Министерство образования и науки Российской Федерации

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

С. В. СУДОПЛАТОВ, Е. В. ОВЧИННИКОВА

МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ

УЧЕБНИК

для дистанционного образования

НОВОСИБИРСК
2006

Рецензенты: *А. Г. Пинус* — д-р физ.-мат. наук, проф.;
В. М. Зыбарев — канд. техн.наук, доц.

Сергей Владимирович Судоплатов
Елена Викторовна Овчинникова

Математическая логика и теория алгоритмов

В книге излагаются основные классические исчисления математической логики: исчисления высказываний и исчисления предикатов, а также элементы теории алгоритмов и неклассических логик.

Книга предназначена для студентов технических вузов, изучающих математическую логику и теорию алгоритмов в рамках дистанционного образования.

Оглавление

ПРЕДИСЛОВИЕ

Книга предназначена для студентов младших курсов технических вузов, изучающих математическую логику и теорию алгоритмов дистанционно, и написана на основе учебника *Судоплатов С. В., Овчинникова Е. В.* Математическая логика и теория алгоритмов: Учебник — М.:ИНФРА-М, Новосибирск: Изд-во НГТУ, 2004, доступного через библиотеку НГТУ, киоск НГТУ или Интернет-магазины России.

Материал учебника составлен в соответствии с рабочими программами курсов математической логики и теории алгоритмов, читаемых в НГТУ, и опирается на учебник *Судоплатов С. В., Овчинникова Е. В.* Дискретная математика : Учебник — М.:ИНФРА-М, Новосибирск: Изд-во НГТУ, 2005.

Учебник включает четыре раздела: исчисления высказываний, логика и исчисления предикатов, элементы теории алгоритмов, неклассические логики. В конце книги приведены варианты контрольной работы.

Вариант N контрольной работы студента вычисляется по формуле $N = k + 25 \cdot m$, где k — число, состоящее из последних двух цифр личного шифра студента, а целое число m выбирается так, чтобы значение N находилось в пределах от 1 до 25.

Перед решением задач контрольной работы полезно прорешать задачи и упражнения, помещенные в конце соответствующих разделов. В конце каждой главы помещены ссылки на примеры, которые являются аналогами соответствующих задач контрольной работы.

Г л а в а 1

ИСЧИСЛЕНИЯ ВЫСКАЗЫВАНИЙ

§ 1.1. Определение формального исчисления

Введем общее понятие формального исчисления. Будем говорить, что *формальное исчисление* I определено, если выполняются следующие четыре условия:

1. Имеется некоторое множество $A(I)$ — *алфавит* исчисления I. Элементы алфавита $A(I)$ называются *символами*. Конечные последовательности символов называются *словами исчисления* I. Обозначим через $W(I)$ множество всех слов алфавита исчисления I.

2. Задано подмножество $E(I) \subseteq W(I)$, называемое *множеством выражений исчисления* I. Элементы множества $E(I)$ называются *формулами* или *секвенциями*.

3. Выделено множество $\mathrm{Ax}(I) \subseteq E(I)$ выражений исчисления I, называемых *аксиомами* исчисления I.

4. Имеется конечное множество $\mathfrak{R}(I)$ частичных операций R_1, R_2, …, R_n на множестве $E(I)$, называемых *правилами вывода исчисления* I.

Итак, исчисление I есть четверка $\langle A(I), E(I), \mathrm{Ax}(I), \mathfrak{R}(I)\rangle$.

Если набор выражений $(\Phi_1, \ldots, \Phi_m, \Phi)$ принадлежит правилу R_i, то выражения $\Phi_1, \ldots, \Phi_m$ называются *посылками*, а выражение Φ — *непосредственным следствием выражений* Φ_1, …, Φ_m *по правилу* R_i, или *заключением правила* R_i. Записываться этот факт будет следующим образом:

$$\frac{\Phi_1; \ldots; \Phi_m}{\Phi} i.$$

Иногда в этой записи символ i будет опускаться, если из контекста ясно, о каком правиле идет речь:

$$\frac{\Phi_1; \ldots; \Phi_m}{\Phi}.$$

Выводом в исчислении I называется последовательность выражений Φ_1, Φ_2, ..., Φ_n такая, что для любого i $(1 \leqslant i \leqslant n)$ выражение Φ_i есть либо аксиома исчисления I, либо непосредственное следствие каких-либо предыдущих выражений.

Выражение Φ называется *теоремой исчисления* I, *выводимым в* I или *доказуемым в* I, если существует вывод Φ_1, ..., Φ_n, Φ, который называется *выводом выражения* Φ или *доказательством* теоремы Φ.

П р и м е р 1.1.1. 1. Пусть $E(I)$ — множество повествовательных предложений русского языка, $\mathrm{Ax}(I)$ — множество истинных в данный момент времени предложений вида "подлежащее сказуемое" с точкой на конце. Имея правила вывода

$$\frac{\Phi.;\, \Psi.}{\Phi \text{ и } \Psi.} \quad \text{и} \quad \frac{\Phi.;\, \Psi.}{\Phi \text{ или } \Psi.},$$

можно из простых предложений (аксиом) составлять более сложные (теоремы), также истинные в данный момент времени.

2. Пусть $E(I)$ — множество программ P_f, производящих вычисления значений одноместных числовых функций f, $\mathrm{Ax}(I)$ — множество "простых" программ. Для программ $P_f, P_g \in E(I)$ обозначим через $P_f \circ P_g$ программу, составленную из программ P_f и P_g так, что по любым начальным данным производятся вычисления значений функции f, а затем полученные значения используются как начальные данные, по которым вычисляется значение функции g. Правило вывода

$$\frac{P_f;\, P_g}{P_f \circ P_g}$$

позволяет задать формальное исчисление, в котором из простых программ (аксиом) можно составлять более сложные (теоремы).

Вообще говоря, может не существовать алгоритма, с помощью которого для произвольного выражения Φ формального исчисления I за конечное число шагов можно определить, является ли Φ выводимым в I или нет. Если такой алгоритм существует, то исчисление называется *разрешимым*, а если такого алгоритма нет — *неразрешимым*.

Исчисление называется *непротиворечивым*, если не все его выражения доказуемы.

Ниже мы проведем построение двух формальных исчислений — исчислений высказываний, в основе которых лежат формулы алгебры логики. Первое из этих исчислений — *исчисление высказываний генценовского типа*, предложенное Генценом, в качестве выражений использует секвенции, построенные из формул алгебры логики. Это исчисление будет обозначаться через ИС. Второе исчисление — *исчисление высказываний гильбертовского типа*, создано Гильбертом, и в

нем выражениями являются непосредственно формулы алгебры логики. Исчисление высказываний гильбертовского типа будет обозначаться через ИВ.

Мы покажем, что оба исчисления эквивалентны в том смысле, что доказуемыми в них будут являться в точности тождественно истинные формулы. Из последнего факта будут вытекать разрешимость и непротиворечивость исчислений высказываний.

Строящиеся в дальнейшем исчисления ИПС и ИП, являющиеся расширениями исчислений ИС и ИВ соответственно, послужат примерами неразрешимых непротиворечивых исчислений.

§ 1.2. Исчисление высказываний генценовского типа

Определим элементы *исчисления высказываний* ИС.

Алфавит ИС состоит из букв A, B, Q, P, R и других, возможно, с индексами (которые называются *пропозициональными переменными*), *логических* символов (связок) отрицания $\neg$, конъюнкции $\wedge$, дизъюнкции $\vee$, импликации $\to$, следования $\vdash$, а также *вспомогательных* символов: левой скобки (, правой скобки), запятой ,.

Множество формул ИС определяется индуктивно:

а) все пропозициональные переменные являются формулами ИС (такие формулы называются *элементарными* или *атомарными*);

б) если φ, ψ — формулы ИС, то $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$ — формулы ИС;

в) выражение является формулой ИС тогда и только тогда, когда это может быть установлено с помощью пунктов “а” и “б”.

Таким образом, любая формула ИС строится из пропозициональных переменных с помощью связок $\neg$, $\wedge$, $\vee$, $\to$.

В дальнейшем при записи формул будем опускать некоторые скобки, используя те же соглашения, что и в алгебре логики.

Секвенциями называются конечные выражения следующих двух видов, где $\varphi_1, \ldots, \varphi_n, \psi$ — формулы ИС:

$\varphi_1, \ldots, \varphi_n \vdash \psi$ (“*из истинности* $\varphi_1, \ldots, \varphi_n$ *следует* ψ”),

$\varphi_1, \ldots, \varphi_n \vdash$ (“*система формул* $\varphi_1, \ldots, \varphi_n$ *противоречива*”).

Последовательности формул $\varphi_1, \ldots, \varphi_n$ в секвенциях будут часто обозначаться через Γ (возможно, с индексами): $\Gamma \vdash \psi$, $\Gamma \vdash$. При этом последовательность Γ считается пустой при $n = 0$. Значит, записи $\vdash \psi$ и $\vdash$ также являются секвенциями, первая из которых может читаться как утверждение о доказуемости формулы ψ. Смысл секвенции $\vdash$ будет указан в § 1.5.

Таким образом, наряду с формулами, символизирующими простые или сложные высказывания, секвенции являются записями утверждений, в которых выделяются посылки и заключение.

Множество *аксиом ИС* определяется следующей *схемой* секвенций: $\varphi \vdash \varphi$, где φ — произвольная формула ИС.

Аксиомами являются, например, секвенции $A \wedge \neg B \vdash A \wedge \neg B$ и $A \to \neg A_1 \to B \vee C \wedge \neg D \vdash A \to \neg A_1 \to B \vee C \wedge \neg D$.

Правила вывода ИС задаются следующими записями, где Γ, Γ_1 — произвольные (возможно пустые) конечные последовательности формул ИС, φ, ψ, χ — произвольные формулы ИС.

1. $\dfrac{\Gamma \vdash \varphi;\ \Gamma \vdash \psi}{\Gamma \vdash (\varphi \wedge \psi)}$ (введение $\wedge$).
2. $\dfrac{\Gamma \vdash (\varphi \wedge \psi)}{\Gamma \vdash \varphi}$ (удаление $\wedge$).
3. $\dfrac{\Gamma \vdash (\varphi \wedge \psi)}{\Gamma \vdash \psi}$ (удаление $\wedge$).
4. $\dfrac{\Gamma \vdash \varphi}{\Gamma \vdash (\varphi \vee \psi)}$ (введение $\vee$).
5. $\dfrac{\Gamma \vdash \psi}{\Gamma \vdash (\varphi \vee \psi)}$ (введение $\vee$).
6. $\dfrac{\Gamma, \varphi \vdash \psi;\ \Gamma, \chi \vdash \psi;\ \Gamma \vdash (\varphi \vee \chi)}{\Gamma \vdash \psi}$ (удаление $\vee$, или правило разбора двух случаев).
7. $\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash (\varphi \to \psi)}$ (введение $\to$).
8. $\dfrac{\Gamma \vdash \varphi;\ \Gamma \vdash (\varphi \to \psi)}{\Gamma \vdash \psi}$ (удаление $\to$).
9. $\dfrac{\Gamma, \neg\varphi \vdash}{\Gamma \vdash \varphi}$ (удаление $\neg$, или доказательство от противного).
10. $\dfrac{\Gamma \vdash \varphi;\ \Gamma \vdash \neg\varphi}{\Gamma \vdash}$ (выведение противоречия).
11. $\dfrac{\Gamma, \varphi, \psi, \Gamma_1 \vdash \chi}{\Gamma, \psi, \varphi, \Gamma_1 \vdash \chi}$ (перестановка посылок).
12. $\dfrac{\Gamma \vdash \varphi}{\Gamma, \psi \vdash \varphi}$ (утончение, или правило лишней посылки).

Вывод $S_0, \ldots, S_n$ в ИС называется *линейным доказательством*. Секвенция S называется *доказуемой* в ИС, или *теоремой* ИС, если существует линейное доказательство $S_0, \ldots, S_n$ в ИС, заканчивающееся секвенцией S: $S_n = S$. Формула φ называется *доказуемой* в ИС, если в ИС доказуема секвенция $\vdash \varphi$.

Определим по индукции понятие *дерева секвенций*:

1) любая секвенция является деревом секвенций;

2) если $D_0, \ldots, D_n$ — деревья секвенций и S — секвенция, то запись

$$\frac{D_0; \ldots; D_n}{S}$$

также является деревом секвенций;

3) любое дерево секвенций строится в соответствии с пп. 1 и 2.

Вхождением секвенции в дерево D называется место, которое секвенция занимает в дереве. Каждая секвенция может иметь несколько вхождений в дерево секвенций. Вхождение секвенции в дерево D, над (под) которым нет горизонтальной черты, называется *начальным* (соответственно *заключительным*).

Из определения дерева секвенций ясно, что начальных секвенций в дереве может быть несколько, а заключительная секвенция единственна.

Часть дерева, состоящая из секвенций, находящихся непосредственно над некоторой чертой, под той же чертой, а также самой черты, называется *переходом*.

Дерево D называется *доказательством* в ИС *в виде дерева*, если все его начальные секвенции суть аксиомы ИС, а переходы осуществляются по правилам 1–12. Дерево D называется *доказательством секвенции* S *в виде дерева* в ИС, или *деревом вывода* S в ИС, если D — доказательство в ИС и S — его заключительная секвенция.

Наличие линейного доказательства секвенции равносильно существованию доказательства секвенции в виде дерева:

Предложение 1.2.1. *Секвенция S имеет доказательство в ИС в виде дерева тогда и только тогда, когда S — теорема ИС.* □

Очевидно, что представление доказательства в виде дерева более наглядно и позволяет проследить все переходы по правилам вывода.

П р и м е р 1.2.1. 1. Следующее дерево демонстрирует доказуемость формулы $\varphi \vee \neg\varphi$ для любой формулы φ:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\neg\varphi\vdash\neg\varphi}{\neg\varphi\vdash\varphi\vee\neg\varphi}5}{\neg\varphi,\neg(\varphi\vee\neg\varphi)\vdash\varphi\vee\neg\varphi}12}{\neg(\varphi\vee\neg\varphi),\neg\varphi\vdash\varphi\vee\neg\varphi;}11 \quad \dfrac{\neg(\varphi\vee\neg\varphi)\vdash\neg(\varphi\vee\neg\varphi)}{\neg(\varphi\vee\neg\varphi),\neg\varphi\vdash\neg(\varphi\vee\neg\varphi)}12}{\dfrac{\neg(\varphi\vee\neg\varphi),\neg\varphi\vdash}{\neg(\varphi\vee\neg\varphi)\vdash\varphi}9}10}{\neg(\varphi\vee\neg\varphi)\vdash\varphi\vee\neg\varphi;}4 \quad \neg(\varphi\vee\neg\varphi)\vdash\neg(\varphi\vee\neg\varphi)}{\dfrac{\neg(\varphi\vee\neg\varphi)\vdash}{\vdash\varphi\vee\neg\varphi}9}10.$$

2. Приведем доказательство секвенций $\varphi \wedge \psi \vdash \psi \wedge \varphi$ в виде дерева для любых формул φ и ψ:

$$\dfrac{\dfrac{\varphi\wedge\psi\vdash\varphi\wedge\psi}{\varphi\wedge\psi\vdash\psi;}3 \quad \dfrac{\varphi\wedge\psi\vdash\varphi\wedge\psi}{\varphi\wedge\psi\vdash\varphi}2}{\varphi\wedge\psi\vdash\psi\wedge\varphi}1.$$

Правило $\dfrac{S_0,\ldots,S_n}{S}$ называется *допустимым* в ИС, если из выводимости в ИС секвенций $S_0,\ldots,S_n$ следует выводимость в ИС секвенции S.

Заметим, что допустимость правила равносильна тому, что его добавление в исчисление ИС не расширяет множество доказуемых секвенций.

Предложение 1.2.2. *Следующие правила допустимы в ИС:*

(а) $\dfrac{\psi_1,\ldots,\psi_m\vdash\varphi}{\chi_1,\ldots,\chi_n\vdash\varphi}$ *(расширение посылок);*

(б) $\dfrac{\psi_1,\ldots,\psi_m\vdash}{\chi_1,\ldots,\chi_n\vdash}$ *(расширение посылок),*

где в пп. (а) *и* (б) *выполняется* $\{\psi_1,\ldots,\psi_m\}\subseteq\{\chi_1,\ldots,\chi_n\}$*;*

(в) $\dfrac{\Gamma\vdash\varphi;\ \Gamma,\varphi\vdash\psi}{\Gamma\vdash\psi}$ *(сечение);*

(г) $\dfrac{\Gamma_1,\varphi,\psi,\Gamma\vdash\chi}{\Gamma_1,\varphi\wedge\psi,\Gamma\vdash\chi}$ *(объединение посылок);*

(д) $\dfrac{\Gamma_1,\varphi\wedge\psi,\Gamma\vdash\chi}{\Gamma_1,\varphi,\psi,\Gamma\vdash\chi}$ *(расщепление посылок);*

(е) $\dfrac{\Gamma,\varphi\vdash\chi;\ \Gamma,\psi\vdash\chi}{\Gamma,\varphi\vee\psi\vdash\chi}$ *(разбор случаев);*

(ж) $\dfrac{\Gamma\vdash\varphi\wedge\neg\varphi}{\Gamma\vdash}$ *(выведение противоречия);*

(з) $\dfrac{\Gamma \vdash}{\Gamma \vdash \psi}$ *(выведение из противоречия);*

(и) $\dfrac{\Gamma, \varphi \vdash}{\Gamma \vdash \neg\varphi}$ *(контрапозиция);*

(к) $\dfrac{\Gamma \vdash \varphi}{\Gamma, \neg\varphi \vdash}$ *(контрапозиция);*

(л) $\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma, \neg\psi \vdash \neg\varphi}$ *(контрапозиция);*

(м) $\dfrac{\Gamma, \neg\psi \vdash \neg\varphi}{\Gamma, \varphi \vdash \psi}$ *(доказательство от противного);*

(н) $\dfrac{\varphi_0, \ldots, \varphi_n \vdash \psi}{\vdash ((\varphi_0 \wedge \ldots \wedge \varphi_n) \to \psi)}$ *(введение $\wedge$ и $\to$);*

(о) $\dfrac{\vdash ((\varphi_0 \wedge \ldots \wedge \varphi_n) \to \psi)}{\varphi_0, \ldots, \varphi_n \vdash \psi}$ *(удаление $\wedge$ и $\to$).*

Доказательство. Допустимость правила $\dfrac{\Gamma, \varphi, \varphi \vdash \psi}{\Gamma, \varphi \vdash \psi}$ показывается следующим деревом:

$$\dfrac{\dfrac{\varphi \vdash \varphi}{\Gamma, \varphi \vdash \varphi;}{}^{11,12} \quad \dfrac{\Gamma, \varphi, \varphi \vdash \psi}{\Gamma, \varphi \vdash \varphi \to \psi}{}^{7}}{\Gamma, \varphi \vdash \psi}{}^{8}.$$

Допустимость правила (а) следует из допустимости указанного правила с помощью правил 11 и 12. Допустимость правила (б) вытекает из правил (а), (ж) и (з).

Докажем допустимость правила (з), а остальные правила оставим читателю в качестве упражнения.

Ясно, что секвенция $\Gamma \vdash$ может быть получена лишь применением правила 10. Поэтому из доказуемости секвенции $\Gamma \vdash$ следует доказуемость секвенций $\Gamma \vdash \varphi$ и $\Gamma \vdash \neg\varphi$ для некоторой формулы φ. Учитывая доказуемость последних секвенций, дерево

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash \varphi}{\Gamma, \neg\psi \vdash \varphi;}{}^{12} \quad \dfrac{\Gamma \vdash \neg\varphi}{\Gamma, \neg\psi \vdash \neg\varphi}{}^{12}}{\Gamma, \neg\psi \vdash}{}^{10}}{\Gamma \vdash \psi}{}^{9}$$

устанавливает доказуемость секвенции $\Gamma \vdash \psi$. $\square$

Использование допустимых правил вывода позволяет во многих случаях приводить сокращенные доказательства секвенций, которые при необходимости можно преобразовать в доказательства секвенций в виде деревьев в ИС.

Например, следующее дерево устанавливает доказуемость секвенции $(\varphi \to \psi), \neg\psi \vdash \neg\varphi$:

$$\dfrac{\dfrac{\dfrac{\varphi \vdash \varphi}{(\varphi \to \psi), \varphi \vdash \varphi;}{}^{11,12} \quad \dfrac{(\varphi \to \psi) \vdash (\varphi \to \psi)}{(\varphi \to \psi), \varphi \vdash (\varphi \to \psi)}{}^{12}}{(\varphi \to \psi), \varphi \vdash \psi}{}^{8}}{(\varphi \to \psi), \neg\psi \vdash \neg\varphi.}{}^{\text{л}}$$

Правило называется *равносильным*, если доказуемость (единственной) секвенции, стоящей над чертой, равносильна доказуемости секвенции, стоящей под чертой. Из предложения 1.2.2 вытекает, что следующие правила равносильны: 9, 11, (г), (д), (ж), (и), (к), (л), (м), (н), (о).

Пусть $V = \{P_i \mid i \in \omega\}$ — множество всех пропозициональных переменных ИС, $\mathcal{F}$ — множество всех формул ИС. Любая функция $s : V \to \mathcal{F}$ называется *подстановкой* пропозициональных переменных. Для любой формулы $\varphi \in \mathcal{F}$ обозначим через $s(\varphi)$ формулу, получающуюся из φ заменой всех пропозициональных переменных P, входящих в φ, на формулы $s(P)$.

П р и м е р 1.2.2. Если $s(P_0) = P_1 \vee \neg P_2$, $s(P_1) = P_1 \to \neg P_3$, $\varphi = \neg P_0 \wedge P_1$, то $s(\varphi) = \neg(P_1 \vee \neg P_2) \wedge (P_1 \to \neg P_3)$. □

Для любой секвенции (последовательности формул) R обозначим через $s(R)$ секвенцию (последовательность формул), получающуюся из R заменой всех пропозициональных переменных P, входящих в R, на формулы $s(P)$.

Таким образом, отображение s естественным образом расширяется на множество всех выражений исчисления ИС.

Следующая теорема утверждает, что подстановкой в доказуемую секвенцию произвольных формул вместо пропозициональных переменных получается также доказуемая секвенция.

Теорема 1.2.3. (теорема о подстановке). *Если s — подстановка, R — секвенция, то $\dfrac{R}{s(R)}$ — допустимое правило.* □

§ 1.3. Эквивалентность формул

Для любых формул φ и ψ ИС обозначим через $(\varphi \leftrightarrow \psi)$ формулу ИС $((\varphi \to \psi) \wedge (\psi \to \varphi))$. Напомним, что таблица истинности последней формулы совпадает с таблицей истинности формулы алгебры логики $(\varphi \leftrightarrow \psi)$.

Лемма 1.3.1. *Секвенция $\Gamma \vdash (\varphi \leftrightarrow \psi)$ доказуема тогда и только тогда, когда доказуемы секвенции $\Gamma, \varphi \vdash \psi$ и $\Gamma, \psi \vdash \varphi$.*

Доказательство. Предположим, что секвенция $\Gamma \vdash (\varphi \leftrightarrow \psi)$ доказуема. Тогда доказуемость секвенции $\Gamma, \varphi \vdash \psi$ устанавливается следующим деревом:

$$\dfrac{\dfrac{\varphi \vdash \varphi}{\Gamma, \varphi \vdash \varphi;}{\scriptstyle 11,12} \quad \dfrac{\Gamma \vdash (\varphi \to \psi) \wedge (\psi \to \varphi)}{\dfrac{\Gamma \vdash \varphi \to \psi}{\Gamma, \varphi \vdash \varphi \to \psi}{\scriptstyle 12}}{\scriptstyle 2}}{\Gamma, \varphi \vdash \psi}{\scriptstyle 8}$$

Доказуемость секвенции $\Gamma, \psi \vdash \varphi$ показывается аналогично.

Установим теперь доказуемость секвенции $\Gamma \vdash (\varphi \leftrightarrow \psi)$, предполагая, что секвенции $\Gamma, \varphi \vdash \psi$ и $\Gamma, \psi \vdash \varphi$ доказуемы:

$$\dfrac{\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi;}{\scriptstyle 7} \quad \dfrac{\Gamma, \psi \vdash \varphi}{\Gamma \vdash \psi \to \varphi}{\scriptstyle 7}}{\Gamma \vdash (\varphi \to \psi) \wedge (\psi \to \varphi)}{\scriptstyle 1}. \ \square$$

Формулы φ и ψ называются *эквивалентными* (обозначается $\varphi \equiv \psi$), если в ИС доказуемы секвенции $\varphi \vdash \psi$ и $\psi \vdash \varphi$.

В силу леммы 1.3.2 условие $\varphi \equiv \psi$ равносильно доказуемости секвенции $\vdash (\varphi \leftrightarrow \psi)$. Покажем, что отношение $\equiv$ образует отношение эквивалентности на множестве формул.

Лемма 1.3.2. *Для любых формул φ, ψ и χ исчисления ИС справедливы следующие утверждения:*

(а) $\varphi \equiv \varphi$;

(б) *если* $\varphi \equiv \psi$, *то* $\psi \equiv \varphi$;

(в) *если* $\varphi \equiv \psi$ *и* $\psi \equiv \chi$, *то* $\varphi \equiv \chi$.

Доказательство. В пункте (а) доказывать нечего, поскольку $\varphi \vdash \varphi$ — аксиома. Пункт (б) следует из симметричности в определении отношения $\equiv$. Пункт (в) вытекает из правила сечения (предложение 1.2.2, в). $\square$

Установим, что эквивалентность формул сохраняется под действием операций $\neg$, $\wedge$, $\vee$ и $\to$.

Лемма 1.3.3. *Если $\varphi_1 \equiv \psi_1$ и $\varphi_2 \equiv \psi_2$, то $\neg\varphi_1 \equiv \neg\psi_1$, $(\varphi_1 \wedge \varphi_2) \equiv (\psi_1 \wedge \psi_2)$, $(\varphi_1 \vee \varphi_2) \equiv (\psi_1 \vee \psi_2)$ и $(\varphi_1 \to \varphi_2) \equiv (\psi_1 \to \psi_2)$.*

Д о к а з а т е л ь с т в о. В силу симметричности отношения $\equiv$ достаточно доказать секвенции $\neg\varphi_1 \vdash \neg\psi_1$, $(\varphi_1 \wedge \varphi_2) \vdash (\psi_1 \wedge \psi_2)$, $(\varphi_1 \vee \varphi_2) \vdash (\psi_1 \vee \psi_2)$ и $(\varphi_1 \to \varphi_2) \vdash (\psi_1 \to \psi_2)$, а доказуемость этих секвенций устанавливается следующими деревьями:

$$\frac{\Gamma, \psi_1 \vdash \varphi_1}{\Gamma, \neg\varphi_1 \vdash \neg\psi_1}\ (\text{контрапозиция});$$

$$\dfrac{\dfrac{\dfrac{\varphi_1 \wedge \varphi_2 \vdash \varphi_1 \wedge \varphi_2}{\varphi_1 \wedge \varphi_2 \vdash \varphi_1;} \quad \dfrac{\varphi_1 \vdash \psi_1}{\vdash \varphi_1 \to \psi_1}}{\varphi_1 \wedge \varphi_2 \vdash \psi_1;} \quad \dfrac{\dfrac{\varphi_1 \wedge \varphi_2 \vdash \varphi_1 \wedge \varphi_2}{\varphi_1 \wedge \varphi_2 \vdash \varphi_2;} \quad \dfrac{\varphi_2 \vdash \psi_2}{\vdash \varphi_2 \to \psi_2}}{\varphi_1 \wedge \varphi_2 \vdash \psi_2}}{\varphi_1 \wedge \varphi_2 \vdash \psi_1 \wedge \psi_2};$$

$$\dfrac{\dfrac{\varphi_1 \vdash \psi_1}{\varphi_1 \vdash \psi_1 \vee \psi_2;} \quad \dfrac{\varphi_2 \vdash \psi_2}{\varphi_2 \vdash \psi_1 \vee \psi_2;} \quad \varphi_1 \vee \varphi_2 \vdash \varphi_1 \vee \varphi_2}{\varphi_1 \vee \varphi_2 \vdash \psi_1 \vee \psi_2};$$

$$\dfrac{\dfrac{\dfrac{\psi_1 \vdash \varphi_1;\ \varphi_1 \to \varphi_2 \vdash \varphi_1 \to \varphi_2}{\varphi_1 \to \varphi_2, \psi_1 \vdash \varphi_2;} \quad \dfrac{\varphi_2 \vdash \psi_2}{\vdash \varphi_2 \to \psi_2}}{\varphi_1 \to \varphi_2, \psi_1 \vdash \psi_2}}{\varphi_1 \to \varphi_2 \vdash \psi_1 \to \psi_2}.\ \square$$

Формула ψ исчисления ИС называется *подформулой* формулы φ исчисления ИС, если ψ является подсловом слова φ. Место, которое занимает подформула ψ в формуле φ, называется *вхождением* ψ в формулу φ.

П р и м е р 1.3.1. Формула A имеет два вхождения в формулу φ, имеющую вид $A \vee (A \to B \wedge C) \to D$. Следующие формулы образуют множество всех подформул формулы φ: A, B, C, D, $B \wedge C$, $A \to B \wedge C$, $A \vee (A \to B \wedge C)$, φ. $\square$

Теорема 1.3.4. (теорема о замене). *Пусть φ — формула исчисления ИС, ψ — ее подформула, а формула φ' получается из φ заменой некоторого вхождения ψ на формулу ψ'. Тогда если $\psi \equiv \psi'$, то $\varphi \equiv \varphi'$.*

Д о к а з а т е л ь с т в о. Если $\varphi = \psi$, то доказывать нечего. Если $\varphi \neq \psi$, то используем индукцию по числу шагов построения формулы φ. Предполагая, что φ — пропозициональная переменная, снова получаем $\varphi = \psi$. Индукционный переход осуществляется рассмотрением четырех случаев: $\varphi = \neg\varphi_1$, $\varphi = \varphi_1 \wedge \varphi_2$, $\varphi = \varphi_1 \vee \varphi_2$, $\varphi = \varphi_1 \to \varphi_2$. В каждом из этих случаев формула ψ входит в φ_1 или φ_2. Поэтому эквивалентность $\varphi \equiv \varphi'$ вытекает из индукционного предположения и леммы 1.3.3. $\square$

§ 1.4. Нормальные формы

В этом параграфе мы покажем, что преобразования формул алгебры логики, приводящие к построению дизъюнктивных и конъюнктивных нормальных форм, имеют место и в исчислении секвенций.

Лемма 1.4.1. *Пусть φ, ψ и χ — формулы ИС. Тогда справедливы следующие эквивалентности:*

(а) $(\varphi\wedge\psi)\wedge\chi \equiv \varphi\wedge(\psi\wedge\chi)$, $(\varphi\vee\psi)\vee\chi \equiv \varphi\vee(\psi\vee\chi)$ *(ассоциативность $\wedge$ и $\vee$);*

(б) $\varphi \wedge \psi \equiv \psi \wedge \varphi$, $\varphi \vee \psi \equiv \psi \vee \varphi$ *(коммутативность $\wedge$ и $\vee$);*

(в) $\varphi \wedge \varphi \equiv \varphi$, $\varphi \vee \varphi \equiv \varphi$ *(идемпотентность $\wedge$ и $\vee$);*

(г) $\varphi \wedge (\psi \vee \chi) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$, $\varphi \vee (\psi \wedge \chi) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \chi)$ *(законы дистрибутивности);*

(д) $\varphi \wedge (\varphi \vee \psi) \equiv \varphi$, $\varphi \vee (\varphi \wedge \psi) \equiv \varphi$ *(законы поглощения);*

(е) $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$, $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$ *(законы де Моргана);*

(ж) $\neg\neg\varphi \equiv \varphi$ *(закон двойного отрицания);*

(з) $\varphi \to \psi \equiv \neg\varphi \vee \psi$.

Д о к а з а т е л ь с т в о. Мы приведем доказательства эквивалентностей $\varphi \wedge (\psi \vee \chi) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$ и $\varphi \to \psi \equiv \neg\varphi \vee \psi$, оставив остальные утверждения читателю в качестве упражнения:

$$\dfrac{\dfrac{\dfrac{\varphi\wedge(\psi\vee\chi)\vdash\varphi;\ \ \psi\vdash\psi}{\varphi\wedge(\psi\vee\chi),\psi\vdash\varphi\wedge\psi}}{\varphi\wedge(\psi\vee\chi),\psi\vdash(\varphi\wedge\psi)\vee(\varphi\wedge\chi);} \quad \dfrac{\dfrac{\varphi\wedge(\psi\vee\chi)\vdash\varphi;\ \ \chi\vdash\chi}{\varphi\wedge(\psi\vee\chi),\chi\vdash\varphi\wedge\chi}}{\varphi\wedge(\psi\vee\chi),\chi\vdash(\varphi\wedge\psi)\vee(\varphi\wedge\chi);} \quad \varphi \wedge (\psi \vee \chi) \vdash \psi \vee \chi}{\varphi \wedge (\psi \vee \chi) \vdash (\varphi \wedge \psi) \vee (\varphi \wedge \chi)};$$

$$\dfrac{\dfrac{\varphi\wedge\psi\vdash\varphi;\ \ \dfrac{\varphi\wedge\psi\vdash\psi}{\varphi\wedge\psi\vdash\psi\vee\chi}}{\varphi\wedge\psi\vdash\varphi\wedge(\psi\vee\chi);} \quad \dfrac{\varphi\wedge\chi\vdash\varphi;\ \ \dfrac{\varphi\wedge\chi\vdash\chi}{\varphi\wedge\chi\vdash\psi\vee\chi}}{\varphi\wedge\chi\vdash\varphi\wedge(\psi\vee\chi);} \quad (\varphi\wedge\psi)\vee(\varphi\wedge\chi)\vdash(\varphi\wedge\psi)\vee(\varphi\wedge\chi)}{(\varphi \wedge \psi) \vee (\varphi \wedge \chi) \vdash \varphi \wedge (\psi \vee \chi)};$$

$$\dfrac{\dfrac{\dfrac{\varphi \vdash \varphi;\ \ \varphi \to \psi \vdash \varphi \to \psi}{\varphi \to \psi, \varphi \vdash \psi}}{\varphi \to \psi, \varphi \vdash \neg\varphi \vee \psi;} \quad \dfrac{\neg\varphi \vdash \neg\varphi}{\neg\varphi \vdash \neg\varphi \vee \psi;} \quad \vdash \varphi \vee \neg\varphi}{\varphi \to \psi \vdash \neg\varphi \vee \psi};$$

$$\dfrac{\dfrac{\dfrac{\varphi \vdash \varphi;\ \ \neg\varphi \vdash \neg\varphi}{\varphi, \neg\varphi \vdash}}{\varphi, \neg\varphi \vdash \psi;} \quad \psi \vdash \psi;\ \ \neg\varphi \vee \psi \vdash \neg\varphi \vee \psi}{\dfrac{\neg\varphi \vee \psi, \varphi \vdash \psi}{\neg\varphi \vee \psi \vdash \varphi \to \psi}}. \ \square$$

Напомним, что *литерой* называется любая атомарная формула A, обозначаемая через A^1, или ее отрицание $\neg A$, которое обозначается через A^0. *Конъюнктом* (*дизъюнктом*) называется литера или конъюнкция (соответственно дизъюнкция) литер. Конъюнкт или дизъюнкция конъюнктов называется *дизъюнктивной нормальной формой* (ДНФ), а дизъюнкт или конъюнкция дизъюнктов — *конъюнктивной нормальной формой* (КНФ).

Следующая теорема является аналогом теоремы о приведении формул алгебры логики к дизъюнктивным и конъюнктивным нормальным формам.

Теорема 1.4.2. *Любая формула ИС эквивалентна некоторой ДНФ и некоторой КНФ.*

Алгоритм приведения формулы ИС к ДНФ аналогичен алгоритму приведения формул алгебры логики к ДНФ и опирается на лемму 1.4.1.

1. Выражаем согласно пункту (з) леммы 1.4.1 все импликации, участвующие в построении формулы, через дизъюнкции и отрицания.

2. Используя законы де Моргана (лемма 1.4.1, п. (е)), переносим все отрицания к переменным и сокращаем двойные отрицания по закону двойного отрицания (лемма 1.4.1, п. (ж)).

3. Используя закон дистрибутивности $\varphi \wedge (\psi \vee \chi) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$, преобразуем формулу так, чтобы все конъюнкции выполнялись раньше дизъюнкций.

В результате применения пп. 1–3 получается ДНФ данной формулы.

Приведение формулы к КНФ производится аналогично приведению ее к ДНФ с применением закона дистрибутивности $\varphi \vee (\psi \wedge \chi) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \chi)$. $\square$

§ 1.5. Семантика исчисления секвенций

Пусть X — некоторое множество, f_X — отображение, которое каждой пропозициональной переменной ставит в соответствие некоторое подмножество множества X. Расширим по индукции отображение f_X до отображения множества формул ИС в булеан $\mathcal{P}(X)$ множества X согласно следующим соотношениям:

$f_X(\neg\varphi) = X \setminus f_X(\varphi)$,
$f_X(\varphi \wedge \psi) = f_X(\varphi) \cap f_X(\psi)$,
$f_X(\varphi \vee \psi) = f_X(\varphi) \cup f_X(\psi)$,
$f_X(\varphi \to \psi) = f_X(\neg\varphi) \cup f_X(\psi)$.

Любое такое отображение f_X, действующее на множестве формул ИС, называется *интерпретацией ИС* в X.

Каждой секвенции S следующим образом ставится в соответствие некоторое утверждение $f_X(S)$ о подмножествах X:
$f_X(\varphi_1, \ldots, \varphi_n \vdash \psi) \Leftrightarrow f_X(\varphi_1) \cap \ldots \cap f_X(\varphi_n) \subseteq f_X(\psi)$,
$f_X(\vdash \psi) \Leftrightarrow f_X(\psi) = X$,
$f_X(\varphi_1, \ldots, \varphi_n \vdash) \Leftrightarrow f_X(\varphi_1) \cap \ldots \cap f_X(\varphi_n) = \varnothing$,
$f_X(\vdash) \Leftrightarrow X = \varnothing$.

Теорема 1.5.1. *Для любой интерпретации f_X ИС и любой доказуемой в ИС секвенции S справедливо утверждение $f_X(S)$.*

Доказательство. Если S — аксиома $\varphi \vdash \varphi$, то истинность утверждения $f_X(S)$, имеющего вид $f_X(\varphi) \subseteq f_X(\varphi)$, очевидна. В общем случае достаточно доказать, что при переходе по любому из 12 правил вывода из справедливости утверждений f_X от секвенций над чертой следует истинность утверждения f_X от секвенции под чертой. Покажем, как проверяются указанные переходы на примере первого правила вывода

$$\frac{\Gamma \vdash \varphi;\ \Gamma \vdash \psi}{\Gamma \vdash (\varphi \wedge \psi)},$$

где $\Gamma = \varphi_1, \ldots, \varphi_n$. Итак, по условию имеем $\bigcap\limits_{i=1}^{n} f_X(\varphi_i) \subseteq f_X(\varphi)$ и $\bigcap\limits_{i=1}^{n} f_X(\varphi_i) \subseteq f_X(\psi)$. Тогда $\bigcap\limits_{i=1}^{n} f_X(\varphi_i) \subseteq f_X(\varphi) \cap f_X(\psi)$. Следовательно, $\bigcap\limits_{i=1}^{n} f_X(\varphi_i) \subseteq f_X(\varphi \wedge \psi)$, т.е. справедливо утверждение $f_X(\Gamma \vdash (\varphi \wedge \psi))$.

Проверка остальных переходов аналогична и предоставляется читателю. □

Следствие 1.5.2. *Исчисление ИС непротиворечиво.*

Доказательство. Пусть X — непустое множество, f_X — произвольная интерпретация ИС, A — некоторая атомарная формула. Покажем, что формула $A \wedge \neg A$ не доказуема в ИС. Действительно, $f_X(A \wedge \neg A) = f_X(A) \cap (X \setminus f_X(A)) = \varnothing$, откуда с учетом непустоты множества X следует, что утверждение $f_X(\vdash A \wedge \neg A)$ ложно. Тогда по теореме 1.5.1 секвенция $\vdash A \wedge \neg A$ не доказуема. □

Понятие интерпретации выходит за рамки самого исчисления и относится к *семантике исчисления*, устанавливающей соответствие действий в исчислении с теоретико-множественными операциями. Сами же понятия формулы, секвенции, правил вывода и доказательства, образующие исчисление, составляют *синтаксис исчисления*.

Определим теперь так называемую *главную интерпретацию ИС*, которая позволяет составлять таблицы истинности формул. Возьмем в качестве множества X одноэлементное множество $\{\varnothing\}$. Тогда для любой атомарной формулы A значение $f_X(A)$ равно $\varnothing$, т.е. $f_X(A) = 0$, или $f_X(A)$ равно $\{\varnothing\}$, т.е. $f_X(A) = 1$ (напомним, что $0 = \varnothing$, а $1 = \{\varnothing\}$). Придавая переменным x и y значения $f_{\{\varnothing\}}(x)$ $f_{\{\varnothing\}}(y)$ из множества $\{0, 1\}$, получаем *таблицы истинности* для логических операций $\wedge$, $\vee$, $\to$ и $\neg$.

x	y	$(x \wedge y)$	$(x \vee y)$	$(x \to y)$	$\neg\, x$
0	0	0	0	1	1
0	1	0	1	1	1
1	0	0	1	0	0
1	1	1	1	1	0

Пусть $A_1, \ldots, A_k$ — пропозициональные переменные, f — отображение множества элементарных формул в $\{0, 1\}$. С помощью таблиц истинности логических связок функция f однозначно продолжается на множество формул $\varphi(A_1, \ldots, A_k)$, которые строятся из пропозициональных переменных $A_1, \ldots, A_k$ и логических связок. При этом для любой формулы φ, равной $\varphi(A_1, \ldots, A_k)$, значение $f(\varphi)$ снова равно 0 или 1. Если $f(\varphi) = 1$ ($f(\varphi) = 0$), то говорят, что формула φ *истинна* (*ложна*) на наборе $(f(A_1), \ldots, f(A_k))$.

