М.И. Дехтярь

1 Логика предикатов (конспект лекций)

1.1 Введение

Утверждения, которые можно сформулировать с помощью средств логики высказываний (пропозициональной логики) всегда относятся к конкретным свойствам конкретных предметов, объектов, ситуаций. Мы уже сталкивались с примерами таких утверждений: "Сегодня идет дождь", "Вася любит Олю", "Если A – преступник, то B – не виновен", "Если цена на нефть растет и страна продает нефть, то растут и доходы ее бюджета"и т.п. Эти утверждения строятся из элементарных высказываний (пропозициональных переменных) с помощью логических связок (операций). Например, последнее из приведенных утверждений может быть формально записано как $(X \wedge Y) \to Z$, где X , Y и Z – это переменные, обозначающие, соответственно, высказывания "цена на нефть растет", "страна продает нефть"и "растут и доходы бюджета".

Для того, чтобы высказывать более общие точные утверждения о свойствах классов (множеств) объектов, предметов, ситуаций и т.п., нужен более богатый формальный язык. Им является язык *логики предикатов* (логики 1-ой ступени).

Прежде, чем переходить к формальным определениям, приведем некоторые содержательные примеры используемых понятий: объектов, свойств, отношений и операций (функций).

Объекты: люди, предприятия, числа, цвета, футбольные матчи, экзамены, дома, столы, компьютеры, фигуры, студенты, ...

Свойства: зеленый, тяжелый, голубоглазый, победный, девятиэтажный, деревянный, отличник, ...

Отношения: быть братом ..., занимать должность ... с зарплатой ..., больше чем ..., находиться внутри ..., любить, иметь цвет ..., случиться после ..., владеть ..., ...

Функции: отец ..., лучший_ друг ..., вдвое_больший_ чем ..., сумма ... и ..., группа_студента ..., самый_любимый_кинофильм ..., ...

В примерах отношений и функций многоточиями отмечены места, на которых должны стоять объекты, к которым они относятся.

Даже этих небольших перечислений достаточно, чтобы понять, что язык логики предикатов позволяет описать почти любой факт и сформулировать нужное утверждение о той или иной рассматриваемой предметной области. Вот несколько простых примеров:

"Два умножить на три равно шесть"
Здесь объекты: два, три и шесть; функция: умножить, отношение: равно.

"Все бармаглоты живут в зеленых домах"
Объекты: бармаглоты, дома; свойство: зеленый; отношение: жить_в.

"Некоторые предприятия в Твери являются банкротами"
Объекты: Тверь, предприятия; свойство: банкрот; отношение: "быть"("являться").

С формализацией этих понятий мы уже знакомы: объекты — это множества; свойства — подмножества; а отношения и функции имеют обычный математический смысл.

Напомним определения. Декартовым (прямым) произведением множеств $A_1, \dots, A_n$ называется множество последовательностей длины n (n-ок)

$$A_1 \times \dots A_n = \{< a_1, \dots, a_n > \ | \ a_1 \in A_1, \dots, a_n \in A_n\}.$$

Если $A_1 = \dots = A_n = A$, то $A_1 \times \dots A_n$ называется декартовой (прямой) степенью множества A и обозначается через A^n. Пусть A^0 – это множество, состоящее из одной пустой последовательности ε длины 0. *n-местным (или n-арным) отношением на множествах* $A_1, \dots, A_n$ называется любое подмножество $P \subseteq A_1 \times \dots \times A_n$. Поэтому понятие свойства (подмножества) совпадает с понятием одноместного отношения. Отношение $f \subseteq A_1 \times \dots \times A_n \times B$ называется *n-местной (или n-арной) функцией* из $A_1 \times \dots \times A_n$ в B, если оно для каждого набора аргументов $< a_1, \dots, a_n >$, $a_i \in A_i$, содержит не более одного набора вида $< a_1, \dots, a_n, b >$. В этом случае пишем $f(a_1, \dots, a_n) = b$.

С каждым отношением $P \subseteq A_1 \times \dots \times A_n$ можно связать его характеристическую функцию, которую мы будем обозначать той же буквой,

$$P(x_1, \dots, x_n) = \begin{cases} 1, & \text{если } < x_1, \dots, x_n > \in P \\ 0, & \text{в противном случае} \end{cases}$$

Как обычно, будем ассоциировать 1 с логическим значением "истина", а 0 — со значением "ложь". Такие характеристические функции отношений будем называть n-местными *предикатами* (используются также термины: предикат размерности n, n-арный предикат, предикат валентности n) и говорить об их истинности или ложности на соответствующих наборах аргументов. Таким образом, предикат – это отображение, сопоставляющее каждому набору своих аргументов одно из логических значений: 1 или 0. Если потребуется, размерность предиката будет указываться соответствующим индексом вверху: $P^{(n)}$ будет означать, что у предиката P имеется n аргументов.

Мы будем рассматривать также функции и предикаты размерности нуль. Множество A^0 одноэлементно (содержит единственную последовательность длины 0). Поэтому функции из A^0 в A отождествляются с элементами множества A. Такие функции называются константными или просто константами. Предикатов размерности нуль ровно два: 1 (истина) и 0 (ложь).

Вообще говоря, типы объектов-аргументов предиката, т.е. типы множеств $A_1, \ldots, A_n$, могут быть различными. Например, у предиката *выпускает(Предприятие, Товар)*, определяющего связь между предприятиями и выпускаемыми ими товарами, первым аргументом является название предприятия, а вторым – одного из выпускаемых им товаров. Но далее для простоты мы будем рассматривать только предикаты и функции, все аргументы которых принадлежат одному множеству объектов A. Это ограничение не очень существенно. Можно выделять в A объекты нужных типов с помощью свойств-одноместных предикатов. В нашем случае выражение *выпускает(X,Y)* можно уточнить, введя предикаты *предприятие*$^{(1)}$ и *товар*$^{(1)}$ и записав конъюнкцию: *предприятие*$(X) \wedge$ *товар*$(Y) \wedge$ *выпускает*(X,Y).

1.2 Синтаксис и семантика: формулы и интерпретации

Формулы языка логики предикатов включают в себя символы (имена) предикатов из некоторого множества $\mathbf{P} = \{P_1^{(n_1)}, P_2^{(n_2)}, \ldots, P_k^{(n_k)}, \ldots\}$, символы (имена) функций из некоторого множества $\mathbf{F} = \{f_1^{(m_1)}, f_2^{(m_2)}, \ldots, f_j^{(m_j)}, \ldots\}$, символы (имена) констант из некоторого множества $\mathbf{C} = \{c_1, c_2, \ldots\}$, логические связки $\{\neg, \wedge, \vee, \rightarrow\}$, кванторы всеобщности $\forall$ и существования $\exists$ и вспомогательные символы (скобки, запятые). Первые три множества образуют *сигнатуру* языка: $\Sigma =< \mathbf{P}, \mathbf{F}, \mathbf{C} >$.

