состоит в сравнении множества хранящихся в системе ПОД и текущего ПОЗ. В результате пользователю выдаются требуемые документы, отвечающие критериям запроса, выводится список найденных документов в порядке убывания релевантности, или отказ. Результат поиска — ссылки на документы (электронные адреса), содержащие требуемую информацию.

## Модели поиска

- Теоретико-множественные (Булева модель)
- Вероятностные
- Алгебраические (Векторная модель)
- Гибридные

В булевой модели документ представляется в виде слов (единица, если слово есть в документе и ноль — иначе). Запросы содержат слова и булевы операции над ними.

В векторной модели учитывается, «насколько сильно» входит в документ каждый термин (вес термина). Документ представляется вектором в  $\mathbb{R}^n$  с помощью *TFiDF*, а запрос в виде бинарного вектора. Близость документа определяется как близость между вектором-документом и вектором-запросом.

Вероятностная модель: определяется вероятность того, что данный документ окажется интересным пользователю.

Подразумевается наличие уже существующего первоначального набора релевантных документов. Рассматривается соотношение встречаемости термов в релевантном наборе и в остальной части коллекции. Документ представляется как множество слов без учета частоты их встречаемости в документе.

**Стратегии поиска** Стратегия поиска — общий план поведения системы или пользователя для выражения и удовлетворения информационной потребности пользователя

Примеры:

- следует по возможности искать специализированную ИПС по своей теме
- следует читать найденные документы и искать наиболее точные термины и связи между ними, т.к. возможно мы не знаем реально употребляющихся терминов
- следует использовать несколько ИПС

**Понятия об ассоциативном поиске и условиях его реализации.** Ассоциативные запросы представляют собой синонимы запроса и близкие по смыслу слова, уточняющие запрос. Они формируются в результате анализа поисковой машиной статистики. Ассоциативный поиск возвращает документы, содержащие не только термины запроса, но и термины, статистически ассоциирующиеся с запросом.

**16.2 Вопрос 2. Базы знаний. Общие принципы моделирования окружающей среды и мышления человека. Методы представления знаний: классификационные тезаурусные, основанные на отношениях, семантические сети и фреймы, продукционные и не продукционные.**

См. 7.1, 8.1

**16.3 Вопрос 3. Теорема Новикова об алгоритмической неразрешимости проблемы тождества слов в теории групп. Примеры других неразрешимых проблем алгебры.**

Проблема тождеств слов для групп, заданных конечным числом образующих, связанных конечным числом определяющих соотношений ставится следующим образом: требуется построить алгоритм, позволяющий для любых двух слов установить, равны они между собой или нет.

**Теорема 27.** *Существует группа с конечным числом образующих элементов и конечным числом определяющих соотношений, для которой не существует алгоритма, решающего проблему тождества слов этой группы.*

**Теорема 28** (О самоприменимости). *Не существует алгоритма, который по коду другого алгоритма, определяет его самоприменимость.*

## 17 Билет 17

17.1 Вопрос 1

17.2 Вопрос 2.

**17.3 Вопрос 3. Мера сложности вычислений. Существование сколь угодно сложных вычислимых функций. Теорема об ускорении. Классы сложности, теорема о пробелах.**

*Успенский, Семёнов, Теория алгоритмов. стр 134*

**Определение 83.** *Для фиксированной геделевой функции  $V : E \times X \rightarrow Y$  мера сложности — вычислимая функция  $C : E \times X \rightarrow \mathbb{N}$ , удовлетворяющая двум аксиомам Блюма:*

- $V(i, x)$  определено  $\Leftrightarrow C(i, x)$  определено;
- множество  $\{(i, x, y) | C(i, x) = y\}$  разрешимо.

**Теорема 29.** Пусть  $C_1, C_2$  - две меры сложности (для одной и той же главной функции). Тогда существует такая вычислимая функция  $D : X \times N \Rightarrow N$  с областью определения  $X \times N$ , что для всех  $i$  из  $I$  неравенство  $C_2(i, x) \leq D(x, C_1(i, x))$  выполнено для всех  $x$ , для которых  $V(i, x)$  определено, кроме конечного их числа.