Функция $f_\varphi : \{0, 1\}^k \to \{0, 1\}$, которая каждому набору $(\delta_1, \ldots, \delta_k) \in \{0, 1\}^k$ сопоставляет значение истинности формулы φ, называется *истинностной функцией формулы* φ. Очевидно, что таблица истинности функции f_φ совпадает с таблицей истинности формулы φ.

Напомним, что формула φ называется *тождественно истинной* (*тождественно ложной*), если функция f_φ тождественно равна единице (тождественно равна нулю).

Секвенция $\Gamma \vdash \varphi$ называется *истинной на наборе* $(\delta_1, \ldots, \delta_k) \in \{0, 1\}^k$, если на этом наборе хотя бы одна формула из Γ ложна или формула φ истинна. Секвенция $\Gamma \vdash$ называется *истинной на наборе* $(\delta_1, \ldots, \delta_k) \in \{0, 1\}^k$, если на этом наборе некоторая формула из Γ ложна. Секвенция $\vdash$ по определению ложна на любом наборе, а истинность секвенции $\vdash \varphi$ совпадает с истинностью формулы φ.

Секвенция S называется *тождественно истинной*, если S истинна на любом наборе $(\delta_1, \ldots, \delta_k)$ значений истинности переменных $A_1, \ldots$ $\ldots, A_k$, среди которых содержатся все переменные, входящие в формулы из S.

Теорема 1.5.3. (теорема о полноте). 1. *Формула φ ИС доказуема в ИС тогда и только тогда, когда φ тождественно истинна.*

2. *Секвенция S ИС доказуема в ИС тогда и только тогда, когда S тождественно истинна.*

По теореме о полноте для того чтобы установить, доказуема ли секвенция $\varphi_1, \ldots, \varphi_n \vdash \psi$ (или $\varphi_1, \ldots, \varphi_n \vdash$), достаточно составить таблицу истинности формулы $\varphi_1 \to (\varphi_2 \to \ldots \to (\varphi_n \to \psi)\ldots)$ (или $\varphi_1 \to (\varphi_2 \to \ldots \to (\varphi_n \to A_0 \wedge \neg A_0)\ldots)$) и проверить ее тождественную истинность. Как известно, существует единый алгоритм построения таблиц истинности, и, значит, ИС *разрешимо*.

Следствие 1.5.4. *Тогда и только тогда выполняется $\varphi \equiv \psi$, когда равны истинностные функции f_φ и f_ψ.*

Доказательство. Предположим, что $\varphi \equiv \psi$. Тогда по лемме 1.3.2 доказуема формула $(\varphi \leftrightarrow \psi)$. По теореме о полноте получаем тождественную истинность этой формулы, что равносильно соотношению $f_\varphi = f_\psi$.

Обратно, из равенства $f_\varphi = f_\psi$ следует тождественная истинность секвенций $\varphi \vdash \psi$ и $\psi \vdash \varphi$. По теореме о полноте эти секвенции доказуемы, и, значит, справедливо $\varphi \equiv \psi$. □

Из следствия 1.5.6 вытекает, что отношение эквивалентности $\sim$ на формулах алгебры логики и отношение эквивалентности $\equiv$ совпадают:

$$\varphi \sim \psi \Leftrightarrow \varphi \equiv \psi.$$

Пример 1.5.1. Так как $\varphi \wedge \psi \sim \neg(\neg\varphi \vee \neg\psi)$, то $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$.

Замечание 1.5.5. *Пусть Φ — множество всех формул ИС с пропозициональными переменными из множества I. Рассмотрим алгебру $\mathfrak{F} = \langle \Phi, \wedge, \vee, \neg, A \wedge \neg A, A \vee \neg A \rangle$, где A — некоторая фиксированная переменная. Тогда фактор-алгебра $\mathfrak{F}/\equiv$ является булевой алгеброй, называемой* алгеброй Линденбаума *для ИС.*

Схема аксиом называется *независимой* в исчислении, если хотя бы один ее частный случай не доказуем в исчислении без этой схемы. Правило вывода называется *независимым* в исчислении, если оно не является допустимым в исчислении, полученном удалением этого правила. Исчисление называется *независимым*, если все его схемы аксиом и правила вывода независимы.

Теорема 1.5.6. *Исчисление ИС независимо.* □

§ 1.6. Исчисление высказываний гильбертовского типа

В этом параграфе мы построим исчисление ИВ, в котором в отличие от ИС не используется понятие секвенции, правило вывода одно, а схем аксиом — несколько.

Формулами ИВ называются формулы ИС.

Аксиомами ИВ являются следующие формулы для любых формул φ, ψ, χ:

1) $\varphi \to (\psi \to \varphi)$;
2) $(\varphi \to \psi) \to ((\varphi \to (\psi \to \chi)) \to (\varphi \to \chi))$;
3) $(\varphi \wedge \psi) \to \varphi$;
4) $(\varphi \wedge \psi) \to \psi$;
5) $(\varphi \to \psi) \to ((\varphi \to \chi) \to (\varphi \to (\psi \wedge \chi)))$;
6) $\varphi \to (\varphi \vee \psi)$;
7) $\varphi \to (\psi \vee \varphi)$;
8) $(\varphi \to \chi) \to ((\psi \to \chi) \to ((\varphi \vee \psi) \to \chi))$;
9) $(\varphi \to \psi) \to ((\varphi \to \neg\psi) \to \neg\varphi)$;
10) $\neg\neg\varphi \to \varphi$.

Указанные формулы называются *схемами аксиом ИВ*. При подстановке конкретных формул в какую-либо схему получается *частный случай схемы аксиом.*

Единственным *правилом вывода в ИВ* является *правило заключения* (*modus ponens*): если φ и $\varphi \to \psi$ — выводимые формулы, то ψ — также выводимая формула. Символически это записывается так:

$$\frac{\varphi;\ \varphi \to \psi}{\psi}.$$

Например, если высказывания $A \wedge B$ и $A \wedge B \to (A \to C)$ выводимы, то высказывание $A \to C$ также выводимо согласно правилу заключения.

Говорится, что формула φ *выводима из формул* $\varphi_1, \ldots, \varphi_m$ (обозначается $\varphi_1, \ldots, \varphi_m \vdash \varphi$), если существует последовательность формул $\psi_1, \ldots, \psi_k, \varphi$, в которой любая формула либо является аксиомой, либо принадлежит списку формул $\varphi_1, \ldots, \varphi_m$, называемых *гипотезами*, либо получается из предыдущих по правилу вывода. Выводимость формулы φ из $\varnothing$ ($\vdash \varphi$) равносильна тому, что φ — теорема ИВ.

П р и м е р 1.6.1. Покажем, что формула $\varphi \to \varphi$ выводима в ИВ. Для этого построим вывод данной формулы:

1) в схеме аксиом 2 формулу ψ заменим на $\varphi \to \varphi$, χ — на φ. Получаем аксиому $(\varphi \to (\varphi \to \varphi)) \to ((\varphi \to ((\varphi \to \varphi) \to \varphi)) \to (\varphi \to \varphi))$;

2) в схеме аксиом 1 формулу ψ заменим на φ. Получаем $\varphi \to (\varphi \to \varphi)$;

3) из 1 и 2 по modus ponens заключаем $(\varphi \to ((\varphi \to \varphi) \to \varphi)) \to (\varphi \to \varphi)$;

4) в схеме аксиом 1 заменяем ψ на $\varphi \to \varphi$. Получаем $\varphi \to ((\varphi \to \varphi) \to \varphi)$;

5) из пп. 3 и 4 по правилу вывода справедливо $\vdash \varphi \to \varphi$. □

Теорема 1.6.1. (теорема о дедукции). *Если* $\Gamma, \varphi \vdash \psi$, *то* $\Gamma \vdash \varphi \to \psi$, *где* Γ — *набор некоторых формул* $\varphi_1, \ldots, \varphi_n$.

Д о к а з а т е л ь с т в о. Рассмотрим минимальный вывод $\psi_1, \ldots, \psi_k$ формулы $\psi = \psi_k$ из формул $\varphi_1, \ldots, \varphi_n$ и φ. Если $k = 1$, то $\psi = \varphi$, ψ — аксиома или входит в Γ. В первом случае в силу примера 1.6.1 формула $\psi \to \psi$ выводима в ИВ из Γ. Во втором и третьем случаях последовательность ψ, $\psi \to (\varphi \to \psi)$, $\varphi \to \psi$ будет выводом в ИВ из Γ.

Предположим, что $k > 1$ и формула ψ получается из формул ψ_i и $\psi_j = \psi_i \to \psi$ (где $i, j < k$) по правилу заключения. Поскольку $i < k$ и $j < k$ по индукции можно считать, что $\Gamma \vdash \varphi \to \psi_i$ и $\Gamma \vdash \varphi \to (\psi_i \to \psi)$. Используя аксиому

$$(\varphi \to \psi_i) \to ((\varphi \to (\psi_i \to \psi)) \to (\varphi \to \psi))$$

и дважды применяя к этой формуле правило заключения с формулами $\varphi \to \psi_i$ и $\varphi \to (\psi_i \to \psi)$, получаем сначала формулу $(\varphi \to (\psi_i \to \psi)) \to (\varphi \to \psi)$, а затем формулу $\varphi \to \psi$. Тем самым, выводимость формулы $\varphi \to \psi$ из Γ доказана. □

Применение теоремы о дедукции позволяет во многих случаях сокращать доказательство формул.

П р и м е р 1.6.2. Покажем, что формула $\varphi \to (\psi \to (\varphi \wedge \psi))$ доказуема в ИВ для любых формул φ и ψ ИВ. По теореме о дедукции достаточно показать $\varphi, \psi \vdash \varphi \wedge \psi$. Построим вывод формулы $\varphi \wedge \psi$ из гипотез φ и ψ:

1) из аксиомы $\varphi \to (\varphi \to \varphi)$ и гипотезы φ получаем формулу $(\varphi \to \varphi)$;

2) из аксиомы $\psi \to (\varphi \to \psi)$ и гипотезы ψ выводим формулу $(\varphi \to \psi)$;

3) применяя правило заключения к формуле $(\varphi \to \varphi)$ и аксиоме $(\varphi \to \varphi) \to ((\varphi \to \psi) \to (\varphi \to (\varphi \wedge \psi)))$, получаем формулу $(\varphi \to \psi) \to (\varphi \to (\varphi \wedge \psi))$;

4) из заключения пп. 2 и 3 выводится формула $\varphi \to (\varphi \wedge \psi)$;

5) из гипотезы φ и заключения п. 5 получаем формулу $\varphi \wedge \psi$. □

Следствие 1.6.2. *Тогда и только тогда* $\varphi_1, \ldots, \varphi_n \vdash \varphi$, *когда* $\vdash (\varphi_1 \to (\varphi_2 \to \ldots \to (\varphi_{n-1} \to (\varphi_n \to \varphi)) \ldots))$.

Следующая теорема устанавливает эквивалентность доказуемости в ИС и доказуемости в ИВ.

Теорема 1.6.3. (теорема об эквивалентности ИС и ИВ) 1. *Секвенция* $\varphi_1, \ldots, \varphi_n \vdash \psi$ *доказуема в ИС тогда и только тогда, когда формула* ψ *выводима в ИВ из формул* $\varphi_1, \ldots, \varphi_n$.

2. *Секвенция* $\varphi_1, \ldots, \varphi_n \vdash$ *доказуема в ИС тогда и только тогда, когда формула* $A \wedge \neg A$ *выводима в ИВ из формул* $\varphi_1, \ldots, \varphi_n$. $\square$

Из теоремы 1.6.3. вытекает непротиворечивость и разрешимость ИВ. Непосредственно проверяется независимость схем аксиом ИВ.

Множество формул Γ называется *противоречивым*, если в ИВ справедливо $\Gamma \vdash A \wedge \neg A$. Если Γ — противоречивое множество формул, будем обозначать этот факт через $\Gamma \vdash$.

Утверждение 1.6.4. *Формула* φ *выводима в ИВ из множества формул* Γ *тогда и только тогда, когда множество* $\Gamma \cup \{\neg\varphi\}$ *— противоречиво:* $\Gamma \vdash \varphi \Leftrightarrow \Gamma \cup \{\neg\varphi\} \vdash$.

§ 1.7. Алгоритмы проверки общезначимости и противоречивости в ИВ

В этом параграфе мы рассмотрим некоторые основные методы, которые используются для исследования множеств формул исчислений ИС и ИВ на тождественную истинность и противоречивость.

1. Алгоритм Квайна. Напомним, что формула φ от пропозициональных переменных A_1, A_2, …, A_k является тождественно истинной, общезначимой или (что то же самое) доказуемой, если булева функция $f_\varphi : \{0,1\}^k \to \{0,1\}$, соответствующая формуле φ, тождественно равна 1. Для проверки значений функции f_φ используется так называемое *семантическое дерево*, т. е. бинарное корневое дерево, удовлетворяющее следующим условиям:

— каждое ребро помечено литерой $A_i^{\delta_j}$;

— литеры, выходящие из одной вершины, *контрарны*: A_i, $\neg A_i$;

— ребра соответствуют литере одной и той же пропозициональной переменной A_i тогда и только тогда, когда они находятся на одинаковом расстоянии от корня (рис. 1.1).

Семантическое дерево имеет 2^k висячих вершин и для проверки общезначимости необходимо пройти 2^k маршрутов от корня до этих вершин.

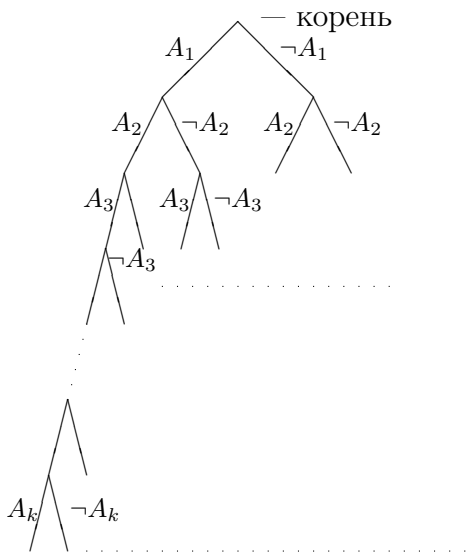

Рис. 1.1

Алгоритм Квайна позволяет проходить не все семантическое дерево, а только его часть. Он состоит в том, что пропозициональным переменным A_i, упорядоченным в набор $(A_1, A_2, \ldots, A_k)$, последовательно придаются значения 0 и 1 и анализируются таблицы истинности формул, содержащих меньшее число переменных.

П р и м е р 1.7.1. Проверить общезначимость формулы

$$\varphi = (((A \wedge B) \to C) \wedge (A \to B)) \to (A \to C).$$

Упорядочим пропозициональные переменные в набор (A, B, C). Придадим первой переменной A значение $f(A) = 1$. Тогда формула φ преобразуется следующим образом: $(((1 \wedge B) \to C) \wedge (1 \to B)) \to (1 \to C) \sim ((B \to C) \wedge B) \to C$. В полученной формуле переменной B придадим значение $f(B) = 1$. Тогда преобразованная формула примет вид $((1 \to C) \wedge 1) \to C \sim C \to C$, т. е. будет общезначимой. В случае $f(B) = 0$ имеем $((0 \to C) \wedge 0) \to C \sim 0 \to C$, что также общезначимо. Рассмотрим теперь случай $f(A) = 0$. Имеем $\varphi(0, B, C) = (((0 \wedge B) \to C) \wedge (0 \to B)) \to (0 \to C) \sim 1 \wedge 1 \to 1 \sim 1$. Таким образом, все возможные случаи приводят к тождественно истинным формулам. Следовательно, формула φ тождественно истинна. На рис. 1.2, *а* изображено семантическое дерево, соответствующее формуле φ, а на рис. 1.2, *б* — часть семантического дерева, которая фактически использовалась для проверки общезначимости. □

2. Алгоритм редукции решает ту же задачу проверки общезначимости формулы, но используется в том случае, когда в формуле

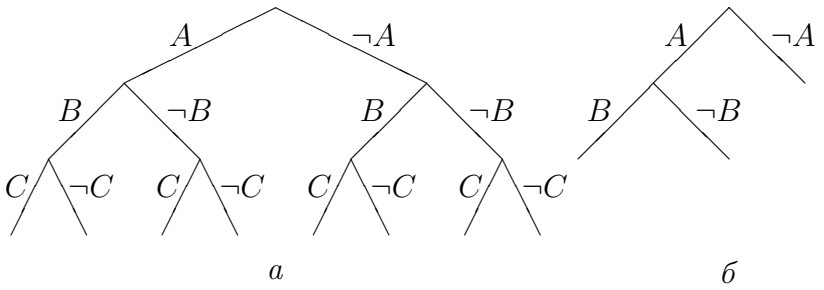

Рис. 1.2

содержится достаточно много импликаций. Идея алгоритма состоит в попытке нахождения значений пропозициональных переменных формулы φ, при которых значение функции f_φ равно 0, на основе того, что импликация является ложной в том и только в том случае, когда посылка истинна, а заключение ложно.

П р и м е р 1.7.2. Проверить тождественную истинность формулы $\varphi = ((A \wedge B) \to C) \to (A \to (B \to C))$.

Предположим, что формула φ ложна при некотором наборе значений переменных A, B, C. Тогда истинностная функция f по этим значениям переменных дает следующие значения формул: $f((A \wedge B) \to C) = 1$, $f(A \to (B \to C)) = 0$. Тогда из второго равенства получаем $f(A) = 1$, $f(B \to C) = 0$, откуда имеем $f(B) = 1$, $f(C) = 0$. Однако при этих значениях справедливо $f((A \wedge B) \to C) = 0$. Получили противоречие. Таким образом, формула φ тождественно истинна.

3. Метод резолюций в ИВ. Пусть $D_1 = D_1' \vee A$, $D_2 = D_2' \vee \neg A$ — дизъюнкты. Дизъюнкт $D_1' \vee D_2'$ называется *резольвентой дизъюнктов* D_1 и D_2 *по литере* A и обозначается через $\mathrm{res}_A(D_1, D_2)$. *Резольвентой дизъюнктов* D_1 и D_2 называется их резольвента по некоторой литере и обозначается через $\mathrm{res}(D_1, D_2)$, $\mathrm{res}(A, \neg A) \rightleftharpoons 0$. В последнем случае нуль будет отождествляться с формулой $A \wedge \neg A$. Если дизъюнкты D_1 и D_2 не содержат контрарных литер, то резольвент у них не существует.

П р и м е р 1.7.3. Если $D_1 = A \vee B \vee C$, $D_2 = \neg A \vee \neg B \vee D$, то $\mathrm{res}_A(D_1, D_2) = B \vee C \vee \neg B \vee D$, $\mathrm{res}_B(D_1, D_2) = A \vee C \vee \neg A \vee D$, $\mathrm{res}_C(D_1, D_2)$ не существует. □

Предложение 1.7.1. 1. *Если резольвента* $\mathrm{res}(D_1, D_2)$ *существует и не равна нулю, то секвенция* $D_1, D_2 \vdash \mathrm{res}(D_1, D_2)$ *доказуема.*

2. *Если резольвента* $\mathrm{res}(D_1, D_2)$ *равна нулю, то доказуема секвенция* $D_1, D_2 \vdash$. □

Пусть $S = \{D_1, D_2, \ldots, D_m\}$ — множество дизъюнктов. Последовательность формул φ_1, φ_2, ..., φ_n называется *резолютивным выводом* из множества S, если для каждой формулы φ_i $(i = 1, \ldots, n)$ выполняется какое-то из следующих условий:

— $\varphi_i \in S$;

— существуют $j, k < i$ такие, что $\varphi_i = \mathrm{res}(\varphi_j, \varphi_k)$.

Теорема 1.7.2. (теорема о полноте метода резолюций). *Множество дизъюнктов S противоречиво в том и только в том случае, когда существует резолютивный вывод из S, заканчивающийся символом* 0.

Отметим, что метод резолюций можно использовать для проверки выводимости формулы φ из данного множества формул $\varphi_1, \ldots, \varphi_n$. Действительно, условие $\varphi_1, \ldots, \varphi_n \vdash \varphi$ равносильно условию $\varphi_1, \ldots, \varphi_n, \neg\varphi \vdash$, что в свою очередь равносильно условию $\psi \vdash$, где $\psi = \varphi_1 \wedge \ldots \wedge \varphi_n \wedge \neg\varphi$. Приведем формулу ψ к КНФ: $\psi \equiv D_1 \wedge D_2 \wedge \ldots \wedge D_m$, тогда

$$\psi \vdash \Leftrightarrow D_1 \wedge D_2 \wedge \ldots \wedge D_m \vdash \Leftrightarrow D_1, D_2, \ldots, D_m \vdash .$$

Таким образом, задача проверки выводимости $\varphi_1, \ldots, \varphi_n \vdash \varphi$ сводится к проверке противоречивости множества дизъюнктов $S = \{D_1, D_2, \ldots, D_m\}$, что равносильно существованию резолютивного вывода 0 из S.

П р и м е р 1.7.4. Проверить методом резолюций доказуемость секвенции

$$A \to (B \to C), C \wedge D \to E, \neg F \to D \wedge \neg E \vdash A \to (B \to F).$$

Согласно утверждению 1.6.4 нужно проверить на противоречивость множество формул

$$S = \{A \to (B \to C), C \wedge D \to E, \neg F \to D \wedge \neg E, \neg(A \to (B \to F))\}.$$

Приведем все формулы из S к КНФ: $S \sim \{\neg A \vee \neg B \vee C, \neg(C \wedge D) \vee E, F \vee D \wedge \neg E, \neg(\neg A \vee \neg B \vee F)\} \sim \{\neg A \vee \neg B \vee C, \neg C \vee \neg D \vee E, (F \vee D) \wedge (F \vee \neg E), A \wedge B \wedge \neg F\}$.

Отсюда получаем множество дизъюнктов $S' = \{\neg A \vee \neg B \vee C, \neg C \vee \neg D \vee E, F \vee D, F \vee \neg E, A, B, \neg F\}$. Построим резолютивный вывод из S', заканчивающийся 0:

1) $\mathrm{res}_A(\neg A \vee \neg B \vee C, A) = \neg B \vee C$;

2) $\mathrm{res}_B(\neg B \vee C, B) = C$;

3) $\mathrm{res}_D(\neg C \vee \neg D \vee E, F \vee D) = \neg C \vee E \vee F$;

4) $\mathrm{res}_E(\neg C \vee E \vee F, F \vee \neg E) = \neg C \vee F$;
5) $\mathrm{res}_C(C, \neg C \vee F) = F$;
6) $\mathrm{res}(F, \neg F) = 0$.

Таким образом, по теореме о полноте метода резолюций множество S противоречиво и, значит, данная секвенция доказуема. $\square$

Отметим, что метод резолюций достаточен для обнаружения возможной выполнимости данного множества дизъюнктов S. Для этого включим в множество S все дизъюнкты, получающиеся при резолютивных выводах из S. Из теоремы о полноте метода резолюций вытекает

Следствие 1.7.3. *Если множество S' состоит из всех элементов множества дизъюнктов S, а также всевозможных резольвент всех своих элементов, то S выполнимо тогда и только тогда, когда $0 \notin S'$.*

Двойственным к правилу резолюций является *правило согласия*. Пусть $K_1 = K_1' \wedge A$, $K_2 = K_2' \wedge \neg A$ — конъюнкты. Положим

$$\overline{\mathrm{res}}_A(K_1, K_2) \rightleftharpoons K_1' \wedge K_2',\ \overline{\mathrm{res}}(A, \neg A) \rightleftharpoons 1.$$

Пусть $S = \{K_1, K_2, \ldots, K_m\}$ — множество конъюнктов. Последовательность формул $\varphi_1, \varphi_2, \ldots, \varphi_n$ называется *выводом из S по правилу согласия*, если для каждой формулы φ_i $(i = 1, \ldots, n)$ выполняется какое-то из следующих условий:
— $\varphi_i \in S$;
— существуют $j, k < i$ такие, что $\varphi_i = \overline{\mathrm{res}}(\varphi_j, \varphi_k)$.

Теорема 1.7.4. *Множество конъюнктов $S = \{K_1, K_2, \ldots, K_m\}$ общезначимо (т.е. выполняется $\models K_1 \vee K_2 \vee \ldots \vee K_m$) тогда и только тогда, когда существует вывод из S по правилу согласия, заканчивающийся символом 1.*

4. Метод резолюций для хорновских дизъюнктов. В общем случае метод резолюций неэффективен, так как количество переборов, которые необходимо сделать для получения ответа, экспоненциально зависит от количества информации (числа дизъюнктов и переменных), содержащейся в множестве дизъюнктов. Однако для некоторых классов дизъюнктов, к которым относится класс так называемых хорновских дизъюнктов, метод резолюций эффективен.

Дизъюнкт D называется *хорновским*, если он содержит не более одной позитивной литеры.

П р и м е р 1.7.5. Хорновскими дизъюнктами являются следующие дизъюнкты: $\neg A \vee \neg B \vee \neg C \vee D$, $\neg A \vee \neg B \vee \neg Q$, $\neg A$, B. $\square$

В общем случае хорновские дизъюнкты имеют вид $\neg A_1 \vee \ldots \vee \neg A_n \vee B$ (что эквивалентно формуле $(A_1 \wedge \ldots \wedge A_n) \to B$) или $\neg A_1 \vee \ldots \vee \neg A_n$. Хорновский дизъюнкт вида $\neg A_1 \vee \cdots \vee \neg A_n \vee B$ называется *точным*. При этом переменные $A_1, \ldots, A_n$ называются *фактами*, а переменная B — *целью*. Хорновский дизъюнкт вида $\neg A_1 \vee \ldots \vee \neg A_n$ называется *негативным*. Дизъюнкт $D = B$ называется *унитарным позитивным дизъюнктом*.

Если S — множество хорновских дизъюнктов, то невыполнимость множества S проверяется следующим образом. Выбираем в S унитарный позитивный дизъюнкт P и дизъюнкт D из S, содержащий $\neg P$. После этого заменяем S на $(S \setminus \{D\}) \cup \{\mathrm{res}(D, P)\}$ и продолжаем процесс до тех пор, пока S не будет содержать 0 или не найдется дизъюнктов P и D указанного вида. Если на заключительном шаге множество дизъюнктов будет содержать 0, то исходное множество S противоречиво, в противном случае S непротиворечиво.

П р и м е р 1.7.6. Проверить на противоречивость множество дизъюнктов $S = \{P \vee \neg R \vee \neg T, Q, R, T \vee \neg P \vee \neg R, T \vee \neg Q, \neg P \vee \neg Q \vee \neg R\}$.

Для доказательства противоречивости запишем дизъюнкты из S в таблицу и применим описанный алгоритм, записывая результат каждого следующего шага в таблицу. Литеры, использующиеся на данном шаге, будем подчеркивать.

Номер шага	$P \vee \neg R \vee \neg T$	Q	R	$T \vee \neg P \vee \neg R$	$T \vee \neg Q$	$\neg P \vee \neg Q \vee \neg R$
1	$P \vee \neg R \vee \neg T$	Q	R	$T \vee \neg P \vee \neg R$	T	$\neg P \vee \neg R$
2	$P \vee \neg T$	Q	R	$T \vee \neg P$	T	$\neg P$
3	P	Q	R	$T \vee \neg P$	T	$\neg P$
4						0

На шаге 4 получаем 0, являющийся резолютивным выводом из S. Следовательно, множество S невыполнимо. $\square$

§ 1.8. Логические задачи

Аппарат исчислений высказываний позволяет решать так называемые “логические” задачи. При этом самым трудным моментом является построение “модели” задачи, т. е. выделение элементарных высказываний и сведение задачи к проверке некоторых свойств высказываний, возникающих из условий задачи.

П р и м е р 1.8.1. На следствии по делу о похищении автомобиля были допрошены четыре гангстера — Андре, Боб, Стив, Том. Андре сказал, что машину похитил Боб, Боб утверждал, что виноват Том.

Том заверил следователя, что Боб лжет. Стив настаивал, что автомобиль угнал не он. Следователю удалось установить, что только один из гангстеров сказал правду. Кто похитил автомобиль?

Р е ш е н и е. Обозначим высказывания "Андре украл", "Боб украл", "Стив украл", "Том украл" через A, B, S и T соответственно. Тогда показания гангстеров имеют вид B, T, $\neg T$, $\neg S$. Поскольку секвенция $\vdash T \vee \neg T$ доказуема, или, что то же самое, формула $T \vee \neg T$ тождественно истинна, то одно из утверждений — T или $\neg T$ обязательно истинно. Значит, Андре и Стив сказали ложь. Так как утверждение Стива "$\neg S$" ложно, то автомобиль украл Стив.

§ 1.9. Задачи и упражнения

1. Построить выводы секвенций в ИС:
   (а) $\vdash (\varphi \to \varphi)$;
   (б) $(\varphi \to \psi), (\psi \to \chi) \vdash (\varphi \to \chi)$;
   (в) $\vdash (\neg\neg\varphi \leftrightarrow \varphi)$;
   (г) $(\varphi \to (\psi \to \chi)), (\varphi \to \psi), \varphi \vdash \chi$;
   (д) $(\varphi \to \psi), \neg\psi \vdash \neg\varphi$;
   (е) $\varphi, \neg\psi \vdash \neg(\varphi \to \psi)$;
   (ж) $\vdash ((\varphi \to \psi) \vee (\psi \to \varphi))$.

2. Доказать допустимость правил (а) – (о) из предложения 1.2.2.

3. Доказать выводимость в ИВ:
   (а) $(\varphi \to \psi), (\psi \to \chi) \vdash (\varphi \to \chi)$;
   (б) $(\varphi \to (\psi \to \chi)) \vdash (\psi \to (\varphi \to \chi))$;
   (в) $(\varphi \to (\psi \to \chi)) \vdash ((\varphi \wedge \psi) \to \chi)$;
   (г) $(\varphi \to \psi) \vdash ((\varphi \wedge \chi) \to (\psi \wedge \chi))$;
   (д) $(\varphi \to \psi) \vdash ((\varphi \vee \chi) \to (\psi \vee \chi))$;
   (е) $\neg\varphi \vdash (\varphi \to \psi)$.

4. Выводимы ли в ИВ следующие формулы:
   (а) $((A \vee B) \to (A \wedge C))$; (б) $(((A \to B) \to B) \to A)$;
   (в) $(((A \to B) \to B) \to B)$; (г) $(\neg(A \vee \neg A) \to (A \vee \neg A))$;
   (д) $(A \to \neg(A \to \neg A))$; (е) $((A \to B) \to (B \to A))$?

5. С помощью алгоритма Квайна и алгоритма редукции доказать тождественную истинность аксиом ИВ.

6. С помощью алгоритма Квайна и алгоритма редукции проверить общезначимость следующих формул:
   (а) $((A \to B) \to (B \to A))$;
   (б) $((A \vee B) \to ((\neg A \wedge B) \vee (\neg B \wedge A)))$.

7. Методом резолюций проверить следующие соотношения:
   (а) $(\overline{A} \vee C), (C \to B), (B \to A) \vdash (A \to (B \to C))$;
   (б) $((A \vee C) \to B), (C \to (A \vee B)), (BC \to (A \vee \overline{B})) \vdash (B \to C)$.

8. Проверить на противоречивость множество хорновских дизъюнктов $\{A \vee \overline{B} \vee \overline{C} \vee \overline{D}, B, \overline{A} \vee B \vee \overline{C}, C, \overline{B} \vee D\}$.

9. Состоялся розыгрыш футбольного кубка между командами “Пламя”, “Рекорд”, “Стрела” и “Трактор”. Было высказано три прогноза: победит “Пламя” или “Рекорд”; не победит “Пламя”; не победит ни “Рекорд”, ни “Трактор”. Известно, что подтвердился только один прогноз. Какая команда выиграла кубок?

После изучения главы 1 выполняется задача 1 контрольной работы. Она решается аналогично примерам 1.2.1, 1.7.1, 1.7.2 и 1.7.4.

Г л а в а 2

ЛОГИКА И ИСЧИСЛЕНИЕ ПРЕДИКАТОВ

В построенной выше логике высказываний в качестве исходных элементов рассматриваются некоторые элементарные высказывания, из которых строятся более сложные высказывания, называемые формулами. При этом не анализируются структура и состав высказываний, а учитываются лишь значения истины или лжи, которые они могут принимать. Однако имеется много мыслей, которые не могут быть рассмотрены таким простым способом. Например, приведем следующее умозаключение:

Каждый человек смертен.
Сократ — человек, следовательно, он смертен.

Очевидно, рассмотренное рассуждение интуитивно корректно. Однако, введя следующие обозначения:

P: Каждый человек смертен,
Q: Сократ — человек,
R: Сократ смертен,

мы получаем формулу $P \wedge Q \to R$, которая не доказуема в ИВ. Указанное несоответствие между утверждениями имеет место потому, что в логике высказываний не используется структура высказываний P, Q и R. В этой главе мы введем *логику предикатов* (*логику первого порядка*) и исчисления предикатов, которые позволяют преодолевать подобные трудности и дают возможность проводить формализацию бóльшей части повседневного и математического языка. Аналогично исчислению секвенций и исчислению высказываний мы будем рассматривать два их расширения — секвенциальное исчисление предикатов данной сигнатуры Σ (ИПС^{Σ}) и исчисление предикатов гильбертовского типа ИП^{Σ}. Построение исчислений предикатов данной сигнатуры мы начнем с понятий синтаксиса и семантики формулы.

§ 2.1. Формулы сигнатуры Σ. Истинность формулы на алгебраической системе

Зафиксируем некоторую сигнатуру Σ и счетное множество $V = \{v_i \mid i \in \omega\}$, элементы которого будем называть *переменными* и обозначать буквами x, y и z, возможно, с индексами. *Алфавит исчисления предикатов сигнатуры* Σ (ИП^Σ) состоит из следующих групп символов:

1. Предикатные и функциональные символы, образующие сигнатуру Σ.

2. Символы переменных, составляющих множество V.

3. Символ равенства: $\approx$.

4. Логические связки: $\neg$, $\wedge$, $\vee$, $\to$.

5. *Кванторы*: $\forall$, $\exists$.

6. Вспомогательные символы: левая скобка (, правая скобка), запятая ,.

Обозначим через $T(\Sigma)$ множество всех термов сигнатуры Σ, в определении которых используются лишь переменные из V. Очевидно, любой терм из $T(\Sigma)$ является словом алфавита ИП^Σ.

Введем понятие *атомарной формулы* сигнатуры Σ:

1) если t_1, $t_2 \in T(\Sigma)$, то $(t_1 \approx t_2)$ — атомарная формула;

2) если $P^{(n)} \in \Sigma$ — предикатный символ, t_1, t_2, …, $t_n \in T(\Sigma)$, то $P(t_1, t_2, \ldots, t_n)$ — атомарная формула;

3) никаких атомарных формул, кроме построенных по пп. 1, 2, нет.

Формула сигнатуры Σ определяется следующим образом:

1) атомарная формула есть формула;

2) если φ, ψ — формулы и $x \in V$, то $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$, $\forall x\, \varphi$, $\exists x\, \varphi$ — формулы;

3) никаких формул, кроме построенных по пп. 1, 2, нет.

Символы $\forall$, $\exists$, использованные в определении, называются соответственно *квантором всеобщности* и *квантором существования.* Запись $\forall x$ (соответственно $\exists x$) читается “для всех x” (“существует x”). Все соглашения относительно расстановок скобок, принятые для формул исчисления высказываний, остаются в силе и для формул логики предикатов. Кроме того, вместо записей $\forall x_1 \ldots \forall x_n\, \varphi$ и $\exists x_1 \ldots \exists x_n\, \varphi$ будем использовать записи $\forall x_1, \ldots, x_n\, \varphi$ и $\exists x_1, \ldots, x_n\, \varphi$. Формула φ сигнатуры Σ называется *бескванторной*, если она не содержит кванторов.

П р и м е р 2.1.1. 1. Рассмотрим предикатную сигнатуру $\Sigma = \{Q^{(1)}, R^{(1)}, P^{(1)}, \text{меньше}^{(2)}\}$ со следующими интерпретациями:

$Q(x) \Leftrightarrow x$ — рациональное число,

$R(x) \Leftrightarrow x$ — вещественное число,

$P(x) \Leftrightarrow x$ — простое число,

$\text{меньше}(x, y) \Leftrightarrow x < y$.

Формула $\forall x\,(Q(x) \to R(x))$ означает, что любое рациональное число является вещественным, а формула $\forall x \exists y\,(\text{меньше}(x, y) \wedge P(y))$ — для любого элемента x найдется бóльший элемент y, являющийся простым числом.

2. Рассмотрим сигнатуру $\Sigma = \{A^{(1)}, B^{(1)}, \text{Сократ}^{(0)}\}$ со следующими интерпретациями: $A(x) \Leftrightarrow x$ — человек, $B(x) \Leftrightarrow x$ смертен, константный символ "Сократ" — известный древнегреческий философ. Формула $\forall x\,(A(x) \to B(x))$ читается как "Каждый человек смертен", а формула $A(\text{Сократ}) \to B(\text{Сократ})$ означает, что Сократ смертен, если он является человеком.

3. Любая бескванторная формула сигнатуры нульместных предикатов может рассматриваться как формула исчисления высказываний. Например, таковой является формула $A \vee B \to \neg C$ сигнатуры $\Sigma = \{A^{(0)}, B^{(0)}, C^{(0)}\}$. □

Определим множество $\mathrm{SF}(\varphi)$ *подформул* формулы φ сигнатуры Σ:

1) если φ — атомарная формула, то φ — ее единственная подформула и $\mathrm{SF}(\varphi) = \{\varphi\}$;

2) если φ имеет вид $\neg\varphi_1$, или $\forall x\,\varphi_1$, или $\exists x\,\varphi_1$, то подформула формулы φ — это либо φ, либо подформула формулы φ_1 и $\mathrm{SF}(\varphi) = \mathrm{SF}(\varphi_1) \cup \{\varphi\}$;

3) если φ имеет вид $\varphi_1 \wedge \varphi_2$, или $\varphi_1 \vee \varphi_2$, или $\varphi_1 \to \varphi_2$, то подформула формулы φ — это либо φ, либо подформула формулы φ_1, либо подформула формулы φ_2 и $\mathrm{SF}(\varphi) = \mathrm{SF}(\varphi_1) \cup \mathrm{SF}(\varphi_2) \cup \{\varphi\}$.

П р и м е р 2.1.2. Пусть $\varphi \rightleftharpoons \forall x\,(\exists y\,(x \approx F(z, y)) \vee \neg P(z))$ — формула сигнатуры $\Sigma = \{F^{(2)}, P^{(1)}\}$. Тогда $\forall x\,(\exists y\,(x \approx F(z, y)) \vee \neg P(z))$, $\exists y\,(x \approx F(z, y)) \vee \neg P(z)$, $\exists y\,(x \approx F(z, y))$, $(x \approx F(z, y))$, $\neg P(z)$, $P(z)$ — все подформулы формулы φ. □

Каждое вхождение квантора $\forall$ ($\exists$) в данную формулу φ однозначно определяет некоторое вхождение $\forall x\,\psi$ ($\exists x\,\psi$) подформулы из $\mathrm{SF}(\varphi)$, первым символом которого является рассматриваемое вхождение соответствующего квантора. Формула $\forall x\,\psi$ ($\exists x\,\psi$), связанная с вхождением квантора $\forall$ ($\exists$), называется *областью действия* этого *вхождения квантора* $\forall$ (квантора $\exists$).

Заметим, что в записи формулы возможно наложение области действия вхождения одного квантора с данной переменной на область действия другого квантора с той же самой переменной. Например, в формуле $\forall x\,(P_2(x,y) \to \exists x\,P_1(x))$ вхождение переменной x в $P_1(x)$ находится одновременно под кванторами $\forall x$ и $\exists x$. Поскольку число переменных, образующих множество V, бесконечно, мы будем избегать подобных коллизий введением новых переменных для меньших областей действия вхождений кванторов. Так, вместо формулы $\forall x\,(P_2(x,y) \to \exists x\,P_1(x))$ будет рассматриваться, например, формула $\forall x\,(P_2(x,y) \to \exists z\,P_1(z))$.

Говорят, что вхождение переменной x в формулу φ *связано* в φ, если оно находится в области действия некоторого вхождения квантора в формулу φ, имеющую вид $\forall x\,\psi$ или $\exists x\,\psi$; в противном случае это вхождение называется *свободным* в φ. Переменная x называется *свободной* (*связанной*), если некоторое вхождение x в φ свободно (связано).

П р и м е р 2.1.3. Рассмотрим следующие формулы сигнатуры $S = \{P_1^{(1)}, P_2^{(2)}\}$:
— $\neg P_1(x)$;
— $P_2(x,y) \to \forall x\,P_1(x)$;
— $\forall x\,(P_2(x,y) \to P_1(x))$.
Переменная x в первой формуле является свободной, во второй — и свободной, и связанной, в третьей — связанной; переменная y во всех формулах свободна. □

Предложением или *замкнутой формулой* сигнатуры Σ называется формула сигнатуры Σ, не имеющая свободных переменных.

П р и м е р 2.1.4. Формула $\forall x, y\,(x+y \approx y+x)$ является предложением сигнатуры $\{+\}$, формула $\exists z \forall x\,(\exists y\,P_2(x,y) \to \neg P_1(x))$ — предложением сигнатуры $\{P_1^{(1)}, P_2^{(2)}\}$, а формула $\forall x\,(P_2(x,y) \to P_1(x))$ предложением не является. □

Запись $\varphi(x_1, \ldots, x_n)$ будет означать, что все свободные переменные формулы φ содержатся в множестве $\{x_1, \ldots, x_n\}$.