Зафиксируем некоторое счетное множество *объектных переменных* **Var** (такие переменные также называют *предметными или индивидными*). Они предназначены для обозначения элементов множества (объектов), на котором определены функции и предикаты; обычно в таком качестве используют латинские буквы $x, y, z, u, v, w, x_i, y_j, z_k$ и т.п. В каждой формуле будет использоваться конечное число переменных, так что счётного набора переменных нам хватит. Чтобы не возникла путаница, будем считать, что переменные отличны от всех имен констант, функций и предикатов.

Определим понятие *терма* данной сигнатуры Σ.

Определение 1. Терм. *Термом называется выражение, состоящее из переменных, запятых, скобок и символов сигнатуры, которое можно построить по следующим правилам:*

- *Объектная переменная из* **Var** *есть терм.*
- *Символ константы из* **C** *есть терм.*
- *Если $t_1, \ldots, t_k$ — термы, а $f^{(k)}$ — символ функции из* **F**, *то $f(t_1, \ldots, t_k)$ есть терм.*

Термы, не содержащие переменных, назовем замкнутыми.

Термы служат для задания объектов. Замкнутые термы задают объекты непосредственно. Например, пусть $\mathbf{F} = \{$*отец* $^{(1)}$, *лучший_ друг*$^{(1)}$, *зарплата* $^{(1)}$, $+^{(2)}\}$, а $\mathbf{C} = \{$*"Петр", "Джон", "Ольга"*, $0, 1, 2, \ldots\}$. Тогда самый простой способ задать объект — это указать соответствующую константу, например, *"Петр", "Джон", 0, 1, 47, ...*. Терм $+(17, 25)$ задает объект 42, терм *лучший_ друг(отец("Петр"))* задает объект – лицо, являющееся лучшим другом отца объекта "Петр", а терм *зарплата(лучший_ друг(отец("Петр")))* задает объект-число, представляющее зарплату этого лица. Значениями переменных также являются объекты. Поэтому термы с переменными задают конкретные объекты при подстановке значений вместо переменых. Приведем еще несколько примеров термов данной сигнатуры: x, *отец* (x), *зарплата(лучший_ друг(отец(отец(z))))*, *+(зарплата(лучший_ друг(отец(z))), +(зарплата ("Ольга"), 1000))*, *отец(5)*, *+(отец("Ольга"), 1000))* . Отметим, что последние два терма построены в соответствии с нашими правилами, но неясно, какие объекты они могут задавать. Это зависит от определения функции *отец* на числах и функции + на парах аргументов вида

(Лицо, Число). Часто во множество объектов включают специальный объект "ОШИБКА", который является значением функций на некорректных аргументах.

Выражения $x+10$, *отец ("Джон","Петр"), +(100), лучший_друг("Мария")* термами данной сигнатуры не являются. *Определите почему?*

Определение 2. Атомная формула.

- *Если t_1 и t_2 — термы, то выражение $(t_1 = t_2)$ является атомной формулой.*
- *Любой предикатный 0-местный символ из $\mathbf{P}$ является атомной формулой.*
- *Если $P^{(k)}$ — предикатный k-местный символ из $\mathbf{P}$, а $t_1, \ldots, t_k$ термы, то выражение $P(t_1, \ldots, t_k)$ является атомной формулой.*

Рассмотрим сигнатуру $\Sigma_1 =< \mathbf{P}, \mathbf{F}, \mathbf{C} >$, в которой $\mathbf{P} = \{$*живут_рядом*$^{(2)}$, *родственники*$^{(2)}$, *человек*$^{(1)}$, *число*$^{(1)}$, $\leq^{(2)}\}$, а функции $\mathbf{F}$ и константы $\mathbf{C}$ определены в примере выше. Тогда следующие выражения являются атомными формулами: *"Джон" = "Петр", "Петр" = 6, отец("Ольга") ="Петр",* $+(3,5) = +(1, +(6,1))$, *зарплата(лучший_ друг(отец(z)) = 5000, живут_рядом("Джон","Ольга"), родственники(лучший_ друг("Джон"), отец(x)),* и т.п.

Формулы строятся по таким правилам:

Определение 3. Формула.

- *Всякая атомная формула есть формула.*
- *Если φ — формула, то $\neg\varphi$ — формула.*
- *Если φ и ψ — формулы, то выражения $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$ также являются формулами.*
- *Если φ — формула, а $x \in$ **Var** – объектная переменная, то выражения $\forall x\varphi$ и $\exists x\varphi$ являются формулами (в этом случае $\forall x\varphi$ и $\exists x\varphi$ называются областью действия квантора $\forall x$ или $\exists x$ соответственно).*

Понятие *подформулы* для формул логики предикатов определяется естественным образом. Во-первых, сама формула является своей подформулой. Во-вторых, если формула имеет вид $\neg\varphi$, $\forall x\varphi$ или $\exists x\varphi$, то ее подформулами являются все подформулы формулы φ, а если она имеет вид $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$ или $(\varphi \to \psi)$, то ее подформулами являются все подформулы формул φ и ψ.

Определим также понятия *связанных и свободных переменных* формулы. Вхождение переменной x в формулу φ называется *связанным*, если оно входит в область действия некоторого квантора $\forall x$ или $\exists x$. В противном случае, оно называется *свободным.* Квантор $\forall x$ ($\exists x$) *связывает* в формуле $\forall x\varphi$ ($\exists x\varphi$) все свободные вхождения переменной x в подформулу φ.

Пусть в некоторой сигнатуре имеются два двуместных предиката $P^{(2)}, Q^{(2)}$. Тогда в формуле
$\forall x(\exists x(P(x,y) \wedge \forall y Q(x,y)) \vee \neg P(x,x))$
оба вхождения x в подформулу $(P(x,y) \wedge \forall y Q(x,y))$ связаны квантором $\exists x$, первое вхождение y является свободным, а второе — связано квантором $\forall y$. Оба вхождения x в подформулу $\neg P(x,x)$ связаны квантором $\forall x$.

Переменная называется *свободной (связанной)* в формуле, если у нее есть хоть одно свободное (связанное) вхождение в эту формулу. Формула, у которой нет свободных переменных называется *замкнутой.* Отметим, что переменная может быть одновременно связанной и свободной в данной формуле.

Рассмотрим примеры формул в сигнатуре Σ_1:
1) $\varphi_1 = \forall x($*человек(x)* $\to \exists y($*человек(y)* $\wedge$ *живут_рядом(x,y))*
В этой формуле все вхождения переменных x и y являются связанными и, следовательно, φ_1 — замкнутая формула. Содержательно, она утверждает, что у всякого человека имется человек-сосед. Понятно, что истинность этого утверждения не зависит от имен переменных, использованных в формуле.