**Теорема 30** (Теорема об ускорении). Пусть  $C$  - мера сложности и пусть  $R$  - всюду определённая вычислимая функция из  $\mathbb{N}$  в  $\mathbb{N}$ . Тогда существует такое разрешимое подмножество  $A$  множества  $X$  (с характеристической функцией  $\chi_A$ ), что для любого  $i \in E$ , для которого  $V(i, x) = \chi_A(x)$  для всех  $x$ , существует такое  $j$ , что  $V(j, x) = \chi_A(x)$  при всех  $x$ , и для всех  $x$ , кроме конечного их числа, имеет место неравенство  $C(i, x) \geq R(C(j, x))$ .

**Определение 84.** Класс сложности задаётся некоторой системой оценок, причём для любых двух оценок  $a, b$  существуют преобразования  $U, V$  из фиксированного множества, т.ч.  $a \leq U(b)$ ,  $b \leq V(a)$ . Функция принадлежит классу, если сложность её вычисления может быть ограничена оценкой из данного класса

**Пример 6.** Все невырожденные полиномы получаются возведением друг друга в ограниченную и отделённую от нуля степень.

## 18 Билет 18

### 18.1 Вопрос 1

### 18.2 Вопрос 2

### 18.3 Вопрос 3. Алгоритмический подход к понятию количества информации. Сложность конечного объекта по А.Н. Колмогорову. Существование оптимального способа описания. Количество информации.

Успенский, Верецагин, Шень, Колмогоровская сложность и алгоритмическая случайность. стр. 28

**Определение 85.** *Способом описания, или декомпрессором*, называется произвольное вычислимое частичное отображение  $D$  из множества двоичных слов  $\Theta$  в себя. (Вычислимость отображения  $D$  означает, что есть алгоритм, который применим к словам из области определения отображения  $D$  и только к ним; результат применения алгоритма к слову  $x$  есть  $D(x)$ .) Если  $D(y) = x$ , говорят, что  $y$  является описанием  $x$  при способе описания  $D$ .

**Определение 86.** Для каждого способа описания  $D$  определена сложность относительно этого способа описания, равная длине кратчайшего описания:

$$KS_D(x) = \min\{l(y) \mid D(y) = x\}.$$

При этом минимум пустого множества считается равным  $+\infty$ .

**Определение 87.** Говорят, что способ описания  $D_1$  не хуже способа описания  $D_2$ , если найдётся такая константа  $c$ , что  $KS_{D_1}(x) \leq KS_{D_2}(x) + c$  для всех слов  $x$ . (Краткая запись:  $KS_{D_1}(x) \leq KS_{D_2}(x) + O(1)$ .)

**Определение 88.** Способ описания называют оптимальным, если он не хуже любого другого способа описания.

**Теорема 31** (Соломонова-Колмогорова). Существует способ оптимальный способ описания  $D$ : для всякого способа описания  $D'$  найдётся такая константа  $c$ , что

$$KS_D(x) \leq KS_{D'}(x) + c$$

для любого слова  $x$ .

*Доказательство.* Успенский, Верещагин, Шень, Колмогоровская сложность и алгоритмическая случайность. стр. 14

Выберем какой-либо универсальный язык программирования. Пусть  $U$  — интерпретатор этого языка:  $U(p, x)$  есть результат работы программы  $p$  на входе  $x$  (программа и вход — двоичные слова). Далее мы полагаем  $D(\hat{p}x) = U(p, x)$ , где вычисляемое отображение  $p \mapsto \hat{p}$  выбрано так, чтобы по слову  $\hat{p}$  можно было определить  $p$ , а также место, где  $\hat{p}$  кончается. (В этом случае слово  $\hat{p}$  не может быть началом слова  $\hat{q}$  при  $p \neq q$ , и это свойство гарантирует, что  $D$  корректно определено.) Тогда для любого способа описания  $D'$  имеем

$$KS_D(x) \leq KS_{D'}(x) + l(\hat{p}),$$

где  $p$  — программа, соответствующая способу описания  $D'$ . (В самом деле, если  $y$  есть описание  $x$  относительно  $D'$ , то  $\hat{p}y$  есть описание  $x$  относительно  $D$ .)  $\square$

Далее рассматриваются только оптимальные способы описания и сложность в них (Обозначение:  $KS(x)$ ).