Дадим индуктивное определение *истинности формулы* $\varphi(x_1, \ldots, x_n)$ сигнатуры Σ на элементах $a_1, \ldots, a_n \in A$ в алгебраической системе $\mathfrak{A} = \langle A; \Sigma\rangle$ (запись $\mathfrak{A} \models \varphi(a_1, \ldots, a_n)$ будет означать, что формула φ *истинна* на элементах $a_1, \ldots, a_n \in A$ в системе $\mathfrak{A}$):

1) если $t_1, t_2 \in T(\Sigma)$, то $\mathfrak{A} \models (t_1(a_1, \ldots, a_n) \approx t_2(a_1, \ldots, a_n)) \Leftrightarrow$ значения термов t_1, t_2 в системе $\mathfrak{A}$ на элементах $a_1, \ldots, a_n \in A$ совпадают;

2) если $P^{(k)}$ — предикатный символ сигнатуры Σ, t_1, …, $t_k \in T(\Sigma)$, то $\mathfrak{A} \models P(t_1(a_1, \ldots, a_n), \ldots, t_k(a_1, \ldots, a_n)) \Leftrightarrow$
$\langle t_1(a_1, \ldots, a_n), \ldots, t_k(a_1, \ldots, a_n)\rangle \in P_{\mathfrak{A}}$;

3) $\mathfrak{A} \models (\psi(a_1, \ldots, a_n) \wedge \chi(a_1, \ldots, a_n)) \Leftrightarrow \mathfrak{A} \models \psi(a_1, \ldots, a_n)$ и $\mathfrak{A} \models \chi(a_1, \ldots, a_n)$;

4) $\mathfrak{A} \models (\psi(a_1, \ldots, a_n) \vee \chi(a_1, \ldots, a_n)) \Leftrightarrow \mathfrak{A} \models \psi(a_1, \ldots, a_n)$ или $\mathfrak{A} \models \chi(a_1, \ldots, a_n)$;

5) $\mathfrak{A} \models (\psi(a_1, \ldots, a_n) \to \chi(a_1, \ldots, a_n)) \Leftrightarrow$ если $\mathfrak{A} \models \psi(a_1, \ldots, a_n)$, то $\mathfrak{A} \models \chi(a_1, \ldots, a_n)$;

6) $\mathfrak{A} \models \neg\psi(a_1, \ldots, a_n) \Leftrightarrow$ неверно, что $\mathfrak{A} \models \psi(a_1, \ldots, a_n)$;

7) $\mathfrak{A} \models \forall x\, \psi(x, a_1, \ldots, a_n) \Leftrightarrow \mathfrak{A} \models \psi(a, a_1, \ldots, a_n)$ для любого $a \in A$;

8) $\mathfrak{A} \models \exists x\, \psi(x, a_1, \ldots, a_n) \Leftrightarrow \mathfrak{A} \models \psi(a, a_1, \ldots, a_n)$ для некоторого $a \in A$.

Если не выполняется $\mathfrak{A} \models \varphi(a_1, \ldots, a_n)$, то будем говорить, что формула φ *ложна* на элементах $a_1, \ldots, a_n$ в системе $\mathfrak{A}$, и писать $\mathfrak{A} \not\models \varphi(a_1, \ldots, a_n)$.

П р и м е р 2.1.5. Рассмотрим формулу $\varphi(x, y)$ функциональной сигнатуры $\{+^{(2)}, 0^{(0)}\}$, имеющую вид $(x + y \approx 0)$. Для алгебраической системы $\mathfrak{A} = \langle \mathbb{Z}; +, 0\rangle$ тогда и только тогда имеет место $\mathfrak{A} \models \varphi(m, n)$, когда $m = -n$. Для формулы $\psi(x) \rightleftharpoons \exists y\, (x + y \approx 0)$ справедливо $\mathfrak{A} \models \psi(n)$ для любого целого числа n, поскольку у любого целого числа имеется к нему противоположное и так же целое число. Следовательно, $\mathfrak{A} \models \forall x \exists y\, (x + y \approx 0)$. Отметим, что $\mathfrak{A} \not\models \exists y \forall x\, (x + y \approx 0)$, так как нет единого целого числа, противоположного ко всем целым числам.

П р и м е р 2.1.6. 1. Записать формулу $\varphi(x)$, истинную в $\langle \omega; +, \cdot\rangle$ тогда и только тогда, когда x четно.

Искомая формула $\varphi(x)$ имеет, например, вид $\exists y\, (x \approx y + y)$.

2. Записать формулу $\varphi'(x, y, z)$, истинную в $\langle \omega; +, \cdot\rangle$ тогда и только тогда, когда z — наименьшее общее кратное чисел x и y.

Положим $\varphi'(x, y, z) \rightleftharpoons \psi(x, y, z) \wedge \chi(x, y, z)$, где формула ψ "говорит" о том, что z делится на x и на y, а формула χ "говорит" о том, что z делит все общие кратные x и y, т. е. является наименьшим из всех общих кратных:

$$\psi \rightleftharpoons \exists u, v\, ((z \approx u \cdot x) \wedge (z \approx v \cdot y)),$$

$$\chi \rightleftharpoons \forall w\, (\exists u, v\, ((w \approx u \cdot x) \wedge (w \approx v \cdot y)) \to \exists w_1\, (w \approx w_1 \cdot z)).$$

Итак, $\varphi'(x, y, z)$ имеет вид $\exists u, v\, ((z \approx u{\cdot}x) \wedge (z \approx v{\cdot}y)) \wedge \forall w\, (\exists u, v\, ((w \approx u \cdot x) \wedge (w \approx v \cdot y)) \to \exists w_1\, (w \approx w_1 \cdot z))$. $\square$

Формула $\varphi(x_1, \ldots, x_n)$ сигнатуры Σ называется *тождественно истинной* или *общезначимой* (*тождественно ложной* или *противоречивой*), если для любой алгебраической системы $\mathfrak{A} = \langle A; \Sigma \rangle$ и любого кортежа элементов $(a_1, \ldots, a_n) \in A^n$ выполнено $\mathfrak{A} \models \varphi(a_1, \ldots, a_n)$ ($\mathfrak{A} \not\models \varphi(a_1, \ldots, a_n)$). Если φ — тождественно истинное предложение, то пишем $\models \varphi$.

Формула $\varphi(x_1, \ldots, x_n)$ называется *выполнимой* в системе $\mathfrak{A}$, если $\mathfrak{A} \models \varphi(a_1, \ldots, a_n)$ для некоторых $a_1, \ldots, a_n \in A$.

П р и м е р 2.1.7. 1. Формула $(x \approx x)$ общезначима, поскольку $\mathfrak{A} \models (a \approx a)$ для любой системы $\mathfrak{A} = \langle A; \Sigma \rangle$ и любого элемента $a \in A$. По этой же причине формула $\neg(x \approx x)$ тождественно ложна.

2. Формула $\varphi \rightleftharpoons \forall x \exists y\, (x + y \approx 0)$ выполнима, но не общезначима, поскольку $\langle \mathbb{Z}; +, 0 \rangle \models \forall x \exists y\, (x + y \approx 0)$ и $\langle \mathbb{N}; +, 0 \rangle \not\models \forall x \exists y\, (x + y \approx 0)$. Тем самым предложение φ описывает одно из характерных свойств, отличающих систему $\langle \mathbb{Z}; +, 0 \rangle$ от системы $\langle \mathbb{N}; +, 0 \rangle$.

3. Формула $\varphi \rightleftharpoons \forall x, y\, (x \cdot y \approx y \cdot x)$ выполнима, но не тождественно истинна, так как $\langle \mathbb{Z}; \cdot \rangle \models \varphi$, а $\langle M_2(\mathbb{Z}); \cdot \rangle \not\models \varphi$, где $M_2(\mathbb{Z})$ — множество матриц порядка 2 с элементами из $\mathbb{Z}$.

4. Определим выполнимость формулы $\varphi(y) \rightleftharpoons \neg \exists x\, (P(x) \wedge Q(x, y))$ в системе $\mathfrak{A} = \langle \{1, 2\}; P, Q \rangle$, где $P = \{1\}$, $Q = \{(1, 1), (1, 2)\}$. Составим таблицу истинности предикатов P и Q:

$P(1)$	$P(2)$	$Q(1,1)$	$Q(1,2)$	$Q(2,1)$	$Q(2,2)$
1	0	1	1	0	0

,

где $P(a) = 1 \Leftrightarrow \mathfrak{A} \models P(a)$, $Q(a, b) = 1 \Leftrightarrow \mathfrak{A} \models Q(a, b)$. По таблице истинности предикатов P и Q составляем таблицу истинности формулы $P(x) \wedge Q(x, y)$:

x	y	$P(x) \wedge Q(x, y)$
1	1	1
1	2	1
2	1	0
2	2	0

из которой следует, что $\mathfrak{A} \models \exists x\, (P(x) \wedge Q(x, 1))$, поскольку $\mathfrak{A} \models P(1) \wedge Q(1, 1)$, а также $\mathfrak{A} \models \exists x\, (P(x) \wedge Q(x, 2))$, так как $\mathfrak{A} \models P(1) \wedge Q(1, 2)$. Таким образом, $\mathfrak{A} \models \neg\varphi(1)$ и $\mathfrak{A} \models \neg\varphi(2)$, т. е. $\mathfrak{A} \not\models \varphi(1)$ и $\mathfrak{A} \not\models \varphi(2)$. Следовательно, формула φ не выполнима в системе $\mathfrak{A}$.

5. Докажем выполнимость формулы $\exists x, y, u, v\, (P(x, y) \wedge \neg P(u, v))$ предикатной сигнатуры $\Sigma = \{P^{(2)}\}$.

Действительно, рассмотрим двухэлементную алгебраическую систему $\mathfrak{A} = \langle \{a, b\}, P \rangle$ с интерпретацией $P = \{(a, b)\}$. Тогда по опреде-

лению выполняется $\mathfrak{A} \models P(a,b)$ и $\mathfrak{A} \not\models P(b,a)$, т.е. $\mathfrak{A} \models P(a,b) \wedge \neg P(b,a)$. Тем самым в системе $\mathfrak{A}$ истинна формула $\varphi(x,y,u,v) \rightleftharpoons P(x,y) \wedge \neg P(u,v)$ на наборе элементов (a,b,b,a). Тогда $\mathfrak{A} \models \exists x,y,u,v\,(P(x,y) \wedge \neg P(u,v))$, т.е. данная формула выполнима. $\square$

Как и раньше, для формул φ и ψ через $(\varphi \leftrightarrow \psi)$ будем обозначать формулу $(\varphi \to \psi) \wedge (\psi \to \varphi)$.

Предложение 2.1.1. 1. *Для любой формулы φ сигнатуры Σ следующие формулы общезначимы:*

(а) $\forall x\,\neg\varphi \leftrightarrow \neg\exists x\,\varphi$;

(б) $\exists x\,\neg\varphi \leftrightarrow \neg\forall x\,\varphi$;

(в) $\forall x,y\,\varphi \leftrightarrow \forall y,x\,\varphi$;

(г) $\exists x,y\,\varphi \leftrightarrow \exists y,x\,\varphi$;

(д) $\exists x \forall y\,\varphi \to \forall y \exists x\,\varphi$.

2. *Для любой формулы φ сигнатуры Σ формула $\forall x\,\varphi \wedge \exists x\,\neg\varphi$ противоречива.* $\square$

Заметим, что на основании примера 2.1.5 общезначимость формулы $\forall y \exists x\,\varphi \to \exists x \forall y\,\varphi$ в общем случае утверждать нельзя. В следующем примере также иллюстрируется отсутствие общезначимости указанной формулы для случая $\varphi = P(x,y)$.

П р и м е р 2.1.8. В графе $\mathfrak{A} = \langle A, P\rangle = \langle\{1,2,3\};$ $\{(1,2),\ (2,3),\ (3,1)\}\rangle$ (рис. 2.1) выполнимо $\mathfrak{A} \models \forall y\,\exists x\,P(x,y)$, но $\mathfrak{A} \not\models \exists x\,\forall y\,P(x,y)$.

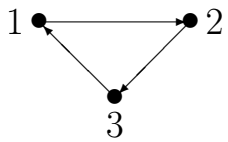

Рис. 2.1

Следующая теорема позволяет создавать общезначимые и противоречивые формулы на основе формул ИВ.

Теорема 2.1.2. *Пусть $\varphi(A_1,\ldots,A_n)$ — общезначимая (противоречивая) формула ИВ, $\varphi_1,\ldots,\varphi_n$ — формулы сигнатуры Σ. Тогда в результате подстановки формул $\varphi_1,\ldots,\varphi_n$ вместо всех соответствующих вхождений пропозициональных переменных $A_1,\ldots,A_n$ образуется общезначимая (противоречивая) формула $\varphi(\varphi_1,\ldots,\varphi_n)$ сигнатуры Σ.*

Теорема 2.1.3. *Если f — изоморфизм системы $\mathfrak{A}$ на систему $\mathfrak{B}$, $\varphi(x_1,\ldots,x_n)$ — формула сигнатуры системы $\mathfrak{A}$, то для любых $a_1,\ldots,a_n \in A$ свойство $\mathfrak{A} \models \varphi(a_1,\ldots,a_n)$ эквивалентно свойству $\mathfrak{B} \models \varphi(f(a_1),\ldots,f(a_n))$.*

Множество формул Γ сигнатуры Σ с множеством свободных переменных X называется *выполнимым*, если существует система $\mathfrak{M} = \langle M;\Sigma\rangle$, элементы $a_x \in M$ для каждого $x \in X$ такие, что для любой формулы $\varphi(x_1,\ldots,x_n) \in \Gamma$ выполнимо $\mathfrak{M} \models \varphi(a_{x_1},\ldots,a_{x_n})$. Система

$\mathfrak{M}$ называется *моделью множества формул* Γ, и этот факт обозначается через $\mathfrak{M} \models \Gamma$.

П р и м е р 2.1.9. Рассмотрим следующее множество формул сигнатуры $\{\leq\}$, описывающих линейные порядки: $\Gamma_{\text{lo}} = \{\forall x\,(x \leq x), \forall x, y, z\,(((x \leq y) \wedge (y \leq z)) \to (x \leq z)), \forall x, y\,(((x \leq y) \wedge (y \leq x)) \to (x \approx y)), \forall x, y\,((x \leq y) \vee (y \leq x))\}$. Очевидно, что множество Γ_{lo} имеет как конечные, так и бесконечные модели. Например, $\langle\{0\}; \leq\rangle \models \Gamma_{\text{lo}}$, где $\leq = \{(0,0)\}$, $\langle\mathbb{N}; \leq\rangle \models \Gamma_{\text{lo}}$. Добавив к множеству Γ_{lo} две формулы $\exists x, y\,\neg(x \approx y)$ и $\forall x, y((x \leq y) \wedge \neg(x \approx y) \to \exists z\,((x \leq z) \wedge \neg(x \approx z) \wedge (z \leq y) \wedge \neg(z \approx y)))$, получаем множество Γ_{dlo}, описывающее бесконечные *плотные линейные порядки*. Моделями множества Γ_{dlo} являются в точности бесконечные линейно упорядоченные множества, у которых между любыми двумя различными элементами имеется некоторый промежуточный. Примерами таких моделей служат $\langle\mathbb{Q}; \leq\rangle$ и $\langle\mathbb{R}; \leq\rangle$. $\square$

§ 2.2. Секвенциальное исчисление предикатов

Зафиксируем некоторую произвольную сигнатуру Σ и определим *секвенциальное исчисление предикатов сигнатуры* Σ (ИПС$^\Sigma$).

Алфавит ИПС$^\Sigma$ получается из алфавита ИП$^\Sigma$ добавлением символа следования $\vdash$, т.е. $A(\text{ИПС}^\Sigma) = \Sigma \cup V \cup \{\approx, \neg, \wedge, \vee, \to, \forall, \exists, (,), , , \vdash\}$. *Формулами* ИПС$^\Sigma$ будут формулы сигнатуры Σ.

Секвенциями ИПС$^\Sigma$ называются конечные последовательности следующих двух видов, где $\varphi_1, \ldots, \varphi_n, \psi$ — формулы ИПС$^\Sigma$:

$$\varphi_1, \ldots, \varphi_n \vdash \psi; \quad \varphi_1, \ldots, \varphi_n \vdash .$$

Примем следующие соглашения. Пусть $x_1, \ldots, x_n$ — переменные, $t_1, \ldots, t_n$ — термы сигнатуры Σ и φ — формула сигнатуры Σ. Запись $(\varphi)^{x_1,\ldots,x_n}_{t_1,\ldots,t_n}$ будет обозначать результат подстановки термов $t_1, \ldots, t_n$ вместо всех свободных вхождений в φ переменных $x_1, \ldots, x_n$ соответственно, причем, если в тексте встречается запись $(\varphi)^{x_1,\ldots,x_n}_{t_1,\ldots,t_n}$, то предполагается, что для всех $i = 1, \ldots, n$ ни одно свободное вхождение в φ переменной x_i не входит в подформулу φ вида $\forall y\,\varphi_1$ или $\exists y\,\varphi_1$, где y — переменная, входящая в t_i. Вместо записи $(\varphi)^{x_1,\ldots,x_n}_{t_1,\ldots,t_n}$ мы будем иногда писать $\varphi(t_1, \ldots, t_n)$. При этом, наряду с тем, что переменные $x_1, \ldots, x_n$ в записи $\varphi(x_1, \ldots, x_n)$ будут всегда предполагаться попарно различными, среди $t_1, \ldots, t_n$ могут быть равные термы.

П р и м е р 2.2.1. 1. Обозначим через φ формулу $\exists x\,R(x, y, z)$, через t_1 — терм $F_1(y, z, u)$, через t_2 — терм $F_2(F_3(z), w)$. Результат подста-

новки $(\varphi)^{y,z}_{t_1,t_2}$ равен $\exists x\, R(x, F_1(y,z,u), F_2(F_3(z), w))$, а результат подстановки $(\varphi)^{y,z}_{t_1,t_1}$ равен $\exists x\, R(x, F_1(y,z,u), F_1(y,z,u))$. Заметим, что результат последовательной подстановки может не совпадать с результатом одновременной подстановки. Например, формула $((\varphi)^{y}_{t_1})^{z}_{t_2}$ равна $\exists x\, R(x, F_1(y, F_2(F_3(z), w), u), F_2(F_3(z),\, w))$ и не совпадает с формулой $(\varphi)^{y,z}_{t_1,t_2}$.

2. Для формулы $\varphi = \exists x\, R(x,y,z)$ и терма $t = F(y,x,u)$ запись $(\varphi)^{y}_{t}$ недопустима, поскольку после подстановки терма t вместо свободной переменной y в формуле $\exists x\, R(x, F(y,x,u), z)$ вхождение переменной x в терм t становится связанным. $\square$

Аксиомами ИПС$^\Sigma$ являются следующие секвенции:

1) $\varphi \vdash \varphi$, где φ — формула сигнатуры Σ;

2) $\vdash (x \approx x)$, где x — переменная;

3) $(t_1 \approx q_1), \ldots, (t_n \approx q_n), (\varphi)^{x_1,\ldots,x_n}_{t_1,\ldots,t_n} \vdash (\varphi)^{x_1,\ldots,x_n}_{q_1,\ldots,q_n}$, где $t_1, \ldots, t_n, q_1, \ldots, q_n$ — термы сигнатуры Σ, φ — формула, удовлетворяющая условиям на записи $(\varphi)^{x_1,\ldots,x_n}_{t_1,\ldots,t_n}$ и $(\varphi)^{x_1,\ldots,x_n}_{q_1,\ldots,q_n}$.

Правила вывода ИПС$^\Sigma$ задаются следующими записями, где Γ, Γ_1 — произвольные (возможно пустые) конечные последовательности формул ИПС$^\Sigma$, φ, ψ, χ — произвольные формулы ИПС$^\Sigma$.

1. $\dfrac{\Gamma \vdash \varphi;\ \Gamma \vdash \psi}{\Gamma \vdash (\varphi \wedge \psi)}$ (введение $\wedge$).

2. $\dfrac{\Gamma \vdash (\varphi \wedge \psi)}{\Gamma \vdash \varphi}$ (удаление $\wedge$).

3. $\dfrac{\Gamma \vdash (\varphi \wedge \psi)}{\Gamma \vdash \psi}$ (удаление $\wedge$).

4. $\dfrac{\Gamma \vdash \varphi}{\Gamma \vdash (\varphi \vee \psi)}$ (введение $\vee$).

5. $\dfrac{\Gamma \vdash \psi}{\Gamma \vdash (\varphi \vee \psi)}$ (введение $\vee$).

6. $\dfrac{\Gamma, \varphi \vdash \psi;\ \Gamma, \chi \vdash \psi;\ \Gamma \vdash (\varphi \vee \chi)}{\Gamma \vdash \psi}$ (удаление $\vee$, или правило разбора двух случаев).

7. $\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash (\varphi \to \psi)}$ (введение $\to$).

8. $\dfrac{\Gamma \vdash \varphi;\ \Gamma \vdash (\varphi \to \psi)}{\Gamma \vdash \psi}$ (удаление $\to$).

9. $\dfrac{\Gamma, \neg\varphi \vdash}{\Gamma \vdash \varphi}$ (удаление $\neg$, или доказательство от противного).

10. $\dfrac{\Gamma \vdash \varphi;\ \Gamma \vdash \neg\varphi}{\Gamma \vdash}$ (выведение противоречия).

11. $\dfrac{\Gamma, \varphi, \psi, \Gamma_1 \vdash \chi}{\Gamma, \psi, \varphi, \Gamma_1 \vdash \chi}$ (перестановка посылок).

12. $\dfrac{\Gamma \vdash \varphi}{\Gamma, \psi \vdash \varphi}$ (утончение, или правило лишней посылки).

13. $\dfrac{\Gamma \vdash \varphi}{\Gamma \vdash \forall x\, \varphi}$, где переменная x не входит свободно в формулы из Γ (введение $\forall$ справа).

14. $\dfrac{\Gamma, (\varphi)^x_t \vdash \psi}{\Gamma, \forall x\, \varphi \vdash \psi}$ (введение $\forall$ слева).

15. $\dfrac{\Gamma \vdash (\varphi)^x_t}{\Gamma \vdash \exists x\, \varphi}$ (введение $\exists$ справа).

16. $\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma, \exists x\, \varphi \vdash \psi}$, где переменная x не входит свободно в ψ и в формулы из Γ (введение $\exists$ слева).

Понятия линейного доказательства в ИПС$^\Sigma$, доказательства в виде дерева в ИПС$^\Sigma$, доказуемой в ИПС$^\Sigma$ секвенции (теоремы ИПС$^\Sigma$) и доказуемой в ИПС$^\Sigma$ формулы определяются аналогично соответствующим понятиям ИС на основе аксиом 1–3 и правил вывода 1–16. Также аналогично предложению 1.2.1 устанавливается

Предложение 2.2.1. *Секвенция S имеет доказательство в ИПС$^\Sigma$ в виде дерева тогда и только тогда, когда S — теорема ИПС$^\Sigma$.*

П р и м е р 2.2.2. Приведем доказательство в виде дерева секвенции

$$\exists x \forall y\, \varphi(x, y) \vdash \forall y \exists x\, \varphi(x, y)$$

для любой формулы $\varphi(x, y)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\varphi(x,y) \vdash \varphi(x,y)}{\varphi(x,y) \vdash \exists x\, \varphi(x,y)}\,15}{\forall y\, \varphi(x,y) \vdash \exists x\, \varphi(x,y)}\,14}{\forall y\, \varphi(x,y) \vdash \forall y \exists x\, \varphi(x,y)}\,13}{\exists x \forall y\, \varphi(x,y) \vdash \forall y \exists x\, \varphi(x,y)}\,16. \quad \square$$

Следующая теорема является синтаксическим аналогом теоремы 2.1.2 и позволяет преобразовывать доказуемые формулы ИС в доказуемые формулы ИПС$^\Sigma$.

Теорема 2.2.2. *Пусть $\varphi(A_1,\dots,A_n)$ — доказуемая формула ИС, $\varphi_1,\dots,\varphi_n$ — формулы сигнатуры Σ. Тогда в результате подстановки формул $\varphi_1,\dots,\varphi_n$ вместо всех соответствующих вхождений пропозициональных переменных $A_1,\dots,A_n$ образуется доказуемая в ИПС$^\Sigma$ формула $\varphi(\varphi_1,\dots,\varphi_n)$.*

Предложение 2.2.3. *Для любой формулы φ, удовлетворяющей условиям на запись $(\varphi)^{x_1,\dots,x_n}_{t_1,\dots,t_n}$, в ИПС$^\Sigma$ доказуемы следующие секвенции:*

(а) $\forall x_1,\dots,x_n\,\varphi \vdash (\varphi)^{x_1,\dots,x_n}_{t_1,\dots,t_n}$;

(б) $(\varphi)^{x_1,\dots,x_n}_{t_1,\dots,t_n} \vdash \exists x_1,\dots,x_n\,\varphi$.

Определение допустимого правила в ИПС$^\Sigma$ совпадает с соответствующим определением допустимого правила в ИС с заменой ИС на ИПС$^\Sigma$.

В следующем предложении расширяется список допустимых правил, составленный для ИС.

Предложение 2.2.4. *В ИПС$^\Sigma$ допустимы правила* (а)–(о) *из предложения* 1.2.2, *а также правила*

(п) $\dfrac{\varphi_1,\dots,\varphi_k \vdash \psi}{(\varphi_1)^{x_1,\dots,x_n}_{t_1,\dots,t_n},\dots,(\varphi_k)^{x_1,\dots,x_n}_{t_1,\dots,t_n} \vdash (\psi)^{x_1,\dots,x_n}_{t_1,\dots,t_n}}$ *(подстановка термов);*

(р) $\dfrac{\Gamma \vdash \forall x\,\varphi}{\Gamma \vdash (\varphi)^x_t}$ *(удаление $\forall$).*

Доказательство допустимости правил (а)–(о) в ИПС$^\Sigma$ совпадает с доказательством допустимости соответствующих правил в ИС.

Для доказательства допустимости правила (п) применим k раз правило 7, начиная с секвенции $\varphi_1,\dots,\varphi_k \vdash \psi$, и получим доказуемость в ИПС$^\Sigma$ секвенции

$$\vdash \varphi_1 \to (\varphi_2 \to \dots (\varphi_k \to \psi)\dots).$$

Затем, применяя k раз правило 13, получаем доказуемость секвенции

$$\vdash \forall x_1,\dots,x_n\,(\varphi_1 \to (\varphi_2 \to \dots (\varphi_k \to \psi)\dots)).$$

Из доказуемости последней секвенции и секвенции (а) предложения 2.2.3 по правилу сечения выводим секвенцию

$$\vdash (\varphi_1 \to (\varphi_2 \to \dots (\varphi_k \to \psi)\dots))^{x_1,\dots,x_n}_{t_1,\dots,t_n}.$$

Из последней секвенции и аксиомы $(\varphi_1)^{x_1,\dots,x_n}_{t_1,\dots,t_n} \vdash (\varphi_1)^{x_1,\dots,x_n}_{t_1,\dots,t_n}$ по правилам 8 и 12 получаем секвенцию

$$(\varphi_1)^{x_1,\dots,x_n}_{t_1,\dots,t_n} \vdash (\varphi_2(\to \dots (\varphi_k \to \psi)\dots))^{x_1,\dots,x_n}_{t_1,\dots,t_n}.$$

Аналогично применяя еще $k-1$ раз правило 8, выводим требуемую секвенцию

$$(\varphi_1)^{x_1,\dots,x_n}_{t_1,\dots,t_n}, \dots, (\varphi_k)^{x_1,\dots,x_n}_{t_1,\dots,t_n} \vdash (\psi)^{x_1,\dots,x_n}_{t_1,\dots,t_n}.$$

Следующее дерево устанавливает допустимость правила (р):

$$\dfrac{\Gamma \vdash \forall x\,\varphi;\ \dfrac{\dfrac{(\varphi)^x_t \vdash (\varphi)^x_t}{\Gamma, \forall x\,\varphi \vdash (\varphi)^x_t}{}^{11,12,14}}{\Gamma \vdash (\forall x\,\varphi \to (\varphi)^x_t)}{}^{7}}{\Gamma \vdash (\varphi)^x_t}{}^{8}.\ \square$$

Покажем, что для отношения $\approx$ на множестве термов доказуемы обычные свойства равенства (рефлексивности, симметричности и транзитивности), т.е. доказуемо в ИПС$^\Sigma$, что $\approx$ — отношение эквивалентности.

Предложение 2.2.5. *Для любых термов $t, q, r \in T(\Sigma)$ следующие секвенции доказуемы в ИПС$^\Sigma$:*

(а) $\vdash (t \approx t)$;

(б) $(t \approx q) \vdash (q \approx t)$;

(в) $(t \approx q), (q \approx r) \vdash (t \approx r)$.

Доказательство. Доказуемость секвенции $\vdash (t \approx t)$ устанавливается применением допустимого правила (п) из предложения 2.2.4:

$$\dfrac{\vdash (x \approx x)}{\vdash (t \approx t)}.$$

Для доказательства секвенции (б) рассмотрим аксиому $(t \approx q), (z \approx t)^z_t \vdash (z \approx t)^z_q$, которая равна секвенции $(t \approx q), (t \approx t) \vdash (q \approx t)$. Выпишем дерево, которое с участием этой аксиомы представляет доказательство свойства симметричности:

$$\dfrac{\vdash (t \approx t);\ \dfrac{(t \approx q), (t \approx t) \vdash (q \approx t)}{(t \approx q) \vdash ((t \approx t) \to (q \approx t))}{}^{7}}{(t \approx q) \vdash (q \approx t)}{}^{8}.$$

Доказуемость секвенции (в) устанавливает следующее дерево:

$$\dfrac{(t \approx q) \vdash (q \approx t);\ \dfrac{(q \approx t), (z \approx r)^z_q \vdash (z \approx r)^z_t}{(q \approx r) \vdash ((q \approx t) \to (t \approx r))}{}^{11,7}}{(t \approx q), (q \approx r) \vdash (t \approx r)}{}^{8,11,12}.\ \square$$

Пусть $\mathfrak{A} = \langle A; \Sigma \rangle$ — алгебраическая система, $S = S(x_1, \ldots, x_n)$ — секвенция ИПС$^\Sigma$, все свободные переменные формул которой содержатся среди $x_1, \ldots, x_n$. Будем говорить, что секвенция S *истинна* на элементах $a_1, \ldots, a_n \in A$ в алгебраической системе $\mathfrak{A}$ и писать $\mathfrak{A} \models S(a_1, \ldots, a_n)$, если выполняется одно из следующих условий:

1) S равна $\varphi_1, \ldots, \varphi_n \vdash \psi$ и из $\mathfrak{A} \models \varphi_i(a_1, \ldots, a_n)$ для всех $i = 1, \ldots n$ следует $\mathfrak{A} \models \psi(a_1, \ldots, a_n)$;

2) S равна $\varphi_1, \ldots, \varphi_n \vdash$ и хотя бы одна из формул $\varphi_1, \ldots, \varphi_n$ ложна на элементах $a_1, \ldots, a_n \in A$ в алгебраической системе $\mathfrak{A}$.

Если секвенция S не истинна на элементах $a_1, \ldots, a_n \in A$ в алгебраической системе $\mathfrak{A}$, то будем говорить, что S *ложна* на элементах $a_1, \ldots, a_n \in A$ в $\mathfrak{A}$. В частности, секвенция $\vdash$ ложна на любой алгебраической системе.

Секвенция $S = S(x_1, \ldots, x_n)$ в ИПС$^\Sigma$ называется *тождественно истинной*, если S истинна на любых элементах $a_1, \ldots, a_n \in A$ в любой алгебраической системе $\mathfrak{A} = \langle A; \Sigma \rangle$.

В дальнейшем нам предстоит доказать *теорему о полноте для исчисления предикатов*, установленную К. Гёделем и утверждающую, что класс доказуемых в ИПС$^\Sigma$ секвенций совпадает с классом тождественно истинных секвенций ИПС$^\Sigma$. Одна часть этого утверждения — это

Теорема 2.2.6. (теорема о непротиворечивости ИПС$^\Sigma$). *Если секвенция S доказуема в ИПС$^\Sigma$, то S тождественно истинна. В частности, не все формулы ИПС$^\Sigma$ доказуемы в ИПС$^\Sigma$.*

Доказательство проводится индукцией по высоте дерева доказательства секвенции S. Тождественная истинность аксиом ИПС$^\Sigma$ очевидна. Проверку сохранения тождественной истинности при переходе по правилам 1–16 мы проведем на примере правила 14

$$\frac{\Gamma, (\varphi)^x_t \vdash \psi}{\Gamma, \forall x\, \varphi \vdash \psi},$$

оставив рассмотрения остальных правил читателю в качестве упражнения. Итак, пусть секвенция $\Gamma, (\varphi)^x_t \vdash \psi$ тождественно истинна, т.е. в предположении истинности в системе $\mathfrak{A} = \langle A; \Sigma \rangle$ на каких-то элементах $a_1, \ldots, a_n \in A$ всех формул из Γ и формулы $(\varphi)^x_t$ мы имеем $\mathfrak{A} \models \psi(a_1, \ldots, a_n)$. Если в системе $\mathfrak{A}$ на каких-то элементах $a_1, \ldots, a_n \in A$ истинны все формулы из Γ и формула $\forall x\, \varphi$, то, в частности, будет справедливо $\mathfrak{A} \models (\varphi)^x_t(a_1, \ldots, a_n)$. Значит, по условию будет верно $\mathfrak{A} \models \psi(a_1, \ldots, a_n)$. Следовательно, секвенция $\Gamma, \forall x\, \varphi \vdash \psi$ тождественно истинна. $\square$

§ 2.3. Эквивалентность формул в ИПС$^\Sigma$

Формулы φ и ψ сигнатуры Σ называются *эквивалентными в ИПС$^\Sigma$* (и пишут $\varphi \equiv \psi$), если секвенции $\varphi \vdash \psi$ и $\psi \vdash \varphi$ доказуемы в ИПС$^\Sigma$.

В силу леммы 1.3.2 условие $\varphi \equiv \psi$ равносильно доказуемости секвенции $\vdash (\varphi \leftrightarrow \psi)$. Аналогично лемме 1.3.3 доказывается, что отношение $\equiv$ является эквивалентностью на множестве формул сигнатуры Σ. При этом отношение $\varphi \equiv \psi$ не зависит от выбора сигнатуры, содержащей все сигнатурные символы формул φ и ψ. Поэтому в дальнейшем мы будем часто говорить о доказуемости в ИПС без упоминания конкретной сигнатуры Σ.

Формулы φ и ψ сигнатуры Σ называются *пропозиционально эквивалентными* (и пишут $\varphi \equiv_P \psi$), если секвенции $\varphi \vdash \psi$ и $\psi \vdash \varphi$ доказуемы в ИПС с использованием лишь правил 1–12.

Предложение 2.3.1. *Пусть $\varphi \rightleftharpoons \varphi(A_1, \ldots, A_n)$, $\psi \rightleftharpoons \psi(A_1, \ldots, A_n)$ — формулы ИС, построенные из пропозициональных переменных, $\chi_1, \ldots, \chi_n$ — формулы сигнатуры Σ, $\varphi' \rightleftharpoons \varphi(\chi_1, \ldots, \chi_n)$, $\psi' \rightleftharpoons \psi(\chi_1, \ldots, \chi_n)$ — формулы сигнатуры Σ, получающиеся из φ и ψ соответственно подстановкой формул χ_i вместо пропозициональных переменных. Если имеет место $\varphi \equiv \psi$ в ИС, то $\varphi' \equiv_P \psi'$.*

Таким образом, для любых формул φ, ψ и χ сигнатуры Σ справедливы все утверждения лемм 1.3.4 и 1.4.1 с заменой $\equiv$ на $\equiv_P$. Например, выполняется $\varphi \wedge (\psi \vee \chi) \equiv_P (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$.

Замечание 2.3.2. *В ИП$^\Sigma$ справедливо утверждение, аналогичное замечанию 1.5.5. Если Φ — множество всех формул сигнатуры Σ с переменными из множества J, то фактор-алгебра $\mathcal{F}/\equiv$ алгебры $\mathcal{F} = \langle \Phi, \wedge, \vee, \neg, \neg(x \approx x), (x \approx x) \rangle$ является булевой алгеброй.*

Предложение 2.3.3. *В ИПС выполнимы все следующие эквивалентности, в которых предполагается, что переменная x не входит свободно в формулу ψ:*

(а) $\neg\exists x\, \varphi \equiv \forall x\, \neg\varphi$;	(б) $\neg\forall x\, \varphi \equiv \exists x\, \neg\varphi$;
(в) $\exists x\, (\varphi \vee \psi) \equiv \exists x\, \varphi \vee \psi$;	(г) $\forall x\, (\varphi \vee \psi) \equiv \forall x\, \varphi \vee \psi$;
(д) $\exists x\, (\varphi \wedge \psi) \equiv \exists x\, \varphi \wedge \psi$;	(е) $\forall x\, (\varphi \wedge \psi) \equiv \forall x\, \varphi \wedge \psi$;
(ж) $\forall x\, \varphi \equiv \forall y\, (\varphi)^x_y$;	(з) $\exists x\, \varphi \equiv \exists y\, (\varphi)^x_y$.

Д о к а з а т е л ь с т в о. Приведем деревья, обосновывающие эквивалентности (а), (в) и (ж), оставляя проверку остальных читателю. При этом в пункте (а) мы будем использовать допустимое правило $\dfrac{\varphi \vdash \neg\psi}{\psi \vdash \neg\varphi}$, которое легко выводится с помощью закона двойного отрица-

ния, а в пункте (ж) — равенство $((\varphi)^x_y)^y_x = \varphi$.

(а)

$$\dfrac{\dfrac{\dfrac{\varphi \vdash \varphi}{\varphi \vdash \exists x\, \varphi}\,15}{\neg\exists x\, \varphi \vdash \neg\varphi}\,\text{Л}}{\neg\exists x\, \varphi \vdash \forall x\, \neg\varphi}\,13, \qquad \dfrac{\dfrac{\dfrac{\neg\varphi \vdash \neg\varphi}{\forall x\, \neg\varphi \vdash \neg\varphi}\,14}{\varphi \vdash \neg\forall x\, \neg\varphi}}{\exists x\, \varphi \vdash \neg\forall x\, \neg\varphi}\,16}{\forall x\, \neg\varphi \vdash \neg\exists x\, \varphi};$$

(в)

$$\dfrac{\dfrac{\dfrac{\varphi \vdash \varphi}{\varphi \vdash \exists x\, \varphi}\,15}{\varphi \vdash \exists x\, \varphi \vee \psi;}\,4 \quad \dfrac{\psi \vdash \psi}{\psi \vdash \exists x\, \varphi \vee \psi;}\,5 \quad \varphi \vee \psi \vdash \varphi \vee \psi}{\dfrac{\varphi \vee \psi \vdash \exists x\, \varphi \vee \psi}{\exists x\, (\varphi \vee \psi) \vdash \exists x\, \varphi \vee \psi}\,16}\,6,$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\varphi \vdash \varphi}{\varphi \vdash \varphi \vee \psi}\,4}{\varphi \vdash \exists x\, (\varphi \vee \psi)}\,15}{\exists x\, \varphi \vdash \exists x\, (\varphi \vee \psi);}\,16 \quad \dfrac{\dfrac{\dfrac{\psi \vdash \psi}{\psi \vdash \varphi \vee \psi}\,5}{\psi \vdash \exists x\, (\varphi \vee \psi);}\,15 \quad \exists x\, \varphi \vee \psi \vdash \exists x\, \varphi \vee \psi}{\exists x\, \varphi \vee \psi \vdash \exists x\, (\varphi \vee \psi)}\,6;$$

(ж)

$$\dfrac{\dfrac{(\varphi)^x_y \vdash (\varphi)^x_y}{\forall x\, \varphi \vdash (\varphi)^x_y}\,14}{\forall x\, \varphi \vdash \forall y\, (\varphi)^x_y}\,13, \qquad \dfrac{\dfrac{((\varphi)^x_y)^y_x \vdash \varphi}{\forall y\, (\varphi)^x_y \vdash \varphi}\,14}{\forall y\, (\varphi)^x_y \vdash \forall x\, \varphi}\,13. \ \square$$

Теорема 2.3.4. (теорема о замене). *Пусть φ — формула сигнатуры Σ, ψ — ее подформула, а формула φ' получается из φ заменой некоторого вхождения ψ на формулу ψ' сигнатуры Σ. Тогда если $\psi \equiv \psi'$, то $\varphi \equiv \varphi'$.*

Доказательство. Если $\varphi = \psi$, утверждение тривиально. Если $\varphi \neq \psi$, воспользуемся индукцией по числу шагов построения формулы φ. Предполагая, что φ — атомарная формула, снова получаем $\varphi = \psi$. Индукционный переход осуществляется рассмотрением шести случаев: $\varphi = \neg\varphi_1$, $\varphi = \varphi_1 \wedge \varphi_2$, $\varphi = \varphi_1 \vee \varphi_2$, $\varphi = \varphi_1 \to \varphi_2$, $\varphi = \forall x\, \varphi_1$, $\varphi = \exists x\, \varphi_1$. В каждом из этих случаев формула ψ входит в φ_1 или φ_2. Поэтому в первых четырех случаях эквивалентность

$\varphi \equiv \varphi'$ вытекает из индукционного предположения и аналога леммы 1.3.3. Тогда в силу индукционного предположения остается рассмотреть случаи, когда формула φ имеет вид $\forall x\,\psi$ или $\exists x\,\psi$, а секвенции $\psi \vdash \psi'$ и $\psi' \vdash \psi$ доказуемы. По симметричности ψ и ψ' достаточно вывести секвенции $\forall x\,\psi \vdash \forall x\,\psi'$ и $\exists x\,\psi \vdash \exists x\,\psi'$. Доказательство этих секвенций устанавливается следующими деревьями:

$$\dfrac{\dfrac{\psi \vdash \psi'}{\forall x\,\psi \vdash \psi'}\,14}{\forall x\,\psi \vdash \forall x\,\psi'}\,13, \qquad \dfrac{\dfrac{\psi \vdash \psi'}{\psi \vdash \exists x\,\psi'}\,15}{\exists x\,\psi \vdash \exists x\,\psi'}\,16. \ \square$$

§ 2.4. Нормальные формы

В этом параграфе мы определим аналоги ДНФ и КНФ, имеющие место в исчислении предикатов, и укажем алгоритмы приведения произвольных формул ИПС к соответствующим формам.

Будем говорить, что бесквантороная формула φ сигнатуры Σ находится в *дизъюнктивной (конъюнктивной) нормальной форме* (сокращенно ДНФ и КНФ соответственно), если φ получается из формулы ИС, находящейся в ДНФ (КНФ), подстановкой вместо пропозициональных переменных некоторых атомарных формул сигнатуры Σ.