2) $\varphi_2 = \exists y[$*человек(y)* $\wedge (\forall x$ *родственники(x, y)*$\vee$ $(\neg$*живут_рядом*$(x,y) \wedge ($*зарплата(y)* $\geq$ 35))].
В формуле φ_2 все вхождения переменной y являются связанными, первое вхождение x также связано, а второе вхождение x свободно, так как не входит в область действия квантора $\forall x$.

Таким образом x в φ_2 является связанной и свободной. Эта формула утверждает что имеется такой человек, который состоит в родственных отношениях со всеми или не живет рядом с x и имеет зарплату ≥ 35. Ясно, что истинность этой формулы зависит от значения свободной переменной x.

Даже этих небольших примеров достаточно, чтобы понять, что семантика (т.е. значение, смысл) формул логики предикатов зависит от состояния дел в той предметной области, о которой идет речь в формулах, и от значений их свободных переменных. Для того, чтобы ее определить, нужно уточнить, свойства какой предметной области мы собираемся описывать с помощью формул и как в этой области интерпретируются символы предикатов, функций и констант нашего языка.

Определение 4. Алгебраической системой *(иногда говорят также, структурой или интерпретацией) сигнатуры Σ называется непустое множество A вместе с отображением, которое каждому n-местному предикатному (функциональному) символу из Σ сопоставляет n-местный предикат (n-местную функцию) на A, а каждой константе из Σ — некоторый элемент из A. A называется основным множеством системы.*

Если множество $\mathbf{F}$ имен функций в Σ пусто, то соответствующую систему часто называют **моделью**.

Определенную таким образом алгебраическую систему будем обозначать как $\mathcal{A} =< A; \Sigma >$, используя для предикатов и функций системы их имена из сигнатуры Σ.

Определим теперь интерпретации свободных переменных.

Определение 5. Состоянием *(оценкой) алгебраической системы $\mathcal{A} =< A; \Sigma >$ называется отображение $\sigma : \mathbf{Var} \to A$, которое каждой переменной $x \in \mathbf{Var}$ сопоставляет ее значение в данном состоянии $\sigma(x)$.*

Имея алгебраическую систему и состояние, можно говорить о значениях термов и формул. Мы будем считать, что в определениях ниже зафиксирована некоторая алгебраическая система $\mathcal{A} =< A; \Sigma >$, будем через $\sigma(t)$ и $\sigma(\varphi)$ обозначать значение терма t (формулы φ) на состоянии σ системы $\mathcal{A}$. И то, и другое определяется индукцией по построению термов и формул соответственно.

Определение 6. Значение терма.

- *Для переменной $x \in Var$ $\sigma(x)$ уже определено в состоянии σ.*
- *Если t является константой $c \in \mathbf{C}$, то $\sigma(c)$ не зависит от σ и равно значению этой константы при данной интерпретации (напомним, в интерпретации каждой константе сопоставлен некоторый элемент основного множества).*
- *Если t имеет вид $f(t_1, \ldots, t_m)$, где $f^{(m)} \in \mathbf{F}$ — символ m-местной функции, а $t_1, \ldots, t_m$ — термы, то $\sigma(t)$ есть $f_A(\sigma(t_1), \ldots, \sigma(t_m))$, где f_A есть функция, соответствующая символу f в алгебраической системе $\mathcal{A}$.*

Таким образом, чтобы определить значение терма на состоянии, нужно подставить в него задаваемые этим состоянием значения переменных и вычислить значение получившейся суперпозиции функций на основном множестве системы.

Определение 7. Значение атомной формулы.

- *Если $\varphi = (t_1 = t_2)$, где t_1 и t_2 — термы, то $\sigma(\varphi) = 1$, если $\sigma(t_1) = \sigma(t_2)$, и $\sigma(\varphi) = 0$, если $\sigma(t_1) \neq \sigma(t_2)$.*
- *Пусть $\varphi = P^{(0)}$ — 0-местный предикатный символ из $\mathbf{P}$ и P_A — это 0-местный предикат, сопоставленный ему в $\mathcal{A}$. Тогда $\sigma(P) = P_A \in \{1, 0\}$.*
- *Пусть $\varphi = P(t_1, \ldots, t_k)$, где $P^{(k)}$ — предикатный k-местный символ из $\mathbf{P}$, $t_1, \ldots, t_k$ — термы, а P_A — это k-местный предикат, сопоставленный $P^{(k)}$ в $\mathcal{A}$. Тогда $\sigma(P(t_1, \ldots, t_k)) = P_A(\sigma(t_1), \ldots, \sigma(t_k))$.*

Таким образом, равенство термов на некотором состоянии истинно, если эти термы имеют на этом состоянии одинаковые значения. Для определения истинности предиката от набора термов нужно вычислить их значения на заданном состоянии и проверить входит ли получившийся набор значений во множество наборов, принадлежащих данному предикату.

Определение 8. Значение формулы.
Если φ — атомная формула, то ее значение $\sigma(\varphi)$ определено выше. Значения сложных формул определяются следующим образом.

- *Если $\varphi = \neg\varphi_1$, то $\sigma(\varphi) = \neg\sigma(\varphi_1)$.*
- *Если φ имеет одну из форм $(\theta\wedge\psi)$, $(\theta\vee\psi)$ или $(\theta \to \psi)$, то значение $\sigma(\varphi)$ равно значению $\sigma(\theta) \wedge \sigma(\psi)$, $\sigma(\theta) \vee \sigma(\psi)$ или $\sigma(\theta) \to \sigma(\psi)$, соответственно.*
- *Если $\varphi = \forall x\psi$, то $\sigma(\varphi) = 1$, если $\sigma'(\psi) = 1$ для любого состояния σ', которое может отличаться от σ только значением переменной x, т.е. такого σ', что $\sigma'(y) = \sigma(y)$ при $y \neq x$, иначе $\sigma(\varphi) = 0$.*
- *Если $\varphi = \exists x\psi$, то $\sigma(\varphi) = 1$, если существует такое состояние σ', которое может отличаться от σ только значением переменной x, для которого $\sigma'(\psi) = 1$, иначе $\sigma(\varphi) = 0$.*

Прокомментируем два последних пункта. Для истинности формулы вида $\forall x\psi$ (она читается: "для всех x имеет место (выполняется, истинно, справедливо) ψ") на состоянии σ нужно убедиться в том, что формула ψ истинна при замене x любым элементом $a \in A$, т.е. на любом состоянии, получающемся из σ изменением значения на переменной x: $\sigma(x) = a$. Для истинности формулы вида $\exists x\psi$ (она читается: "существует такое x, что имеет место (выполняется, истинно, справедливо) ψ") на состоянии σ достаточно найти такое значение $a \in A$, что при подстановке a вместо x в формулу ψ она будет истинна при неизменившихся значениях остальных свободных переменных.