**Определение 89.** Если слово  $x$  имеет сложность  $k$ , мы говорим, что  $x$  содержит  $k$  битов информации.

**Теорема 32.**  $\exists c \forall x : KS(x) \leq l(x) + c$

Из теоремы, в частности, следует, что сложность любого слова при оптимальном способе описания конечна.

**Теорема 33.** Для любого алгоритма  $A$  существует такая константа  $c$ , что

$$KS(A(x)) \leq KS(x) + c$$

для всех  $x$ , при которых  $A(x)$  определено.

## 19 Билет 19

### 19.1 Вопрос 1

### 19.2 Вопрос 2

### 19.3 Вопрос 3. Разрешимые теории. Элиминация кванторов. Разрешимость теорий алгебраически замкнутых полей и вещественно замкнутых полей.

Кейслер, Чэн, Теория моделей, стр 52

**Определение 90.** *Теорией* (первого порядка) в некотором языке называется множество истинных формул (предложений) в этом языке. **Множеством аксиом теории**  $T$  называется всякое множество предложений, обладающее теми же следствиями, что и  $T$ . Теория называется **конечно аксиоматизируемой**, если она обладает конечным множеством аксиом.

**Определение 91.** *Теория разрешима*, если существует регулярный способ (алгоритм) проверки, принадлежит ли некоторое высказывание заданной теории или нет.

**Определение 92.** *Интерпретация* — задание значения (смысла) математических выражений (символов, формул и т.д.) В математике такими значениями служат математические объекты (математические операции, выражения и т.п.). **Моделью** называется интерпретация, в которой выполнен заданный набор аксиом.

**Пример 7.** Пусть  $\mathcal{L}$  содержит только двуместный предикатный символ  $\leq$ . Будем писать  $x \leq y$ , имея в виду  $\leq(x, y)$ . Теория частичного порядка имеет три аксиомы:

1.  $(\forall xyz)(x \leq y \wedge y \leq z \rightarrow x \leq z)$ ,
2.  $(\forall xy)(x \leq y \wedge y \leq x \rightarrow x \equiv y)$ ,
3.  $(\forall x)(x \leq x)$ .

Всякая модель  $\langle A, \leq \rangle$  состоит из непустого множества  $A$  и отношения частичного порядка  $\leq$  на  $A$ .

**Определение 93.** *Теория булевых алгебр* задаётся в языке  $L = \{+, \times, \neg, 0, 1\}$  стандартными аксиомами. Всякая модель этой теории называется *булевой алгеброй*.

**Определение 94.** *Атомом* булевой алгебры называется всякий элемент  $x \neq 0$ , такой, что ни один элемент  $y$  не может лежать строго между  $0$  и  $x$ , т.е. ни для какого  $y$  невозможно  $0 \leq y \leq x$ ,  $0 \neq y$ ,  $y \neq x$ . Булева алгебра называется *атомной*, если всякий её ненулевой элемент содержит некоторый атом. Булеву алгебру называют **безатомной**, если она не содержит атомов.

**Определение 95.** Формула  $\varphi(v_0, \dots, v_n)$  является следствием предложений  $\Sigma$  тогда и только тогда, когда следствием множества  $\Sigma$  является предложение  $(\forall v_0, \dots, v_n) \varphi(v_0, \dots, v_n)$ .

Элиминация кванторов состоит в следующем: выбирается некоторое исходное множество формул, называемых базисными и доказывается, что всякая формула  $T$ -эквивалентна некоторой булевой комбинации базисных формул. Булева комбинация — это соединение с помощью связок  $\wedge$  и  $\neg$ .

**Лемма 24.** Пусть  $T$  — теория, а  $\Sigma$  — множество формул, называемых базисными. Чтобы доказать, что всякая формула  $T$ -эквивалентна некоторой булевой комбинации базисных формул, достаточно проверить следующие утверждения:

Всякая атомная формула  $T$ -эквивалентна булевой комбинации базисных формул.

Если  $\theta$  — булева комбинация базисных формул, то формула  $(\exists v_m)\theta$  оказывается  $T$ -эквивалентной булевой комбинации базисных формул.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

## 20 Билет 20

### 20.1 Вопрос 1

### 20.2 Вопрос 2

### 20.3 Вопрос 3. Задача распознавания образов. Основные подходы: геометрический, вероятностный и комбинаторно-логический. Примеры задач распознавания.