Будем говорить, что формула φ сигнатуры Σ находится в *пренексной (предклазуальной) нормальной форме* (сокращенно ПНФ и ПКНФ соответственно), если φ имеет вид

$$Q_1 x_1 \ldots Q_n x_n\,\psi,$$

где $Q_1, \ldots, Q_n$ — кванторы, а ψ находится в ДНФ (КНФ). При этом формула ψ называется *дизъюнктивным (конъюнктивным) ядром*, а слово $Q_1 x_1 \ldots Q_n x_n$ — *кванторной приставкой* формулы φ.

Теорема 2.4.1. *Для любой формулы φ сигнатуры Σ существует формула ψ сигнатуры Σ, находящаяся в ПНФ (ПКНФ) и эквивалентная формуле φ.*

Доказательство. Для приведения формулы φ к ПНФ (ПКНФ) на основании теоремы о замене используются эквивалентности, описанные в предложении 2.3.3, а также эквивалентность $(\varphi \to \psi) \equiv (\neg\varphi \vee \psi)$. Сначала формула φ преобразуется в эквивалентную ей формулу φ', не содержащую символа импликации. Затем формула φ' последовательным вынесением кванторов (при этом, если необходимо, переименовываются связанные переменные) приводится к виду $Q_1 x_1 \ldots Q_n x_n\,\varphi''$, где $Q_i \in \{\exists, \forall\}$, $1 \leqslant i \leqslant n$, φ'' — бесквантороная фор-

мула. Наконец, формула φ'' приводится к ДНФ (КНФ), как показано в § 6.4* и в § 1.4. □

П р и м е р 2.4.1. Считая формулы φ и ψ атомарными, привести к пренексной и предклазуальной нормальным формам формулу $\chi \rightleftharpoons \exists x\,\forall y\,\varphi(x,y) \to \exists x\,\forall y\,\psi(x,y)$.

Избавясь от импликации, получаем

$$\chi \equiv \neg\exists x\,\forall y\,\varphi(x,y) \vee \exists x\,\forall y\,\psi(x,y).$$

Используя пп. (а, б) предложения 2.3.3 и теорему о замене, получаем

$$\chi \equiv \forall x\,\exists y\,\neg\varphi(x,y) \vee \exists x\,\forall y\,\psi(x,y).$$

Так как в формуле $\exists x\,\forall y\,\psi(x,y)$ переменные x, y являются связанными, то по пп. (в, г) предложения 2.3.3. имеем

$$\chi \equiv \forall x\,\exists y\,(\neg\varphi(x,y) \vee \exists x\,\forall y\,\psi(x,y)).$$

Пусть u, v — некоторые новые переменные. Тогда по пунктам (ж, з) предложения 2.3.3 получаем

$$\chi \equiv \forall x\,\exists y\,(\neg\varphi(x,y) \vee \exists u\,\forall v\,\psi(u,v)),$$

откуда

$$\chi \equiv \forall x\,\exists y\,\exists u\,\forall v\,(\neg\varphi(x,y) \vee \psi(u,v)).$$

Формула $\neg\varphi(x,y) \vee \psi(u,v)$ находится в ДНФ и в КНФ одновременно, а значит, формула

$$\forall x\,\exists y\,\exists u\,\forall v\,(\neg\varphi(x,y) \vee \psi(u,v))$$

находится и в ПНФ, и в ПКНФ. □

§ 2.5. Теорема о существовании модели

В этом параграфе мы сформулируем основные теоремы исчисления предикатов: теорему Мальцева — Гёделя о существовании модели для любого непротиворечивого множества формул, теорему Гёделя о полноте и теорему Мальцева о компактности.

Множество Γ формул сигнатуры Σ называется *противоречивым* или *несовместным*, если для некоторых формул $\varphi_1, \ldots, \varphi_n \in \Gamma$ в ИПС^Σ доказуема секвенция $\varphi_1, \ldots, \varphi_n \vdash$. Множество формул сигнатуры Σ, не являющееся противоречивым, называется *непротиворечивым* или *совместным*.

Формула ψ сигнатуры Σ называется *логическим следствием* множества Γ формул сигнатуры Σ, если в ИПС$^\Sigma$ доказуема секвенция $\varphi_1, \ldots, \varphi_n \vdash \psi$ для некоторых формул $\varphi_1, \ldots, \varphi_n$ из Γ.

Из допустимого правила (з) (правила выведения из противоречия) вытекает, что любая формула ψ сигнатуры Σ (и, в частности, тождественно ложная формула $\neg(x \approx x)$) является логическим следствием противоречивого множества формул сигнатуры Σ. С другой стороны, любая модель множества формул Γ является моделью для любого его логического следствия. Таким образом, противоречивое множество формул Γ не может иметь модели. Верно и обратное утверждение:

Теорема 2.5.1. (теорема о существовании модели). *Если* Γ *— совместное множество формул сигнатуры* Σ*, то* Γ *имеет модель мощности, не превосходящей* $\max\{\omega, |\Gamma|\}$.

Следствие 2.5.2. (теорема Гёделя о полноте). *Любая тождественно истинная секвенция ИПС доказуема в ИПС.*

Доказательство. Аналогично доказательству теоремы о полноте для ИС (теорема 1.5.5) утверждение для секвенций сводится к проверке доказуемости тождественно истинных формул. Итак, пусть $\varphi \rightleftharpoons \varphi(x_1, \ldots, x_n)$ — произвольная тождественно истинная формула, все свободные переменные которой содержатся среди переменных $x_1, \ldots, x_n$.

Предположим, что формула φ не доказуема. Установим тогда непротиворечивость множества, состоящего из предложения $\exists x_1, \ldots, x_n \neg\varphi$. Действительно, предполагая его несовместность, получаем доказуемость секвенции $\exists x_1, \ldots, x_n \neg\varphi \vdash$, из которой по правилу контрапозиции выводим $\vdash \neg\exists x_1, \ldots, x_n \neg\varphi$. Тогда в силу предложения 2.3.3 доказуема секвенция $\vdash \forall x_1, \ldots, x_n \varphi$, откуда по предложению 2.2.4 (пункт (р)) выводится секвенция $\vdash \varphi$, что противоречит предположению о недоказуемости формулы φ.

Из совместности множества $\{\exists x_1, \ldots, x_n \neg\varphi\}$ в силу теоремы о существовании модели найдется алгебраическая система $\mathfrak{A}$, для которой $\mathfrak{A} \models \exists x_1, \ldots, x_n \neg\varphi$. Последнее соотношение означает, что найдутся элементы $a_1, \ldots, a_n \in A$, опровергающие предположение о тождественной истинности формулы φ. $\square$

Таким образом, по теореме Гёделя проверка доказуемости формулы φ сводится к проверке ее тождественной истинности. Однако в отличие от ИВ в общем случае не существует алгоритма распознавания доказуемости формул ИПС$^\Sigma$, т. е. ИПС$^\Sigma$ неразрешимо. Тем не менее если в формуле φ “записать”, что каждая переменная может принимать конечное число значений, то перебором всех возможных систем

можно установить, является ли формула тождественно истинной или нет. В § 2.8 будет описан метод резолюций в исчислении предикатов, который, как и метод резолюций в ИВ, позволяет определять невыполнимость формул.

Множество формул Γ называется *локально выполнимым*, если любое конечное подмножество Γ_0 множества Γ выполнимо.

Следствие 2.5.3. (теорема Мальцева о компактности). *Каждое локально выполнимое множество* Γ *формул сигнатуры* Σ *выполнимо.*

Доказательство. Допустим, что выполнимо любое конечное подмножество формул множества Γ, а само Γ невыполнимо. Тогда по теореме о существовании модели оно противоречиво, т. е. найдутся такие формулы $\varphi_1, \ldots, \varphi_n \in \Gamma$, что доказуема секвенция $\varphi_1, \ldots, \varphi_n \vdash$. Следовательно, нашлось несовместное, а значит, невыполнимое конечное множество формул из Γ. Полученное противоречие доказывает теорему компактности. □

Следствие 2.5.4. *Если для любого* $n \in \omega$ *множество формул* Γ *сигнатуры* Σ *имеет модель мощности* $\geqslant n$, *то* Γ *имеет бесконечную модель.*

Доказательство. Рассмотрим множество $\Gamma' = \Gamma \cup \{\exists x_1, \ldots, x_n\,(\neg(x_1 \approx x_2) \wedge \neg(x_1 \approx x_3) \wedge \ldots \wedge \neg(x_{n-1} \approx x_n)) \mid n \in \omega\}$. По предположению для любого $n \in \omega$ множество $\Gamma \cup \{\exists x_1, \ldots, x_n\,(\neg(x_1 \approx x_2) \wedge \neg(x_1 \approx x_3) \wedge \ldots \wedge \neg(x_{n-1} \approx x_n))\}$ выполнимо. Значит, множество Γ′ локально выполнимо. По теореме компактности получаем выполнимость множества Γ′. Но Γ′ может иметь только бесконечные модели. Следовательно, найдется бесконечная модель $\mathfrak{M}$ множества Γ′, которая, в частности, является моделью для Γ. □

§ 2.6. Исчисление предикатов гильбертовского типа

Зафиксируем произвольную сигнатуру Σ и определим *исчисление предикатов гильбертовского типа,* относящееся к сигнатуре Σ (ИП^Σ). Затем мы установим эквивалентность исчислений ИП^Σ и ИПС^Σ подобно тому, как была показана эквивалентность исчислений ИС и ИВ.

Формулами ИП^Σ будут формулы сигнатуры Σ. Секвенций в ИП^Σ нет.

Аксиомами ИП^Σ являются следующие формулы для любых формул φ, ψ, χ сигнатуры Σ, переменных x, y, z и термов $t \in T(\Sigma)$, удовлетворяющих условиям на записи $(\varphi)^x_t$, $(\varphi)^z_x$, $(\varphi)^z_y$:

1) $\varphi \to (\psi \to \varphi)$;
2) $(\varphi \to \psi) \to ((\varphi \to (\psi \to \chi)) \to (\varphi \to \chi))$;
3) $(\varphi \wedge \psi) \to \varphi$;
4) $(\varphi \wedge \psi) \to \psi$;
5) $(\varphi \to \psi) \to ((\varphi \to \chi) \to (\varphi \to (\psi \wedge \chi)))$;
6) $\varphi \to (\varphi \vee \psi)$;
7) $\varphi \to (\psi \vee \varphi)$;
8) $(\varphi \to \chi) \to ((\psi \to \chi) \to ((\varphi \vee \psi) \to \chi))$;
9) $(\varphi \to \psi) \to ((\varphi \to \neg\psi) \to \neg\varphi)$;
10) $\neg\neg\varphi \to \varphi$;
11) $\forall x\, \varphi \to (\varphi)^x_t$;
12) $(\varphi)^x_t \to \exists x\, \varphi$;
13) $(x \approx x)$;
14) $(x \approx y) \to \big((\varphi)^z_x \to (\varphi)^z_y\big)$.

Формулы 1–14 называются *схемами аксиом* ИП^Σ.

Правила вывода $\textit{ИП}^\Sigma$:

$$1)\ \frac{\varphi,\ \varphi \to \psi}{\psi};\quad 2)\frac{\psi \to \varphi}{\psi \to \forall x\, \varphi};\quad 3)\frac{\varphi \to \psi}{\exists x\, \varphi \to \psi},$$

где в правилах 2 и 3 переменная x не входит свободно в ψ.

Понятия доказательства формулы φ ($\vdash \varphi$) и вывода формулы φ из множества гипотез Γ ($\Gamma \vdash \varphi$) в ИП^Σ определяются аналогично соответствующим понятиям в ИВ.

Предложение 2.6.1. *Для любых формул φ и ψ сигнатуры Σ в $\textit{ИП}^\Sigma$ справедливы следующие соотношения:*

(а) $\varphi \vdash \forall x\, \varphi$;
(б) $((\varphi)^x_t \to \psi) \vdash (\forall x\, \varphi \to \psi)$;
(в) $(\psi \to (\varphi)^x_t) \vdash (\psi \to \exists x\, \varphi)$;
(г) $(\varphi)^x_t \vdash \exists x\, \varphi$.

Доказательство. Для доказательства соотношения (а) рассмотрим некоторое доказуемое предложение ψ, например, $\chi \to \chi$ для произвольного предложения χ сигнатуры Σ (доказательство формулы $\chi \to \chi$ повторяет с заменой φ на χ доказательство из примера 1.6.1). Применяя правило 1 к гипотезе φ и аксиоме $\varphi \to (\psi \to \varphi)$, выводим формулу $\psi \to \varphi$. Поскольку переменная x не входит свободно в ψ, по правилу 2 из $\psi \to \varphi$ выводим $\psi \to \forall x\, \varphi$. Поскольку предложение ψ доказуемо, по правилу 1 выводим формулу $\forall x\, \varphi$.

Доказательство соотношения (б) состоит в выводе формулы $(\forall x\, \varphi \to \psi)$ из аксиомы $\forall x\, \varphi \to (\varphi)^x_t$ и гипотезы $((\varphi)^x_t \to \psi)$ по правилу 1 с помощью аксиом 1 и 2.

Доказательство соотношения (в) заключается в выводе формулы $\psi \to \exists x\, \varphi$ из гипотезы $\psi \to (\varphi)^x_t$ и аксиомы $(\varphi)^x_t \to \exists x\, \varphi$ также по правилу 1 с помощью аксиом 1 и 2.

Соотношение (г) получается применением правила 1 к гипотезе $(\varphi)^x_t$ и аксиоме $(\varphi)^x_t \to \exists x\, \varphi$. $\square$

Теорема 2.6.2. (теорема о дедукции). *Если* $\Gamma \cup \{\varphi, \psi\}$ — *множество формул сигнатуры* Σ, *то в ИП*$^\Sigma$ *из* $\Gamma, \varphi \vdash \psi$ *следует* $\Gamma \vdash \varphi \to \psi$.

Доказательство. Рассмотрим минимальный вывод $\psi_1, \ldots,$ ψ_k формулы $\psi = \psi_k$ из $\Gamma \cup \{\varphi\}$. Если $k = 1$ или ψ_k получается по правилу 1, то повторяем доказательство теоремы 1.6.1. В силу минимальности вывода формулы ψ остается рассмотреть случай, когда ψ получается из ψ_{k-1} по правилам 2 или 3. При этом по индукционному предположению установлено соотношение $\Gamma \vdash \varphi \to \psi_{k-1}$.

Пусть формула $\psi = (\chi_1 \to \forall x\, \chi_2)$ получается из формулы $\psi_{k-1} = (\chi_1 \to \chi_2)$ по правилу 2, где по определению вывода переменная x не входит свободно в формулы из $\Gamma \cup \{\varphi, \chi_1\}$. Так как $\varphi \to (\theta_1 \to \theta_2) \vdash (\varphi \wedge \theta_1) \to \theta_2$ и $(\varphi \wedge \theta_1) \to \theta_2 \vdash \varphi \to (\theta_1 \to \theta_2)$ для любых формул θ_1, θ_2, следующая последовательность формул дополняется до вывода формулы $\varphi \to \psi$ из Γ:

$\varphi \to \psi_{k-1}$, $(\varphi \wedge \chi_1) \to \chi_2$, $(\varphi \wedge \chi_1) \to \forall x\, \chi_2$, $\varphi \to (\chi_1 \to \forall x\, \chi_2)$.

Пусть теперь формула ψ получается по правилу 3, т.е. $\psi_{k-1} = (\chi_1 \to \chi_2)$ и $\psi = (\exists x\, \chi_1 \to \chi_2)$, где переменная x не входит свободно в формулы из $\Gamma \cup \{\varphi, \chi_2\}$. Так как $\varphi \to (\chi_1 \to \chi_2) \vdash \chi_1 \to (\varphi \to \chi_2)$ и $\exists x\, \chi_1 \to (\varphi \to \chi_2) \vdash \varphi \to (\exists x\, \chi_1 \to \chi_2)$, следующая последовательность формул дополняется до вывода формулы $\varphi \to \psi$ из Γ:

$\varphi \to \psi_{k-1}$, $\chi_1 \to (\varphi \to \chi_2)$, $\exists x\, \chi_1 \to (\varphi \to \chi_2)$, $\varphi \to (\exists x\, \chi_1 \to \chi_2)$. $\square$

Таким образом, как и в исчислении высказываний, в силу теоремы о дедукции проверка выводимости $\varphi_1, \ldots, \varphi_n \vdash \varphi$ в ИП$^\Sigma$ равносильна проверке доказуемости в ИП$^\Sigma$ формулы $(\varphi_1 \to (\varphi \to \ldots \to (\varphi_n \to \varphi) \ldots))$.

Пример 2.6.1. Из соотношения $\varphi \vdash \forall x\, \varphi$, установленного в предложении 2.6.1., по теореме о дедукции в ИП$^\Sigma$ доказуема формула $\varphi \to \forall x\, \varphi$ для любой формулы φ сигнатуры Σ.

Теорема 2.6.3. (теорема об эквивалентности ИПС$^\Sigma$ и ИП$^\Sigma$). 1. *Секвенция* $\varphi_1, \ldots, \varphi_n \vdash \psi$ *доказуема в ИПС*$^\Sigma$ *тогда и только тогда, когда формула* ψ *выводима в ИП*$^\Sigma$ *из формул* $\varphi_1, \ldots, \varphi_n$.

2. *Секвенция* $\varphi_1, \ldots, \varphi_n \vdash$ *доказуема в ИПС*$^\Sigma$ *тогда и только тогда, когда формула* $\neg(x \approx x)$ *выводима в ИП*$^\Sigma$ *из формул* $\varphi_1, \ldots, \varphi_n$.

Из теоремы 2.6.3 вытекает непротиворечивость ИП$^\Sigma$. Непосредственно проверяется независимость схем аксиом ИП$^\Sigma$.

§ 2.7. Скулемизация алгебраических систем

В этом параграфе мы определим конструкцию, предложенную Скулемом и позволяющую расширять сигнатуру данной алгебраической системы так, чтобы появилась возможность "убирать" кванторы у формул. Необходимость такого преобразования объясняется тем, что работать с формулами, содержащими кванторы, значительно трудней, чем с бескванторными. В следующем параграфе будет изложен метод резолюций в исчислении предикатов, использующий *скулемизацию*.

Алгебраическая система $\mathfrak{A} = \langle A; \Sigma \rangle$ называется *обогащением алгебраической системы* $\mathfrak{A}' = \langle A'; \Sigma' \rangle$, если $A = A'$, $\Sigma \supseteq \Sigma'$ и совпадают интерпретации всех сигнатурных символов из Σ' в системах $\mathfrak{A}$ и $\mathfrak{A}'$.

Если система $\mathfrak{A}$ сигнатуры Σ является обогащением системы $\mathfrak{A}'$ сигнатуры Σ', то $\mathfrak{A}'$ называется *обеднением алгебраической системы* $\mathfrak{A}$ и обозначается через $\mathfrak{A} \restriction \Sigma'$.

Пример 2.7.1. Система $\mathfrak{A} = \langle \mathbb{Z}; +, \cdot, 0, 1 \rangle$ является обогащением системы $\mathfrak{B} = \langle \mathbb{Z}; +, 0, 1 \rangle$, а система $\mathfrak{C} = \langle \mathbb{Z}; +, 0 \rangle$ — обеднением системы $\mathfrak{B}$. $\square$

Пусть Σ — некоторая сигнатура, Σ^S — сигнатура, полученная из Σ добавлением:

а) новых константных символов c_φ для каждой формулы φ сигнатуры Σ, имеющей вид $\exists x_0 \, \psi(x_0)$;

б) новых n-местных функциональных символов f_φ для каждой формулы $\varphi = \exists x_0 \, \psi(x_0, x_1, \ldots, x_n)$ сигнатуры Σ, имеющей $n > 0$ свободных переменных.

Тогда сигнатура Σ^S называется *скулемизацией сигнатуры* Σ.

Через $S(\Sigma)$ обозначим множество следующих предложений сигнатуры Σ^S, называемых *аксиомами Скулема*:

а) $\exists x_0 \, \psi(x_0) \to \psi(c_\varphi)$ для каждой формулы $\varphi = \exists x_0 \, \psi(x_0)$ сигнатуры Σ;

б) $\forall x_1 \ldots x_n (\exists x_0 \, \psi(x_0, x_1, \ldots, x_n) \to \psi(f_\varphi(x_1, \ldots, x_n), x_1, \ldots, x_n))$ для каждой формулы $\varphi = \exists x_0 \, \psi(x_0, x_1, \ldots, x_n)$ $(n > 0)$ сигнатуры Σ.

Согласно аксиомам Скулема из существования элемента, который можно подставить вместо переменной x_0 в формулу ψ, следует возможность подстановки значения некоторой функции f_φ (константы c_φ), зависящего от оставшихся свободных переменных.

Если $\mathfrak{A}$ — алгебраическая система сигнатуры Σ, то любое ее обогащение $\mathfrak{A}^S = \langle A; \Sigma^S \rangle$, являющееся моделью множества $S(\Sigma)$, называется *скулемизацией системы* $\mathfrak{A}$. Возникающие при обогащении константы и операции, соответствующие символам c_φ и f_φ, называются *скулемовскими константами* и *скулемовскими функциями*.

Отметим, что в отличие от Σ^S и $S(\Sigma)$ скулемизация $\mathfrak{A}^S$ не определяется однозначно, поскольку из существования элементов, которые можно подставлять вместо переменных x_0, вообще говоря, не следует их единственность.

П р и м е р 2.7.2. Рассмотрим алгебраическую систему $\mathfrak{A} = \langle\{0,1\}; P^{(1)}, R^{(2)}\rangle$ с интерпретациями $P_{\mathfrak{A}} = \{0,1\}$, $R_{\mathfrak{A}} = \{(0,0), (0,1), (1,1)\}$. В скулемизации $\mathfrak{A}^S$ для формулы $\varphi \rightleftharpoons \exists x\, P(x)$ константный символ c_φ может быть проинтерпретирован как 0 или как 1, поскольку $\mathfrak{A} \models P(0)$ и $\mathfrak{A} \models P(1)$. Для функционального символа $f_{\varphi'}$, где $\varphi' \rightleftharpoons \exists x_0 R(x_0, x_1)$, возможны интерпретации $f_{\varphi'} = \{(0,0), (1,0)\}$ или $f_{\varphi'} = \{(0,0), (1,1)\}$. $\square$

Предложение 2.7.1. *Любая алгебраическая система* $\mathfrak{A}$ *имеет некоторую скулемизацию* $\mathfrak{A}^S$.

В следующем параграфе при работе с методом резолюций нам предстоит с помощью скулемизации снимать кванторы с формул, находящихся в предклазуальной нормальной форме. Говорят, что формула φ находится в *клазуальной нормальной форме*, если она получается из формулы ψ, находящейся в предклазуальной нормальной форме, удалением всех кванторов существования с одновременной заменой соответствующих переменных на термы, определяемые аксиомами Скулема, и последующим удалением всех кванторов всеобщности.

П р и м е р 2.7.3. 1. В примере 2.4.1 найдена формула

$$\forall x\, \exists y\, \exists u\, \forall v\, (\neg\varphi(x,y) \vee \psi(u,v)),$$

находящаяся в ПКНФ. С помощью скулемовских функций $f_1(x)$ и $f_2(x)$, заменяющих переменные y и u соответственно, эта формула преобразуется к следующей формуле, находящейся в КлНФ:

$$(\neg\varphi(x, f_1(x)) \vee \psi(f_2(x), v)).$$

2. Формула

$$\exists x\, \forall y\, \forall z\, \forall u\, \exists v\, ((P(x,y,z,u) \vee \neg Q(z,u,v)) \wedge R(x,z,v))$$

сигнатуры $\{P^{(4)}, Q^{(3)}, R^{(3)}\}$ находится в ПКНФ и приводится к следующей КлНФ с помощью скулемовской константы c, заменяющей переменную x, и скулемовской функции $f(y,z,u)$, заменяющей переменную v:

$$(P(c,y,z,u) \vee \neg Q(z,u,f(y,z,u))) \wedge R(c,z,f(x,z,u)).\ \square$$

§ 2.8. Метод резолюций в исчислении предикатов

Зафиксируем некоторую сигнатуру Σ.

Подстановкой сигнатуры Σ называется конечное множество вида $\{t_1/x_1, \ldots, t_n/x_n\}$, где t_i — терм сигнатуры Σ, отличный от переменных x_i $(1 \leqslant i \leqslant n)$, и все переменные $x_1, \ldots, x_n$ различны. Подстановка, которая не содержит элементов, называется *пустой* и обозначается через ε.

Мы будем использовать греческие буквы для записи подстановок.

П р и м е р 2.8.1. Множества $\{F_1(z)/x, y/z\}$, $\{c_1/x,\ F_2(y)/y, F_1(F_2(c_2))/z\}$ — подстановки сигнатуры $\Sigma = \{F_1^{(1)}, F_2^{(1)}, c_1^{(0)}, c_2^{(0)}\}$. □

Пусть $\theta = \{t_1/x_1, \ldots, t_n/x_n\}$ — подстановка сигнатуры Σ, W — множество формул (термов) сигнатуры Σ. Тогда $W\theta$ — множество формул (термов) сигнатуры Σ, полученных из формул (термов) множества W заменой в них одновременно всех вхождений x_i $(1 \leqslant i \leqslant n)$ на термы t_i $(1 \leqslant i \leqslant n)$. При этом предполагается, что выполняются все условия на записи формул $(\varphi)^{x_1,\ldots,x_n}_{t_1,\ldots,t_n}$, $\varphi \in W$.

Если $W = \{\Phi\}$ или $W = \{t\}$, где Φ — формула, t — терм сигнатуры Σ, то вместо $\{\Phi\}\theta$ и $\{t\}\theta$ будем писать $\Phi\theta$ и $t\theta$ соответственно.

П р и м е р 2.8.2. Пусть $\theta = \{c_1/x, F(c_2)/y, y/z\}$ — подстановка сигнатуры $\Sigma = \{c_1^{(0)}, c_2^{(0)}, F^{(1)}, F_1^{(3)}\}$, $t = F_1(x, y, z)$, $\Phi = (F(x) \approx F_1(x, c_1, z))$. Тогда $t\theta = F_1(c_1, F(c_2), y)$, $\Phi\theta = (F(c_1) \approx F_1(c_1, c_1, y))$.

Пусть $\theta = \{t_1/x_1, \ldots, t_n/x_n\}$ и $\lambda = \{q_1/y_1, \ldots, q_m/y_m\}$ — подстановки сигнатуры Σ. Тогда *композиция подстановок* θ и λ $(\theta \circ \lambda)$ есть подстановка, которая получается из множества $\{t_1\lambda/x_1, \ldots, t_n\lambda/x_n, q_1/y_1, \ldots, q_m/y_m\}$ вычеркиванием всех элементов $t_j\lambda/x_j$, для которых $t_j\lambda = x_j$, и всех элементов q_i/y_i, таких, что $y_i \in \{x_1, \ldots, x_n\}$.

П р и м е р 2.8.3. Пусть $\theta = \{t_1/x_1, t_2/x_2\} = \{F(y)/x,\ z/y\}$, $\lambda = \{q_1/y_1, q_2/y_2, q_3/y_3\} = \{c_1/x, c_2/y, y/z\}$ — подстановки сигнатуры $\Sigma = \{c_1^{(0)}, c_2^{(0)}, F^{(1)}\}$. Тогда $\{t_1\lambda/x_1, t_2\lambda/x_2, q_1/y_1,\ q_2/y_2,\ q_3/y_3\} = \{F(c_2)/x, y/y, c_1/x, c_2/y, y/z\}$. Так как $t_2\lambda = y$, то y/y должно быть вычеркнуто. Так как $x, y \in \{x, y\}$, то c_1/x, c_2/y также должны быть вычеркнуты. Таким образом, $\theta \circ \lambda = \{F(c_2)/x,\ y/z\}$. □

У п р а ж н е н и е. Доказать: 1) ассоциативность композиции подстановок, т. е. $(\theta \circ \lambda) \circ \mu = \theta \circ (\lambda \circ \mu)$ для любых подстановок θ, λ, μ; 2) $\theta \circ \varepsilon = \varepsilon \circ \theta$ для любой подстановки θ. □

Подстановка θ сигнатуры Σ называется *унификатором* для множества $\{\Phi_1, \ldots, \Phi_k\}$ формул сигнатуры Σ, если $\Phi_1\theta = \ldots = \Phi_k\theta$. Множество формул $\{\Phi_1, \ldots, \Phi_k\}$ сигнатуры Σ называется *унифицируемым*, если для него существует унификатор сигнатуры Σ.

П р и м е р 2.8.4. Множество $\{P(c_1, y), P(x, F(c_2))\}$ формул сигнатуры $\Sigma = \{c_1^{(0)}, c_2^{(0)}, P^{(2)}, F^{(1)}\}$ унифицируемо, так как подстановка $\theta = \{c_1/x, F(c_2)/y\}$ является его унификатором. □

Унификатор σ для множества $\{\Phi_1, \ldots, \Phi_k\}$ формул сигнатуры Σ называется *наиболее общим унификатором* (НОУ), если для каждого унификатора θ сигнатуры Σ этого множества существует подстановка λ сигнатуры Σ такая, что $\theta = \sigma \circ \lambda$.

Пусть $W = \{\Phi_1, \ldots, \Phi_k\}$ — непустое множество атомарных формул сигнатуры Σ. *Множеством рассогласований* в W называется множество термов $\{t_1, \ldots, t_k\}$, где t_i входит в Φ_i и начинается с символа (который есть либо сигнатурный символ, либо переменная), стоящего на первой слева позиции в Φ_i, на которой не для всех формул $\Phi_1, \ldots, \Phi_k$ находится один и тот же символ.

П р и м е р 2.8.5. Пусть $W = \{P(x, F(y, z)), P(x, c), P(x, F(y, F_1(z)))\}$ — множество формул сигнатуры $\Sigma = \{P^{(2)}, F^{(2)}, F_1^{(1)}\}$. Во всех трех формулах первые четыре символа $P(x$, совпадают, а на пятом месте в первой и второй формулах стоят разные символы: F, c. Таким образом, множество рассогласований в W — $\{F(y, z), c, F(y, F_1(z))\}$. □

Алгоритм унификации предназначен для распознавания того, является ли данное конечное непустое множество атомарных формул унифицируемым, и нахождения НОУ для этого множества в случае его унифицируемости.

Пусть W — конечное непустое множество атомарных формул. *Алгоритм унификации* для множества W:

Шаг 1. Полагаем $k = 0$, $W_k = W$, $\sigma_k = \varepsilon$.

Шаг 2. Если W_k — одноэлементное множество, то остановка: σ_k — НОУ для W. В противном случае найдем множество D_k рассогласований для W_k.

Шаг 3. Если существуют $x_k, t_k \in D_k$ такие, что x_k — переменная, не входящая в терм t_k, то перейти к шагу 4. В противном случае остановка: множество W не унифицируемо.

Шаг 4. Полагаем $\sigma_{k+1} \rightleftharpoons \sigma_k \circ \{t_k/x_k\}$ и $W_{k+1} \rightleftharpoons W_k\{t_k/x_k\}$ (заметим, что $W_{k+1} = W\sigma_{k+1}$).

Шаг 5. Присвоить k значение $k + 1$ и перейти к шагу 2.

Теорема 2.8.1. *Если W — конечное непустое унифицируемое множество атомарных формул, то алгоритм унификации будет всегда заканчивать работу на шаге 2 и последняя подстановка σ_k будет НОУ для W.* □

П р и м е р 2.8.6. Найти НОУ для множества $W = \{P(c, x, F_2(F_1(y))), P(z, F_2(z), F_2(u))\}$.

1. Положим $k = 0$, $W_0 = W$, $\sigma_0 = \varepsilon$.

2. Так как W — неодноэлементное множество, то σ_0 не является НОУ для W. Множество рассогласований для W_0 есть $D_0 = \{c, z\}$.

3. В D_0 существует переменная $x_0 = z$, которая не встречается в терме $t_0 = c$. Поэтому переходим к шагу 4.

4. Полагаем $\sigma_1 \rightleftharpoons \sigma_0 \circ \{c/z\} = \varepsilon \circ \{c/z\} = \{c/z\}$, $W_1 \rightleftharpoons W_0\{c/z\} = \{P(c, x, F_2(F_1(y))), P(c, F_2(c), F_2(u))\}$.

5. Присваиваем $k = 1$.

6. Так как W_1 — неодноэлементное множество, множество рассогласований для W_1 есть $D_1 = \{x, F_2(c)\}$.

7. Из D_1 находим, что $x_1 = x$, $t_1 = F_2(c)$.

8. Полагаем $\sigma_2 \rightleftharpoons \sigma_1 \circ \{F_2(c)/x\}$, $W_2 \rightleftharpoons W_1\{F_2(c)/x\} = \{P(c, F_2(c), F_2(F_1(y))), P(c, F_2(c), F_2(u))\}$.

9. Присваиваем $k = 2$.

10. Так как W_2 — неодноэлементное множество, множество рассогласований для W_2 есть $D_2 = \{F_1(y), u\}$.

11. Из D_2 найдем, что $x_2 = u$, $t_2 = F_1(y)$.

12. Полагаем $\sigma_3 \rightleftharpoons \sigma_2 \circ \{F_1(y)/u\} = \{c/z, F_2(c)/x, F_1(y)/u\}$, $W_3 \rightleftharpoons W_2\{F_1(y)/u\} = \{P(c, F_2(c), F_2(F_1(y))), P(c, F_2(c), F_2(F_1(y)))\} = \{P(c, F_2(c), F_2(F_1(y)))\}$.

13. Множество W_3 одноэлементно. Поэтому $\sigma_3 = \{c/z, F_2(c)/x, F_1(y)/u\}$ — НОУ для W. □

П р и м е р 2.8.7. Определить, унифицируемо ли множество $W = \{P(F_1(c), F_2(x)), P(y, y)\}$.

1. Положим $k = 0$, $W_0 = W$, $\sigma_0 = \varepsilon$.

2. Так как W_0 — неодноэлементное множество, множество рассогласований для W_0 есть $D_0 = \{F_1(c), y\}$.

3. Из D_0 находим, что $x_0 = y$, $t_0 = F_1(c)$.

4. Полагаем $\sigma_1 \rightleftharpoons \sigma_0 \circ \{F_1(c)/y\} = \{F_1(c)/y\}$, $W_1 \rightleftharpoons \{P(F_1(c), F_2(x)), P(F_1(c), F_1(c))\}$.

5. Присваиваем $k = 1$.

6. Так как W_1 — неодноэлементное множество, множество рассогласований для W_1 есть $D_1 = \{F_2(x), F_1(c)\}$.

7. В D_1 нет элемента, который был бы переменной. Поэтому множество W не унифицируемо. □

Литерой сигнатуры Σ называется атомарная формула или отрицание атомарной формулы сигнатуры Σ. *Дизъюнктом сигнатуры* Σ называется литера сигнатуры Σ или дизъюнкция литер сигнатуры Σ.

Примеры дизъюнктов сигнатуры $\Sigma = \{P^{(1)}, F_1^{(1)}, F_2^{(2)}, c^{(0)}\}$ — $P(F_1(x))$, $\neg P(F_1(x))$, $P(F_1(x)) \vee \neg(F_1(x) \approx F_2(x,y))$, $P(F_2(F_1(x),z)) \vee \neg P(F_2(x,y)) \vee (x \approx c)$.

Пусть Φ — дизъюнкт сигнатуры Σ вида $\psi_1 \vee \cdots \vee \psi_n \vee \chi$ или $\neg\psi_1 \vee \cdots \vee \neg\psi_n \vee \chi$, где ψ_i — атомарные формулы сигнатуры Σ $(1 \leqslant i \leqslant n)$. Предположим, что множество формул $\{\psi_1, \ldots, \psi_n\}$ имеет НОУ σ. Тогда $\psi_1\sigma \vee \chi\sigma$ или соответственно $\neg\psi_1\sigma \vee \chi\sigma$ называется *склейкой* Φ. Полученную формулу в дальнейшем будем обозначать через $\Phi\sigma$.

П р и м е р 2.8.8. В формуле $\Phi = P(x) \vee P(F(y)) \vee \neg P_2(x)$ подформулы $P(x)$ и $P(F(y))$ имеют НОУ $\sigma = \{F(y)/x\}$. Следовательно, $\Phi\sigma = P(F(y)) \vee \neg P_2(F(y))$ — склейка Φ. □

Пусть Φ_1, Φ_2 — два дизъюнкта, не имеющих общих переменных, L_1, L_2 — литеры в Φ_1 и Φ_2 соответственно. Если литеры L_1 и $L_2' \equiv \neg L_2$ имеют НОУ σ, то дизъюнкт, получаемый из дизъюнкта $\Phi_1\sigma \vee \Phi_2\sigma$ вычеркиванием $L_1\sigma$ и $L_2\sigma$, называется *бинарной резольвентой* Φ_1 и Φ_2, а литеры L_1 и L_2 называются *отрезаемыми литерами*. Если $\Phi_1\sigma = L_1$ и $\Phi_2\sigma = L_2$, то полагаем бинарную резольвенту Φ_1 и Φ_2 равной 0.

Если Φ_1 и Φ_2 имеют общие переменные, то, заменив в формуле Φ_2 эти общие переменные на переменные, не встречающиеся в Φ_1 и Φ_2, получим формулу Φ_2', которая не имеет общих переменных с формулой Φ_1. *Бинарной резольвентой* формул Φ_1 и Φ_2 называется бинарная резольвента формул Φ_1 и Φ_2'.

П р и м е р 2.8.9. Найти бинарную резольвенту формул $\Phi_1 \rightleftharpoons P_1(x) \vee P_2(x)$ и $\Phi_2 \rightleftharpoons \neg P_1(c) \vee P_3(x)$.

Заменив переменную x в Φ_2 на y, получим $\Phi_2' = \neg P_1(c) \vee P_3(y)$. Выбираем $L_1 = P_1(x)$, $L_2 = \neg P_1(c)$. Так как $\neg L_2 \equiv L_2' = P_1(c)$, то L_1 и L_2' имеют НОУ $\sigma = \{c/x\}$. Бинарная резольвента формул Φ_1 и Φ_2' получается из $\Phi_1\sigma \vee \Phi_2'\sigma = P_1(c) \vee P_2(c) \vee \neg P_1(c) \vee P_3(y)$ вычеркиванием $P_1(c)$ и $\neg P_1(c)$. Следовательно, $P_2(c) \vee P_3(y)$ — бинарная резольвента Φ_1 и Φ_2, а $P_1(x)$ и $\neg P_1(c)$ — отрезаемые литеры. □

Резольвентой дизъюнктов Φ_1 и Φ_2 $(\mathrm{res}(\Phi_1, \Phi_2))$ является одна из следующих бинарных резольвент:

— бинарная резольвента Φ_1 и Φ_2;

— бинарная резольвента склейки Φ_1 и Φ_2;

— бинарная резольвента Φ_1 и склейки Φ_2;

— бинарная резольвента склейки Φ_1 и склейки Φ_2.

П р и м е р 2.8.10. Найти $\mathrm{res}(\Phi_1, \Phi_2)$, где $\Phi_1 = P(x) \vee P(F(y)) \vee P_1(F_1(y))$, $\Phi_2 = \neg P(F(F_1(c_1))) \vee P_2(c_2)$.

Склейка Φ_1 есть $\Phi_1' = \Phi_1\{F(y)/x\} = P(F(y)) \vee P_1(F_1(y))$. Бинарная резольвента Φ_1' и Φ_2 есть $P_1(F(F_1(c_1))) \vee P_2(c_2)$. Следовательно, $\mathrm{res}(\Phi_1, \Phi_2) = P_1(F(F_1(c_1))) \vee P_2(c_2)$. □

Пусть S — множество дизъюнктов сигнатуры Σ. *Резолютивный вывод* формулы Φ из S есть такая конечная последовательность Φ_1, ..., Φ_k дизъюнктов, что $\Phi_k = \Phi$ и каждый дизъюнкт Φ_i или принадлежит S, или является резольвентой дизъюнктов, предшествующих Φ_i.

Универсальным замыканием формулы $\Phi(x_1, \ldots, x_n)$ называется предложение $\forall x_1, \ldots, x_n\, \Phi(x_1, \ldots, x_n)$.

Теорема 2.8.2. (теорема о полноте метода резолюций). *Если S — множество дизъюнктов сигнатуры Σ, то множество универсальных замыканий формул из S невыполнимо тогда и только тогда, когда существует резолютивный вывод нуля из S.* $\square$

Пример 2.8.11. Доказать невыполнимость множества формул $W = \{\Phi_1, \ldots, \Phi_6\}$, где

$\Phi_1 = P_1(c_1, F(c_2), F(c_3))$,

$\Phi_2 = P_2(c_1)$,

$\Phi_3 = P_1(x, x, F(x))$,

$\Phi_4 = \neg P_1(x, y, z) \vee P_3(x, z)$,

$\Phi_5 = \neg P_2(x) \vee \neg P_1(y, z, u) \vee \neg P_3(x, u) \vee P_3(x, y) \vee P_3(x, z)$,

$\Phi_6 = \neg P_3(c_1, c_3)$.

Построим резолютивный вывод 0 из W:

$\Phi_7 = \mathrm{res}(\Phi_2, \Phi_5) = \mathrm{res}(\Phi_2, \Phi_5\{z/y\}) = \neg P_1(z, z, u) \vee \neg P_3(c_1,\ u) \vee P_3(c_1, z)$;

$\Phi_8 = \mathrm{res}(\Phi_3, \Phi_7) = \neg P_3(c_1, F(x)) \vee P_3(c_1, x)$;

$\Phi_9 = \mathrm{res}(\Phi_6, \Phi_8) = \neg P_3(c_1, F(c_3))$;

$\Phi_{10} = \mathrm{res}(\Phi_4, \Phi_9) = \neg P_1(c_1, y, F(c_3))$;

$\Phi_{11} = \mathrm{res}(\Phi_1, \Phi_{10}) = 0$. $\square$

Пример 2.8.12. Выполнимо ли множество предложений $\{\Phi_1, \Phi_2\}$? Если множество выполнимо, построить систему, на которой предложения Φ_1 и Φ_2 истинны:

$$\Phi_1 \rightleftharpoons \exists y\, \forall x\, z((P_1(x, z) \to (P_2(x) \wedge P_3(y))) \wedge P_4(y)),$$

$$\Phi_2 \rightleftharpoons \forall x\, ((P_4(x) \to \neg P_3(x)) \wedge \exists y\, P_1(x, y)).$$

Приведем формулы Φ_1, Φ_2 к предклазуальной нормальной форме:

$\Phi_1 \equiv \exists y\, \forall x, z\, ((\neg P_1(x, z) \vee P_2(x)) \wedge (\neg P_1(x, z) \vee P_3(y)) \wedge P_4(y))$,

$\Phi_2 \equiv \forall x\, \exists y\, ((\neg P_4(x) \vee \neg P_3(x)) \wedge P_1(x, y))$.