Из приведенных определений нетрудно вывести, что значение формулы на некотором состоянии зависит лишь от значений ее свободных переменных в этом состоянии.

Предложение 1. *Пусть $\mathcal{A}$ — алгебраическая система сигнатуры Σ, φ — формула той же сигнатуры, $V(\varphi)$ — множество ее свободных переменных. Тогда для любых двух состояний $\mathcal{A}$ σ_1 и σ_2, совпадающих на $V(\varphi)$ (т.е. таких, что для всякой $x \in V(\varphi)$ $\sigma_1(x) = \sigma_2(x)$), имеет место равенство $\sigma_1(\varphi) = \sigma_2(\varphi)$.*

Доказательство получается индукцией по построению формулы φ. Действительно, утверждение, очевидно, справедливо для значений термов и атомных формул. Из предположения о его справедливости для формул φ_1 и φ_2 его легко перенести на формулы вида $\neg\varphi_1$ и $\varphi_1 \circ \varphi_2$, где $\circ \in \{\wedge, \vee, \to\}$. (*Проведите это рассуждение самостоятельно!*). Пусть теперь $\varphi = Qx\varphi_1$, где $Q \in \{\forall, \exists\}$, а $x \in \mathbf{Var}$. Тогда $V(\varphi) = V(\varphi_1) \setminus \{x\}$. Пусть σ_1 и σ_2 — состояния, совпадающие на $V(\varphi)$. Рассмотрим множества состояний, отличающихся от них только на x: $\sigma_\mathbf{1}^\mathbf{x} = \{\sigma' \mid \sigma'(y) = \sigma_1(y) \textit{ при } y \neq x\}$ и $\sigma_\mathbf{2}^\mathbf{x} == \{\sigma'' \mid \sigma''(y) = \sigma_2(y) \textit{ при } y \neq x\}$. Для каждого состояния $\sigma' \in \sigma_\mathbf{1}^\mathbf{x}$ в $\sigma_\mathbf{2}^\mathbf{x}$ имеется состояние σ'', совпадающее с σ' на $V(\varphi_1)$. Верно и обратное. Поэтому по предположению индукции в $\sigma_\mathbf{1}^\mathbf{x}$ имеется состояние σ', для которого $\sigma'(\varphi_1) = 1$ тогда и только тогда, когда в $\sigma_\mathbf{2}^\mathbf{x}$ имеется состояние σ'', для которого $\sigma''(\varphi_1) = 1$. Отсюда и из определения значения формулы $\varphi = Qx\varphi_1$ следует, что $\sigma_1(\varphi) = \sigma_2(\varphi)$.

Из этого предложения немедленно вытекает
Следствие. *Значение замкнутой формулы не зависит от состояния, на котором она оценивается, т.е. либо на всех состояниях алгебраической системы она истинна, либо на всех состояниях алгебраической системы эта формула ложна.*

Определение 9. *Формула φ называется* **истинной** *на алгебраической системе $\mathcal{A}$, если для любого состояния σ этой системы $\sigma(\varphi) = 1$. В этом случае пишем $\mathcal{A} \models \varphi$.*
Формула φ называется **тождественно истинной** *(общезначимой), если она истинна на всех алгебраических системах своей сигнатуры. В этом случае пишем $\models \varphi$. Тождественно истинные формулы называют также законами логики.*

В качестве примера рассмотрим алгебраическую систему $\mathcal{A}_1 =< A_1; \Sigma_1 >$ для определенной выше сигнатуры Σ_1. Пусть A_1 состоит из натуральных чисел $\{0, 1, 2, \ldots\}$, для которых истинен предикат *число* и людей {*петр, джон, ольга, иван, мария*}, для которых истинен предикат *человек* (мы используем для объектов-людей имена, начинающиеся со строчных букв, чтобы отличить их от имен констант из сигнатуры). Пусть, кроме того, A_1 включает специальный объект ∞, соответствующий неопределенному (или ошибочному) значению. Константы интерпретируются естественным образом: числовые — соответствующими числами, имена людей — людьми с теми же именами. Функция + на числах интерпретируется обычным сложением, а в остальных случаях ее значением является ∞. Интерпретации остальных функций заданы в следующей таблице (для числовых аргументов все они равны ∞) .

x	*отец(x)*	*лучший_друг(x)*	*зарплата(x)*
петр	иван	джон	20
джон	петр	ольга	35
ольга	джон	петр	20
иван	∞	мария	40
мария	иван	петр	30

Предикат $\leq$ интерпретируется обычным образом на числах и ложен, если хоть один его аргумент не число (мы будем для него использовать стандартную форму записи $x \leq y$ вместо $\leq (x, y)$). Остальные предикаты зададим перечислением пар, на которых они истинны. *живут_рядом* = {*(ольга,петр), (петр, ольга), (иван, джон), (джон, иван), (мария, ольга), (ольга, мария)* }
родственники = { *(ольга,петр), (петр, ольга), (джон, петр), (петр, джон), (ольга, джон), (джон, ольга), (иван, мария), (мария, иван)* } .

Пусть множество переменных $\mathbf{Var} = \{x, y, z, i, n, k, \ldots\}$ и состояние σ определяет их значения следующим образом: $\sigma(x) =$ *петр*, $\sigma(y) =$ *ольга*, $\sigma(z) =$ *иван*, $\sigma(i) = 1, \sigma(n) = 30, \sigma(k) = 2$.

Определим значения некоторых термов на σ:
$\sigma(+(7, +(2, 6)) = 15, \quad \sigma(+(i, +(2, n)) = 33,$
σ(*отец(лучший_друг("Мария"))) = джон* ,
σ(*зарплата(лучший_друг(отец(x))) = 30,*
$\sigma(+($*зарплата*$(y), k)) = 22, \quad \sigma(+(x, k)) = \infty.$

Рассмотрим теперь введенные выше формулы
$\varphi_1 = \forall x$(*человек(x)* $\to \exists y$(*человек(y)* $\wedge$ *живут_рядом(x,y))* и
$\varphi_2 = \exists y$[*человек(y)* $\wedge$ ($\forall x$ *родственники(x, y)*$\vee$ ($\neg$*живут_рядом*$(x, y) \wedge (35 \leq$ *зарплата(y)*)))].