см. билеты ниже

## 21 Билет 21

### 21.1 Вопрос 1

### 21.2 Вопрос 2

### 21.3 Вопрос 3. Задача распознавания образов. Вероятностный подход. Условия кластеризации. Основные процедуры построения кластеров. Метод скрытых марковских процессов.

(Гасанов. Введение в теорию интеллектуальных систем, стр. 47) простой билет.

## 22 Билет 22

### 22.1 Вопрос 1

### 22.2 Вопрос 2

### 22.3 Вопрос 3. Задача распознавания образов. Комбинаторно-логический подход. Линейные процедуры и информационные веса. Условия эффективности распознавания.

Имеется некоторое множество объектов  $x_i$ ,  $i \in \{1, 2, \dots, m\}$ . Каждый характеризуется параметрами-признаками  $x_i(j)$ ,  $j \in \{1, 2, \dots, n\}$ . Тогда множество  $M$  можно задать матрицей  $T_{m,n} = (x_i(j))$ . Пусть  $M$  разбито на два класса  $K_1$  и  $K_2$ , а тем самым и матрица  $T_{m,n}$  состоит из двух подматриц  $T_{K_1}$  и  $T_{K_2}$ . Классы  $K_1$  и  $K_2$  неизвестны, но есть обучающей выборка в виде подматриц  $T_1$  и  $T_2$ . Требуется для любого набора признаков, который есть описание объекта из  $M$ , определить класс.

Будем считать, что значения признаков принадлежат множеству  $\{0, 1\}$  и эти признаки выбраны так, что в матрице  $T$  нет двух одинаковых столбцов, отвечающих объектам разных классов.

Если  $T$  - некоторая матрица, а  $\tau$  - некоторый набор признаков (столбцов), то через  $T(\tau)$  обозначим подматрицу матрицы  $T$ , полученную удалением всех столбцов кроме последнего из  $\tau$ .

Набор столбцов назовём **тестом**, если не существует строки, содержащейся в  $T_1(\tau)$  и  $T_2(\tau)$  одновременно. **Минимальный тест** — тест, содержащий наименьшее число признаков. **Тупиковый тест** — тест, который при удалении любого признака перестает быть тестом. Минимальный тест является также и тупиковым.

Для пары матриц  $T_1$  и  $T_2$  информационным весом  $p_i$  признака  $i$ ,  $i \in \{1, 2, \dots, n\}$ , будем называть отношение числа тупиковых тестов, в которых вошёл признак  $i$ , к общему числу тупиковых тестов.

Чем больше вес  $p_i$ , тем важнее признак  $i$  с точки зрения различения классов  $K_1$  и  $K_2$ . Вектор  $p = (p_1, \dots, p_n)$  назовём **вектором информационных весов** или **весовым вектором**.

Приведём пример алгоритма распознавания  $A_1$  на основе информационных весов.

Сначала в алгоритме  $A_1$  для каждого признака  $i$  вычисляется его информационный вес  $p_i$ ,  $i \in \{1, 2, \dots, n\}$ . Если  $T_1 = (a_{i,j})$ ,  $1 \leq i \leq m_1$ ,  $T_2 = (b_{i,j})$ ,  $1 \leq i \leq m_2$  и  $x = (x_1, x_2, \dots, x_n)$  — объект, подлежащий классификации, то вычисляются две величины:

$$r_1 = \frac{1}{m_1} \sum_{i=1}^{m_1} \sum_{j=1}^n p_j (a_{i,j} \oplus x_j)$$

и

$$r_2 = \frac{1}{m_2} \sum_{i=1}^{m_2} \sum_{j=1}^n p_j (b_{ij} \oplus x_j).$$

Если  $r_1 \geq r_2$ , то относим объект к первому классу, иначе — ко второму.

### Условия эффективности распознавания

## 23 Билет 23

### 23.1 Вопрос 1

### 23.2 Вопрос 2

### 23.3 Вопрос 3. Тестовые процедуры распознавания. Алгоритм голосования. Оценки длины минимальных тестов

Тестовые процедуры распознавания описаны выше.