Введением символов скулемовской константы c и скулемовской функции F получаем, что выполнимость множества формул $\{\Phi_1, \Phi_2\}$ сигнатуры $\Sigma = \{P_1^{(2)}, P_2^{(1)}, P_3^{(1)}, P_4^{(1)}\}$ равносильна выполнимости множества

формул

$$\{\forall x, z\,((\neg P_1(x,z) \vee P_2(x)) \wedge (\neg P_1(x,z) \vee P_3(c)) \wedge P_4(c)),\\ \forall x\,((\neg P_4(x) \vee \neg P_3(x)) \wedge P_1(x, F(x)))\}, \tag{2.1}$$

сигнатуры $\Sigma' = \Sigma \cup \{c, F^{(1)}\}$, что в свою очередь равносильно выполнимости множества формул

$$\{\forall x, z\,(\neg P_1(x,z) \vee P_2(x)), \forall x, z\,(\neg P_1(x,z) \vee P_3(c)),\\ P_4(c), \forall x\,(\neg P_4(x) \vee \neg P_3(x)), \forall x\,P_1(x, F(x))\}. \tag{2.2}$$

Действительно, пусть множество формул $\{\Phi_1, \Phi_2\}$ выполнимо. Тогда существует алгебраическая система $\mathfrak{A} = \langle A, \Sigma\rangle$ и $c' \in A$, для которых $\mathfrak{A} \models \forall x\,z\,(\neg P_1(x,z) \vee P_2(x)) \wedge \forall x\,z\,(\neg P_1(x,z) \vee P_3(c')) \wedge P_4(c')$ и $\mathfrak{A} \models \forall x\,(\neg P_4(x) \vee \neg P_3(x))$.

Кроме того, для любого $a \in A$ найдется элемент в A, который обозначим через $G(a)$, такой, что $\mathfrak{A} \models P_1(a, G(a))$, т. е. $\mathfrak{A} \models \forall x\,P_1(x, G(x))$. Тогда в системе $\mathfrak{A}' = \langle A, \Sigma'\rangle$, где c' является интерпретацией $c \in \Sigma'$, G — интерпретацией $F \in \Sigma'$, истинны формулы из (2.2), а значит, и формулы из (2.1).

Напротив, если все формулы из (2.1) истинны в системе $\mathfrak{A}' = \langle A, \Sigma'\rangle$, то, очевидно, формулы (2.2) и Φ_1, Φ_2 будут истинны и в системе $\mathfrak{A} = \langle A, \Sigma\rangle$.

Приведем формулы из (2.2) к КлНФ и исследуем на выполнимость с помощью метода резолюций получившееся множество дизъюнктов

$$\{\neg P_1(x,z) \vee P_2(x), \neg P_1(x,z) \vee P_3(c), P_4(c), \neg P_4(x) \vee \neg P_3(x),\\ P_1(x, F(x))\}. \tag{2.3}$$

Имеем

$\mathrm{res}(\neg P_1(x,z) \vee P_3(c), P_1(x, F(x))) = P_3(c)$,

$\mathrm{res}(P_3(c), \neg P_4(x) \vee \neg P_3(x)) = \neg P_4(c)$,

$\mathrm{res}(\neg P_4(c), P_4(c)) = 0$.

Построили резолютивный вывод нуля. Следовательно, множество дизъюнктов (2.3) невыполнимо. Тогда и множество предложений (2.1) невыполнимо, что равносильно невыполнимости множества предложений $\{\Phi_1, \Phi_2\}$. $\square$

П р и м е р 2.8.13. Выполнимо ли множество предложений $\{\Phi_1, \Phi_2, \Phi_3\}$? Если выполнимо, построить систему, на которой эти предложения истинны: $\Phi_1 \rightleftharpoons \exists x\,(P_1(x) \wedge \forall y(P_2(y) \to P_3(x,y)))$,

$\Phi_2 \rightleftharpoons \forall x\,(P_1(x) \to \forall y\,(P_4(y) \to \neg P_3(x,y)))$,

$\Phi_3 \rightleftharpoons \forall x\,(P_2(x) \to \neg P_4(x))$.

Приведем формулы Φ_1, Φ_2, Φ_3 к ПКНФ:
$\Phi_1 \equiv \exists x\, \forall y\, (P_1(x) \wedge (\neg P_2(y) \vee P_3(x,y))$,
$\Phi_2 \equiv \forall x\, y\, (\neg P_1(x) \vee \neg P_4(y) \vee \neg P_3(x,y))$,
$\Phi_3 \equiv \forall x\, (\neg P_2(x) \vee \neg P_4(x))$.

Из полученных формул получаем следующие формулы, находящиеся в КлНФ:
$(P_1(c) \wedge (\neg P_2(y) \vee P_3(c,y))$,
$(\neg P_1(x) \vee \neg P_4(y) \vee \neg P_3(x,y))$,
$(\neg P_2(x) \vee \neg P_4(x))$.

Так же, как в примере 2.8.12, строим множество дизъюнктов:

$$\{P_1(c), \neg P_2(y) \vee P_3(c,y), \neg P_1(x) \vee \neg P_4(y) \vee \neg P_3(x,y), \neg P_2(x) \vee \neg P_4(x)\}. \tag{2.4}$$

Исследуем это множество на выполнимость с помощью метода резолюций:

$$\mathrm{res}(P_1(c), \neg P_1(x) \vee \neg P_4(y) \vee \neg P_3(x,y)) = \neg P_4(y) \vee \neg P_3(c,y),$$
$$\mathrm{res}(\neg P_2(y) \vee P_3(c,y), \neg P_4(y) \vee \neg P_3(c,y)) = \neg P_2(y) \vee \neg P_4(y). \tag{2.5}$$

Других резольвент для множества (2.4) нет, поэтому резолютивный вывод 0 из (2.4) не существует. Рассмотрим множество, составленное из констант, входящих в формулы (2.4), т. е. множество $\{c\}$. Определим на $\{c\}$ предикаты P_1, P_2, P_3, P_4 так, чтобы множество формул из (2.4) и (2.5) выполнялось на системе $\langle \{c\}; P_1, P_2, P_3, P_4\rangle$. Из (2.5) следует, что необходимо потребовать $c \notin P_4$, или $\langle c, c\rangle \notin P_3$ и $c \notin P_2$. Положим $c \notin P_4$, $\langle c, c\rangle \notin P_3$, $c \notin P_2$. Из (2.4) следует, что необходимо потребовать $c \in P_1$. Таким образом, на системе $\langle \{c\}; P_1, P_2, P_3, P_4, c\rangle$ выполняются все формулы из (2.4). Более того, на ней истинны все формулы из (2.4) с навешанными на них кванторами всеобщности по переменным x, y, что равносильно истинности формул Φ_1, Φ_2, Φ_3 на системе $\langle \{c\}; P_1, P_2, P_3\rangle$.

П р и м е р 2.8.14. Выполнимо ли множество предложений $\{\Phi_1, \Phi_2\}$? Если выполнимо, построить систему, на которой эти предложения истинны:

$\Phi_1 \rightleftharpoons \exists u\, \forall x\, \exists z\, \forall y\, (P_3(z) \wedge ((P_2(x,z) \wedge \neg P_1(u)) \vee \neg((P_3(y) \to P_1(y)) \to P_1(u))))$,

$\Phi_2 \rightleftharpoons \forall x\, (\exists y\, P_2(x,y) \to \neg P_3(x))$.

Преобразуем формулы Φ_1 и Φ_2 к предклазуальной нормальной форме:
$\Phi_1 \equiv \exists u\, \forall x\, \exists z\, \forall y\, (P_3(z) \wedge (P_2(x,z) \vee \neg P_3(y) \vee P_1(y)) \wedge \neg P_1(u))$,
$\Phi_2 \equiv \forall x, y\, (\neg P_2(x,y) \vee \neg P_3(x))$.

Исследуем на выполнимость множество дизъюнктов

$$\{P_3(F(x)), P_2(x, F(x)) \vee \neg P_3(y) \vee P_1(y), \neg P_1(c),$$
$$\neg P_2(x, y) \vee \neg P_3(x)\}, \tag{2.6}$$

которое получается из преобразованных формул после введения символов скулемовской константы c и скулемовской функции F. Имеем

$\mathrm{res}(\neg P_1(c), P_2(x, F(x)) \vee \neg P_3(y) \vee P_1(y)) = P_2(x, F(x)) \vee \neg P_3(c)$,
$\mathrm{res}(P_2(x, F(x)) \vee \neg P_3(c), \neg P_2(x, y) \vee \neg P_3(x)) = \neg P_3(c)$,
$\mathrm{res}(P_2(x, F(x)) \vee \neg P_3(y) \vee P_1(y), \neg P_2(x, y) \vee \neg P_3(x)) =$
$\neg P_3(y) \vee P_1(y)$,
$\mathrm{res}(\neg P_3(y) \vee P_1(y), P_3(F(x))) = P_1(F(x))$,

$$\mathrm{res}(\neg P_2(x, y) \vee \neg P_3(x), P_3(F(x))) = \neg P_2(F(x), y). \tag{2.7}$$

Таким образом, резолютивного вывода 0 из множества (2.6) не существует. Построим алгебраическую систему $\mathfrak{A} = \langle A; P_1, P_2, P_3, F, c\rangle$, в которой будут истинны формулы (2.6) и (2.7) с навешанными на них кванторами всеобщности по переменным x, y. Ясно, что $c \in A$. Так как $\mathfrak{A} \models \forall x(\neg P_3(c) \wedge P_3(F(x))$, то $F(c) \neq c$. Положим $A \rightleftharpoons \{c, c'\}$ и $F(c) \rightleftharpoons c'$. Так как $\mathfrak{A} \models \forall x P_3(F(x))$, то необходимо, чтобы $F(c') = c'$ и $c' \in P_3$. Из (2.7) следует, что $c \notin P_3$. Поскольку $\mathfrak{A} \models \forall x, y\,(P_1(F(x)) \vee \neg P_2(F(x), y))$, полагаем $c' \in P_1$, $(c', c) \notin P_2$, $(c', c') \notin P_2$. Предикаты P_1 и P_2 доопределяются произвольно. Таким образом, в системе $\langle\{c, c'\}; P_1, P_2, P_3\rangle$ такой, что $c' \in P_1$, $(c', c), (c', c') \notin P_2$, $c \notin P_3$, $c' \in P_3$, истинны формулы Φ_1 и Φ_2. $\square$

Следующий пример показывает, как формализуются предложения и методом резолюций эффективно доказываются теоремы при переходе к соответствующим формализациям.

П р и м е р 2.8.15. Установить, что из *посылки* "Студенты суть граждане" следует *заключение* "Голоса студентов суть голоса граждан".

Пусть формулы $S(x), C(x)$ и $V(x, y)$ означают "x — студент", "x — гражданин" и "x есть голос y" соответственно. Тогда посылка и заключение запишутся следующим образом:

$\forall y\,(S(y) \to C(y))$ (посылка),
$\forall x\,(\exists y(S(y) \wedge V(x, y)) \to \exists z(C(z) \wedge V(x, z)))$ (заключение).

Формула, соответствующая посылке, эквивалентна дизъюнкту $\neg S(y) \vee C(y)$. Поскольку

$$\neg\forall x(\exists y(S(y) \wedge V(x,y)) \to \exists z(C(z) \wedge V(x,z))) \equiv$$
$$\equiv \neg\forall x(\forall y(\neg S(y) \vee \neg V(x,y)) \vee \exists z(C(z) \wedge V(x,z))) \equiv$$
$$\equiv \neg\forall x\forall y\exists z(\neg S(y) \vee \neg V(x,y) \vee (C(z) \wedge V(x,z))) \equiv$$
$$\equiv \exists x\exists y\forall z(S(y) \wedge V(x,y) \wedge (\neg C(z) \vee \neg V(x,z))),$$

имеем три дизъюнкта, определяющие отрицание заключения:

$$S(b), V(a,b), \neg C(z) \vee \neg V(a,z).$$

Доказательство заканчивается следующим образом:
$\mathrm{res}(\neg S(y) \vee C(y), S(b)) = C(b)$,
$\mathrm{res}(C(b), \neg C(z) \vee \neg V(a,z)) = \neg V(a,b)$,
$\mathrm{res}(V(a,b), \neg V(a,b)) = 0$. $\square$

§ 2.9. Логические программы

Приведенный метод резолюций служит основой языков логического программирования, главным отличием которых от "процедурных" языков является то, что программа не указывает, *как* что-либо следует делать для решения задачи, а описывает некоторые элементы и связи между ними (модель определенной сигнатуры) и ставит цель, т.е. задает вопрос об этой системе (на формальном языке это означает проверить истинность предложения на данной системе или найти элемент, удовлетворяющий заданной формуле). При этом компьютер самостоятельно ищет стратегию для решения поставленных вопросов.

Логическая программа представляет собой конечный набор выражений, каждая из которых имеет один из следующих видов:

$$P(t_1, \ldots, t_n)., \tag{2.8}$$

$$Q(s_1, \ldots, s_k) : -Q_1(s_1, \ldots, s_k), \ldots, Q_m(s_1, \ldots, s_k)., \tag{2.9}$$

где $P(t_1, \ldots, t_n)$, $Q(s_1, \ldots, s_k)$, $Q_1(s_1, \ldots, s_k)$, $\ldots$, $Q_m(s_1, \ldots, s_k)$ — атомарные формулы сигнатуры Σ. При этом в конце каждого выражения ставится точка. Выражения первого вида называются *фактами*, а второго — *правилами*.

Каждый факт интерпретируется отношением между объектами, а правило (2.9) читается как "если истинны $Q_1(s_1, \ldots, s_k), \ldots, Q_m(s_1, \ldots, s_k)$, то истинно $Q(s_1, \ldots, s_k)$". Формула $Q(s_1, \ldots, s_k)$ называется *заголовком* правила (2.9). Правила позволяют выводить новые факты из уже имеющихся.

Таким образом, логическая программа состоит из конечного числа фактов и правил:

$$\begin{array}{l}\mathcal{A}.\\ \ldots\\ \mathcal{M}.\\ \mathcal{N} :- \mathcal{N}_1, \ldots, \mathcal{N}_m.\\ \ldots\\ \mathcal{Z} :- \mathcal{Z}_1, \ldots, \mathcal{Z}_m.\end{array}$$

В фактах описывается алгебраическая система, для которой с помощью правил выводятся некоторые ее свойства.

Логическая программа задает множество следствий, которые являются результатом программы. Таким образом, выполнение логической программы — это вывод следствий.

Для *выполнения программы* требуется обратиться к *целевому запросу* (*цели*), который представляет собой последовательность атомарных формул (с точкой в конце!) вида

$$R_1(q_1, \ldots, q_l), \ldots, R_p(q_1, \ldots, q_l).. \tag{2.10}$$

Выполнение программы состоит в попытке решить задачу, т.е. доказать целевое утверждение (2.10), используя факты и правила, заданные в логической программе.

Семантика логической программы представляется в двух видах — логическая семантика и процедурная семантика. Определим сначала *логическую семантику*.

Каждому факту (2.8) поставим в соответствие предложение

$$\varphi \rightleftharpoons \forall x_1, \ldots, x_s \, P(t_1, \ldots, t_n),$$

где $x_1, \ldots, x_s$ — все переменные, входящие в формулу $P(t_1, \ldots, t_n)$.

Каждому правилу (2.9) поставим в соответствие предложение

$$\psi \rightleftharpoons \forall y_1, \ldots, y_r \, (Q_1(s_1, \ldots, s_k) \wedge \ldots \wedge Q_m(s_1, \ldots, s_k) \rightarrow$$

$$Q(s_1, \ldots, s_k)),$$

где $y_1, \ldots, y_s$ — все переменные, входящие в формулы $Q_1(s_1, \ldots, s_k)$, $\ldots$, $Q_m(s_1, \ldots, s_k)$, $Q(s_1, \ldots, s_k)$.

Запрос (2.10) получит в соответствие формулу

$$\chi \rightleftharpoons \exists z_1, \ldots, z_d \, (R_1(q_1, \ldots, q_l) \wedge \ldots \wedge R_p(q_1, \ldots, q_l)),$$

где кванторы существования связывают все переменные.

Пусть $\varphi_1, \ldots, \varphi_a$ — предложения, соответствующие всем фактам, $\psi_1, \ldots, \psi_b$ — всем правилам. Тогда значение пары

$$\langle \textbf{программа}, \textbf{запрос} \rangle$$

в логической семантике есть утверждение о том, что секвенция

$$\varphi_1, \ldots, \varphi_a, \psi_1, \ldots, \psi_b \vdash \chi$$

доказуема. Для того чтобы выяснить, так ли это, применяется метод резолюций.

Операционная семантика определяет действия компьютера при ответе на запрос. При этом предполагается наличие логической машины, называемой *интерпретатором*, которая осуществляет процесс логического вывода. Механизм этого вывода использует алгоритм унификации.

П р и м е р 2.9.1. Рассмотрим действия интерпретатора на примере следующей логической программы:

$$\begin{array}{l} R(a,b). \\ Q(b,g(c)). \\ P(x,f(y)) :- R(x,z), Q(z,f(y)). \\ P(x,f(y)) :- R(x,z), Q(z,g(y)). \\ R(x,z) :- Q(f(x),g(z)). \end{array}$$

Здесь a, b, c — константы, x, y, z — переменные.

Предположим, что целевой запрос имеет вид

$$P(u, f(v)). \tag{2.11}$$

При вычислении ответа на этот запрос, интерпретатор формулирует цель $P(u, f(v))$ и пытается ее достичь, унифицируя цель с фактами. В нашем случае цель $P(u, f(v))$ не унифицируется ни с одним из фактов. Тогда интерпретатор пытается ее унифицировать с заголовком одного из правил.

Это можно сделать с заголовком первого правила с помощью подстановки $\{u/x, v/y\}$. Но предварительно нужно проверить истинность посылок первого правила. Для этого интерпретатор формирует запрос

$$R(x,z), Q(z,f(y)).$$

Цель $R(x,z)$ достигается за счет унификации с первым фактом посредством подстановки $\sigma = \{a/x, b/z\}$. Теперь следующим запросом

является $Q(b, f(y))$. Но эта цель не достижима, поскольку не унифицируется ни с одним из фактов, ни с заголовками правил.

После этого происходит возврат к запросу $P(u, f(v))$ и цели $R(x, z)$. Делается попытка достичь этой цели при помощи второго правила. Цель $R(x, z)$ снова достигается унификацией с первым фактом при подстановке σ, а цель $Q(b, g(y))$ достигается в силу того, что унифицируется со вторым фактом при помощи подстановки $\{c/y\}$. Следовательно, цель (2.11) достигается подстановкой $\{a/u, c/v\}$. Поэтому интерпретатор на этом заканчивает свою работу и выдает найденную подстановку. Выполнение программы считается успешно завершенным, и при этом найдено доказательство формулы $\exists u, v\, P(u, f(v))$. □

Для реализации идей логического программирования разработаны различные языки логического программирования, среди которых наиболее заметным является язык ПРОЛОГ. В его реализации ТУРБО ПРОЛОГ структура логической программы имеет следующий вид:

domains<структуры и типы данных>
global domains<внешние структуры и типы данных>
data base<глобальные предикаты динамической базы данных>
predicates<определение предикатов>
global predicates<внешние предикаты>
goal<цели>

clauses<факты и правила>

П р и м е р 2.9.2. Приведем логическую программу для выяснения родственных связей:

```
domains
name=symbol

predicates
parent(name,name)
father(name,name)
mother(name,name)
grandfather(name,name)
grandmother(name,name)

clauses
mother(Людмила,Сергей).
father(Александр,Сергей).
mother(Раиса,Людмила).
father(Виктор,Людмила).
mother(Валентина,Александр).
father(Василий,Александр).
parent(X,Y):- mother(X,Y).
parent(X,Y):- father(X,Y).
grandfather(X,Y):- father(X,Z),parent(Z,Y).
grandmother(X,Y):- mother(X,Z),parent(Z,Y).
```

Рассматривая запрос

?- grandfather(X,Сергей).

(“Кто является дедушкой Сергея?”), программа выдает первый правильный ответ “Виктор”, а при дополнительном обращении — и второй ответ “Василий”.

§ 2.10. Элементарные теории

Две алгебраические системы $\mathfrak{A}$ и $\mathfrak{B}$ сигнатуры Σ называются *элементарно эквивалентными* (обозначаются $\mathfrak{A} \equiv \mathfrak{B}$), если для любого предложения φ сигнатуры Σ истинность $\mathfrak{A} \models \varphi$ равносильна истинности $\mathfrak{B} \models \varphi$. Множество предложений $\{\varphi \mid \mathfrak{A} \models \varphi\}$ сигнатуры Σ называется *элементарной теорией* или просто *теорией* системы $\mathfrak{A}$ и обозначается через $\mathrm{Th}(\mathfrak{A})$.

Очевидно, что условие $\mathfrak{A} \equiv \mathfrak{B}$ равносильно равенству $\mathrm{Th}(\mathfrak{A}) = \mathrm{Th}(\mathfrak{B})$.

Заметим, что слово “элементарная” в терминах “элементарная эквивалентность” и “элементарная теория” объясняется тем, что любая формула исчисления предикатов выражает свойства систем через указание связей между их элементами, поскольку кванторы навешиваются лишь на переменные из множества V.

П р и м е р 2.10.1. Рассмотрим систему $\mathfrak{A}_1 \rightleftharpoons \langle \mathbb{Z}; s^{(1)} \rangle$ с одной одноместной *функцией следования* $s : \mathbb{Z} \leftrightarrow \mathbb{Z}$, где $s(n) = n + 1$ для каждого $n \in \mathbb{Z}$. Обозначим через $\mathfrak{A}_n$ систему сигнатуры $\{s\}$, имеющую n компонент связности, каждая из которых изоморфна системе $\mathfrak{A}_1$, $n \in \omega \setminus \{0\}$. Нетрудно заметить, что все системы $\mathfrak{A}_n$ имеют одну и ту же элементарную теорию. При этом модели $\mathfrak{A}_m$ и $\mathfrak{A}_n$ изоморфны тогда и только тогда, когда $m = n$. $\square$

Множество T предложений сигнатуры Σ, замкнутое относительно выводимости (т.е. содержащее все предложения, выводимые из T в ИП^Σ), называется *элементарной теорией* или просто *теорией сигнатуры* Σ. Теория T называется *непротиворечивой (противоречивой)*, если множество предложений T непротиворечиво (противоречиво). Непротиворечивая теория T называется *полной*, если $\varphi \in T$ или $\neg\varphi \in T$ для любого предложения сигнатуры Σ.

Очевидно, что для любой алгебраической системы $\mathfrak{A}$ теория $\mathrm{Th}(\mathfrak{A})$ полна и непротиворечива.

Любое подмножество Ax теории T, из которого выводимы все предложения из T, называется *системой аксиом для теории* T.

Если $\mathcal{F}$ — некоторое множество предложений сигнатуры Σ, то множество $T(\mathcal{F})$ предложений, выводимых из $\mathcal{F}$ в ИП^Σ, является теорией

с системой аксиом $\mathcal{F}$. При этом теория T называется *теорией, порожденной множеством* $\mathcal{F}$.

П р и м е р 2.10.2. 1. Теория функциональной сигнатуры $\{\cdot^{(2)}\}$, порожденная системой аксиом $\mathcal{F} = \{\varphi_{sg}\}$, где

$$\varphi_{sg} \rightleftharpoons \forall x, y, z\,((x \cdot y) \cdot z \approx x \cdot (y \cdot z))\,,$$

представляет собой *теорию полугрупп* T_{sg}. Любая полугруппа является моделью теории T_{sg}. Теория T_{sg} неполна, поскольку, например, ее моделями являются как полугруппы, удовлетворяющие закону коммутативности $\forall x, y\,(x \cdot y \approx y \cdot x)$ (такие как $\langle \omega; \cdot \rangle$), так и полугруппы, в которых этот закон нарушается. К примеру, в системе $\langle E_3; \cdot \rangle$, где E_3 — множество геометрических векторов, а $\cdot$ — операция векторного произведения, истинно предложение $\neg\forall x, y\,(x \cdot y \approx y \cdot x)$.

2. Теория полей T_F порождается системой следующих аксиом сигнатуры $\{+, -, \cdot, {}^{-1}, 0, 1\}$:

С1) $\forall x, y, z\ ((x + y) + z \approx x + (y + z))$,
С2) $\forall x, y\,(x + y \approx y + x)$,
С3) $\forall x\,(x + 0 \approx x)$,
С4) $\forall x\,(x + (-x) \approx 0)$,
У1) $\forall x, y, z\ ((x \cdot y) \cdot z \approx x \cdot (y \cdot z))$,
У2) $\forall x, y\,(x \cdot y \approx y \cdot x)$,
У3) $\forall x\,(x \cdot 1 \approx x)$,
У4) $\forall x\,(\neg(x \approx 0) \to (x \cdot x^{-1} \approx 1))$,
СУ) $\forall x, y, z\ ((x + y) \cdot z \approx (x \cdot z) + (y \cdot z))$,
Д) $\neg(0 \approx 1)$.

Моделями теории T_F являются в точности все поля.

3. Теория сигнатуры $\{\leq^{(2)}\}$, порожденная системой аксиом Γ_{dlo} из примера 2.1.9 является *теорией плотных линейных порядков* T_{dlo}. Добавляя к системе аксиом Γ_{dlo} аксиому $\forall x \exists y, z\,((y \leq x) \wedge \neg(y \approx x) \wedge (x \leq z) \wedge \neg(x \approx z))$ отсутствия наименьшего и наибольшего элементов, получаем систему аксиом Γ^*_{dlo}, порождающую *теорию* T^*_{dlo} *плотных линейных порядков без концевых элементов*. Теория T^*_{dlo} непротиворечива и полна.

4. Пусть $\mathfrak{A} = \langle A; C^{(3)} \rangle$ — алгебраическая система, состоящая из множества точек A и *отношения коллинеарности* C ($(a, b, c) \in C \Leftrightarrow$ точки a, b и c лежат на одной линии), удовлетворяющего следующим условиям:

а) $(a, a, a) \in C$ для любой точки $a \in A$;

б) любая линия $l \subseteq A$ однозначно определяется любыми двумя различными точками $a, b \in l$: $l = \{c \mid \mathfrak{A} \models C(a, b, c)\}$.

Система $\mathfrak{A}$ называется *геометрической системой* и определяет *геометрическую теорию* $\mathrm{Th}(\mathfrak{A})$. При этом система $\mathfrak{A}$ может быть как конечной, так и бесконечной, удовлетворять или не удовлетворять *аксиоме параллельности* о существовании для любой точки не более одной линии, проходящей через эту точку и параллельной данной линии.

Пусть $\mathfrak{G}_1 = \langle G_1; \cdot, e\rangle$ — группа с единицей e, позволяющая измерять расстояния между точками на линиях: если a и b — точки на линии l, то определено *расстояние* $|a\hat{}b|_l \in G_1$ от точки a до точки b на линии l. При этом будем считать, что $|a\hat{}a|_l = e$ и для любой точки $c \in l$ выполняется одно из равенств $(|a\hat{}c|_l)^{\pm 1} = |a\hat{}b|_l \cdot (|b\hat{}c|_l)^{\pm 1}$. Пусть $\mathfrak{G}_2 = \langle G_2; \cdot, e\rangle$ — группа с единицей e, позволяющая измерять расстояния между пересекающимися линиями: если l_1 и l_2 — линии, пересекающиеся в точке a, то определен *угол* $\angle(l_1, l_2)_a \in G_2$ от линии l_1 до линии l_2 относительно точки a. При этом будем считать, что $\angle(l_1, l_1)_a = e$ и для любой линии l_3, содержащей точку a, выполняется одно из равенств $(\angle(l_1, l_3)_a)^{\pm 1} = \angle(l_1, l_2)_a \cdot (\angle(l_2, l_3)_a)^{\pm 1}$. Группы $\mathfrak{G}_1$ и $\mathfrak{G}_2$ называются *группой сторон* и *группой углов* соответственно.

Последовательность точек $S \rightleftharpoons (a_1, \ldots, a_n)$ называется *многоугольником* или *n-угольником*, если существуют линии $l_1, \ldots, l_n$, для которых справедливо $a_1, a_2 \in l_1$, $a_2, a_3 \in l_2$, …, a_{n-1}, $a_n \in l_{n-1}$, $a_n, a_1 \in l_n$. Параметры $g_{11} \rightleftharpoons |a_1\hat{}a_2|_{l_1}$, …, $g_{1,n-1} \rightleftharpoons |a_{n-1}\hat{}a_n|_{l_{n-1}}$, $g_{1n} \rightleftharpoons |a_n\hat{}a_1|_{l_n}$, $g_{21} \rightleftharpoons \angle(l_1, l_2)_{a_2}$, …, $g_{2,n-1} \rightleftharpoons \angle(l_{n-1}, l_n)_{a_n}$, $g_{2n} \rightleftharpoons \angle(l_n, l_1)_{a_1}$ и углов n-угольника S задаются в виде матрицы $\begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \end{pmatrix}$. При $n = 3$ n-угольник называется *треугольником*, при $n = 4$ — *четырехугольником* и т.д.

Обогатим геометрическую систему $\mathfrak{A} = \langle A; C\rangle$ двухместными отношениями $Q_{g_1} \rightleftharpoons \{(a, b) \mid |a\hat{}b|_l = g_1$ для некоторой линии $l\}$ для всех $g_1 \in G_1$, а также трехместными отношениями $R_{g_2} = \{(a, b, c) \mid \angle(l_1, l_2)_b = g_2,$ где $a \in l_1, b \in l_2\}$ для всех $g_2 \in G_2$. Полученная система $\mathfrak{A}(\mathfrak{G}_1, \mathfrak{G}_2) \rightleftharpoons \langle A; C, Q_{g_1}, R_{g_2}\rangle_{g_1 \in G_1, g_2 \in G_2}$ называется *полигонометрической системой* над парой групп $(\mathfrak{G}_1, \mathfrak{G}_2)$. Если через любые две точки проходит линия, т.е. любой многоугольник разбивается на треугольники, то полигонометрическая система называется *тригонометрической*. Полигонометрическая (тригонометрическая) система $\mathfrak{A}(\mathfrak{G}_1, \mathfrak{G}_2)$ определяет *полигонометрическую* (соответственно *тригонометрическую*) *теорию* $\mathrm{Th}(\mathfrak{A}(\mathfrak{G}_1, \mathfrak{G}_2))$.

5. Как известно, для любой конечной системы $\mathfrak{A}$ конечной сигнатуры существует одна (может быть, достаточно длинная) формула $\varphi_{\mathfrak{A}}$, описывающая все взаимоотношения элементов из $\mathfrak{A}$. Тогда моделями

теории, порожденной системой аксиом $\{\varphi_{\mathfrak{A}}\}$, являются в точности все системы, изоморфные системе $\mathfrak{A}$. $\square$

Формулы φ и ψ сигнатуры Σ называются *T-эквивалентными* (обозначается $\varphi \equiv_T \psi$), где T — теория сигнатуры Σ, если в ИП^Σ из T выводима формула $(\varphi \leftrightarrow \psi)$.

Возможность нахождения "несложных" формул, представляющих все классы T-эквивалентности, существенно упрощает понимание структур моделей данной теории T. Простейшим здесь является класс теорий T, у которых любая формула T-эквивалентна некоторой бескванторной.

Теория T сигнатуры Σ называется *теорией с элиминацией кванторов* или *теорией, допускающей элиминацию кванторов*, если любая формула сигнатуры Σ T-эквивалентна некоторой бескванторной формуле.

Пусть Δ — некоторое множество формул сигнатуры Σ. Под *булевой комбинацией формул* из множества Δ понимается всякая формула, полученная из формул множества Δ в результате их соединения произвольным числом связок $\neg$ и $\wedge$.

Очевидно, у любой теории T с элиминацией кванторов каждая формула данной сигнатуры T-эквивалентна некоторой булевой комбинации атомарных формул.

Следующие теории обладают элиминацией кванторов.

П р и м е р 2.10.3. 1. Теория $T_\varnothing$ любой бесконечной модели пустой сигнатуры. Каждая формула пустой сигнатуры $T_\varnothing$-эквивалентна булевой комбинации формул вида $(x \approx y)$.

2. Теория T^*_{dlo} плотных линейных порядков без концевых элементов. Каждая формула сигнатуры $\{\leq\}$ T^*_{dlo}-эквивалентна булевой комбинации формул вида $(x \approx y)$ и $(x \leq y)$.

3. *Теория T_{ACF} алгебраически замкнутых полей*, которая получается из теории полей T_F добавлением бесконечной системы аксиом существования корней любого многочлена ненулевой степени:

$$\forall y_1, \ldots, y_n \exists x \, (x^n + y_1 x^{n-1} + \ldots y_{n-1} x + y_n \approx 0),$$

$n \in \omega \setminus \{0\}$. Поле комплексных чисел $\langle \mathbb{C}; +, -, \cdot, {}^{-1}, 0, 1 \rangle$ является моделью теории T_{ACF}.

4. *Теория T_{RCF} вещественно замкнутых полей*, которая получается из теории полей T_F добавлением двухместного предикатного символа $<$ и следующих аксиом:

П1) $\forall x \, \neg(x < x)$,

П2) $\forall x, y, z \, ((x < y) \wedge (y < z) \rightarrow (x < z))$,

П3) $\forall x, y\,((x < y) \vee (x \approx y) \vee (y < x))$,
П4) $\forall x, y\,((0 < x) \wedge (0 < y) \to (0 < x \cdot y))$,
П5) $\forall x, y, z\,((x < y) \to (x + z < y + z))$,
П6) $\forall x\,((0 < x) \to \exists y\,(x \approx y^2)$,
Р) $\forall y_1, \ldots, y_n \exists x(x^n + y_1 x^{n-1} + \ldots y_{n-1}x + y_n \approx 0)$ для каждого нечетного $n > 0$.

Упорядоченное поле вещественных чисел $\langle \mathbb{R}; +, -, \cdot, {}^{-1}, 0, 1, < \rangle$ является моделью теории T_{RCF}. $\square$

Консервативным расширением или *консервативным обогащением* теории T сигнатуры Σ называется всякая теория $T' \supseteq T$ сигнатуры $\Sigma' \supseteq \Sigma$, такая, что любая модель $\mathfrak{M}$ теории T может быть обогащена до модели $\mathfrak{M}'$ теории T'.

Существует два основных вида консервативных расширений теорий, приводящих к теориям с элиминацией кванторов. Первая операция — операция *полной скулемизации*, которая теорию T сигнатуры Σ преобразует в теорию T^{cS} сигнатуры Σ^{cS} добавлением аксиом Скулема для каждой формулы $\varphi = \exists x_0\,\psi(x_0, x_1, \ldots, x_n)$ сигнатуры Σ^{cS}.

Предложение 2.10.1. *Теория T^{cS} допускает элиминацию кванторов.*

Предикатное консервативное расширение теории T сигнатуры Σ, допускающее элиминацию кванторов, предложено Морли и называется *морлизацией*. Оно получается следующими преобразованиями.

Обозначим набор переменных $x_1, \ldots, x_n$ через $\overline{x}$. Определим сигнатуру Σ^M добавлением к сигнатуре Σ для каждой формулы $\varphi(\overline{x})$ сигнатуры Σ нового предикатного символа $R_\varphi^{(n)}$. *Морлизация теории T* есть теория T^M сигнатуры Σ^M, порожденная теорией T и *аксиомами Морли* $\forall \overline{x}\,(\varphi(\overline{x}) \leftrightarrow R_\varphi(\overline{x}))$ для всех формул $\varphi(\overline{x})$ сигнатуры Σ.

Предложение 2.10.2. *Теория T^M допускает элиминацию кванторов.*

Доказательство. Очевидно, что, заменяя в формуле $\varphi(\overline{x})$ сигнатуры Σ^M подформулы, начинающиеся с символов R_ψ, на T^M-эквивалентные им формулы сигнатуры Σ, получаем формулу $\varphi'(\overline{x})$ сигнатуры Σ, для которой $\varphi(\overline{x}) \equiv_{T^M} \varphi'(\overline{x})$. Так как $\varphi'(\overline{x}) \equiv_{T^M} R_{\varphi'}(\overline{x})$, формула $\varphi(\overline{x})$ T^M-эквивалентна бесквантворной формуле $R_{\varphi'}(\overline{x})$. $\square$

§ 2.11. Типы. Основные классы моделей

В этом параграфе мы определим классы моделей, играющих ключевую роль в описании всех моделей данных теорий, и установим важнейшие свойства этих моделей.

Поскольку каждая непротиворечивая теория расширяется до полной, информация о моделях данной теории T сводится к информации о моделях полных расширений теории T. Поэтому для определения классов моделей всюду в этом параграфе будем считать, что рассматриваемые теории полны. Кроме того, ограничимся случаем, когда сигнатура Σ не более чем счетна, а значит, счетна сама теория T.

Для множества X и кортежа $\overline{a} = (a_1, \ldots, a_n)$ запись $\overline{a} \in X$ будет означать, что все координаты кортежа $\overline{a}$ принадлежат множеству X.

Пусть T — теория сигнатуры Σ, $\mathfrak{M}$ — модель теории T, также имеющая сигнатуру Σ, A — некоторое подмножество множества M, $\overline{a} = (a_1, \ldots, a_n)$ — кортеж элементов из M, $\overline{x} = (x_1, \ldots, x_n)$ — кортеж переменных. *Типом от переменных $\overline{x}$ кортежа $\overline{a}$ над множеством A в модели $\mathfrak{M}$* называется множество формул

$$\mathrm{tp}^{\overline{x}}_{\mathfrak{M}}(\overline{a}/A) \rightleftharpoons \{\varphi(\overline{x}, \overline{b}) \mid \mathfrak{M} \models \varphi(\overline{a}, \overline{b}), \overline{b} \in A\}.$$

В дальнейшем, если из контекста ясно, о каком кортеже $\overline{x}$ и о какой модели $\mathfrak{M}$ идет речь, мы будем опускать соответствующие индексы и обозначать тип через $\mathrm{tp}(\overline{a}/A)$. Если $A = \varnothing$, то вместо записи $\mathrm{tp}(\overline{a}/\varnothing)$ используется запись $\mathrm{tp}(\overline{a})$.

Множество всех типов $\mathrm{tp}_{\mathfrak{M}}(\overline{a})$, $\overline{a} \in M$, называется *конечной диаграммой модели* $\mathfrak{M}$ и обозначается через $\mathrm{FD}(\mathfrak{M})$.

Тип кортежа $\overline{a}$ над множеством A содержит полную информацию о взаимоотношениях элементов, составляющих кортеж $\overline{a}$ и множество A. Таким образом, равенство $\mathrm{tp}(\overline{a}/A) = \mathrm{tp}(\overline{b}/A)$ означает, что информация о кортеже $\overline{a}$ над множеством A совпадает с информацией о кортеже $\overline{b}$ над множеством A. При этом равенство $\mathrm{tp}_{\mathfrak{M}}(\overline{a}/A) = \mathrm{tp}_{\mathfrak{M}}(\overline{b}/A)$ равносильно равенству $\mathrm{tp}_{\mathfrak{M}_A}(\overline{a}) = \mathrm{tp}_{\mathfrak{M}_A}(\overline{b})$. В конечной диаграмме $\mathrm{FD}(\mathfrak{M})$ перечисляются все информационные представители, присутствующие в модели $\mathfrak{M}$.

Теперь мы дадим несколько более общее понятие типа, не "привязанное" к конкретной модели.

Множеством A в теории T сигнатуры Σ называется подмножество $A \subseteq M$ носителя некоторой модели $\mathfrak{M}$ теории T вместе со следующим множеством формул сигнатуры Σ_A:

$$\mathrm{tp}(A, \mathfrak{M}) \rightleftharpoons \{\varphi(\overline{a}) \mid \mathfrak{M} \models \varphi(\overline{a}), \overline{a} \in A\}.$$

Пусть $\varphi(\overline{x},\overline{y})$ — формула сигнатуры Σ, $\overline{a}$ — кортеж элементов в T *длины* $l(\overline{a})$, равной длине $l(\overline{y})$ кортежа $\overline{y}$. Тогда формула $\varphi(\overline{x},\overline{a})$ — формула сигнатуры $\Sigma_{\overline{a}}$ — называется *формулой в теории* T.

Пусть T — теория сигнатуры Σ, A — некоторое множество в T, $\overline{x}$ — набор переменных длины $l(\overline{x}) = n$. Множество $\Phi(\overline{x},A)$ формул сигнатуры Σ_A называется *множеством формул в теории* T *над множеством* A, если каждая формула из $\Phi(\overline{x},A)$ имеет вид $\varphi(\overline{x},\overline{a})$, где $\overline{a}\in A$. Множество формул $\Phi(\overline{x},A)$ называется *совместным с теорией* T, если совместно множество $T\cup \mathrm{tp}(A,\mathfrak{M})\cup\Phi(\overline{x},A)$.

Под n-*типом* или просто *типом* над множеством A от переменных $\overline{x}$ понимается любое совместное с T множество формул $\Phi(\overline{x},A)$. Множество всех n-типов от переменных $\overline{x}$ над множеством A обозначается через ${}^{\subseteq}S^n_{\overline{x}}(A)$.

По теореме компактности из совместности любого типа $\Phi(\overline{x},\ A)$ с теорией T следует существование модели $\mathfrak{N}$ теории T, элементарно расширяющей модель $\mathfrak{M}$, $A\subseteq M$, и кортежа элементов $\overline{b}\in N$, $l(\overline{b}) = l(\overline{x})$, таких, что $\mathfrak{N}\models\varphi(\overline{b},\overline{a})$ для всех $\varphi(\overline{b},\overline{a})$ из $\Phi(\overline{x},A)$. В этом случае будем говорить, что тип $\Phi(\overline{x},A)$ *реализуется в модели* $\mathfrak{N}$, и писать $\mathfrak{N}\models\Phi(\overline{b},A)$. Если в модели $\mathfrak{N}$ нет кортежа $\overline{b}$, удовлетворяющего всем формулам из $\Phi(\overline{x},A)$, будем говорить, что тип $\Phi(\overline{x},A)$ *опускается в модели* $\mathfrak{N}$.