Чтобы оценить первую из них на состоянии σ, мы должны проверить ее подформулу (*человек(x)* $\to \exists y$(*человек(y)* $\wedge$ *живут_рядом(x,y))* на всех состояниях σ', совпадающих с σ на всех переменных кроме, быть может, x. Если $\sigma'(x)$ — число или ∞, то σ'(*человек(x))=0*, а вся импликация = 1. Если же *человек*$(\sigma'(x)) = 1$, т. е $\sigma'(x) \in$ { *петр, джон, ольга, иван, мария*}, то легко проверить, что для каждого из этих значений x найдется такое значение $\sigma''(y)$, для которого выполнено *человек*$(\sigma''(y))$ и что имеет место *живут_рядом*$(\sigma'(x), \sigma''(y))$. Например, если $\sigma'(x) =$ *джон*, то в качестве $\sigma''(y)$ следует взять *иван.* Таким образом формула (*человек(y)* $\wedge$ *живут_рядом*(x, y)) будет иметь значение 1 на состоянии σ'', совпадающем с σ' на всех переменных кроме, быть может, y. Следовательно, мы показали, что $\sigma(\varphi_1) = 1$. Заметим, что так как φ_1 – замкнутая формула, то на самом деле, мы показали, что формула она имеет значение 1 на любом состоянии алгебраической системы $\mathcal{A}_1$, т.е. что $\mathcal{A}_1 \models \varphi_1$.

Для оценки значения $\sigma(\varphi_2)$ нужно постараться найти такого человека $\tilde{y}$, для которого выполняется хотя бы одна из подформул $\forall x$ *родственники*$(x, \tilde{y})$ или ($\neg$*живут_рядом(петр,* $\tilde{y}$) $\wedge$(*зарплата*$(\tilde{y}) \geq 35$)) (напомним, что $\sigma(x) =$ *петр*). Из определения предиката *родственники* в $\mathcal{A}_1$ следует, что первая из этих формул не выполняется ни для какого $\tilde{y}$. Что касается второй подформулы, то она верна при $\tilde{y} =$ *иван* или $\tilde{y} =$ *джон*, поскольку оба они не являются соседями *петр*а и имеют зарплату ≥ 35. Таким образом, $\sigma(\varphi_2) = 1$.

Задача 1. *Определите значение $\sigma_1(\varphi_2)$ формулы φ_2 на состоянии σ_1, отличающемся от σ только значением $\sigma_1(x)$ =иван.*

Задача 2. *Выделите в следующих формулах свободные и связанные вхождения переменных и определите значения этих формул на состоянии σ алгебраической системы $\mathcal{A}_1$.*
a) $\varphi_3 = \forall x \forall z$(*живут_рядом*$(x, z) \to$ *родственники*(x, z)).
b) $\varphi_4 = \forall x$(*живут_рядом*$(x, y) \to$ (*отец(x)* = *"Иван"*)).
c) $\varphi_5 = \forall u \exists v$[*живут_рядом(u, лучший_друг(v))* $\vee$ (*зарплата*$(x) \leq$ *зарплата*(v))]
d) $\varphi_6 = \exists v$[$\neg$*живут_рядом*$(v, z) \to$ (*зарплата(отец(v)* $\leq n$)].

Задача 3. *Запишите формулы, выражающие следующие свойства алгебраической системы $\mathcal{A}_1$ и проверьте их истинность на этой системе.*
а) Рядом с каждым человеком A живет некто, отец которого является лучшим другом A.
б) У каждого человека не более одного соседа.
в) Если у соседа некоторого человека A зарплата больше чем у A, то этот сосед является родственником A.

г) Есть человек, у которого не менее 2-х детей.
д) Никто не является лучшим другом для более чем 2-х людей.

Задача 4. *Запишите формулы, которые истинны на состояниях σ системы $\mathcal{A}_1$, удовлетворяющих следующим условиям.*
а) $\sigma(x)$ — человек, зарплата которого больше 20 и не больше 35.
б) $\sigma(x)$ и $\sigma(y)$ — это люди, у которых один и тот же лучший друг.
в) $\sigma(x)$ — человек с максимальной зарплатой, равной $\sigma(n)$.
г) $\sigma(x)$ — человек, получающий большую зарплату, чем его отец.
Найдите все значения переменных, на которых истинны соответствующие формулы.

Задача 5. *Арифметикой называется система $(\mathbf{N}; +^{(2)}, *^{(2)}; 0, 1)$, где основное множество $\mathbf{N} = \{0, 1, 2, ...\}$ — множество натуральных чисел, функции + и * — обычное сложение и умножение, а 0 и 1 — соответствующие числа.*

Запишите формулы со свободными переменными из $\{x, y, z\}$, которые истинны в арифметике тогда и только тогда, когда
а) x меньше y;
б) x является четным числом;
в) y является простым числом;
г) x и y взаимно просты;
д) z лежит в интервале между x и y;
е) существует бесконечно много простых чисел;
ж) сложение (умножение) коммутативно.

Задача 6. *Представьте каждое из следующих предложений формулой логики предикатов, определив в каждом случае подходящую сигнатуру.*
а) Не все студенты изучают и анализ, и историю.
б) Только один студент не сдал экзамен по дискретной математике.
в) Только один студент сдал все экзамены на отлично.
г) Максимальные баллы, полученные по дискретной математике, превышают максимальные баллы, полученные по информатике.
д) Каждый, кто не любит всех вегетарианцев, является странным.
е) Имеется брадобрей, который бреет только тех жителей города, которые не бреются сами.
ж) Политики могут обманывать всех людей некоторое время, они могут обманывать некоторых людей все время, но они не могут обманывать всех людей все время.

Задача 7. *Напишите формулу логики предикатов, которая истинна только на всех системах, основное множество которых состоит из одного элемента (2-х , 3-х,..., k элементов).*

1.3 Эквивалентные формулы и нормальные формы

С понятием эквивалентности формул мы уже знакомились, когда рассматривали булевы формулы (формулы логики высказываний). Для логики предикатов эквивалентность определяется следующим образом.

Определение 10. *Формулы φ и ψ сигнатуры Σ называются эквивалентными, если для любой алгебраической системы $\mathcal{A}$ этой сигнатуры и любого ее состояния σ имеет место равенство $\sigma(\varphi) = \sigma(\psi)$. В этом случае пишем $\varphi \equiv \psi$.*

Отметим сначала, что все эквивалентности булевых формул – законы ассоциативности и коммутативности для конъюнкции и дизъюнкции, законы дистрибутивности, двойного отрицания, де Моргана и др. – остаются справедливыми и для формул логики предикатов. Также остается справедливым **правило замены эквивалентных подформул:**
если ψ является подформулой формулы φ, $\psi \equiv \psi_1$ и формула φ_1 получена из формулы φ заменой некоторого вхождения ψ на ψ_1, то $\varphi \equiv \varphi_1$.

Задача 8. *Докажите правило замены эквивалентных подформул, используя индукцию по построению формул и подформул.*

Новые интересные эквивалентности связаны с кванторами.