**Алгоритм Кора (комбинаторного распознавания)** — алгоритм классификации (взвешенного голосования правил):

[http://ru.wikipedia.org/wiki/КОРА\\_\(алгоритм\)](http://ru.wikipedia.org/wiki/КОРА_(алгоритм))

В таблице  $\|a_{ij}\|_{m \times n}$ , задающей объекты с известной классовой принадлежностью, пусть  $S_1, \dots, S_q \in K_1, S_{q+1}, \dots, S_m \in K_2$ . Просматриваем все тройки признаков  $\{r, u, v\}$  (число таких троек, очевидно, равно  $C_n^3$  и анализируем часть таблицы информационных векторов  $T_1$  из обучающей выборки, составленную из столбцов  $r, u, v$ :

$a_{1r}$	$a_{1u}$	$a_{1v}$
$a_{2r}$	$a_{2u}$	$a_{2v}$
...	...	...
$a_{ir}$	$a_{iu}$	$a_{iv}$
...	...	...
$a_{qr}$	$a_{qu}$	$a_{qv}$
<hr/>		
$a_{q+1r}$	$a_{q+1u}$	$a_{q+1v}$
...	...	...
$a_{jr}$	$a_{ju}$	$a_{jv}$
...	...	...
$a_{mr}$	$a_{mu}$	$a_{mv}$

Среди первых  $q$  строк выделяем и фиксируем все тройки, не совпадающие ни с одной из троек в строках  $q+1, \dots, m$ . Формируем множество таких троек  $\{(a_{ir}, a_{iu}, a_{iv})\}$ . Аналогично выделяем все тройки  $\{(a_{jr}, a_{ju}, a_{jv})\}$ , не совпадающие ни с одной из первых  $q$  троек. Множества  $\{(a_{ir}, a_{iu}, a_{iv})\}, \{(a_{jr}, a_{ju}, a_{jv})\}$  назовем, соответственно, характери-

ками классов  $K_1, K_2$ . Такие характеристики формируем для всех троек  $(r, u, v)$ . Пусть задан для распознавания объект  $S = (b_1 \dots b_r \dots b_u \dots b_v \dots b_n)$ . Сравниваем все характеристики всех троек для  $K_1$  с соответствующими тройками в распознаваемом объекте  $S$ . Число совпадений  $(a_{ir}, a_{iu}, a_{iv}) = (b_r, b_u, b_v)$  обозначаем  $\Gamma(S, K_1)$  — число голосов, поданных для  $S$  за класс  $K_1$ . Аналогично формируем величину  $\Gamma(S, K_2)$  — число совпадений  $(a_{jr}, a_{ju}, a_{jv}) = (b_r, b_u, b_v)$ . Вводим пороговый параметр  $\nu$ . Если  $\Gamma(S, K_1) - \nu > \Gamma(S, K_2)$ , относим  $S$  классу  $K_1$ , при  $\Gamma(S, K_2) - \nu > \Gamma(S, K_1)$  — в класс  $K_2$ . В остальных случаях алгоритм отказывается от классификации. На практике часто полагают  $\nu = 0$ .

**Теорема 34.** Пусть  $A(M)$  — длина минимального теста для таблицы признаков  $M$ , поделённой на  $t$  классов. Тогда

$$\log_2(t) \leq A(M) \leq t - 1.$$

*Доказательство.* Докажем нижнюю оценку. Предположим, что тест содержит  $k < \log_2(t)$  наборов. Но тогда различных строк не более  $2^k \leq t$ , чего не может быть, поскольку по условию все строки разных классов различны.

В доказательстве верхней оценки используется

**Теорема 35.** Для произвольной булевой матрицы  $M$ , содержащей  $t$  попарно различных строк, существует тест, длина которого не превосходит  $t - 1$ .

*Доказательство.* Докажем индукцией по числу строк. Если строка одна, то тест пустой. Если строк две, то обязательно найдётся столбец, по которой они отличаются. Предположим, что утверждение теоремы имеет место для числа строк равного  $k$  и менее. Покажем его справедливость для  $k + 1$  строк. Возьмём произвольный  $j$ -й столбец  $S_j$ , в котором различны, например, первая и вторая компоненты. Будем считать для определённости, что первая компонента нулевая, а вторая единичная. Рассмотрим теперь матрицы  $M_0$  и  $M_1$ , составленные соответственно из  $m_0$  строк матрицы  $M$  с нулевой  $j$ -й компонентой и  $m_1$  строк с единичной  $j$ -й компонентой. Понятно, что  $m_i \leq k$ ,  $i = 0, 1$  и  $m_0 + m_1 = k + 1$ . Тогда по предположению индукции существуют тесты  $T_0$  для  $M_0$  и  $T_1$  для  $M_1$  такие, что