Тип $\Phi(\overline{x},A)$ называется *полным*, если $\Phi(\overline{x},A)$ — максимальное совместное с T множество формул $\varphi(\overline{x},\overline{a})$ сигнатуры Σ_A. Множество всех полных n-типов от переменных $\overline{x}$ над множеством A обозначается через $S^n_{\overline{x}}(A)$. Обозначим через ${}^{\subseteq}S^n(A)$ и $S^n(A)$ соответственно множество всех типов и полных типов от n переменных над множеством A. Положим

$$ {}^{\subseteq}S(A) \rightleftharpoons \bigcup_{n\in\omega\setminus\{0\}} {}^{\subseteq}S^n(A),\quad S(A)\rightleftharpoons \bigcup_{n\in\omega\setminus\{0\}} S^n(A). $$

Множество $D(T)\rightleftharpoons S(\varnothing)$ называется *конечной диаграммой теории* T.

В силу теоремы компактности каждый тип из $D(T)$ реализуется в некоторой модели теории T. Таким образом, конечная диаграмма теории T представима в виде объединения конечных диаграмм ее моделей.

Будем говорить, что тип $\Phi(\overline{x},A)$ *определяет* или *изолирует* тип $\Psi(\overline{x},A)$, и писать $\Phi(\overline{x},A)\vdash\Psi(\overline{x},A)$, если любая формула типа $\Psi(\overline{x},A)$ выводима из множества $T\cup\mathrm{tp}(A,\mathfrak{M})\cup\Phi(\overline{x},A)$. Тип $\Phi(\overline{x},A)$ называется *главным* или *изолированным*, если $\Phi(\overline{x},A)$ изолируется конечным

или, что эквивалентно, одноэлементным множеством. При этом формула $\varphi(\overline{x},\overline{a}) \in \Phi(\overline{x},A)$, для которой $\{\varphi(\overline{x},\overline{a})\} \vdash \Phi(\overline{x},A)$ (если такая существует), называется *главной* или *изолирующей* для типа $\Phi(\overline{x},A)$.

П р и м е р 2.11.1. Рассмотрим теорию $T = \mathrm{Th}(\langle\mathbb{Z};<,s\rangle)$, где s — функция следования. Множество $S^1_x(\varnothing)$ одноэлементно, т.е. любой 1-тип над $\varnothing$ изолируется формулой вида $(x \approx x)$. Множество $S^2_{(x,y)}(\varnothing)$ состоит из главных типов $p_n(x,y)$, определяемых формулами $(y \approx s^n(x))$, где $s^n(x) = x+n$, $n \in \mathbb{Z}$, и неглавного типа $p_\infty(x,y)$, изолируемого множеством формул $\neg(y \approx s^n(x))$, $n \in \mathbb{Z}$. При этом типы $p_n(x,y)$, $n \in \mathbb{Z}$, реализуются в модели $\mathfrak{M} = \langle\mathbb{Z};<,s\rangle$ $(p_n(x,y) = \mathrm{tp}_{\mathfrak{M}}(k,k+n)$, где $k \in \mathbb{Z})$, а тип $p_\infty(x,y)$ опускается в модели $\mathfrak{M}$.

Для любого элемента a некоторой модели теории T множество $S^1_x(\{a\})$ состоит из типов $p_\nu(x,a)$, где $\nu \in \mathbb{Z} \cup \{\infty\}$. □

Очевидно, любой главный тип реализуется в любой модели данной теории T. С другой стороны, для любого неглавного типа существует модель, в которой он опускается. Более того, справедлива следующая теорема.

Теорема 2.11.1. (теорема об опускании типов). *Для любого счетного множества A в теории T сигнатуры Σ и любой последовательности $(p_n(\overline{x}^n))_{n\in\omega}$ неглавных типов из $^{\subseteq}S(A)$ существует модель $\mathfrak{M}$ теории T, $M \supseteq A$, опускающая все типы $p_n(\overline{x}^n)$, $n \in \omega$.* □

Мы теперь готовы определить основные классы моделей. Начнем определение с “малых” моделей.

Подсистема $\mathfrak{A} = \langle A;\Sigma\rangle$ системы $\mathfrak{B} = \langle B;\Sigma\rangle$ называется *элементарной подсистемой* (обозначается $\mathfrak{A} \preccurlyeq \mathfrak{B}$), если для любой формулы $\varphi(x_1,\ldots,x_n)$ сигнатуры Σ и любых элементов $a_1,\ldots,a_n \in A$ условие $\mathfrak{A} \models \varphi(a_1,\ldots,a_n)$ равносильно условию $\mathfrak{B} \models \varphi(a_1,\ldots,a_n)$. При этом система $\mathfrak{B}$ называется *элементарным расширением* системы $\mathfrak{A}$.

Модель $\mathfrak{M}_0$ теории T называется *простой*, если для любой модели $\mathfrak{M}_1$ теории T существует элементарная подмодель$\mathfrak{M}_2 \preccurlyeq \mathfrak{M}_1$, изоморфная модели $\mathfrak{M}_0$.

П р и м е р 2.11.2. 1. Система $\mathfrak{M}_0 = \langle\mathbb{Z};<,s\rangle$ является простой моделью теории $\mathrm{Th}(\mathfrak{M}_0)$.

2. Простой моделью теории алгебраически замкнутого поля $\langle\mathbb{C};+,-,\cdot,\ ^{-1},0,1\rangle$ является *поле алгебраических чисел*, т.е. расширение поля рациональных чисел решениями уравнений вида $x^n+a_1x^{n-1}+\ldots+a_{n-1}x+a_n=0$. □

Очевидно, что простая модель счетна, и по теореме об опускании типов ее конечная диаграмма содержится в конечной диаграмме любой модели данной теории и состоит из изолированных типов. Верно

и обратное утверждение, т.е. указанные условия характеризуют простоту модели.

Теорема 2.11.2. (теорема Воота). *Модель $\mathfrak{M}$ теории T проста, если и только если $\mathfrak{M}$ — счетная модель и любой тип из $S(\varnothing)$, реализующийся в $\mathfrak{M}$, является изолированным.* $\square$

Теорема 2.11.3. (теорема единственности для простых моделей). *Любые две простые модели полной теории T изоморфны.* $\square$

Теорема 2.11.4. (теорема существования для простых моделей). *Теория T сигнатуры Σ имеет простую модель, если и только если любая формула сигнатуры Σ, совместная с T, принадлежит некоторому изолированному типу.*

Следующий класс, который мы рассмотрим, класс *однородных* моделей. Эти модели обладают тем свойством, что любые кортежи, имеющие одинаковый тип, связаны автоморфизмом, т.е. наборы элементов, обладающие одинаковой информацией, можно перемещать друг в друга с сохранением всех свойств модели.

Счетная алгебраическая система $\mathfrak{A}$ называется *однородной*, если для любых элементов $a_1, \ldots, a_n, b_1, \ldots, b_n, a \in A$ из элементарной эквивалентности моделей $\mathfrak{A}_{\{a_1,\ldots,a_n\}}$ и $\mathfrak{A}_{\{b_1,\ldots,b_n\}}$ с выделенными константами $a_1, \ldots, a_n$ и $b_1, \ldots, b_n$ соответственно следует $\mathfrak{A}_{\{a_1,\ldots,a_n,a\}} \equiv \mathfrak{A}_{\{b_1,\ldots,b_n,b\}}$ для некоторого $b \in A$.

Последовательным расширением частичных изоморфизмов $\{(a_1, b_1), \ldots, (a_n, b_n)\}$ однородной системы $\mathfrak{A}$ строятся автоморфизмы системы $\mathfrak{A}$, содержащие данный частичный изоморфизм $f : \overline{a} \leftrightarrow \overline{b}$, который характеризуется соотношением $\mathrm{tp}_{\mathfrak{A}}(\overline{a}) = \mathrm{tp}_{\mathfrak{A}}(\overline{a})$. Этим свойством расширения обладает любая простая модель.

Предложение 2.11.5. *Любая простая модель является однородной.*

Теорема 2.11.6. (теорема об изоморфизме однородных моделей). *Пусть $\mathfrak{M}$ и $\mathfrak{N}$ — счетные однородные модели теории T. Тогда следующие условия эквивалентны:*

(1) $\mathfrak{M} \simeq \mathfrak{N}$;

(2) *в $\mathfrak{M}$ и $\mathfrak{N}$ реализуются одни и те же типы из $S(\varnothing)$, т.е.* $\mathrm{FD}(\mathfrak{M}) = \mathrm{FD}(\mathfrak{N})$.

Теорема 2.11.7. (теорема существования для однородных моделей). *Для любой счетной системы $\mathfrak{A}$ существует счетное элементарное однородное расширение $\mathfrak{B} \succcurlyeq \mathfrak{A}$.*

Следующие две разновидности систем — системы, содержащие "большое" количество информации.

Счетная алгебраическая система $\mathfrak{A}$ называется *универсальной*, если в $\mathfrak{A}$ реализуются все типы из $S(\varnothing)$ теории $\mathrm{Th}(\mathfrak{A})$.

Счетная алгебраическая система $\mathfrak{A}$ называется *насыщенной*, если для любого конечного множества $A_0 \subset A$ в $\mathfrak{A}$ реализуются все типы из $S^1(A_0)$ теории $\mathrm{Th}(\mathfrak{A})$.

П р и м е р 2.11.3. Система $\mathfrak{M}$, состоящая из счетного числа компонент связности системы $\langle \mathbb{Z}; <, s\rangle$, является насыщенной моделью теории $\mathrm{Th}(\langle \mathbb{Z}; <, s\rangle)$. □

Предложение 2.11.8. *Для любой счетной алгебраической системы $\mathfrak{A}$ следующие условия эквивалентны:*

(1) *система $\mathfrak{A}$ насыщена;*

(2) *система $\mathfrak{A}$ универсальна и однородна.*

Следствие 2.11.9. (теорема единственности для насыщенных моделей). *Если $\mathfrak{A}$ и $\mathfrak{B}$ — насыщенные элементарно эквивалентные системы, то $\mathfrak{A} \simeq \mathfrak{B}$.*

Теорема 2.11.10. (теорема существования для насыщенных моделей). *Для любой теории T, имеющей бесконечные модели, следующие условия эквивалентны:*

(1) *T имеет насыщенную модель;*

(2) *конечная диаграмма $D(T)$ счетна.*

П р и м е р 2.11.4. Теория T счетного множества независимых одноместных предикатов не имеет насыщенной модели, поскольку $|D(T)| = 2^\omega$. □

Следствие 2.11.11. *Если теория T имеет насыщенную модель, то T имеет простую модель.*

Нетрудно заметить, что если $|D(T)| > \omega$, то $|D(T)| = 2^\omega$. Таким образом, конечная диаграмма любой счетной теории счетна или континуальна. Если $|D(T)| = \omega$, то теория T называется *малой*.

§ 2.12. Категоричность. Спектры моделей полных теорий

В этом параграфе мы будем рассматривать непротиворечивые счетные теории T сигнатуры Σ, имеющие бесконечные модели. Зафиксируем такую теорию. Для каждой бесконечной мощности λ обозначим через $I(T, \lambda)$ число попарно неизоморфных моделей теории T, имеющих мощность λ. Функция $I(T, \cdot)$, сопоставляющая каждой бесконеч-

ной мощности λ значение $I(T,\lambda)$, называется *спектральной функцией* или *спектром теории* T.

Очевидно, что значение $I(T,\lambda)$ не может превосходить мощности множества всех подмножеств данного множества мощности λ, т.е. $I(T,\lambda) \leq 2^{\lambda}$. С другой стороны, для любой бесконечной мощности λ существует по крайней мере одна модель теории T, имеющая мощность λ. Таким образом, $1 \leq I(T,\lambda) \leq 2^{\lambda}$.

Из неравенства $1 \leq I(T,\lambda)$ следует, что получить однозначное с точностью изоморфизма или *категоричное* описание модели в виде элементарной теории можно лишь для конечных моделей. Следующее понятие выделяет класс теорий, однозначно описывающих модели фиксированной мощности.

Теория T называется *категоричной в мощности* λ или λ-*категоричной*, если $I(T,\lambda) = 1$.

Категоричность теории в некоторой бесконечной мощности позволяет устанавливать полноту теории.

Теорема 2.12.1. (теорема Лося — Воота). *Если теория* T *категорична в некоторой бесконечной мощности* λ *и содержит формулы*

$$\exists x_1, \ldots, x_n \, (\bigwedge_{i<j\leq n} \neg(x_i \approx x_j)),$$

$n \in \omega$, *то теория* T *полна.*

П р и м е р 2.12.1. Рассмотрим теорию T^*_{dlo} плотных линейных порядков без концевых элементов (см. пример 2.10.2). Пошаговыми расширениями конечных изоморфизмов доказывается, что любые две счетные модели теории T^*_{dlo} изоморфны, т.е. теория T^*_{dlo} ω-категорична. Из отсутствия конечных моделей теории T^*_{dlo} по теореме Лося — Воота получаем, что теория T^*_{dlo} полна. $\square$

Далее в этом параграфе будут рассматриваться полные теории. Следующая теорема представляет синтаксическую характеризацию ω-категоричности.

Теорема 2.12.2. (теорема Рыль-Нардзевского). *Теория* T ω-*категорична, если и только если для любого* $n \in \omega \setminus \{0\}$ *множество* n-*типов* $S^n_{(x_1,\ldots,x_n)}(\varnothing)$ *конечно.*

П р и м е р 2.12.2. 1. Теория T^*_{dlo} ω-категорична и допускает элиминацию кванторов. Поэтому для любого $n \in \omega$ конечное множество типов от переменных $x_1, \ldots, x_n$ определяется относительным порядком элементов $x_1, \ldots, x_n$.

2. Рассмотрим теорию $T = \mathrm{Th}(\langle \mathbb{Z}; s^{(1)} \rangle)$ с функцией следования s.

Как показано в примере 2.11.1, теория T имеет счетное число 2-типов $p(x,y) \in S(\varnothing)$. В силу теоремы Рыль-Нардзевского теория T не ω-категорична. $\square$

Для теорий, категоричных в несчетных мощностях, справедлива следующая

Теорема 2.12.3. (теорема Морли). *Если теория T категорична в некоторой несчетной мощности λ, то T категорична в любой несчетной мощности.* $\square$

Таким образом, если $I(T,\lambda) = 1$ для некоторой мощности $\lambda > \omega$, то $I(T,\lambda) = 1$ для любой мощности $\lambda > \omega$. Теория T, категоричная в некоторой (любой) несчетной мощности, называется *несчетно категоричной*.

Теория T называется *стабильной в мощности* λ или λ-*стабильной*, если для любого множества A в T мощности λ имеет место $|S(A)| \leq |A|$. Теория T называется *стабильной* (*суперстабильной*), если T стабильна в некоторой бесконечной мощности λ (T стабильна во всех мощностях $\lambda \geq 2^{\omega}$).

Теорема 2.12.4. (теорема Шелаха). *Если теория T несуперстабильна (в частности, нестабильна), то $I(T,\lambda) = 2^{\lambda}$ для любой несчетной мощности λ.* $\square$

Нестабильность теории T характеризуется существованием некоторой формулы $\varphi(\overline{x},\overline{y})$ и такой бесконечной последовательности кортежей $\overline{a}_n$ в T, $l(\overline{x}) = l(\overline{y}) = l(\overline{a_n})$, $n \in \omega$, что в некоторой модели $\mathfrak{M}$ теории T выполняется $\mathfrak{M} \models \varphi(\overline{a}_m,\overline{a}_n) \Leftrightarrow m \leq n$. В частности, с помощью формулы $(x \leq y)$ проверяется нестабильность теории T^*_{dlo} плотных линейных порядков, и в силу теоремы Шелаха имеет место $I(T^*_{\mathrm{dlo}},\lambda) = 2^{\lambda}$ для любой несчетной мощности λ.

§ 2.13. Система аксиом арифметики Пеано. Нестандартные модели арифметики

Представим аксиомы сигнатуры $\Sigma_0 = \{s^{(1)}, +^{(2)}, \cdot^{(2)}, 0^{(0)}, \leq^{(2)}\}$, порождающие элементарную теорию арифметики:

S1) $\forall x,y\,((s(x) \approx s(y)) \to (x \approx y))$,
S2) $\forall x\,\neg(s(x) \approx 0)$,
S3) $\forall x\,(\neg(x \approx 0) \to \exists y\,(x \approx s(y)))$,
C1) $\forall x\,((x+0 \approx x) \wedge (0+x \approx x))$,
C2) $\forall x,y\,((x+s(y) \approx s(x+y)) \wedge (s(x)+y \approx s(x+y)))$,
У1) $\forall x\,((x \cdot 0 \approx 0) \wedge (0 \cdot x \approx 0))$,

У2) $\forall x, y\,(x \cdot s(y) \approx x \cdot y + x)$,

П) $\forall x, y, z\,((x \leq x) \wedge ((x \leq y) \wedge (y \leq x) \to (x \approx y)) \wedge ((x \leq y) \wedge (y \leq z) \to (x \leq z)) \wedge ((x \leq y) \vee (y \leq x)))$,

ПС) $\forall x, y\,((x \leq y) \leftrightarrow \exists z\,(x + z \approx y))$.

Обозначим множество всех перечисленных аксиом через $\mathbb{A}_0$. Отметим, что приведенные аксиомы согласуются с аксиомами Дедекинда — Пеано, приведенными в § 1.3 учебника "Дискретная математика". Аксиома математической индукции является единственной в аксиоматике Дедекинда — Пеано, не выразимой в логике первого порядка, поскольку в этой аксиоме "навешивается" квантор не по элементам, а по предикатам. Элементарным аналогом аксиомы математической индукции будет следующая схема аксиом для всех формул $\varphi(x)$ сигнатуры Σ с одной свободной переменной x:

И$_\varphi$) (*аксиома математической индукции для формулы* $\varphi(x)$)

$$\varphi(0) \wedge \forall x\,(\varphi(x) \to \varphi(s(x))) \to \forall y\,\varphi(y).$$

Множество аксиом $\mathbb{A}_0$ вместе с аксиомами математической индукции для формул составляют *систему аксиом арифметики Пеано*, обозначаемую через $\mathbb{P}$. Теория $T_{\mathbb{P}}$, порожденная системой аксиом $\mathbb{P}$, называется *теорией арифметики Пеано*.

Очевидно, теория $T_{\mathbb{P}}$ непротиворечива и система $\mathfrak{N} = \langle \mathbb{N}; s, +, \cdot, 0, \leq \rangle$ является ее моделью, называемой *стандартной моделью арифметики*.

Обозначим через Δ_0 константный символ 0, через Δ_{n+1} — терм $s(\Delta_n)$, $n \in \omega$.

Перечислим некоторые основные свойства, выводящиеся из системы аксиом $\mathbb{P}$. В частности, покажем, что арифметика на термах Δ_n согласуется с обычной арифметикой на натуральных числах.

Предложение 2.13.1. *Из аксиом* $\mathbb{A}_0$ *выводимы следующие предложения:*

(1) $\forall x\,(0 \leq x)$;

(2) $\forall x\,((x \leq \Delta_n) \to (x \approx \Delta_0) \vee \ldots \vee (x \approx \Delta_n))$;

(3) $\neg(\Delta_m \approx \Delta_n)$, *если* $m \neq n$;

(4) $(\Delta_k + \Delta_m \approx \Delta_n)$, *если* $k + m = n$;

(5) $\neg(\Delta_k + \Delta_m \approx \Delta_n)$, *если* $k + m \neq n$;

(6) $(\Delta_k \cdot \Delta_m \approx \Delta_n)$, *если* $k \cdot m = n$;

(7) $\neg(\Delta_k \cdot \Delta_m \approx \Delta_n)$, *если* $k \cdot m \neq n$.

Из предложения 2.11.2 вытекает, что стандартная модель арифметики $\mathfrak{N}$ является простой моделью теории $T = \mathrm{Th}(\mathfrak{N})$. Вместе с тем описательные возможности логики первого порядка не позволяют го-

ворить о категоричности теории T ни в какой мощности. Более того, в каждой бесконечной мощности число попарно неизоморфных моделей теории T максимально.

Теорема 2.13.2. *Для любой бесконечной мощности λ имеет место равенство $I(\mathrm{Th}(\mathfrak{N}), \lambda) = 2^\lambda$.*

Доказательство. Покажем сначала, что для теории $T = \mathrm{Th}(\mathfrak{N})$ справедливо $I(T, \omega) = 2^\omega$. Для этого достаточно показать, что $|S^1(\varnothing)| = 2^\omega$. Так как модель $\mathfrak{N}$ проста и любой ее элемент представляется в виде константного терма (т.е. терма, не содержащего переменных), достаточно установить, что $|S^1(N)| = 2^\omega$. Каждому множеству $P = (p_n)_{n\in\omega}$, состоящему из бесконечного числа простых чисел, поставим в соответствие множество формул

$$q_P(x) = \{\exists y\,(p_n \cdot y \approx x) \mid n \in \omega\} \cup$$

$$\{\neg\exists y\,(p \cdot y \approx x) \mid p \text{ — простое число и } p \notin P\}.$$

Локально совместное, а значит, в силу теоремы компактности совместное множество $q_P(x)$ реализуется элементами, делящимися на все простые числа из P и не делящимися на простые числа, не входящие в P. Тогда число попарно несовместных типов $q_P(x)$ совпадает с 2^ω и, следовательно, $|S^1(\varnothing)| = |S^1(N)| = 2^\omega$. Пусть теперь λ — произвольный несчетный кардинал. Наличие бесконечного линейного порядка в модели $\mathfrak{N}$ влечет нестабильность теории T и в силу теоремы Шелаха получаем $I(T, \lambda) = 2^\lambda$. □

Из теоремы 2.13.2 вытекает существование континуума счетных нестандартных моделей арифметики. Каждая такая модель содержит по крайней мере один *аномальный* элемент, т.е. элемент, не интерпретирующийся никаким натуральным числом.

В завершение покажем, что добавление к системе аксиом $\mathbb{A}_0$ аксиомы математической индукции в полном объеме гарантирует однозначность описания модели арифметики. Алгебраическая система $\mathfrak{M} = \langle M; s, +, \cdot, 0, \leq\rangle$ называется *системой Дедекинда — Пеано* или *DP-системой*, если $\mathfrak{M}$ является моделью теории $T_{\mathbb{P}}$ и удовлетворяет аксиоме математической индукции

$$\forall A \subseteq M((0 \in A) \wedge \forall x\,((x \in A) \to s(x) \in A) \to A = M).$$

Из аксиомы математической индукции следует отсутствие аномальных элементов в любой DP-системе. Таким образом, справедлива

Теорема 2.13.3. *Любые две DP-системы изоморфны.* □

§ 2.14. Задачи и упражнения

1. Пусть $f^{(1)}$, $g^{(2)}$, $h^{(3)}$ — функциональные, а $P^{(1)}$, $Q^{(3)}$ — предикатные символы. Являются ли формулами следующие слова:
   а) $P(f(x)) \wedge \forall x\, \neg Q(g(y,z), x, h(z,y,x))$;
   б) $P(Q(x, g(x,y), h(x,y,z)))$;
   в) $h(x, g(x,y), f(x))$?

2. Выписать все подформулы формулы:
   а) $Q(x, g(x,y), h(x,y,z))$; б) $(\forall x\, \neg P(x) \to \neg\exists y\, (\forall z\, P(x) \wedge Q(x,y,z)))$.

3. Перечислить свободные и связанные вхождения в следующих формулах:
   а) $\exists x\, (P(x,y) \vee \neg\forall y\, Q(x,y))$;
   б) $(\neg\forall x\, P(x,y) \to Q(f(x,y)))$.

4. Написать предложение, истинное в системе $\langle \omega; \cdot \rangle$ и ложное в системе $\langle \mathbb{Z}; \cdot \rangle$.

5. Определим на множестве людей следующие отношения: отец(x,y) — x является отцом для y, мать(x,y) — x является матерью для y, муж(x,y) — x является мужем для y, жена(x,y) — x является женой для y, м(x) — x имеет мужской пол, ж(x) — x имеет женский пол. В сигнатуре указанных отношений описать следующие отношения:
   а) брат(x,y) — x является братом для y,
   б) сестра(x,y) — x является сестрой для y,
   в) сын(x,y) — x является сыном для y,
   г) дочь(x,y) — x является дочерью для y,
   д) племянник(x,y) — x является сыном брата или сестры для y,
   е) племянница(x,y) — x является дочерью брата или сестры для y,
   ж) зять(x,y) — x является мужем дочери или сестры для y,
   з) невестка(x,y) — x является женой сына или брата для y.

6. Доказать выполнимость формулы
$$\forall x\, \forall y\, \exists z\, (P(x,z) \wedge P(z,y) \wedge \neg(x \approx z) \wedge \neg(z \approx y)).$$

7. Доказать тождественную истинность формулы
$$\forall x\, (\varphi(x) \to \neg\psi(x)) \to \neg(\exists x\, \varphi(x) \wedge \forall x\, \psi(x)).$$

8. Доказать выводимость в ИПС следующих секвенций:
   а) $\vdash (\forall x \forall y\, \varphi(x,y) \to \forall x\, \varphi(x,x))$;
   б) $\vdash (\exists x\, \varphi(x,x) \to \exists x \exists y\, \varphi(x,y))$.

9. Проверить, выводима ли в ИПС формула
$$\vdash (\exists x\, (\varphi(x) \to \psi(x)) \leftrightarrow (\forall x\, (\varphi(x) \to \exists x\, \psi(x)))).$$

10. *Гипотеза четырех красок* утверждает, что любую карту можно раскрасить четырьмя красками так, что никакие соседние страны не будут иметь одинаковый цвет. Показать, что если гипотеза четырех красок справедлива для карт с конечным числом стран, то она справедлива для всех карт.

11. Привести к ПНФ, ПКНФ и КлНФ формулу

$$(\neg(\forall x\,\forall y\,P(x,y) \to \forall x\,\forall y\,R(x,y)) \vee \forall x\,\forall y\,P(x,y)).$$

12. Построить все попарные композиции $\theta_i \circ \theta_j$ подстановок

$$\theta_1 = \{F_1(y)/x, F_2(x,z)/y, c_1/z\},$$
$$\theta_2 = \{F_2(F_1(x),y)/x, F_1(c_1)/y, F_1(z)/z\},$$
$$\theta_3 = \{F_1(c_1)/x, c_2/y, x/z\},$$
$$\theta_4 = \{y/x, z/y, x/z\}$$

сигнатуры $\Sigma = \{c_1^{(0)}, c_2^{(0)}, F_1^{(1)}, F_2^{(2)}\}$.

13. Определить, унифицируемо ли множество W. В случае унифицируемости найти наиболее общий унификатор:
    а) $W = \{P(c,x), P(c,c)\}$;
    б) $W = \{P(c,x,F(x)), P(c,y,y)\}$;
    в) $W = \{(F(u,F_1(x,y)) \approx x), (F(y,z) \approx x), (F(u,F_1(c,z)) \approx x)\}$.

14. Определить, имеют ли склейки следующие дизъюнкты. Если да, построить их:
    а) $P_1(x) \vee P_2(y) \vee P_1(F(x))$;
    б) $(F_1(x) \approx F_2(y)) \vee (F_1(F_2(c)) \approx F_2(y)) \vee (F_1(z) \approx F_2(z))$.

15. Найти все возможные резольвенты следующих пар дизъюнктов:
    а) $\neg P_1(x) \vee P_2(x,c_1)$, $P_1(c_2) \vee P_2(c_2,c_1)$;
    б) $\neg P_1(x) \vee P_2(x,x)$, $\neg P_2(c,F(c))$;
    в) $(F_1(x) \approx F_2(y,c_1)) \vee P_1(x)$, $\neg(F_1(F_1(y)) \approx z) \vee \neg P_1(F_1(y))$.

16. Проверить невыполнимость следующих множеств формул:
    а) $\{\neg P_1(x) \vee (F_1(x) \approx x), P_1(F_2(c)) \vee \neg(y \approx x)\}$;
    б) $\{P_1(c_1), \neg P_2(y) \vee P_3(c_1,y), \neg P_1(x) \vee \neg P_4(y) \vee \neg P_3(x,y), P_2(c_2), P_4(c_2)\}$.

17. Установить, выполнимо ли множество предложений $\{\Phi_1, \ldots, \Phi_n\}$. Если множество выполнимо, построить для него модель:
    а) $\Phi_1 \rightleftharpoons \neg\forall x\,(P(x) \to \forall y\,(P(y) \to ((Q(x) \to \neg Q(y)) \vee \forall z\,P(z))))$;
    б) $\Phi_1 \rightleftharpoons \forall x\,\forall y\,(P_1(x,y) \to P_2(x,y))$,
    $\Phi_2 \rightleftharpoons \forall x\,\forall y\,(P_2(x,y) \to P_3(x,y))$,
    $\Phi_3 \rightleftharpoons \exists x\,\exists y\,P_1(x,y)$;
    в) $\Phi_1 \rightleftharpoons \forall x\,((P_1(x) \wedge \neg P_2(x)) \to \exists y\,(P_3(x,y) \wedge P_4(y)))$,
    $\Phi_2 \rightleftharpoons \exists x\,(P_5(x) \wedge P_1(x) \wedge \forall y\,(P_3(x,y) \to P_5(y)))$,
    $\Phi_3 \rightleftharpoons \forall x\,(P_5(x) \to \neg P_2(x))$.

18. Пусть T — теория плотных линейных порядков. Показать, что T имеет ровно четыре полных расширения сигнатуры $\{\leq\}$.

19. Пусть T — теория константной сигнатуры $\{c_n \mid n \in \omega\}$. Показать, что T допускает элиминацию кванторов.

20. Описать теории сигнатуры $\{E\}$ одного отношения эквивалентности. Найти все возможные функции спектра для этих теорий.

21. Определить интерпретации сигнатурных символов теории арифметики Пеано так, чтобы моделью теории $T_{\mathbb{P}}$ стала система с носителем:

    (а) $\mathbb{N} \cup \{a\}$, где $a \notin \mathbb{N}$; (б) $\mathbb{N} \cup \{a, b\}$, где $a, b \notin \mathbb{N}$;

    (в) $\mathbb{N} \cup \{a_n \mid n \in \omega\}$, где $a_n \notin \mathbb{N}$ для всех $n \in \omega$.

После изучения главы 2 выполняются задачи 2–6 контрольной работы. Задача 2 решается аналогично примеру 2.1.7, задача 3 — аналогично примеру 2.2.2, п.2, задача 4 — аналогично примеру 2.2.2, пп.4, 5, задача 5 — аналогично примерам 2.4.1 и 2.7.3, а задача 6 — аналогично примерам 2.8.12–2.8.14.

Г л а в а 3

ЭЛЕМЕНТЫ ТЕОРИИ АЛГОРИТМОВ

При изучении дискретной математики и формальных исчислений мы рассматривали большое количество различных алгоритмов. Это и алгоритм Евклида нахождения наибольших общих делителей, и алгоритмы нахождения кратчайших маршрутов во взвешенном графе, и алгоритм распознавания доказуемости формул исчисления высказываний.

Отметим несколько основных общих черт алгоритмов.

1. *Алгоритм* — это процесс последовательного построения (вычисления) величин, протекающий в дискретном времени так, что в начальный момент времени задается исходная конечная система величин, а в каждый последующий момент система величин получается по определенному закону (*программе*) из системы величин, имевшихся в предыдущий момент времени (*дискретность алгоритма*).

2. Система величин, получаемых в любой не начальный момент времени, однозначно определяется системой величин, полученных в предыдущие моменты времени (*детерминированность алгоритма*).

3. Закон получения последующей системы величин из предшествующей должен быть простым (*элементарность шагов алгоритма*).

4. Если способ получения последующей величины из какой-нибудь заданной величины не дает результата, то должно быть указано, что́ надо считать результатом алгоритма (*направленность алгоритма*).

5. Алгоритм должен быть пригоден для решения всех задач из заданного класса (*массовость алгоритма*).

Приведенные свойства определяют, конечно, не строгое понятие алгоритма, которое называется *интуитивным*. Оно практически не вызывает разногласий относительно того, является ли данный процесс алгоритмическим. Однако ситуация существенно изменяется, если есть предположение об *алгоритмической неразрешимости* задачи, т.е. о невозможности решить ее алгоритмическими методами.

В 20-х — 30-х годах двадцатого века предпринимались попытки формализовать понятие алгоритма. В результате было предложено несколько моделей алгоритмов (машины Поста и Тьюринга, рекурсивные функции, нормальные алгорифмы Маркова (1954 г.) и др.). Впоследствии было установлено, что классы решаемых ими задач совпадают, и на основании этого появился тезис о моделях алгоритмов, опубликованный впервые Чёрчем в 1936 г. для класса рекурсивных функций и носящий его имя в следующем современном виде.

Тезис Чёрча. Класс задач, решаемых в любой формальной алгоритмической модели, совпадает с классом задач, которые могут быть решены интуитивно алгоритмическими методами.

Тезис Чёрча доказать нельзя, поскольку интуитивное понятие алгоритма строго не определяется.

Любое вычисление по алгоритму A можно представить в виде функции $f_A : X \to \mathbb{N}$, где $X \subseteq \mathbb{N}^n$, такой, что для любых числовых данных $(x_1, \ldots, x_n) \in X$ значение $f_A(x_1, \ldots, x_n)$ совпадает с результатом $A(x_1, \ldots . x_n)$ работы алгоритма A на данных $x_1, \ldots, x_n$. Таким образом, в классе всех функций $f : X \to \mathbb{N}$, где $X \subseteq \mathbb{N}^n$, выделяется подкласс *вычислимых функций*, т.е функций вида f_A. Тем самым тезис Чёрча утверждает, что совпадают классы вычислимых и рекурсивных функций.

Формальное определение понятия алгоритма создало предпосылки для разработки теории алгоритмов. Прогресс вычислительной техники стимулировал дальнейшее развитие этой теории.

В настоящей главе мы рассмотрим две основные модели вычислимости — машины Тьюринга и рекурсивные функции, установим эквивалентность этих моделей и на основе этих моделей укажем некоторые пределы вычислимости.

§ 3.1. Машины Тьюринга

1. Определение и примеры. *Машина Тьюринга* T представляет собой систему, работающую в дискретные моменты времени $t = 0, 1, 2, \ldots$ и состоящую из следующих частей (рис. 3.1).

1. *Конечная лента*, разбитая на конечное число *ячеек*. При этом в каждый момент времени t в ячейках записаны буквы из некоторого алфавита $A = \{a_0, a_1, \ldots, a_m\}$ (где $a_0 = 0$, $a_1 = 1$, $m \geq 1$), называемого *внешним алфавитом машины*. Ячейка, в которой записан символ 0, называется *пустой*. В процессе работы машины каждая ячейка может менять свое состояние путем замены приписанного к ней сим-

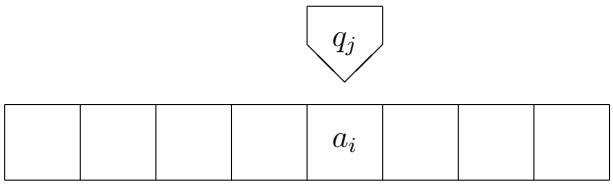

Рис. 3.1

вола a_i на другой символ $a_l \in A$. К существующим ячейкам можно пристраивать неограниченное число дополнительных ячеек, которые изначально считаются пустыми. Лента считается направленной, и ее ячейки будут просматриваться слева направо. Таким образом, если в какой-то момент времени лента имеет r ячеек, то *состояние ленты* полностью описывается словом $a_{i_1}a_{i_2}\dots a_{i_r}$, где a_{i_1} — состояние первой (слева) ячейки, a_{i_2} — состояние второй ячейки и т.д.

2. *Управляющая головка*, представляющая собой устройство, которое может перемещаться вдоль ленты так, что в каждый рассматриваемый момент времени оно находится напротив определенной ячейки и имеет некоторое состояние q_j из конечного *множества внутренних состояний* $Q = \{q_0, q_1, \dots, q_n\}$, $Q \cap A = \varnothing$. Состояние q_0 называется *заключительным* и означает завершение работы машины, а состояние q_1 — *начальным* и означает начало работы машины.

3. *Программа* Π, т.е. совокупность выражений $T(i, j)$ (где $i = 0, \dots, m$, $j = 1, \dots, n$), каждое из которых имеет один из следующих видов:

— $a_i q_j \to L\, q_k$ (сдвиг головки, находящейся в состоянии q_j напротив ячейки с буквой a_i, на одну ячейку *влево* с заменой состояния q_j на q_k);

— $a_i q_j \to R\, q_k$ (сдвиг головки, находящейся в состоянии q_j напротив ячейки с буквой a_i, на одну ячейку *вправо* с заменой состояния q_j на q_k);

— $a_i q_j \to a_l q_k$ (замена буквы a_i в текущей ячейке на букву a_l, а также замена состояния q_j головки на состояние q_k).

Выражения $T(i, j)$ называются *командами*. При этом команды не могут начинаться со слов $a_i q_0$. Символы L и R не принадлежат множеству $A \cup Q$.

Таким образом, машина Тьюринга T есть пятерка $\langle A, Q, \Pi, q_0, q_1\rangle$, а работа машины T состоит в изменении ее *конфигурации* K, состоящей из состояния ленты, состояния головки и положения текущей ячейки. Дискретным моментам времени $t = 0, 1, 2, \dots$ соответствуют конфигурации $K_0, K_1, K_2, \dots$ Конфигурация K_0 называется *начальной*, а конфигурация K_t, содержащая q_0, — *заключительной*. Конфигурации

будут задаваться в виде *машинных слов* $M = \alpha q_j a_i \beta$, где $\alpha a_i \beta$ — состояние ленты, q_j — состояние головки, находящейся напротив ячейки с состоянием a_i, занимающей то же положение среди других ячеек, что и буква a_i в слове $\alpha q_j a_i \beta$.

Опишем *преобразование* $M \to^T M'$ *машинного слова* M *в машинное слово* M' *за один шаг работы машины* T.

Пусть команда $T(i,j)$ равна $a_i q_j \to L\, q_k$. Если α — пустое слово Λ, то $M_2 = q_k a_0 a_i \beta$. Если же $\alpha = \alpha' a_l$, то $M_2 = \alpha' q_k a_l a_i \beta$.

Если $T(i,j) = a_i q_j \to R\, q_k$, то $M_2 = \alpha a_i q_k a_0$ при $\beta = \Lambda$ и $M_2 = \alpha a_i q_k a_l \beta'$ при $\beta = a_l \beta'$.

Если $T(i,j) = a_i q_j \to a_l q_k$, полагаем $M_2 \rightleftharpoons \alpha q_k a_l \beta$.

Будем говорить, что *машинное слово* M' *получается из машинного слова* M *с помощью машины* T, и писать $M \Rightarrow^T M'$, если существует последовательность преобразований $M_i \to^T M_{i+1}$, $i = 0, \ldots, k-1$, для которой $M_0 = M$, $M_k = M'$.

Преобразование $M \Rightarrow^T M'$, при котором на каждом переходе $M_i \to^T M_{i+1}$ *не достраиваются ячейки слева*, обозначается через $M \Rrightarrow^T M'$.

Преобразование $M \Rightarrow^T M'$, при котором на каждом переходе $M_i \to^T M_{i+1}$ *не достраиваются ячейки ни слева, ни справа*, обозначается через $M \Lrightarrow^T M'$.

В дальнейшем в записях $\to^T$, $\Rightarrow^T$, $\Rrightarrow^T$ и $\Lrightarrow^T$ индекс ^{T} будет опускаться, если из контекста будет ясно, о какой машине T идет речь.

Зафиксируем алфавит $A = \{a_0, a_1, a_2, \ldots, a_m\}$. Через a_i^x будем обозначать слово $a_i a_i \ldots a_i$ алфавита A, состоящее из x букв a_i. Приведем список машин, которые будут использоваться в дальнейшем в качестве составляющих для других машин.

1. **А** (*перенос нуля*): $q_1 001^x 0 \Lrightarrow q_0 01^x 00$.
2. **Б**$^+$ (*сдвиг вправо*): $q_1 a_i 1^x 0 \Lrightarrow a_i 1^x q_0 0$.
3. **Б**$^-$ (*сдвиг влево*): $01^x q_1 a_i \Lrightarrow q_0 01^x a_i$.
4. **В** (*транспозиция*): $q_1 01^x 01^y 0 \Lrightarrow q_0 01^y 01^x 0$.
5. **К** (*копирование*): $q_1 01^x 00^x 0 \Lrightarrow q_0 01^x 01^x 0$.
6. **Л** (*стирающая машина*): $q_1 01^x 0 \Lrightarrow q_0 00^x 0$.
7. **R** (*удаление* 1): $q_1 01^{x+1} 0 \Lrightarrow q_0 01^x 00$.
8. **S** (*добавление* 1): $q_1 01^x 0 \Lrightarrow q_0 01^{x+1}$.
9. **Сложение**: $q_1 01^{x+1} 01^{y+1} 0 \Lrightarrow q_0 01^{x+y+1} 000$.
10. **Умножение**: $q_1 01^{x+1} 01^{y+1} 0 \Rrightarrow q_0 01^{x \cdot y+1} 0$.

Приведем примеры программ для машин $\mathbf{Б}^+$ и **Сложение**, оставив написание остальных программ читателю в качестве упражнения.