Теорема 1. *Для любых формул $\varphi(x)$ и ψ таких, что ψ не содержит свободно переменной x, имеют место следующие эквивалентности.*

(1) $\neg\forall x\varphi(x) \equiv \exists x\neg\varphi(x)$,	(2) $\neg\exists x\varphi(x) \equiv \forall x\neg\varphi(x)$,
(3) $(\psi\wedge\forall x\varphi(x)) \equiv \forall x(\psi\wedge\varphi(x))$,	(4) $(\psi\wedge\exists x\varphi(x)) \equiv \exists x(\psi\wedge\varphi(x))$,
(5) $(\psi\vee\forall x\varphi(x)) \equiv \forall x(\psi\vee\varphi(x))$,	(6) $(\psi\vee\exists x\varphi(x)) \equiv \exists x(\psi\vee\varphi(x))$,
(7) $(\psi\to\forall x\varphi(x)) \equiv \forall x(\psi\to\varphi(x))$,	(8) $(\psi\to\exists x\varphi(x)) \equiv \exists x(\psi\to\varphi(x))$,
(9) $(\forall x\varphi(x)\to\psi) \equiv \exists x(\varphi(x)\to\psi)$,	(10) $(\exists x\varphi(x)\to\psi) \equiv \forall x(\varphi(x)\to\psi)$

Доказательства всех этих эквивалентностей получаются непосредственно из определения значения формул на состояниях. Рассмотрим, например, первую из них. Для любого состояния σ имеем $\sigma(\neg\forall x\varphi(x)) \equiv \neg\sigma(\forall x\varphi(x))$. Это значение равно 1 тогда и только тогда, когда $\sigma(\forall x\varphi(x)) = 0$ или, по определению значения квантора $\forall$, когда найдется такое состояние σ', совпадающее с σ вне x, для которого $\sigma'(\varphi(x)) = 0$. Но это эквивалентно истинности формулы $\exists x\neg\varphi(x)$ на состоянии σ.

Рассмотрим еще эквивалентности (9) и (10), которые, на первый взгляд, могут удивить — квантор всеобщности превращается при вынесении за скобки в квантор существования и наоборот. Доказательство (9) получается с помощью булевых эквивалентностей и пунктов (1) и (4) настоящей теоремы:
$(\forall x\varphi(x)\to\psi) \equiv (\neg\forall x\varphi(x)\vee\psi) \equiv (\exists x\neg\varphi(x)\vee\psi) \equiv \exists x(\neg\varphi(x)\vee\psi) \equiv \exists x(\varphi(x)\to\psi)$.
Эквивалентность (10) доказывается аналогично.

Задача 9. *Проведите доказательство эквивалентностей (2) - (8).*

Задача 10. *Доказажите следующие эквивалентности, показывающие, что одноименные кванторы можно менять местами:*
а) $\forall x\forall y\varphi \equiv \forall y\forall x\varphi$,
б) $\exists x\exists y\varphi \equiv \exists y\exists x\varphi$.

Задача 11. *Докажите, что разноименные кванторы менять местами нельзя, т.е., что формулы $\forall x\exists y\varphi$ и $\exists x\forall y\varphi$ не эквивалентны.*

Останутся ли эквивалентности (2) - (8) справедливыми, если мы не будем требовать, чтобы формула ψ не содержала свободных вхождений переменной x? Нет! Рассмотрим, например, для арифметической сигнатуры формулы $\varphi = (x = x)$ и $\psi = (x = 0)$. Тогда формула $(\psi\wedge\forall x\varphi) = ((x = 0)\wedge\forall x(x = x))$, представляющая левую часть эквивалентности (3) истинна на состоянии σ, в котором $\sigma(x) = 0$. А формула $\forall x((x = 0)\wedge(x = x))$, представляющая ее правую часть ложна на всех состояниях. Чтобы корректно вынести за скобки квантор в формуле $(\psi(x)\wedge\forall x\varphi(x))$, в которой $\psi(x)$ содержит x свободно, нужно предварительно переименовать связанную переменную в $\forall x\varphi(x)$. Это позволяют сделать следующие эквивалентности, справедливость которых следует из предложения 1.

Лемма 1. О замене связанных переменных.
1) $\forall x\varphi(x) \equiv \forall y\varphi(y)$, *где $\varphi(x)$ не содержит y, а $\varphi(y)$ получается заменой всех свободных вхождений x в $\varphi(x)$ на y.*
2) $\exists x\varphi(x) \equiv \exists y\varphi(y)$, *где $\varphi(x)$ не содержит , а $\varphi(y)$ получается заменой всех свободных вхождений x в $\varphi(x)$ на y.*

Используя эту лемму, можем получить следующую цепочку эквивалентностей: $((x = 0)\wedge\forall x(x = x)) \equiv ((x = 0)\wedge\forall y(y = y)) \equiv \forall y((x = 0)\wedge(y = y))$. В общем случае, если переменная y не входит в $\varphi(x)$, справедлива эквивалентность $(\psi(x)\wedge\forall x\varphi(x)) \equiv \forall y(\psi(x)\wedge\varphi(y))$.

Определение 11. *Формула вида $Q_1x_1Q_2x_2\dots Q_nx_n\varphi$, где φ — бескванторная формула, а каждый символ Q_i — это один из кванторов $\forall$ или $\exists$, называется* предваренной (или пренексной) *нормальной формой. Последовательность кванторов $Q_1x_1Q_2x_2\dots Q_nx_n$ называется ее приставкой, а бескванорная формула φ — матрицей этой нормальной формы.*

Теорема 2. О предваренной форме.
Для всякой формулы φ существует формула ψ в предваренной нормальной форме такая, что $\varphi \equiv \psi$.

Доказательство следует из теоремы 1 и леммы 1. Мы приведем его в виде процедуры, которая позволяет построить предваренную фому ψ по исходной формуле φ. Процедура состоит из двух этапов.

На первом этапе, используя лемму 1, заменим все связанные переменные в φ на новые так, чтобы (1) разные кванторы связывали разные переменные и

(2) имена всех связанных переменных не пересекались с именами свободных переменных формулы φ.

На втором этапе используя эквивалентности (3) - (10) теоремы 1 поочередно выносим кванторы за скобки. Именно, последовательно заменяем каждую подформулу вида $(\varphi_1 \circ Qz\varphi_2)$, где $Q \in \{\forall, \exists\}, \circ \in \{\wedge, \vee, \rightarrow\}$ на эквивалентную подформулу $Qz(\varphi_1 \circ \varphi_2)$, и каждую подформулу вида $(Qz\varphi_1 \rightarrow \varphi_2)$ на подформулу $Q'z(\varphi_1 \rightarrow \varphi_2)$, где $Q' = \exists$ при $Q = \forall$ и $Q' = \forall$ при $Q = \exists$. Если после этого вынесенный квантор стоит сразу после операции отрицания $\neg$, то применяя эквивалентности (1), (2) проносим отрицание в глубь формулы. Заметим, что порядок выноса кванторов определен неоднозначно, в зависимости от него в результате могут получиться эквивалентные формулы с разными кванторными приставками.