$$|T_0| \leq m_0 - 1, \quad |T_1| \leq m_1 - 1.$$

Пусть  $T = T_0 \cup T_1 \cup \{S_j\}$ . Покажем, что  $T$  — тест для  $M$ . Действительно, рассмотрим произвольные строки  $A$  и  $B$  матрицы  $M$ . Если их компоненты из  $S_j$  неодинаковы, то столбцы уже различены с помощью  $T$ , если обе их компоненты из  $S_j$  нулевые (единичные), то столбцы содержатся в  $M_0$  ( $M_1$ ) и различены по предположению индукции. Оценим длину теста:  $|T| \leq m_0 - 1 + m_1 - 1 + 1 = k$ . Теорема доказана.  $\square$

$\square$

## 24 Билет 24

### 24.1 Вопрос 1

### 24.2 Вопрос 2

### 24.3 Вопрос 3. Оценка числа тупиковых тестов для почти всех таблиц. Асимптотически оптимальный алгоритм построения множества коротких тестов. Полиномиальный характер решающих правил распознавания.

*Теория тестового распознавания В. Б. Кудрявцев, А. Е. Андреев Жесть, не читал.*

## 25 Билет 25

### 25.1 Вопрос 1

### 25.2 Вопрос 2. Защита информационных технологий, систем и прав на них. Информационное оружие в информационной войне. Модели безопасности и их применение. Таксономия нарушений информационной безопасности. Основные технологии построения защищённых информационных систем.

Под угрозой безопасности вычислительной системе понимаются воздействия на систему, которые прямо или косвенно могут нанести ущерб ее безопасности.

Выделяют три вида угроз:

- угрозы нарушения конфиденциальности обрабатываемой информации;
- угрозы нарушения целостности обрабатываемой информации;
- угрозы нарушения работоспособности системы (отказа в обслуживании).

Угрозы конфиденциальности направлены на разглашение секретной информации, т. е. информация становится известной лицу, которое не должно иметь к ней доступ. Иногда для обозначения этого явления используется понятие «несанкционированный доступ» (НСД).

Угрозы целостности представляют собой любое искажение или изменение неавторизованным на это действие лицом хранящейся в вычислительной системе или передаваемой информации. Целостность информации может быть нарушена как злоумышленником, так и результате объективных воздействий со стороны среды эксплуатации системы. Наиболее актуальна эта угроза для систем передачи информации — компьютерных сетей и систем телекоммуникаций.

Угрозы нарушения работоспособности (отказ в обслуживании) направлены на создание ситуаций, когда в результате преднамеренных действий ресурсы вычислительной системы становятся недоступными, или снижается ее работоспособность.

Реализация угрозы это атака. Любая атака на вычислительную систему (ВС) опирается на определенные особенности построения и функционирования последней, иными словами — использует имеющиеся недостатки средств обеспечения безопасности. В контексте изучения истории атак на вычислительные системы и построения таксономии причин нарушений информационной безопасности наиболее точно отражения сущности этого явления предлагается ввести термин "изъян защиты" (ИЗ). Изъян защиты — совокупность причин, условий и обстоятельств, наличие которых в конечном итоге может привести к нарушению нормального функционирования ВС и нарушению безопасности (несанкционированный доступ, ознакомление, уничтожение или искажение данных).

Таксономия, в отличие от классификации, представляющей собой абстрактную структуру разнесенных по категориям экземпляров, включает в себя комплексное исследование предметной области и создание модели полного множества изучаемых объектов, что позволяет определить признаки, которые могут быть положены в основу той или иной классификации. С точки зрения технологии создания защищенных систем наибольшее значение имеют следующие вопросы, на которые должна дать ответ таксономия ИЗ:

1. Каким образом ошибки, приводящие к появлению ИЗ, вносятся в систему защиты? (классификация ИЗ по источнику появления).
2. Когда, на каком этапе, они вносятся? (классификация ИЗ по этапу возникновения.)
3. Где, в каких компонентах, системы защиты (или ВС в целом) они возникают и проявляются? (классификация ИЗ по размещению в системе.)