Программа для машины $\mathbf{Б}^+$, например, выглядит так:

$$0q_1 \to Rq_2,\ 0q_2 \to 0q_0,$$

$$a_iq_1 \to Rq_2,\ a_iq_2 \to Rq_2,\ i = 1, \ldots, m.$$

Для написания программы машины **Сложение** приведем сначала упрощенные преобразования, позволяющие перерабатывать машинное слово $q_1 01^{x+1} 01^{y+1} 0$ в слово $q_0 01^{x+y+1} 000$:

$$q_1 01^{x+1} 01^{y+1} 0 \Rrightarrow 01^{x+1} 01^{y+1} q_\alpha 0 \Rrightarrow$$

$$\begin{cases} 01^{x+1} q_\beta 01^{y-1} 000 \Rrightarrow q_0 01^{x+y+1} 000, \text{ если } y \neq 0, \\ q_0 01^{x+1} 000, \text{ если } y = 0. \end{cases}$$

Программу, соответствующую указанной схеме, представим в виде следующей *таблицы команд*, в которой на пересечении строки с символом $a_i \in A$ и столбца с символом $q_j \in Q$, $j > 0$, имеется запись $*q_k$ $(* \in A \cup \{L, R\})$, содержащаяся в команде $a_iq_j \to *q_k$, а пустая ячейка соответствует произвольной записи $*q_k$ и для определенности будет означать $0q_0$:

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9
0	Rq_2	Rq_3	Lq_4	Lq_5	Lq_6 (где $y = 0$)	$0q_0$	Lq_8	$1q_9$	$0q_0$
1		Rq_2	Rq_3	$0q_4$	$0q_7$ (где $y \neq 0$)	Lq_6		Lq_8	Lq_9

Определим основные операции над машинами Тьюринга.

2. Композиция машин Тьюринга. Пусть Π — некоторая программа. Обозначим через $[\Pi]_{q_j}^{q_i}$ результат замены во всех командах из Π вхождений символов q_i на символы q_j.

Пусть $T_i = \langle A_i, Q_i, \Pi_i, q_0^{T_i}, q_1^{T_i} \rangle$, $i = 1, 2$, — машины Тьюринга с условиями $A_1 = A_2 = A$, $Q_1 \cap Q_2 = \varnothing$. Машина

$$T = \left\langle A, \left(Q_1 \setminus \{q_0^{T_1}\}\right) \cup Q_2, [\Pi_1]_{q_1^{T_2}}^{q_0^{T_1}} \cup \Pi_2, q_0^{T_2}, q_1^{T_1} \right\rangle$$

называется *композицией машин* T_1 *и* T_2 и обозначается через $T_1 \circ T_2$ или через T_1T_2.

Очевидно, для любого машинного слова M машины T_1 преобразование $M \Rightarrow^T M'$ означает, что найдется слово $M'' = \alpha q_0^{T_1} \beta$ с условиями $M \Rightarrow^{T_1} M''$ и $\alpha q_1^{T_2} \beta \Rightarrow^{T_2} M'$.

Условимся считать, что для машин T_1, T_2, T_3 запись $T_1T_2T_3$ означает композицию машин T_1T_2 и T_3, а запись $(T_1)^n$ совпадает с записью n машин $T_1T_1\ldots T_1$.

Следующие машины представляются в виде композиции машин из списка 1 – 10.

11. $\mathbf{Ц}_n$, $n \geq 2$ (*циклическая перестановка n аргументов*):

$$q_1 01^{x_1} 01^{x_2} 01^{x_3} \ldots 01^{x_n} 0 \Rrightarrow q_0 01^{x_n} 01^{x_1} 01^{x_2} \ldots 01^{x_{n-1}} 0.$$

12. $\mathbf{К}_n$, $n \geq 2$ (*удвоение n аргументов*):

$$q_1 01^{x_1} 01^{x_2} \ldots 01^{x_n} 0 \Rrightarrow q_0 01^{x_1} 01^{x_2} \ldots 01^{x_n} 01^{x_1} 01^{x_2} \ldots 01^{x_n} 0.$$

Машина $\mathbf{Ц}_2$ совпадает с машиной $\mathbf{В}$. Машина $\mathbf{Ц}_3$ соответствует следующей цепочке преобразований:

$$q_1 01^{x_1} 01^{x_2} 01^{x_3} 0 \Rrightarrow^{\mathbf{Б}^+} 01^{x_1} q_\alpha 01^{x_2} 01^{x_3} 0 \Rrightarrow^{\mathbf{Ц}_2}$$

$$01^{x_1} q_\beta 01^{x_3} 01^{x_2} 0 \Rrightarrow^{\mathbf{Б}^-} q_\gamma 01^{x_1} 01^{x_3} 01^{x_2} 0 \Rrightarrow^{\mathbf{Ц}_2}$$

$$q_0 01^{x_3} 01^{x_1} 01^{x_2} 0.$$

Таким образом, $\mathbf{Ц}_3 = \mathbf{Б}^+\mathbf{Ц}_2\mathbf{Б}^-\mathbf{Ц}_2$. В общем случае по индукции выводится равенство $\mathbf{Ц}_{n+1} = \mathbf{Б}^+\mathbf{Ц}_n\mathbf{Б}^-\mathbf{В}$.

Машина $\mathbf{К}_2$ представляется в виде следующей композиции:

$$\mathbf{К}_2 = \mathbf{Б}^+\mathbf{К}\mathbf{Б}^-(\mathbf{Ц}_3)^2(\mathbf{Б}^+)^2\mathbf{К}(\mathbf{Б}^-)^2\mathbf{Ц}_4(\mathbf{Б}^+)^2\mathbf{Ц}_2(\mathbf{Б}^-)^2,$$

а машина $\mathbf{К}_{n+1}$ выражается через $\mathbf{К}_n$ по следующей формуле:

$$\mathbf{К}_{n+1} = \mathbf{Б}^+\mathbf{К}_n\mathbf{Б}^-(\mathbf{Ц}_{2n+1})^{2n}(\mathbf{Б}^+)^{2n}\mathbf{К}(\mathbf{Б}^-)^{2n}\cdot$$

$$\mathbf{Ц}_{2n+2}(\mathbf{Б}^+)^{n+1}\mathbf{Ц}_{n+1}(\mathbf{Б}^-)^{n+1}.$$

3. Условные операторы. Пусть T_1, T_2 — машины Тьюринга, удовлетворяющие условиям предыдущего пункта. Определим машину T, содержащую начальное состояние q_1^T, заключительное состояние q_0^T и обладающую следующими свойствами для любых символов $a, b, c \in A$, слов $\alpha, \beta \in W(A)$, машинных слов M_i, содержащих заключительные состояния $q_0^{T_i}$, и машинных слов $[M_i]_{q_0^T}^{q_0^{T_i}}$, получающихся из M_i заменой $q_0^{T_i}$ на q_0^T, $i = 1, 2$:

а) если $abc = 010$ и $\alpha q_1^{T_1} abc\beta \Rightarrow^{T_1} M_1$, то $\alpha q_1^T abc\beta \Rightarrow^T [M_1]_{q_0^T}^{q_0^{T_1}}$;

б) если $abc \neq 010$ и $\alpha q_1^{T_2} abc\beta \Rightarrow^{T_2} M_2$, то $\alpha q_1^T abc\beta \Rightarrow^T [M_2]_{q_0^T}^{q_0^{T_2}}$.

Положим $T \rightleftharpoons \langle A, Q, \Pi, q_0^T, q_1^T \rangle$, где

$$Q = (Q_1 \setminus \{q_0^{T_1}\}) \cup (Q_2 \setminus \{q_0^{T_2}\}) \cup \{q_0^T, q_1^T, q_2^T, q_3^T, q_4^T, q_5^T\},$$

$$\Pi = [\Pi_1]_{q_0^T}^{q_0^{T_1}} \cup [\Pi_2]_{q_0^T}^{q_0^{T_2}} \cup \Pi',$$

Π' — программа, осуществляющая проверку условия $abc = 010$ и задание альтернатив по следующей таблице команд:

	q_1^T	q_2^T	q_3^T	q_4^T	q_5^T
0	Rq_2^T	$Lq_1^{T_2}$	Lq_4^T		$Lq_1^{T_2}$
1	$1q_1^{T_2}$	Rq_3^T	Lq_5^T	$Lq_1^{T_1}$	$Lq_1^{T_2}$
a_2	$a_2q_1^{T_2}$	$Lq_1^{T_2}$	Lq_5^T		$Lq_1^{T_2}$
...	...	...	...	...	...
a_m	$a_mq_1^{T_2}$	$Lq_1^{T_2}$	Lq_5^T		$Lq_1^{T_2}$

Отображение $E(\cdot, \cdot)$, ставящее в соответствие машинам T_1 и T_2 машину T, называется *условным оператором* и обозначается через $E(T_1, T_2)$ или $E \left\{ \begin{array}{l} T_1 \\ T_2 \end{array} \right.$.

Пусть $T_i = \langle A_i, Q_i, \Pi_i, q_0^{T_i}, q_1^{T_i} \rangle$, $i = 0, 1, 2$, — машины Тьюринга с условиями $A_0 = A_1 = A_2 = A$, $Q_0 \cap Q_1 = \varnothing$, $Q_0 \cap Q_2 = \varnothing$, $Q_1 \cap Q_1 = \varnothing$, $T = E(T_1, T_2)$, $T = \langle A, Q, \Pi, q_0^T, q_1^T \rangle$. *Условным оператором с циклом* называется оператор, ставящий в соответствие машинам T_0, T_1, T_2 машину $T^* = \left\langle A, Q^*, \Pi^*, q_0^T, q_1^{T_0} \right\rangle$ с множеством внутренних состояний $Q^* = (Q_0 \setminus \{q_0^{T_0}\}) \cup Q$ и программой $\Pi^* = [\Pi_0]_{q_1^T}^{q_0^{T_0}} \cup [\Pi_1]_{q_0^T}^{q_0^{T_1}} \cup [\Pi_2]_{q_1^{T_0}}^{q_0^{T_2}} \cup \Pi'$.

Значение T^* условного оператора с циклом от машин T_0, T_1, T_2 обозначается через $\dot{T}_0 E \left\{ \begin{array}{l} T_1 \\ \dot{T}_2 \end{array} \right.$.

Работа машины T^* с машинным словом M_0, содержащим начальное состояние $q_1^{T_0}$ машины T^*, состоит в следующем. Сначала машина T_0 перерабатывает слово M_0 в слово $\alpha q_1^T abc\beta$, соответствующее слову $\alpha q_0^{T_0} abc\beta$, на котором завершается работа машины T_0. Затем с помощью программы Π' проводится проверка равенства $abc = 010$. В случае равенства машина T_1 перерабатывает слово $\alpha q_1^T abc\beta$ в некоторое слово M_1, содержащее q_0^T, и завершает свою работу или работает неограниченное время.

Если же $abc \neq 010$, машина T_2 перерабатывает слово $\alpha q_1^T abc\beta$ в некоторое слово M_2, содержащее $q_1^{T_0}$, соответствующее слову, на котором завершается работа машины T_2, или работает неограниченное время. При получении слова M_2 снова применяется машина T_0 и процесс продолжается неограниченное время или после прохождения некоторого числа циклов останавливается после работы машины T_1.

4. Функции, вычислимые на машинах Тьюринга. Для любого натурального числа $n \in \mathbb{N}$ обозначим через $\overline{n}$ слово, состоящее из $n+1$ числа единиц: $11\ldots1$.

Функция $f : X \to \mathbb{N}$, где $X \subseteq \mathbb{N}^k$, называется *вычислимой на машине Тьюринга* $T_f = \left\langle A, Q, \Pi, q_0^{T_f}, q_1^{T_f} \right\rangle$, если выполняются следующие условия:

а) из $(n_1, \ldots, n_k) \in X$ следует

$$q_1^{T_f} 0\overline{n_1}0\overline{n_2}\ldots0\overline{n_k}0 \Rightarrow^{T_f} \alpha q_0^{T_f} 0\overline{f(n_1, n_2, \ldots, n_k)}0\beta;$$

б) из $(n_1, \ldots, n_k) \in \mathbb{N}^k \setminus X$ следует

$$q_1^{T_f} 0\overline{n_1}0\overline{n_2}\ldots0\overline{n_k}0 \Rightarrow^{\infty},$$

т.е. начиная со слова $q_1^{T_f} 0\overline{n_1}0\overline{n_2}\ldots0\overline{n_k}0$ машина T_f работает неограниченно долго, не приходя в заключительное состояние $q_0^{T_f}$.

Таким образом, вычислимость функции f на машине Тьюринга T_f означает, что по любому набору $(n_1, \ldots, n_k)$ из области определения δ_f (закодированному в виде наборов единиц длин $n_i + 1$, разделенных нулями) машина T_f выдает значение $f(n_1, \ldots, n_k)$ (закодированное в виде набора единиц длины $f(n_1, \ldots, n_k) + 1$), а для любого набора $(n_1, \ldots, n_k) \in \mathbb{N}^k \setminus X$ машина T_f не завершает работу за конечное время.

Функция f называется *правильно вычислимой на машине Тьюринга* $T_f = \left\langle A, Q, \Pi, q_0^{T_f}, q_1^{T_f} \right\rangle$, если выполняются следующие условия:

а) из $(n_1, \ldots, n_k) \in X$ следует

$$q_1^{T_f} 0\overline{n_1}0\overline{n_2}\ldots0\overline{n_k}0 \Rrightarrow^{T_f} q_0^{T_f} 0\overline{f(n_1, n_2, \ldots, n_k)}0\beta$$

и ячейка, напротив которой находится головка в начальный момент времени, совпадает с ячейкой, напротив которой находится головка в заключительный момент времени;

б) из $(n_1, \ldots, n_k) \in \mathbb{N}^k \setminus X$ следует

$$q_1^{T_f} 0\overline{n_1}0\overline{n_2}\ldots0\overline{n_k}0 \Rrightarrow^{\infty},$$

т.е. начиная со слова $q_1^{T_f}0\overline{n_1}0\overline{n_2}\ldots 0\overline{n_k}0$ машина T_f работает неограниченно долго, не приходя в заключительное состояние $q_0^{T_f}$, и при этом не достраиваются ячейки слева.

Функция f называется *правильно вычислимой* (сокращенно ПВФ), если f правильно вычислима на некоторой машине Тьюринга.

Преимущество правильной вычислимости по сравнению с обычной вычислимостью на машинах Тьюринга состоит в том, что левее ячейки, напротив которой стоит головка в начальном состоянии, можно записывать информацию, которая не изменяется в процессе "правильного вычисления".

П р и м е р 3.1.1. 1. 0-*Функция* $o : \mathbb{N} \to \mathbb{N}$, для которой $o(x) = 0$, $x \in \mathbb{N}$, правильно вычислима на машине **ЛS**.

2. *Функция следования* $s : \mathbb{N} \to \mathbb{N}$, для которой $s(x) = x + 1$, $x \in \mathbb{N}$, правильно вычислима на машине **S**.

3. n-*Местная функция проекции на* m-*ю координату* $I_m^n : \mathbb{N}^n \to \mathbb{N}$, для которой $I_m^n(x_1, \ldots, x_n) = x_m$, $x_1, \ldots, x_n \in \mathbb{N}$, $1 \le m \le n$ правильно вычислима на машине $\mathbf{Ц}_m(\mathbf{Б}^+)^{n-1}(\mathbf{ЛБ}^-)^{n-1}$.

4. Функции $+ : \mathbb{N}^2 \to \mathbb{N}$, $+(x, y) = x + y$, и $\cdot : \mathbb{N}^2 \to \mathbb{N}$, $\cdot(x, y) = x \cdot y$, правильно вычислимы на машинах **Сложение** и **Умножение** соответственно. При этом машина **Умножение** представима в виде следующей машины:

$$\mathbf{Б}^+E\left\{\begin{array}{l}\mathbf{Б}^-\mathbf{В}\mathbf{Б}^+\mathbf{ЛБ}^- \\ \mathbf{RБ}^-\mathbf{В}\mathbf{Б}^+\mathbf{КБ}^-\mathbf{В}\mathbf{Б}^+\dot{E}\end{array}\right.$$

$$\left\{\begin{array}{l}\mathbf{Б}^-\mathbf{Ц}_3(\mathbf{Б}^+)^2(\mathbf{ЛБ}^-)^2 \\ \mathbf{RБ}^+\mathrm{A}(\mathbf{Б}^-)^2\mathbf{Ц}_3^2(\mathbf{Б}^+)^2\mathbf{К}(\mathbf{Б}^-)^2\mathbf{Ц}_3(\mathbf{Б}^+)^2\,\mathbf{Сложение}\,\dot{\mathbf{Б}}^-\end{array}\right. .$$

5. *Функция усеченной разности* $\dot{-} : \mathbb{N}^2 \to \mathbb{N}$, где

$$x\dot{-}y = \left\{\begin{array}{r} 0, \text{ если } x \le y, \\ x - y, \text{ если } x > y,\end{array}\right.$$

правильно вычислима на машине

$$\mathbf{Б}^+\dot{E}\left\{\begin{array}{l}\mathbf{ЛБ}^- \\ \mathbf{RБ}^-E\left\{\begin{array}{l}\mathbf{Б}^+\mathbf{ЛБ}^- \\ \mathbf{RБ}^+\dot{\mathbf{А}}\end{array}\right.\end{array}\right. . \ \square$$

Теорема 3.1.1. *Если функции* $f_1(x_1, \ldots, x_n)$, $\ldots$, $f_m(x_1, \ldots, x_n)$ *и* $g(y_1, \ldots, y_m)$ *правильно вычислимы, то функция*

$$h(x_1, \ldots, x_n) = g(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n))$$

правильно вычислима.

Д о к а з а т е л ь с т в о. Для простоты рассмотрим случай $m = 2$. Пусть T_{f_1}, T_{f_2}, T_g — машины Тьюринга, правильно вычисляющие функции f_1, f_2 и g соответственно. В качестве машины T_h для правильного вычисления функции h годится машина

$$\mathbf{К}_n(\mathbf{Б}^+)^n T_{f_1}(\mathbf{Б}^-)^n \mathbf{Ц}_{n+1}\mathbf{Б}^+ T_{f_2}\mathbf{Б}^- T_g. \ \square$$

Таким образом, любая суперпозиция правильно вычислимых функций правильно вычислима, и список правильно вычислимых функций, представленных в примере 3.1.1, существенно расширяется.

§ 3.2. Рекурсивные функции и отношения

1. Примитивно рекурсивные функции. Рассмотрим сначала *всюду определенные* числовые функции $f : \mathbb{N}^k \to \mathbb{N}$, $k \in \mathbb{N} \setminus \{0\}$, т.е не нуль-местные операции на множестве $\mathbb{N}$, и определим понятие примитивно рекурсивной функции (сокращенно ПРФ).

Функции $o(x)$, $s(x)$, I_m^n, $1 \leq m \leq n$, называются *простейшими примитивно рекурсивными функциями*.

Пусть f — m-местная операция, $g_1, \ldots, g_m$ — n-местные операции на множестве $\mathbb{N}$. Оператор S, ставящий в соответствие операциям f и $g_1, \ldots, g_m$ n-местную операцию h, удовлетворяющую тождеству

$$h(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)),$$

называется *оператором суперпозиции* (*подстановки*). При этом всюду определенная функция

$$h \rightleftharpoons S(f, g_1, \ldots, g_m)$$

является, очевидно, суперпозицией функций f и $g_1, \ldots, g_m$.

Оператор примитивной рекурсии R каждой $(n+2)$-местной операции f и n-местной операции g на множестве $\mathbb{N}$ ставит в соответствие $(n+1)$-местную операцию

$$h \rightleftharpoons R(f, g),$$

удовлетворяющую следующей *схеме примитивной рекурсии*:

$$\begin{cases} h(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n), \\ h(x_1, \ldots, x_n, y+1) = f(x_1, \ldots, x_n, y, h(x_1, \ldots, x_n, y)). \end{cases}$$

Для $n = 0$ схема примитивной рекурсии имеет следующий вид:

$$\begin{cases} h(0) = a, \\ h(y+1) = f(y, h(y)), \end{cases}$$

где a — постоянная одноместная функция, равная числу a.

Схема примитивной рекурсии образует некоторый индукционный процесс построения функции h, при котором на нулевом шаге используется функция g, а на каждом последующем шаге — значение функции f от аргументов $x_1, \ldots, x_n$, номера y предыдущего шага и значения функции h, вычисленного на предыдущем шаге.

Функция f называется *примитивно рекурсивной*, если существует последовательность функций $f_0, \ldots, f_n$, в которой $f_n = f$ и всякая функция f_i является простейшей ПРФ или получается из предыдущих функций с помощью оператора суперпозиции S или оператора примитивной рекурсии R.

П р и м е р 3.2.1. 1. Функция *сложения* $x + y$ примитивно рекурсивна в силу схемы примитивной рекурсии

$$\begin{cases} x + 0 = I_1^1(x), \\ x + (y + 1) = s(x + y). \end{cases}$$

2. Схема примитивной рекурсии

$$\begin{cases} x \cdot 0 = o(x), \\ x \cdot (y + 1) = x \cdot y + x \end{cases}$$

обосновывает примитивную рекурсивность функции *умножения* $x \cdot y$.

Доказательство примитивной рекурсивности следующих функций оставляется читателю в качестве упражнения.

3. x^y, где $0^0 = 1$, — функция *возведения в степень*.
4. $x!$, где $0! = 1$, — *x-факториал*.
5. $\operatorname{sg}(x) = \begin{cases} 0, & \text{если } x = 0, \\ 1, & \text{если } x > 0, \end{cases}$ — *знак числа x*.
6. $\overline{\operatorname{sg}}(x) = \begin{cases} 1, & \text{если } x = 0, \\ 0, & \text{если } x > 0, \end{cases}$ — *дополнение знака числа x*.
7. $x \dot{-} 1 = \begin{cases} 0, & \text{если } x = 0, \\ x - 1, & \text{если } x > 0, \end{cases}$ — *усеченное вычитание единицы*.
8. $x \dot{-} y = \begin{cases} 0, & \text{если } x \le y, \\ x - y, & \text{если } x > y, \end{cases}$ — *усеченная разность*.
9. $|x - y|$ — *модуль разности*.
10. $\max(x, y)$ — *максимум чисел x и y*.
11. $\min(x, y)$ — *минимум чисел x и y*.
12. $[x/y]$, где $[x/0] = x$, — *частное от деления x* на y.
13. $\operatorname{rest}(x, y)$, где $\operatorname{rest}(x, 0) = x$, — *остаток от деления x* на y.
14. p_x — *x-е простое число* ($p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $\ldots$).

15. $\mathrm{ex}(x, y)$, где $\mathrm{ex}(0, y) = 0$, — *показатель степени y-го простого числа* p_y в разложении числа x на неприводимые сомножители (т.е. если $x = p_0^{n_0} \cdot p_1^{n_1} \cdot \ldots \cdot p_k^{n_k}$, то $\mathrm{ex}(x, y) = n_y$).

16. $\mathrm{long}(x)$ — *номер наибольшего простого делителя* числа x.

Пусть $f(x_1, \ldots, x_n, i)$ — $(n+1)$-местная операция на множестве $\mathbb{N}$. Определим операторы суммирования и произведения g_1, g_2, g_3, g_4:

$$g_1(x_1, \ldots, x_n, y) = \sum_{i=0}^{y} f(x_1, \ldots, x_n, i),$$

$$g_2(x_1, \ldots, x_n, y) = \prod_{i=0}^{y} f(x_1, \ldots, x_n, i),$$

$$g_3(x_1, \ldots, x_n, y, z) = \sum_{i=y}^{z} f(x_1, \ldots, x_n, i) =$$

$$\begin{cases} f(x_1, \ldots, x_n, y) + \ldots + f(x_1, \ldots, x_n, z), & \text{если } y \le z, \\ 0, & \text{если } y > z, \end{cases}$$

$$g_4(x_1, \ldots, x_n, y, z) = \prod_{i=y}^{z} f(x_1, \ldots, x_n, i) =$$

$$\begin{cases} f(x_1, \ldots, x_n, y) \cdot \ldots \cdot f(x_1, \ldots, x_n, z), & \text{если } y \le z, \\ 1, & \text{если } y > z. \end{cases}$$

Предложение 3.2.1. *Если f — примитивно рекурсивная функция, то функции g_1, g_2, g_3, g_4 примитивно рекурсивны.*

2. Примитивно рекурсивные отношения. Отношение $P \subseteq \mathbb{N}^k$ называется *примитивно рекурсивным* (сокращенно ПРО), если примитивно рекурсивна *характеристическая функция*

$$\chi_P(\overline{x}) = \begin{cases} 1, & \text{если } \overline{x} \in P, \\ 0, & \text{если } \overline{x} \notin P, \end{cases}$$

где $\overline{x} = (x_1, \ldots, x_k)$.

П р и м е р 3.2.2. Отношения равенства $x = y$ и делимости $x|y$ (x делит y) примитивно рекурсивны. Действительно, характеристические функции $\chi_=(x, y) = \overline{\mathrm{sg}}(|x - y|)$ и $\chi_|(x, y) = \overline{\mathrm{sg}}(\mathrm{rest}(y, x))$ примитивно рекурсивны. □

Покажем, что теоретико-множественные операции над ПРО сохраняют примитивную рекурсивность.

Пусть $P, Q \subseteq \mathbb{N}^k$ — некоторые отношения. Положим $P \wedge Q \rightleftharpoons P \cap Q$, $P \vee Q \rightleftharpoons P \cup Q$, $\neg P \rightleftharpoons \mathbb{N}^k \setminus P$, $P \Rightarrow Q \rightleftharpoons \neg P \vee Q$.

Предложение 3.2.2. *Если $P, Q \subseteq \mathbb{N}^k$ и $R \subseteq \mathbb{N}^l$ — примитивно рекурсивные отношения, то отношения $P \wedge Q$, $P \vee Q$, $\neg P$, $P \Rightarrow Q$ и $P \times R$ примитивно рекурсивны.*

Доказательство. Поскольку по условию примитивно рекурсивны функции $\chi_P(\overline{x})$, $\chi_Q(\overline{x})$, $\chi_R(\overline{y})$, примитивно рекурсивными являются также характеристические функции $\chi_{P \wedge Q}(\overline{x}) = \chi_P(\overline{x}) \cdot \chi_Q(\overline{x})$, $\chi_{\neg P}(\overline{x}) = \overline{\mathrm{sg}}(\chi_P(\overline{x}))$, $\chi_{P \times R}(\overline{x}, \overline{y}) = \chi_P(\overline{x}) \cdot \chi_R(\overline{y})$. Теперь из соотношений $P \vee Q = \neg(\neg P \wedge \neg Q)$ и $P \Rightarrow Q = \neg(P \wedge \neg Q)$ вытекает примитивная рекурсивность характеристических функций $\chi_{P \vee Q}(\overline{x})$ и $\chi_{P \Rightarrow Q}(\overline{x})$. $\square$

Записи $(\forall z \leq y)$ и $(\exists z \leq y)$ называются *ограниченными кванторами*. Пусть $P \subseteq \mathbb{N}^{k+1}$ — некоторое отношение. Обозначим через $(\forall z \leq y)\, P(x_1, \ldots, x_k, z)$ $(k+1)$-местное отношение

$$\{(a_1, \ldots, a_k, b) \mid (a_1, \ldots, a_k, c) \in P \text{ для любого } c \leq b\},$$

а через $(\exists z \leq y)\, P(x_1, \ldots, x_k, z)$ — $(k+1)$-местное отношение

$$\{(a_1, \ldots, a_k, b) \mid (a_1, \ldots, a_k, c) \in P \text{ для некоторого } c \leq b\}.$$

Следующее утверждение показывает, что примитивная рекурсивность отношений сохраняется при навешивании ограниченных кванторов.

Предложение 3.2.3. *Если $P \subseteq \mathbb{N}^{k+1}$ — примитивно рекурсивное отношение, то отношения $Q_1 \rightleftharpoons (\forall z \leq y)\, P(x_1, \ldots, x_k, z)$ и $Q_2 \rightleftharpoons (\exists z \leq y)\, P(x_1, \ldots, x_k, z)$ примитивно рекурсивны.*

Доказательство. Примитивная рекурсивность отношений Q_1 и Q_2 вытекает из следующих соотношений:

$$\chi_{Q_1}(\overline{x}, y) = \prod_{i=0}^{y} \chi_P(\overline{x}, i),$$

$$\chi_{Q_2}(\overline{x}, y) = \mathrm{sg}\left(\sum_{i=0}^{y} \chi_P(\overline{x}, i)\right),$$

где $\overline{x} = (x_1, \ldots, x_k)$. $\square$

П р и м е р 3.2.3. 1. Поскольку справедливо соотношение $x \le y \Leftrightarrow (\exists z \le y)\,(x + z = y)$, отношение $x \le y$ примитивно рекурсивно в силу предложения 3.2.3.

2. Аналогично из $x < y \Leftrightarrow (\exists z \le y)\,(s(x + z) = y)$ вытекает примитивная рекурсивность отношения $x < y$.

3. Множество простых чисел совпадает с примитивно рекурсивным отношением $(x \ge 2) \wedge (\forall z \le x)\ (z|x \Rightarrow ((z = x) \vee (z = 1)))$. $\square$

Пусть $\alpha(\overline{x})$ — некоторая k-местная операция на множестве $\mathbb{N}$, $P \subseteq \mathbb{N}^{k+1}$. Обозначим через $(\mu y \le \alpha(\overline{x}))P(\overline{x}, y)$ k-местную операцию, значение которой на наборе $\overline{x}$ равно

$$\begin{cases} \min\{y \mid (y \le \alpha(\overline{x})) \wedge P(\overline{x}, y)\}, & \text{если } P(\overline{x}, y) \\ & \text{для некоторого } y \le \alpha(\overline{x}), \\ \alpha(\overline{x}) + 1 & \text{в противном случае.} \end{cases}$$

Оператор, ставящий в соответствие каждой операции $\alpha \subseteq \mathbb{N}^{k+1}$ и отношению $P \subseteq \mathbb{N}^{k+1}$ операцию $\mu(\alpha, P) \rightleftharpoons (\mu y \le \alpha(\overline{x}))P(\overline{x}, y)$, называется *оператором ограниченной минимизации*.

Предложение 3.2.4. *Если α — ПРФ, $P \subseteq \mathbb{N}^{k+1}$ — ПРО, то $\mu(\alpha, P)$ — ПРФ.*

Д о к а з а т е л ь с т в о. Рассмотрим примитивно рекурсивную функцию

$$G(\overline{x}, z) \rightleftharpoons \sum_{y=0}^{z} \left(\prod_{i=0}^{y} \overline{\mathrm{sg}}\,(\chi_P(\overline{x}, i)) \right).$$

Примитивная рекурсивность функции $\mu(\alpha, P)$ вытекает из следующих соотношений:

$$\mu(\alpha, P)(\overline{x}) = \sum_{y=0}^{\alpha(\overline{x})} \left(\prod_{i=0}^{y} \overline{\mathrm{sg}}\,(\chi_P(\overline{x}, i)) \right) = G(\overline{x}, \alpha(\overline{x})). \ \square$$

П р и м е р 3.2.4. Функция $[\sqrt{x}]$, которая каждому натуральному числу x ставит в соответствие целую часть от квадратного корня из x, примитивно рекурсивна в силу предложения 4.2.4, поскольку $[\sqrt{x}] = (\mu y \le x)\left((y + 1)^2 > x\right)$. $\square$

3. Нумерации кортежей натуральных чисел. *Нумерацией множества X* называется любая сюръекция $\nu : \mathbb{N} \xrightarrow{\text{на}} X$. В предложении 1.4.3* представлена "диагональная" нумерация множества $\mathbb{N}^2$, устанавливающая биекцию между множествами $\mathbb{N}$ и $\mathbb{N}^2$. Отображение

$c : \mathbb{N}^2 \to \mathbb{N}$, ставящее в соответствие каждой паре натуральных чисел (x, y) ее "диагональный" номер $c(x, y)$, называется *канторовской нумерующей функцией*. По номеру $n = c(x, y)$ однозначно определяются его *левая координата* $l(n) \rightleftharpoons x$ и *правая координата* $r(n) \rightleftharpoons y$.

Функции c, l и r удовлетворяют следующим тождествам:

$$c(l(n), r(n)) = n, \quad l(c(x, y)) = x, \quad r(c(x, y)) = y.$$

Примитивная рекурсивность функций c, l и r вытекает из следующих равенств:

$$c(x, y) = \left[\frac{(x + y) \cdot (x + y + 1)}{2}\right] + x,$$

$$l(n) = n \dot{-} \left[\frac{\left(\left[\frac{\left[\sqrt{8n+1}\right]+1}{2}\right] \dot{-} 1\right) \cdot \left[\frac{\left[\sqrt{8n+1}\right]+1}{2}\right]}{2}\right],$$

$$r(n) = \left[\frac{\left[\sqrt{8n+1}\right] + 1}{2}\right] \dot{-} (l(n) + 1).$$

Используя канторовскую функцию c, определяем последовательность примитивно рекурсивных функций $c^1, c^2, \ldots, c^k, \ldots$ такую, что c^k — k-местная функция, осуществляющая взаимно однозначное отображение $\mathbb{N}^k$ на $\mathbb{N}$:

$$\begin{aligned}
&c^1(x_1) \rightleftharpoons x_1,\\
&c^2(x_1, x_2) \rightleftharpoons c(x_1, x_2),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&c^{k+1}(x_1, \ldots, x_k, x_{k+1}) \rightleftharpoons c(c^k(x_1, \ldots, x_k), x_{k+1}),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots
\end{aligned}$$

Координаты $x_1, \ldots, x_k, x_{k+1}$, для которых $c^{k+1}(x_1, \ldots, x_k, x_{k+1}) = n$, определяются соответствующими одноместными примитивно рекурсивными функциями $c_1^{k+1}, \ldots, c_k^{k+1}, c_{k+1}^{k+1}$, $k \in \mathbb{N}$, для которых

$$\begin{aligned}
&c_1^1(n) = n,\\
&c_1^2(n) = l(n), \quad c_2^2(n) = r(n),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&c_1^{k+1}(n) = c_1^k(l(n)), \quad , c_k^{k+1}(n) = c_k^k(l(n)), \quad c_{k+1}^{k+1}(n) = r(n),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots
\end{aligned}$$

Функция c^k называется *k-сверткой*, а набор $(c_1^k, \ldots, c_k^k)$ из k одноместных функций — *k-развертокой*. Очевидно, свертки и развертки удовлетворяют следующим тождествам:

$$c^k(c_1^k(n), \ldots, c_k^k(n)) \approx n, \quad c_i^k(c^k(x_1, \ldots, x_k)) \approx x_i, \quad i = 1, \ldots, k.$$

4. Частично рекурсивные функции. Будем говорить, что функция $f(\overline{x})$ получается из функций $g(\overline{x}, y)$ и $h(\overline{x}, y)$ с помощью *оператора минимизации* M, и обозначать

$$f(\overline{x}) = \mu y \, (g(\overline{x}, y) \leq h(\overline{x}, y))$$

или $f(\overline{x}) = M(g(\overline{x}, y), h(\overline{x}, y))$, если выполнено следующее условие: $f(\overline{x})$ определено и равно y тогда и только тогда, когда определены $g(\overline{x}, 0), h(\overline{x}, 0), \ldots, g(\overline{x}, y), h(\overline{x}, y)$ и справедливы соотношения $g(\overline{x}, i) > h(\overline{x}, i)$, $i = 0, \ldots, y-1$, $g(\overline{x}, y) \leq h(\overline{x}, y)$.

Функция $f : X \to \mathbb{N}^k$ называется *частично рекурсивной* (ЧРФ), если она может быть получена из простейших ПРФ с помощью конечного числа применений операторов S, R и M. Частично рекурсивная функция называется *рекурсивной* (РФ), если она всюду определена.

Обозначим через $M^o(g, h)$ значение $M(g, h)$, если функции g, h и $M(g, h)$ всюду определены.

Функция $f : X \to \mathbb{N}^k$ называется *общерекурсивной* (ОРФ), если она может быть получена из простейших ПРФ с помощью конечного числа применений операторов S, R и M^o.

Класс ЧРФ строго содержит класс РФ, поскольку, например, нигде не определенная функция $f(x) = \mu y \, (s(x) \leq x)$ частично рекурсивна.

Класс РФ совпадает с классом ОРФ. Включение ОРФ $\subseteq$ РФ очевидно. Обратное включение будет вытекать из теоремы о нормальной форме (теорема 4.3.7).

Класс ОРФ строго содержит класс ПРФ. В качестве примеров общерекурсивных, но не примитивно рекурсивных функций можно взять так называемые *быстрорастущие* функции, т.е. такие общерекурсивные функции $f(x)$, что для каждой ПРФ $g(x)$ существует число a с условием $g(x) < f(x)$ при $x \geq a$.

Рассмотрим функцию $B(n, x)$, удовлетворяющую следующим тождествам:

$$B(0, x) = x + 2, \quad B(n+1, 0) = \mathrm{sg}(n),$$
$$B(n+1, x+1) = B(n, B(n+1, x)).$$

Быстроту роста значений функции $B(n,x)$ можно проследить, например, на основе следующих соотношений:

$$B(3,0)=1,\quad B(3,1)=2,\quad B(3,2)=2^2,\quad B(3,3)=2^{2^2},\ldots$$

Функции $B(n,x)$ при фиксированных n называются *функциями Аккермана*, а функция $A(x) \rightleftharpoons B(x,x)$ — *диагональной функцией Аккермана*.

Теорема 3.2.5. (теорема Аккермана). *Функция $A(x)$ является рекурсивной быстрорастущей функцией.* □

Следующая теорема показывает, что при построении частично рекурсивных функций вместо оператора примитивной рекурсии можно использовать арифметические операции.

Теорема 3.2.6. (теорема об элиминации примитивной рекурсии). *Любая ЧРФ может быть получена из простейших ПРФ и функций $+$ и $\cdot$ с помощью операторов S и M.* □

§ 3.3. Эквивалентность моделей алгоритмов

В предыдущих двух параграфах мы определили два класса вычислимых функций: класс ПВФ функций, правильно вычислимых на машинах Тьюринга, и класс ЧРФ частично рекурсивных функций. В этом параграфе устанавливается совпадение этих классов, на основе которого делается вывод о совпадении класса интуитивно вычислимых функций с классом вычислимых функций в любой из приведенных выше алгоритмических моделей.

1. Правильная вычислимость частично рекурсивных функций. По теореме об элиминации примитивной рекурсии доказательство правильной вычислимости для ЧРФ сводится к проверке правильной вычислимости функций o, s, I_m^n, $+$, $\cdot$, а также к проверке сохранения правильной вычислимости при переходе к результатам действия операторов суперпозиции и минимизации.

Правильная вычислимость вышеперечисленных функций установлена в примере 3.1.1. Сохранение правильной вычислимости при переходе к суперпозиции функций показано в теореме 3.1.1. Таким образом, импликация ЧРФ $\Rightarrow$ ПВФ сводится к доказательству следующей теоремы.

Теорема 3.3.1. *Если функции $g(x_1,\ldots,x_n,y)$ и $h(x_1,\ldots,x_n,y)$ правильно вычислимы, то функция*

$$f(x_1,\ldots,x_n)=\mu y\,(g(x_1,\ldots,x_n,y)\le h(x_1,\ldots,x_n,y))$$

правильно вычислима.

Д о к а з а т е л ь с т в о. Пусть T_g и T_h – машины, правильно вычисляющие функции g и h соответственно. Представим машину T_f, правильно вычисляющую функцию f, в виде следующей схемы преобразований:

$$
\begin{array}{l}
q_1 0\overline{x_1}0\overline{x_2}\ldots 0\overline{x_n}0 \Rrightarrow^{(\mathbf{Б}^+)^n\overset{\bullet}{\mathbf{S}}} \\
0\overline{x_1}0\overline{x_2}\ldots 0\overline{x_n}q_\alpha 0\overline{0}0 \Rrightarrow^{(\mathbf{Б}^-)^n} \\
q_\beta 0\overline{x_1}0\overline{x_2}\ldots 0\overline{x_n}0\overline{0}0 \Rrightarrow^{\mathbf{К}_{n+1}} \\
q_\gamma 0\overline{x_1}0\overline{x_2}\ldots 0\overline{x_n}0\overline{0}0\overline{x_1}0\overline{x_2}\ldots 0\overline{x_n}0\overline{0}0 \Rrightarrow^{(\mathbf{Б}^+)^{n+1}T_g(\mathbf{Б}^-)^{n+1}} \\
q_\delta 0\overline{x_1}0\overline{x_2}\ldots 0\overline{x_n}0\overline{0}0\overline{g(x_1,\ldots,x_n,0)}0 \Rrightarrow^{\mathbf{Ц}_{n+2}\mathbf{Б}^+\mathbf{К}^{n+1}(\mathbf{Б}^+)^{n+1}T_h} \\
0\overline{g(x_1,\ldots,x_n,0)}0\overline{x_1}0\overline{x_2}\ldots 0\overline{x_n}0\overline{0}q_\varepsilon 0\overline{h(x_1,\ldots,x_n,0)}0 \Rrightarrow \\
\Rrightarrow^{(\mathbf{Б}^-)^{n+2}(\mathbf{Ц}_{n+2})^{n+1}(\mathbf{Б}^+)^{n+1}} \\
0\overline{x_1}0\overline{x_2}\ldots 0\overline{x_n}0\overline{0}q_\zeta 0\overline{g(x_1,\ldots,x_n,0)}0\overline{h(x_1,\ldots,x_n,0)}0 \Rrightarrow^{T_{\dot{-}}}
\end{array}
$$

$$
0\overline{x_1}0\overline{x_2}\ldots 0\overline{x_n}0\overline{0}q_\eta 0\overline{g(x_1,\ldots,x_n,0)\dot{-}h(x_1,\ldots,x_n,0)}0 \Rrightarrow
$$

$$
\begin{cases}
q_0 0\overline{y}0, & \text{если } g(x_1,\ldots,x_n,0)\dot{-}h(x_1,\ldots,x_n,0)=0, \\
\mathbf{Л}\overset{\bullet}{\mathbf{Б}}^-, & \text{если } g(x_1,\ldots,x_n,0)\dot{-}h(x_1,\ldots,x_n,0)>0.
\end{cases}
$$

Здесь для наглядности в условном операторе с циклом использован символ $\bullet$ вместо $\dot{}$.