Рассмотрим пример. Пусть $\varphi = (\exists y P(x, y) \wedge \neg(\forall x P(x, y) \rightarrow \exists x Q(x, z)))$. Ее свободные переменные: x, y и z (у x, y имеются также и связанные вхождения). Выберем в качестве новых имен для связанных переменных u, v и w и заменим ими "старые"связанные переменные. Получим эквивалентную формулу $\varphi_1 = (\exists u P(x, u) \wedge \neg(\forall v P(v, y) \rightarrow \exists w Q(w, z)))$. На следующем этапе выносим кванторы за скобки (в качестве индексов у знаков эквивалентности указаны номера применяемых эквивалентностей):
$\varphi_1 \equiv_{(4)} \exists u(P(x, u) \wedge \neg(\forall v P(v, y) \rightarrow \exists w Q(w, z))) \equiv_{(8)} \exists u(P(x, u) \wedge \neg\exists w(\forall v P(v, y) \rightarrow Q(w, z))$
$\equiv_{(9)} \exists u(P(x, u) \wedge \neg\exists w \exists v(P(v, y) \rightarrow Q(w, z)) \equiv_{(2)(2)} \exists u(P(x, u) \wedge \forall w \forall v \neg(P(v, y) \rightarrow Q(w, z))$
$\equiv_{(3)(3)} \forall v \forall w \exists u(P(x, u) \wedge \neg(P(v, y) \rightarrow Q(w, z))$.
Далее можно упрощать матрицу формулы, используя известные эквивалентности булевой логики. В данном случае получим формулу $\forall v \forall w \exists u(P(x, u) \wedge P(v, y) \wedge \neg Q(w, z))$.

Задача 12. *Приведите к предваренной нормальной форме следующие формулы, считая, что φ и ψ — бескванторные.*
а) $\neg\forall z \neg\forall x \exists y \forall u \varphi$,
б) $\exists x \forall y \psi(x, y) \rightarrow \exists x \forall y \psi(x, y)$,
в) $\neg(\forall x \forall y \varphi(x, y, z) \vee (\forall z \psi(x, z) \wedge \neg\exists y \psi(x, y)))$.

1.4 Логика предикатов и базы данных

Большинство современных промышленных баз данных являются *реляционными* — данные в них представляют конечные отношения (relations), которые хранятся в таблицах. *Схема отношения* $R(A_1, A_2, \ldots, A_n)$ включает имя отношения R и имена его атрибутов $A_1, A_2, \ldots, A_n$. Для каждого атрибута A_i определено множество $dom(A_i)$ его допустимых значений. *Схема базы данных* состоит из перечня схем отношений, входящих в эту базу. В приложениях отношения чаще называют *таблицами*, их атрибуты — *столбцами*, строки таблиц — *записями*, а их элементы — *полями*. В каждый момент времени состояние базы данных — это набор (конечных) таблиц с соответствующими схемами.

Пусть, например, база данных со сведениями о сотрудниках некоторой организации имеет схему: *Сотрудники(Номер, ФИО, Отдел, Должность, Оклад), Комнаты(НомерСотрудника, Этаж, НомерКомнаты)*. Рассмотрим некоторое состояние этой базы данных.

Сотрудники

Номер	**ФИО**	**Отдел**	**Должность**	**Оклад**
1	Иванов А.А.	торговый	менеджер	7000
2	Сидоров Н.П.	плановый	экономист	5000
3	Сидорова М.И.	торговый	зав.складом	6000
4	Ольгина Н.А.	плановый	экономист	5500
5	Горев С.В.	плановый	зав.отделом	10000

Комнаты

НомерСотрудника	**Этаж**	**НомерКомнаты**
3	2	17
1	2	17
7	2	18
5	3	7
2	3	27

С точки зрения логики предикатов это состояние не что иное, как некоторая модель сигнатуры $\Sigma_2 = \{\text{\textit{Сотрудники}}^{(5)}, \text{\textit{Комнаты}}^{(3)}\}$ с основным множеством, включающим строки и числа. Первая из приведенных таблиц задает интерпретацию предиката $\text{\textit{Сотрудники}}^{(5)}$, а вторая — интерпретацию предиката *Комнаты*$^{(3)}$.

Запросы

Пользователи извлекают информацию из баз данных с помощью запросов. Одним из самых популярных языков запросов является язык SQL (Structured Query Language – Структурный Язык Запросов), фактически он сейчас является стандартным средством для всех промышленных баз данных. Мы опишем здесь лишь простую форму основного оператора этого языка SELECT (ВЫБРАТЬ). Она имеет вид:

SELECT <список полей (атрибутов)>
FROM <список таблиц>
WHERE <условия выбора>,
где <список полей (атрибутов)> — список тех полей (или атрибутов) таблиц, значения которых будут входить в результаты запроса, <список таблиц> — список таблиц, из которых берутся значения полей (если некоторые из них имеют поля с одинаковыми именами, то их именуют как <имя таблицы>.<имя поля>), <условия выбора> — условия, которым должны удовлетворять выбираемые значения полей, они представляют из себя бескванторную формулу, построенную из атомных формул, в которых логические знаки $\neg, \wedge, \vee$ заменены на их английские аналоги **NOT, AND, OR**, соответственно. В качестве переменных используются имена полей, в качестве предикатов — кроме $=$, для числовых полей используются отношения неравенства: $>$, $\geq$, $<$, $\leq$. В термах для числовых полей можно использовать арифметические функции $+, -, *, /$. В термах для полей других типов можно, обычно, использовать некоторый набор стандартных функций соответствующих типов.

Пусть, например, мы хотим получить список фамилий всех сотрудников, работающих на 2-ом этаже. Соответствующий запрос выглядит так:

SELECT ФИО
FROM Сотрудники, Комнаты
WHERE Номер = НомерСотрудника **AND** Этаж = 2.

Нетрудно построить формулу логики предикатов, реализующую этот запрос:

$\varphi_1 = \exists$*Номер* $\exists$*Отдел* $\exists$*Должность* $\exists$*Оклад* $\exists$*НомерСотрудника* $\exists$*Этаж* $\exists$*НомерКомнаты*[*(Сотрудники(Номер, ФИО, Отдел, Должность, Оклад)* $\wedge$ *Комнаты(НомерСотрудника, Этаж, НомерКомнаты))* $\wedge$ ((*Номер* = *НомерСотрудника*) $\wedge$ (*Этаж* $= 2$))].