### **25.3 Вопрос 3. Задача распознавания образов. Геометрический подход. Линейные процедура распознавания. Перцептроны. Теорема Новикова. Метод потенциальных функций.**

В геометрический подходе мы ищем оптимальную разделяющую плоскость для объектов двух классов пространства  $\mathcal{R}^n$ .

Пороговым элементом (персептроном) называется функция  $\mathcal{R}^n \mapsto \{0, 1\}$  следующего вида:

$$y = \begin{cases} 1 & \text{если } \sum_{i=1}^n \lambda_i x^i \geq \lambda_0, \\ 0 & \text{если } \sum_{i=1}^n \lambda_i x^i < \lambda_0, \end{cases}$$

где  $\Lambda = \lambda_1, \dots, \lambda_n$  — коэффициенты усиления сигналов  $x^1, \dots, x^n$ , а  $\lambda_0$  — величина порога срабатывания элемента.

Будем считать, что  $\lambda_0 = 0$  (мы можем сделать это без ограничения общности, повысив размерность входных сигналов на 1 и потребовав, чтобы первый элемент каждого сигнала

был равен единице).

Обучение. Пусть в некоторый момент времени на вход элемента  $R$  поступает сигнал, описываемый вектором  $x$ , на котором ожидается выходной сигнал  $y \in \{1, -1\}$ . Тогда реакция элемента может либо соответствовать ему  $\text{sgn}(R(x)) = y$ , либо нет. В первом случае оставили веса без изменения. Во втором случае выполним преобразование:

$$\lambda_i \leftarrow \lambda_i + x^i y$$

**Теорема 36.** Пусть  $X = \mathcal{R}^{n+1}$ ,  $Y = \{+1, -1\}$  и выборка  $X^l$  линейно разделима, т.е. существуют  $\delta > 0$  и единичный вектор  $\Lambda^*$ , т.ч.  $(x_i, \Lambda^*)y_i > \delta$ . Тогда за конечное число итераций для начального вектора  $\Lambda = 0$  и для произвольного случайного порядка предъявления элементов начальной выборки персептрон научится различать объекты.

*Доказательство.* Пусть  $\Lambda_i$  последовательность получающихся конфигураций для некоторого произвольного порядка предъявления элементов  $x_i$ .

Если на  $i$ -м шаге персептрон угадал, то

$$\Lambda_{i+1} = \Lambda_i$$

.

Иначе, производится исправление

$$\Lambda_{i+1} = \Lambda_i + x_{i+1}y_{i+1}$$

и

$$\Lambda_{i+1}^2 = \Lambda_i^2 + 2(\Lambda_i, x_{i+1})y_{i+1} + x_{i+1}y_{i+1}^2.$$

Пусть  $\forall j: |y_j| < D$ , тогда, т.к.  $(x_{i+1}, \Lambda_i)y_{i+1} \leq 0$  и  $\Lambda_0 = 0$ :

$$\Lambda_{i+1}^2 \leq \Lambda_i^2 + 2(\Lambda_i, x_{i+1})y_{i+1} + x_{i+1}y_{i+1}^2 \leq \Lambda_i^2 + 2(\Lambda_i, x_{i+1})y_{i+1} + D^2 \leq \Lambda_i^2 + D^2 \leq (i+1)D^2$$

По условию существуют  $\delta > 0$  и единичный вектор  $\Lambda^*$ , т.ч.  $(x_i, \Lambda^*)y_i > \delta$ . Оценим  $(\Lambda_i, \Lambda^*)$ . В начальный момент  $(\Lambda_0, \Lambda^*) = 0$ . Если исправление не происходит, то ничего не меняется. В противном случае:

$$(\Lambda_{i+1}, \Lambda^*) = (\Lambda_i, \Lambda^*) + (x_{i+1}, \Lambda^*)y_{i+1} \geq (\Lambda_i, \Lambda^*) + \delta$$

Итого:

$$(\Lambda_{i+1}, \Lambda^*) \geq (i+1)\delta$$

В силу неравенства Коши:

$$(\Lambda_k, \Lambda^*) \leq |\Lambda_k| \times |\Lambda^*| = |\Lambda_k|$$

Комбинируя оценки на  $\Lambda_k$ :

$$(k\delta)^2 \leq |\Lambda_k|^2 \leq kD^2 \Rightarrow k = \frac{D^2}{\delta}$$

□