В качестве переменных в этой формуле используются имена полей (атрибутов) таблиц. Свободными переменными являются выбираемые поля (в списке после SELECT). Все остальные переменные связаны кванторами существования в кванторной приставке. Матрица формулы состоит из конъюнкции двух формул: первая представляет собой конъюнкцию предикатов-таблиц, а вторая — это формула из условий выбора. Построенная таким образом формула определяет множество состояний,на которых она истинна, а каждое из этих состяний задает набор значений свободных переменных формулы, т.е. тех полей, о которых задан запрос. Этот набор и является ответом на запрос (результатом запроса).

В нашем случае, φ_1 истинна на состояниях со значениями ФИО *Иванов А.А* и *Сидорова М.И.*, которые и являются ответом на исходный запрос.

Рассмотрим еще один пример. Пусть необходимо получить список сотрудников планового отдела с указанием комнат, в которых они трудятся. Запрос на SQL выглядит так:

SELECT ФИО, НомерКомнаты
FROM Сотрудники, Комнаты
WHERE Номер = НомерСотрудника **AND** Отдел = “плановый".

По описанным выше правилам построим формулу логики предикатов, реализующую этот запрос. При этом заменим имена связанных переменных более короткими.

$\varphi_2 = \exists x_1 \exists x_2 \exists x_3 \exists x_4 \exists x_5 \exists x_6[$ *(Сотрудники*$(x_1,$ *ФИО*$, x_2, x_3, x_4) \wedge$ *Комнаты*$(x_5, x_6,$ *НомерКомнаты*$)) \wedge$ $((x_1 = x_5) \wedge (x_2 =$ *“плановый"))]*.

Эта формула истинна на состояниях со следующими парами свободных переменных *(ФИО, НомерКомнаты)* (мы представили их в виде новой таблицы):

ФИО	**НомерКомнаты**
Сидоров Н.П.	27
Горев С.В.	7

Заметим, что в нашей базе данных нет сведений о комнате сотрудницы планового отдела Ольгиной Н.А.

Таким образом, описанные выше SQL-запросы являются ни чем иным, как “синтаксическим сахаром", за которым скрываются формулы логики предикатов весьма специального вида (такие формулы называются $\exists$-формулами). Имеются и более сложные формы SQL-запросов, которым соответствуют формулы логики предикатов более общего вида. Но в любом случае SQL-запросы не выходят за рамки логики предикатов.

Задача 13. *Напишите SQL-запросы и соответствующие формулы для получения информации из базы данных* (Сотрудники, Комнаты) *и получите ответы на эти запросы.*
а) Найти всех сотрудников с окладом больше 5500.
б) Найти все отделы, в которых есть сотрудники с окладом $>$ 8000.
в) Составить список должностей и получаемых по ним окладов.
г) Составить список сотрудников торгового отдела, получающих зарплату от 6000 до 6500 и работающих не на 3-ем этаже.
д) Составить список комнат, в которых есть сотрудники с окладом меньше 5500 или больше 7500.

Ограничения целостности

Состояние базы данных постоянно изменяется. В нее добавляются новые записи, изменяются и удаляются старые. При этом нужно, чтобы всех модификациях состояние оставалось корректным (например, для каждого сотрудника имелась лишь одна запись в таблице Сотрудники). *Ограничение целостности* — это условие, задаваемое для схемы базы данных, которое ограничивает множество возможных состояний базы данных. Вот типичные виды ограничений целостности:
- *ограничение на ключи*: в таблице не должно быть двух строк с одинаковым значением некоторого поля (оно назывется ключом). (В более общем виде оно утверждает, что в таблице не должно быть двух строк с одинаковыми значениями нескольких заданных полей (такой набор полей также называется ключом таблицы);
- *ограничение на ссылки* : значение некоторого поля в одной таблице должно быть среди значений некоторого поля в другой таблице. Это ограничение не позволит удалить из второй таблицы запись, на которую имеется ссылка из первой таблицы;
- *ограничение на значение* : значение некоторого поля в таблице должно удовлетворять заданному условию (принадлежать определенному интервалу, быть больше (меньше) заданного числа, быть строкой длины не больше (не меньше) заданной, быть строкой, удовлетворяющей некоторому образцу и т.п.).

Современные системы управления базами данных позволяют задавать такие ограничения при конструировании базы данных, а затем автоматически поддерживают их выполнение, не давая пользователям производить модификации, которые могут нарушить эти ограничения.

С точки зрения логики предикатов, ограничение целостности — это замкнутая формула, которая должна быть истинна на допустимых состояниях базы данных. Рассмотрим, как можно задать ограничения целостности указанных видов для нашей базы данных *(Сотрудники, Комнаты).*

1) В таблице *Сотрудники* ключом является поле *Номер*:

$$\Phi_1 = \forall\ \textit{Номер}\ \forall x\ \forall y\ \forall z\ \forall v\ \forall x_1\ \forall y_1\ \forall z_1\ \forall v_1\ [(\textit{Сотрудники}(\ \textit{Номер}, x, y, z, v) \wedge \textit{Сотрудники}(\ \textit{Номер}, x_1, y_1, z_1, v_1)) \to ((x = x_1)\ \wedge\ (y = y_1)\ \wedge\ (z = z_1)\ \wedge\ (v = v_1))].$$

2) Каждый сотрудник, для которого определена комната в таблице *Комнаты*, должен присутствовать в таблице *Сотрудники*:

$$\Phi_2 = \forall\ \textit{НомерСотрудника}\ \forall x\ \forall y[\textit{Комнаты(НомерСотрудника}, x, y) \to \exists z\ \exists u\ \exists v\ \exists w(\textit{Сотрудники}(\ \textit{Номер}, z, u, v, w)\ \wedge\ (\ \textit{Номер} =\ \textit{НомерСотрудника}))].$$

3) Оклад каждого сотрудника должен лежать в интервале (1000, 25000):

$$\Phi_3 = \forall x\ \forall y\ \forall z\ \forall v\ \forall\ \textit{Оклад}\ (\textit{Сотрудники}(x, y, z, v,\ \textit{Оклад})\ \to ((1000 <\ \textit{Оклад}) \wedge (\ \textit{Оклад} < 25000))).$$

Задача 14. *Определите, какие из приведенных ограничений целостности Φ_1, Φ_2, Φ_3 выполняются для приведенного выше состояния базы данных* (Сотрудники, Комнаты).

Задача 15. *Напишите формулы, выражающие следующие ограничения целостности для базы* (Сотрудники, Комнаты) *и определите, какие из них выполняются для приведенного выше ее состояния.*
а) В таблице Комнаты *набор полей (НомерСотрудника, Комната) является ключом.*
б) Для каждого сотрудника из таблицы Сотрудники *в таблице* Комнаты *определено его место работы.*
в) Номера всех комнат на 2-ом этаже больше 10, но меньше 20, а всех комнат на 3-ем этаже больше 20.