Таким образом, машина T_f равна

$$
\begin{array}{c}
(\mathbf{Б}^+)^n\,\overset{\bullet}{\mathbf{S}}\,(\mathbf{Б}^-)^n\mathbf{К}_{n+1}(\mathbf{Б}^+)^{n+1}T_g(\mathbf{Б}^-)^{n+1}\mathbf{Ц}_{n+2} \\
\mathbf{Б}^+\mathbf{К}^{n+1}(\mathbf{Б}^+)^{n+1}T_h(\mathbf{Б}^-)^{n+2}(\mathbf{Ц}_{n+2})^{n+1}(\mathbf{Б}^+)^{n+1}T_{\dot{-}} \\
E\begin{cases}
(\mathbf{Б}^-)^{n+1}\mathbf{Ц}_{n+1}(\mathbf{Б}^+)^{n+1}(\mathbf{ЛБ}^-)^{n+1} \\
\mathbf{Л}\overset{\bullet}{\mathbf{Б}}^-
\end{cases}. \quad \square
\end{array}
$$

2. Частичная рекурсивность правильно вычислимых функций. Определим числовую кодировку машин Тьюринга, которая позволит по коду машины Тьюринга T_f, правильно вычисляющей функцию f, построить частично рекурсивную схему для вычисления функции f.

Пусть $A=\{a_m \mid m\in\mathbb{N}\}$ — некоторый счетный алфавит, $W(A)$ — множество всех слов алфавита A, включающее пустое слово Λ. Определим *кодирующую функцию* код : $W(A)\to\mathbb{N}$ для множества $W(A)$ по следующим правилам:

$$
\text{код}(\Lambda)=2^1,\quad \text{код}(a_{i_1}a_{i_2}\ldots a_{i_k})=2^1\cdot p_1^{i_1+1}\cdot p_2^{i_2+1}\cdot\ldots\cdot p_k^{i_k+1}.
$$

Лемма 3.3.2. *Множество* код$(W(A)) \rightleftharpoons \{m \mid m = \text{код}(w)$ *для некоторого слова* $w \in W(A)\}$ *является примитивно рекурсивным отношением.*

Рассмотрим теперь алфавит A вместе со счетным множеством $Q = \{q_n \mid n \in \mathbb{N}\}$ внутренних состояний и расширим функцию код на множество всех машинных слов $\alpha q_j a_i \beta$, полагая

$$\text{код}(\alpha q_j a_i \beta) \rightleftharpoons 2^2 \cdot 3^j \cdot 5^i \cdot 7^{\text{код}(\alpha)} \cdot 11^{\text{код}(\beta)}.$$

Лемма 3.3.3. *Множество* M_0 *всех кодов машинных слов является примитивно рекурсивным отношением.* □

Снова расширим функцию код теперь на множество машинных команд $a_i q_j \to d q_k$, где $d \in \{L, R\} \cup A$, полагая

$$\text{код}(a_i q_j \to d q_k) \rightleftharpoons p_{c(i,j)+2}^{2^{\text{код}(d)} \cdot 3^k},$$

$$\text{код}(d) \rightleftharpoons \begin{cases} 0, & \text{если } d = L, \\ 1, & \text{если } d = R, \\ m+2, & \text{если } d = a_m. \end{cases}$$

Лемма 3.3.4. *Множество* K_0 *всех кодов машинных команд является примитивно рекурсивным отношением.*

Последнее расширение функции код позволяет примитивно рекурсивно закодировать все машины Тьюринга. Для машины

$$T = \langle \{a_0, \ldots, a_m\}, \{q_0, \ldots, q_n\}, \{T(i,j) \mid 0 \le i \le m, 1 \le j \le n\}, q_0, q_1 \rangle$$

положим

$$\text{код}(T) \rightleftharpoons 2^3 \cdot 3^m \cdot 5^n \cdot \prod_{i=0}^{m} \prod_{j=1}^{n} \text{код}(T(i,j)).$$

Лемма 3.3.5. *Множество* T_0 *всех кодов машин Тьюринга является примитивно рекурсивным отношением.* □

Теорема 3.3.6. *Следующие отношения примитивно рекурсивны:*

(0) *отношение* T_0*, состоящее из всех наборов* (n, x, y, t) *таких, что машина Тьюринга с кодом* n*, начиная работу на слове с кодом* x*, заканчивает свою работу на слове с кодом* y *в состоянии* q_0 *не более чем за* t *шагов;*

(k) *отношение* T_k*,* $k \ge 1$*, состоящее из всех наборов* $(n, x_1, \ldots, x_k, y, t)$ *таких, что машина Тьюринга с кодом* n*, начиная работу на слове* $q_1 0 \overline{x_1} 0 \overline{x_2} 0 \ldots 0 \overline{x_k}$*, заканчивает свою работу не более чем за* $l(t)$ *шагов на слове* $\alpha q_0 0 \overline{y} 0 \beta$*, где* $c(\text{код}(\alpha), \text{код}(\beta)) = r(t)$.

Теорема 3.3.7. (теорема Клини о нормальной форме). *Если функция $f(x_1, \ldots, x_k)$ вычислима на некоторой машине Тьюринга, то существует номер $n \in \mathbb{N}$, для которого*

$$f(x_1, \ldots, x_k) = l(\mu t\, (1 \leq \chi_{T_k}(n, x_1, \ldots, x_k, l(t), r(t)))).$$

.

Д о к а з а т е л ь с т в о. Пусть n — код машины T_f, вычисляющей функцию f. Если $f(x_1, \ldots, x_k) = y$, то

$$q_1 0\overline{x_1} 0\overline{x_2} 0 \ldots 0\overline{x_k} \Rightarrow^{T_f} \alpha q_0 0\overline{y} 0\beta$$

и переход из состояния q_1 в состояние q_0 (т.е вычисление значения $f(x_1, \ldots, x_k)$) происходит за некоторое t_0 число шагов. Тогда для числа $t = c(y, c(t_0, c(\text{код}(\alpha), \text{код}(\beta))))$ выполняется отношение $T_k(n, x_1, \ldots, x_k, l(t), r(t))$. Взяв наименьшее число t_1 с условием $(n, x_1, \ldots, x_k, l(t_1), r(t_1)) \in T_k$, получаем

$$l(\mu t\, (1 \leq \chi_{T_k}(n, x_1, \ldots, x_k, l(t), r(t)))) = l(t_1) = y.$$

Предположим теперь, что значение $f(x_1, \ldots, x_k)$ не определено. Тогда машина T_f, начиная со слова $q_1 0\overline{x_1} 0\overline{x_2} 0 \ldots 0\overline{x_k}$, работает неограниченно долго и не приходит в состояние q_0. Тогда $\chi_{T_k}(n, x_1, \ldots, x_k, l(t), r(t)) = 0$ для любого t и значение $l(\mu t\, (1 \leq \chi_{T_k}(n, x_1, \ldots, x_k, l(t), r(t))))$ также не определено. □

Следствие 3.3.8. *Для любой функции $f : X \to \mathbb{N}$, где $X \subseteq \mathbb{N}^k$, следующие условия эквивалентны:*

(1) *функция f вычислима на некоторой машине Тьюринга;*

(2) *функция f правильно вычислима на некоторой машине Тьюринга;*

(3) *функция f частично рекурсивна.*

Д о к а з а т е л ь с т в о. Импликация (1) ⇒ (3) вытекает из теоремы Клини о нормальной форме. Импликация (3) ⇒ (2) установлена в п. 1 настоящего параграфа, а импликация (2) ⇒ (1) очевидна. □

Из теоремы Клини о нормальной форме и следствия 4.3.8 вытекает, что любую ЧРФ $f(x_1, \ldots, x_k)$ можно представить в виде функции

$$U_k(n, x_1, \ldots, x_k) \rightleftharpoons l(\mu t\, (1 \leq \chi_{T_k}(n, x_1, \ldots, x_k, l(t), r(t)))),$$

где оператор минимизации используется только однажды, а значение n фиксировано. Таким образом, справедливо

Следствие 3.3.9. *Класс ОРФ совпадает с классом РФ.*

§ 3.4. Универсальные частично рекурсивные функции. Теорема Райса

Пусть K — некоторый класс k-местных функций $f : X \to \mathbb{N}$, $X \subseteq \mathbb{N}^k$. $(k+1)$-Местная функция $f(n, x_1, \ldots, x_k)$ называется *универсальной для класса* K, если выполняются следующие условия:

1) для любого фиксированного числа $n_0 \in \mathbb{N}$ k-местная функция $f(n_0, x_1, \ldots, x_k)$ принадлежит классу K;

2) для любой функции $g \in K$ существует такое число $n_0 \in \mathbb{N}$ (называемое *f-номером* функции g), что $g(x_1, \ldots, x_k) = f(n_0, x_1, \ldots, x_k)$.

Следующая теорема показывает, что функция $U_1(n, x_1)$ универсальна для класса всех одноместных ЧРФ и для любого $k \geq 1$ порождает универсальные функции U^k для классов всех k-местных ЧРФ.

Теорема 3.4.1. (теорема об универсальности). *Для любого натурального числа $k \geq 1$ функция $U^k(n, x_1, \ldots, x_k) \rightleftharpoons U_1(n, c^k(x_1, \ldots, x_k))$ (где c^k — k-свертка) универсальна для класса всех k-местных ЧРФ.*

Следствие 3.4.2. *Существует ЧРФ $v(x)$, которая не может быть доопределена до рекурсивной функции.*

Доказательство. Покажем, что ЧРФ $v(x) \rightleftharpoons \overline{\text{sg}}(U^1(x, x))$ не доопределяется до ОРФ. Предположим противное и рассмотрим рекурсивную функцию $v_0(x)$, которая доопределяет $v(x)$. Тогда в силу универсальности функции U^1 существует такой номер n, что $v_0(x) = U^1(n, x)$ для всех x. В частности, имеем $v_0(n) = U^1(n, n)$, откуда получаем $\overline{\text{sg}}(U^1(n, n)) = U^1(n, n)$, а это невозможно. □

Теорема 3.4.3. (теорема Райса). *Пусть F — некоторое непустое семейство n-местных ЧРФ, не совпадающее с совокупностью всех n-местных ЧРФ. Тогда множество $A_F \rightleftharpoons \{a \mid \kappa^n(a) \in F\}$ всех клиниевских номеров функций, входящих в F, нерекурсивно.*

Из теоремы Райса непосредственно вытекает

Следствие 3.4.4. *Каково бы ни было нетривиальное свойство P n-местных ЧРФ (т.е. свойство, которым обладают некоторые, но не все n-местные ЧРФ), задача распознавания этого свойства алгоритмически неразрешима, т.е. не существует машины Тьюринга T^0, для которой выполняются следующие условия:*

1) $q_1\text{код}(T_f) \Rightarrow^{T^0} q_0 0110$, *если n-местная ЧРФ f обладает свойством P;*

2) $q_1\text{код}(T_f) \Rightarrow^{T^0} q_0 010$, *если n-местная ЧРФ f не обладает свойством P.*

Теорема Райса объясняет природу многих трудностей, возникающих в практике программирования и создания алгоритмических языков. Из этой теоремы, в частности, вытекает, что невозможно алгоритмически распознавать, является ли данная одноместная функция:

— тождественно равной нулю,

— нигде не определенной,

— рекурсивной,

— монотонной,

— взаимно однозначной и т.д.

На практике это означает, что если имеется некоторая программа, по ней, вообще говоря, автоматически нельзя ничего сказать о функции, реализуемой программой. Точно также нельзя установить, реализуют ли две заданные программы одну и ту же функцию. Наконец, в любом нетривиальном алгоритмическом языке неразрешима задача обнаружения "бессмысленных" программ, т.е. программ, задающих нигде не определенные функции.

§ 3.5. Неразрешимость исчисления предикатов. Теорема Гёделя о неполноте. Разрешимые и неразрешимые теории

Следующей нашей целью является установление неразрешимости исчисления предикатов арифметической сигнатуры $\Sigma_0 = \{s^{(1)}, +^{(2)}, .^{(2)}, 0^{(0)}, \leq^{(2)}\}$.

Напомним, что формальное исчисление I называется *разрешимым*, если существует алгоритм, позволяющий по любому выражению Φ исчисления I узнавать, доказуемо ли Φ в исчислении I или нет. В противном случае исчисление I называется *неразрешимым*.

Гёделевской нумерацией множества $W(A)$ слов алфавита A называется такая разнозначная функция $g : W(A) \to \mathbb{N}$, что существует алгоритм G, вычисляющий по слову $w \in W(A)$ его номер $g(w)$, и существует алгоритм G', выписывающий по числу $n \in \mathbb{N}$ слово w, если $n = g(w)$, и выдающий число 0, если $n \in \mathbb{N} \setminus g(W(A))$.

Ясно, что в силу тезиса Чёрча вопрос о разрешимости исчисления I, имеющего гёделевскую нумерацию g для всех выражений I, сводится к вопросу о рекурсивности множества гёделевских номеров всех теорем исчисления I.

Поставим в соответствие каждой формуле ИП^{Σ_0} некоторый *гёделевский* номер γ, по которому можно эффективно распознать структуру самой формулы.

Для множества $T(\Sigma_0)$ термов сигнатуры Σ_0 и множества $F(\Sigma_0)$ формул сигнатуры Σ_0 определим индукцией по числу шагов построения термов и формул *гёделевскую нумерацию* $\gamma : T(\Sigma_0) \cup F(\Sigma_0) \to \mathbb{N}$ в соответствии со следующими правилами:

0) $\gamma(0) = c(0,1)$;
1) $\gamma(v_n) = c(1,n)$, где v_n — n-я переменная из множества V;
2) $\gamma(s(t)) = c(2,\gamma(t))$, где $t \in T(\Sigma_0)$;
3) $\gamma(t_1 + t_2) = c(3,c(\gamma(t_1),\gamma(t_2)))$, где $t_1, t_2 \in T(\Sigma_0)$;
4) $\gamma(t_1 \cdot t_2) = c(4,c(\gamma(t_1),\gamma(t_2)))$, где $t_1, t_2 \in T(\Sigma_0)$;
5) $\gamma(t_1 \approx t_2) = c(5,c(\gamma(t_1),\gamma(t_2)))$, где $t_1, t_2 \in T(\Sigma_0)$;
6) $\gamma(t_1 \le t_2) = c(6,c(\gamma(t_1),\gamma(t_2)))$, где $t_1, t_2 \in T(\Sigma_0)$;
7) $\gamma(\varphi \wedge \psi) = c(7,c(\gamma(\varphi),\gamma(\psi)))$, где $\varphi, \psi \in F(\Sigma_0)$;
8) $\gamma(\varphi \vee \psi) = c(8,c(\gamma(\varphi),\gamma(\psi)))$, где $\varphi, \psi \in F(\Sigma_0)$;
9) $\gamma(\varphi \to \psi) = c(9,c(\gamma(\varphi),\gamma(\psi)))$, где $\varphi, \psi \in F(\Sigma_0)$;
10) $\gamma(\neg\varphi) = c(10,\gamma(\varphi))$, где $\varphi \in F(\Sigma_0)$;
11) $\gamma(\exists v_n\, \varphi) = c(11,c(n,\gamma(\varphi)))$, где $v_n \in V$, $\varphi \in F(\Sigma_0)$;
12) $\gamma(\forall v_n\, \varphi) = c(12,c(n,\gamma(\varphi)))$, где $v_n \in V$, $\varphi \in F(\Sigma_0)$.

Из примитивной рекурсивности функций c, l и r вытекает следующее

Предложение 3.5.1. 1. *Множество* $\gamma_{T_0} \rightleftharpoons \{\gamma(t) \mid t \in T(\Sigma_0)\}$ *гёделевских номеров термов сигнатуры* Σ_0 *примитивно рекурсивно.*

2. *Множество* $\gamma_{F_0} \rightleftharpoons \{\gamma(\varphi) \mid \varphi \in F(\Sigma_0)\}$ *гёделевских номеров формул сигнатуры* Σ_0 *примитивно рекурсивно.* $\square$

Обозначим через Fr характеристическую функцию двухместного отношения, состоящего из всех пар (m,n), для которых m является гёделевским номером некоторой формулы φ, в которую входит свободно переменная v_n. Для функции Fr справедливо следующее соотношение:

$$\mathrm{Fr}(m,n) = \begin{cases} 1, & \text{если } l(m) \in \{5,6\} \text{ и } v_n \text{ входит} \\ & \text{в терм } t_1 \text{ с условием } \gamma(t_1) = l(r(n)) \\ & \text{или в терм } t_2 \text{ с условием } \gamma(t_1) = r(r(n)); \\ 1, & \text{если } l(m) \in \{7,8,9\}, m \in \gamma_{F_0}, \\ & \text{и } \mathrm{Fr}(l(r(m)),n) = 1 \text{ или } \mathrm{Fr}(r(r(m)),n) = 1; \\ 1, & \text{если } l(m) = 10, m \in \gamma_{F_0} \text{ и } \mathrm{Fr}(r(m),n) = 1; \\ 1, & \text{если } l(m) \in \{11,12\}, m \in \gamma_{F_0}, l(r(m)) \neq n \\ & \text{и } \mathrm{Fr}(r(r(m)),n) = 1; \\ 0 & \text{в противном случае.} \end{cases}$$

Из приведенного соотношения вытекает примитивная рекурсивность функции Fr, а также функции $\mathrm{Sb}(m,n,k)$, устанавливающей по гёделев-

скому номеру m терма q и гёделевскому номеру k терма t гёделевский номер результата подстановки $(q)_t^{v_n}$ в терм q терма t вместо переменной v_n, а по гёделевскому номеру m формулы φ и гёделевскому номеру k терма t гёделевский номер результата подстановки $(\varphi)_t^{v_n}$ в формулу φ терма t вместо переменной v_n.

Из примитивной рекурсивности множества γ_{F_0}, функций Fr, Sb следует примитивная рекурсивность множеств гёделевских номеров формул для каждой из 14 аксиом ИП^{Σ_0}:

Предложение 3.5.2. *Для любого $i \in \{1, \ldots, 14\}$ множество A_i гёделевских номеров для множества i-х аксиом ИП^{Σ_0} примитивно рекурсивно.* □

В качестве примера приведем пример примитивно рекурсивного описания характеристической функции для множества A_1 гёделевских номеров аксиом, имеющих вид $\varphi \to (\psi \to \varphi)$:

$$\chi_{A_1}(n) = \begin{cases} 1, & \text{если } \gamma_{F_0}(n) = 1, l(n) = 9, l(r(r(n))) = 9 \\ & \text{и } l(r(n)) = r(r(r(r(n)))), \\ 0 & \text{в противном случае.} \end{cases}$$

Для каждого из трех правил вывода исчисления ИП^{Σ_0} определим характеристические функции $\chi_{R_1}(x, y, z)$, $\chi_{R_2}(x, y)$, $\chi_{R_3}(x, y)$ соответствующих отношений R_1, R_2, R_3 такие, что

$$\chi_{R_1}(n, m, k) = \begin{cases} 1, & \text{если формула } \psi \text{ с номером } \gamma(\psi) = k \\ & \text{получается из формулы } \varphi \\ & \text{с номером } \gamma(\varphi) = n \text{ и формулы } (\varphi \to \psi) \\ & \text{с номером } \gamma(\varphi \to \psi) = m \text{ по правилу 1,} \\ 0 & \text{в противном случае,} \end{cases}$$

$$\chi_{R_2}(n, m) = \begin{cases} 1, & \text{если формула } (\psi \to \forall x\, \varphi) \\ & \text{с номером } \gamma(\psi \to \forall x\, \varphi) = m \\ & \text{получается из формулы } (\psi \to \varphi) \\ & \text{с номером } \gamma(\psi \to \varphi) = n \text{ по правилу 2,} \\ 0 & \text{в противном случае,} \end{cases}$$

$$\chi_{R_3}(n, m) = \begin{cases} 1, & \text{если формула } (\exists x\, \varphi \to \psi) \\ & \text{с номером } \gamma(\exists x\, \varphi \to \psi) = m \\ & \text{получается из формулы } (\varphi \to \psi) \\ & \text{с номером } \gamma(\varphi \to \psi) = n \text{ по правилу 3,} \\ 0 & \text{в противном случае.} \end{cases}$$

Из приведенных описаний вытекает примитивная рекурсивность отношений R_1, R_2, R_3.

Отношение $P \subseteq \mathbb{N}^k$ называется *рекурсивным* (сокращенно РО), если рекурсивна характеристическая функция χ_P. Отношение $P \subseteq \mathbb{N}^k$ называется *рекурсивно перечислимым* (сокращенно РПО), если существует такое рекурсивное отношение $Q \subseteq \mathbb{N}^{k+1}$, что $P = \exists y\, Q(x_1, \ldots, x_k, y)$, где

$$\exists y\, Q(x_1, \ldots, x_k, y) \rightleftharpoons$$

$$\{(x_1, \ldots, x_k) \mid (x_1, \ldots, x_k, y) \in Q \text{ для некоторого } y \in \mathbb{N}\}.$$

Теорема 3.5.3. *Множество гёделевских номеров доказуемых формул* ИП^{Σ_0} *рекурсивно перечислимо.*

Следствие 3.5.4. *Если* X *— множество формул сигнатуры* Σ_0*, у которого множество* $\gamma(X)$ *рекурсивно перечислимо, то множество* $\{\gamma(\varphi) \mid X \vdash \varphi\}$ *рекурсивно перечислимо.*

Теорема 3.5.5. *Если* X *— непротиворечивое множество формул сигнатуры* Σ_0*, содержащее множество* $\mathbb{A}_0$*, то множество* $\gamma(T_X)$*, где* $T_X = \{\varphi \mid X \vdash \varphi\}$*, нерекурсивно.*

Непротиворечивая теория T сигнатуры Σ_0 называется *разрешимой*, если множество гёделевских номеров предложений из T рекурсивно. Из теоремы 4.6.7 вытекает

Следствие 3.5.6. (теорема о неразрешимости теории элементарной арифметики). *Для любого непротиворечивого множества* X *предложений сигнатуры* Σ_0*, содержащего множество* $\mathbb{A}_0$*, теория* T_X*, порожденная множеством* X*, неразрешима.*

Приведенное утверждение означает, что не существует универсального алгоритма решения всех математических задач.

Следствие 3.5.7. (теорема Гёделя о неполноте теории арифметики Пеано). *Пусть* X *— непротиворечивое множество формул сигнатуры* Σ_0*, содержащее множество* $\mathbb{A}_0$*, и* $\gamma(X)$ *— рекурсивно перечислимое отношение. Тогда теория* T_X*, состоящая из всех предложений, выводимых в* ИП^{Σ_0} *из множества* X*, неполна.*

Следствие 3.5.8. (теорема Чёрча о неразрешимости ИП^{Σ_0}). *Множество гёделевских номеров теорем исчисления* ИП^{Σ_0} *нерекурсивно.*

В силу теоремы о неразрешимости элементарной теории арифметики возникла задача нахождения разрешимых теорий. К таким теориям относятся полные теории, имеющие множества аксиом, у которых множества гёделевских номеров рекурсивно перечислимы. В частности, разрешимыми являются следующие теории:

— $\mathrm{Th}(\langle \mathbb{Q}; < \rangle)$,
— $\mathrm{Th}(\langle \mathbb{C}; +, \cdot, 0, 1 \rangle)$,
— $\mathrm{Th}(\langle \mathbb{R}; +, \cdot, <, 0, 1 \rangle)$,
— $\mathrm{Th}(\langle \mathbb{Q}; +, -, 0 \rangle)$,
— $\mathrm{Th}(\langle \mathbb{Z}; +, -, 0 \rangle)$,
— $\mathrm{Th}(\langle \mathbb{N}; +, s, 0 \rangle)$,
— $\mathrm{Th}(\langle \mathbb{N}; \cdot, 0 \rangle)$.

§ 3.6. Характеристики сложности алгоритмов

1. Определение и связь основных характеристик. До сих пор мы рассматривали принципиальную возможность алгоритмического решения той или иной задачи. Однако существуют задачи, которые хотя и могут быть решены на машине, требуют столь большого объема вычислений, что их решение практически недоступно. Настоящий параграф будет посвящен изучению и классификации алгоритмических задач относительно их вычислительной сложности.

Поскольку основные сложностные характеристики переносятся от одной модели вычисления к другой, в качестве модели вычислительного устройства достаточно рассмотреть машину Тьюринга, а в качестве характеристик сложности — необходимое время (число шагов вычисления) и память (размер используемой ленты).

Пусть T — машина Тьюринга, вычисляющая функцию $f(x)$. Обозначим через $t_T(x)$ число шагов работы машины T, вычисляющей значение $f(x)$, если $f(x)$ определено, и будем считать, что $t_T(x)$ не определено, если не определено значение $f(x)$. Функция $t_T(x)$ называется *временно́й сложностью машины* T.

Пусть значение $f(x)$ определено. *Активной зоной* при работе машины T на числе x называется минимальное множество $S_T(x)$ ячеек ленты, содержащее все ячейки, которые являются активными, т.е. используются при вычислении $f(x)$.

Ленточной сложностью машины T называется функция $s_T(x)$, которая равна мощности активной зоны $S_T(x)$, если значение $f(x)$ определено, и $s_T(x)$ не определено, если не определено значение $f(x)$.

Если в машине Тьюринга $T = \langle A, Q, \Pi, q_0, q_1 \rangle$ внешний алфавит A состоит из $m+1$ элементов, множество внутренних состояний Q — из $n+1$ элементов и значение $f(x)$ определено, то справедливы следующие неравенства, позволяющие оценивать величину ленточной сложности через временну́ю и наоборот:

$$s_T(x) \le x + 1 + t_T(x), \quad t_T(x) \le m \cdot s_T^2(x)(n+1)^{s_T(x)}.$$

Приведенные оценки показывают, что для исследования сложности алгоритма в качестве основной характеристики достаточно рассматривать временну́ю сложность.

2. О верхней границе сложности вычислений. Следующая теорема дает отрицательный ответ на вопрос о существовании единой временно́й рекурсивной границы для вычисления значений функций, т.е. о существовании такой рекурсивной функции $h(x)$, что для любой вычислимой функции $f(x)$ существует машина T, на которой время вычисления значения $f(x)$ (если оно определено) не превосходит $h(x)$.

Теорема 3.6.1. *Для любой рекурсивной функции $h(x)$ существует такая рекурсивная функция $f(x)$, что $\rho_f = \{0,1\}$, и для любой машины T, вычисляющей функцию $f(x)$, найдется число $n \in \mathbb{N}$, для которого $t_T(x) > h(n)$.*

Д о к а з а т е л ь с т в о. Зафиксируем рекурсивную функцию $h(x)$. Рассмотрим универсальную функцию $U_1(n,x)$ для класса всех одноместных ЧРФ. Для каждого числа $n_0 \in \mathbb{N}$, являющегося кодом некоторой машины Тьюринга, обозначим через T_{n_0} машину Тьюринга с кодом n_0, вычисляющую одноместную функцию $U_1(n_0,x)$. Определим рекурсивную функцию

$$f(x) = \begin{cases} \overline{\mathrm{sg}}(U_1(x,x)), & \text{если } U_1(x,x) \text{ определено и } t_{T_x}(x) \le h(x), \\ 0 & \text{в противном случае.} \end{cases}$$

Пусть $T = T_n$ — машина Тьюринга, вычисляющая функцию $f(x)$. Тогда условие $t_{T_n}(n) \le h(n)$ выполниться не может, поскольку $f(x) = U_1(n,x)$, а $U_1(n,x) \ne \overline{\mathrm{sg}}(U_1(n,x))$. $\square$

Несложная модификация доказательства теоремы 4.7.1 приводит к ее следующему усилению.

Теорема 3.6.2. *Для любой рекурсивной функции $h(x)$ существует такая рекурсивная функция $f(x)$, что $\rho_f = \{0,1\}$, и для любой машины T, вычисляющей функцию $f(x)$, найдется число $n \in \mathbb{N}$, начиная с которого справедливо неравенство $t_T(x) > h(x)$.* $\square$

Задача получения нижних оценок сложности вычислений важна, поскольку на основе этих оценок можно делать заключения о качестве используемых методов вычислений, устанавливать их оптимальность.

3. О наилучших вычислениях. Следующая теорема показывает, что любое вычисление некоторой рекурсивной функции можно улуч-

шить, начиная с некоторого шага. Тем самым не существует вычисления, наилучшего в абсолютном смысле.

Теорема 3.6.3. (теорема Блюм об ускорении). *Для любой рекурсивной функции $r(x)$ существует такая рекурсивная функция $f(x)$, что $\rho_f = \{0, 1\}$, и для любой машины T, вычисляющей функцию $f(x)$, найдутся машина T', вычисляющая функцию $f(x)$, и число $n \in \mathbb{N}$, начиная с которого справедливо неравенство $t_T(x) > r(t_{T'}(x))$.* $\square$

П р и м е р 3.6.1. Если $r(x) = 2^x$, то для функции $f(x)$ и машины T, о которых идет речь в теореме 4.7.3, найдется машина T' с условием $t_{T'}(x) < \log_2(t_T(x))$, начиная с некоторого числа n. В свою очередь для машины T' найдется машина T'' с условием $t_{T''}(x) < \log_2(t_{T'}(x)) < \log_2(\log_2(t_T(x)))$, начиная с некоторого $n' \geq n$, и т.д. $\square$

§ 3.7. Задачи и упражнения

1. Какую функцию $f(x)$ вычисляет машина Тьюринга со следующей программой:
$$0q_1 \to Rq_2, \quad 1q_1 \to 1q_0,$$
$$0q_2 \to 1q_3, \quad 1q_2 \to Rq_2,$$
$$0q_3 \to 0q_0, \quad 1q_3 \to Lq_3?$$

2. Пусть машина Тьюринга T имеет следующую программу: $0q_1 \to 0q_0$. Какие функции $f(x)$, $f(x_1, x_2)$, $\ldots$, $f_n(x_1, \ldots, x_n)$, $\ldots$ вычисляет эта машина?

3. Построить машину Тьюринга, которая правильно вычисляет функцию $o(x) = 0$.

4. Построить машины Тьюринга **А**, **Б**$^-$, **В**, **К**, **Л**, **R**.

5. Построить машины Тьюринга для правильного вычисления функций
(а) $x \dot{-} 1$; (б) $\mathrm{sg}(x)$; (в) $\overline{\mathrm{sg}}(x)$; (г) $x \dot{-} y$;
(д) $x - y$; (е) $x/2$; (ж) $[x/2]$.

6. Доказать, что если $f(x_1, \ldots, x_n)$ — ПРФ, то следующие функции примитивно рекурсивны:
(а) $f_1(x_1, x_2, x_3, \ldots, x_n) \rightleftharpoons f(x_2, x_1, x_3, \ldots, x_n)$ (перестановка аргументов);
(б) $f_2(x_1, x_2, \ldots, x_n) \rightleftharpoons f(x_2, \ldots, x_n, x_1)$ (циклическая перестановка аргументов);
(в) $f_3(x_1, \ldots, x_n, x_{n+1}) \rightleftharpoons f(x_1, \ldots, x_n)$ (введение фиктивного аргумента);
(г) $f_4(x_1, \ldots, x_{n-1}) \rightleftharpoons f(x_1, x_1, x_2, \ldots, x_{n-1})$ (отождествление аргументов).

7. Доказать, что следующие функции примитивно рекурсивны:
(а) $f(x) = x + n$; (в) $f(x) = x!$, где $0! \rightleftharpoons 1$,
(б) $f(x) = n$; (г) $f(x, y) = x^y$, где $0^0 \rightleftharpoons 1$.

8. Записать аналитическое выражение для функции $R(f, g)$, если
   (а) $f(x, y, z) = z^x$, $g(x) = x$; (б) $f(x, y, z) = x^z$, $g(x) = x$.

9. Доказать, что следующие функции примитивно рекурсивны:
   (а) $\mathrm{sg}(x)$;
   (б) $\overline{\mathrm{sg}}(x)$;
   (в) $x \dot{-} 1$;
   (г) $x \dot{-} y$;
   (д) $|x - y|$;
   (е) $\max(x, y)$;
   (ж) $\min(x, y)$;
   (з) $[x/y]$;
   (и) $x \oplus y$ (сумма по модулю 2);
   (к) $\mathrm{rest}(x, y)$;
   (л) p_x (x-е простое число);
   (м) $\mathrm{ex}(x, y)$;
   (н) $\mathrm{long}(x)$;
   (о) $\left[x\sqrt{2}\right]$;
   (п) $\left[e^x\right]$.

10. Доказать, что:

    (а) множество всех частично рекурсивных функций счетно;

    (б) существует частичная числовая функция, не являющаяся частично рекурсивной;

    (в) существует всюду определенная числовая функция, не являющаяся общерекурсивной.

11. Доказать, что частично рекурсивны следующие функции:

    (а) $f(x, y) = \begin{cases} x - y, \text{ если } x \geq y, \\ \text{не определена в противном случае;} \end{cases}$

    (б) $f(x, y) = \begin{cases} x/y, \text{ если } y \text{ делит } x, \\ \text{не определена в противном случае;} \end{cases}$

    (в) $f(x, y) = \begin{cases} z, \text{ если } x = z^y, \\ \text{не определена в противном случае;} \end{cases}$

    (г) функция с конечной областью определения.

12. Доказать, что если $f^{(n+1)}$ и $g^{(n+1)}$ — ЧРФ, то частично рекурсивны следующие функции:
    (а) $\mu y\,(f(\overline{x}, y) = g(\overline{x}, y))$; (в) $\mu y\,(f(\overline{x}, y) < g(\overline{x}, y))$;
    (б) $\mu y\,(f(\overline{x}, y) \neq g(\overline{x}, y))$; (г) $\mu y\,(f(\overline{x}, y) = 0$ и $g(\overline{x}, y) = 0)$.

13. Доказать, что следующие предикаты примитивно рекурсивны:
    (а) $x + y = z$; (в) x четно;
    (б) $x \cdot y = z$; (г) x и y взаимно просты.

14. Доказать, что любое конечное множество натуральных чисел примитивно рекурсивно.

15. Доказать, что существует множество $X \subset \mathbb{N}$, не являющееся рекурсивно перечислимым.

После изучения главы 3 выполняются задачи 7 и 8 контрольной работы. Задача 7 решается написанием программы, аналогичной программе машин Тьюринга $\mathbf{Б}^+$ и **Сложение** из п.1 параграфа 3.1, а задача 8 — аналогично примеру 3.2.1.

Г л а в а 4

НЕКЛАССИЧЕСКИЕ ЛОГИКИ

В предыдущих разделах мы изучили основные формальные исчисления — исчисление высказываний и исчисление предикатов. Рассмотрение этих исчислений не случайно. Практика математических исследований позволяет сформулировать тезис о выразимости всех утверждений и доказательств классической математики с помощью исчисления предикатов. Таким образом, эмпирически справедлив

Тезис Гильберта. Любое математическое утверждение может быть записано на языке исчисления предикатов, а любое математическое доказательство можно провести в рамках исчисления предикатов.

Вместе с тем формализация утверждений и доказательств вызывает известные неудобства и на практике рассматриваются различные модификации логики высказываний и логики предикатов (*неклассические логики*), приспособленные для решения соответствующих им задач.

Ниже мы рассмотрим основные неклассические логики, которые делятся на *пропозициональные* (т.е модифицирующие логику высказываний) и *предикатные* (т.е видоизменяющие логику предикатов). Кроме того, мы изложим некоторые элементы темпоральных и алгоритмических логик, которые используются для создания и анализа программ.

§ 4.1. Пропозициональные логики

1. Интуиционистские логики — логики, с помощью которых описываются способы вывода высказываний, истинных с точки зрения интуиционизма, т.е. совокупности идей и методов, для которой основным критерием истинности математического суждения является интуитивная убедительность возможности мысленного эксперимента, связываемого с этим суждением. Основное отличие *интуиционистского исчисления высказываний* (ИИВ) состоит в замене в ИВ *закона исключенного третьего* (состоящего в доказуемости всевозможных

формул $\varphi \vee \neg\varphi$) или эквивалентного ему *закона двойного отрицания* ($\neg\neg\varphi \to \varphi$) более слабым *принципом противоречия*: $\varphi \to (\neg\varphi \to \psi)$. Таким образом, ИИВ получается из ИВ заменой десятой схемы аксиом на принцип противоречия.

Всякая формула, выводимая в ИИВ, приемлема с интуиционистской точки зрения.

Аналогично ИИВ интуиционистское исчисление предикатов является ослаблением ИП. При этом становятся недоказуемыми формулы вида $\forall x\,(\varphi(x) \vee \neg\varphi(x))$, а из $\neg\exists x\,\varphi(x)$ не выводится $\forall x\neg\varphi(x)$.

2. Многозначные логики. Рассмотренная в главе 6* *двузначная логика* допускает следующее обобщение на *k-значный* случай.

Функцией k-значной логики от n переменных $x_1, x_2, \ldots, x_n$ называется любая функция

$$f : \{0, 1, \ldots, k-1\}^n \to \{0, 1, \ldots, k-1\}.$$

В качестве формул k-значной логики рассматриваются термы сигнатуры $\Sigma_k = \{\wedge^{(2)}, \vee^{(2)}, \cdot^{\delta}\}_{\delta\in\{0,1,\ldots,k-1\}}$, интерпретации которых определяются по индукции согласно следующим соотношениям для любых формул φ и ψ:

— $f(\varphi^\delta) = \begin{cases} k-1, & \text{если } f(\varphi) = \delta, \\ 0, & \text{если } f(\varphi) \neq \delta, \end{cases}$

— $f(\varphi \wedge \psi) = \min\{f(\varphi), f(\psi)\}$,

— $f(\varphi \vee \psi) = \max\{f(\varphi), f(\psi)\}$.

Аналогично теореме о функциональной полноте для двузначной логики доказывается

Теорема 4.1.1. *Всякая функция f k-значной логики представляется в виде*

$$f(x_1, \ldots, x_n) = \bigvee_{(\delta_1,\ldots,\delta_n)\in\{0,\ldots,k-1\}^n} f(\delta_1, \ldots, \delta_n) \wedge x_1^{\delta_1} \wedge \ldots \wedge x_n^{\delta_n}. \quad \square$$

Правая часть в последнем равенстве называется *совершенной дизъюнктивной нормальной формой* (СДНФ).

По аналогии с двузначной логикой в k-значной логике система функций $\{f_i \mid i \in I\}$ называется *полной*, если любая функция k-значной логики представима в виде терма сигнатуры $\{f_i \mid i \in I\}$.

Из представления произвольной функции k-значной логики в виде СДНФ следует, что совокупность функций $x \wedge y$, $x \vee y$, x^δ, где $\delta \in \{0, 1, \ldots, k-1\}$, образует полную систему.

Обобщениями k-значной логики являются *счетнозначная* и *континуумзначная* логики, в которых функции f имеют соответственно счетное или континуальное число значений.

3. Нечеткие логики и нечеткие подмножества. Одним из важнейших классов, который можно рассматривать как модификацию класса многозначных логик, является класс *вероятностных* или *нечетких логик*, составляющий основу теории вероятностей. Каждая нечеткая логика представляет исчисление высказываний, у которого всякая пропозициональная переменная A интерпретируется некоторым значением $P(A) = c$ (где c — элемент числового интервала $[0, 1]$), называемым *вероятностью* для переменной A. Число c задает *степень определенности* или *степень четкости* для переменной A. При этом, если значение c равно или близко к нулю, считается, что степень четкости, т.е вероятность для переменной A, мала, а если c равно или близко к единице, то степень четкости или вероятность для переменной A велика.

В нечетких логиках формулы исчисления высказываний называются *событиями*, а их интерпретации P — *вероятностями*. Если $P(\varphi) = 1$ (соответственно $P(\varphi) = 0$), то событие φ называется *достоверным* (*невозможным*).

При этом вероятности событий определяются в соответствии со следующими соотношениями для любых событий φ и ψ:

1) $P(\varphi \vee \neg\varphi) = 1$ ($\varphi \vee \neg\varphi$ — достоверное событие);

2) $P(\varphi \wedge \neg\varphi) = 0$ ($\varphi \wedge \neg\varphi$ — невозможное событие);

3) если $P(\varphi\wedge\psi) = 0$ (т.е. события φ и ψ *несовместны*), то $P(\varphi\vee\psi) = P(\varphi) + P(\psi)$.

4) если секвенция $\varphi \vdash \psi$ доказуема (т.е событие φ влечет событие ψ), то $P(\varphi) \leq P(\psi)$.

Из соотношений 1–4 выводится следующее

Предложение 4.1.2. *Для любых событий φ и ψ справедливы следующие соотношения:*

(а) $P(\neg\varphi) = 1 - P(\varphi)$ *(вероятность события φ и вероятность* дополнительного *события $\neg\varphi$ в сумме равна единице);*

(б) *если секвенция $\vdash \varphi$ доказуема, то $P(\varphi) = 1$ (доказуемое событие достоверно);*

(в) *если секвенция $\varphi \vdash$ доказуема, то $P(\varphi) = 0$ (противоречивое событие невозможно);*

(г) $P(\varphi \vee \psi) \leq P(\varphi) + P(\psi)$;

(д) $P(\varphi \vee \psi) = P(\varphi) + P(\psi) - P(\varphi \wedge \psi)$.

С нечеткими логиками тесно связаны нечеткие подмножества данного множества. Пусть M — некоторое множество, $A \subseteq M$. Принадлежность элементов из M подмножеству A полностью определяется характеристической функцией

$$\mu_A(x) = \begin{cases} 1, & \text{если } x \in A, \\ 0, & \text{если } x \in M \setminus A. \end{cases}$$

Теперь допустим, что функция μ_A может принимать любое значение в интервале $[0, 1]$. В соответствии с этим элемент $x \in M$ может не принадлежать множеству A ($\mu_A(x) = 0$), может быть элементом A ($\mu_A(x) = 1$), а может принадлежать множеству A в некоторой небольшой (когда $\mu_A(x)$ близко к 0) или большой (когда $\mu_A(x)$ близко к 1) степени. Теперь, как и функции μ_A с условием $\rho_{\mu_A} \subseteq \{0, 1\}$, функции μ_A с условием $\rho_{\mu_A} \subseteq [0, 1]$ дают полную информацию о степени вхождения каждого элемента множества M в множество A. На основании последнего замечания любая функция $\mu : M \to [0, 1]$ называется *нечетким подмножеством* множества M. Таким образом, нечеткое множество A — это множество пар $\{(x, \mu(x)) \mid x \in M\}$.

П р и м е р 4.1.1. 1. В нечетком множестве

$$A = \{(x_1, 0.9), (x_2, 0), (x_3, 0.2), (x_4, 1), (x_5, 0.3)\}$$

элемент x_1 содержится в значительной степени, x_2 не содержится, x_3 содержится в небольшой степени, x_4 содержится полностью, а x_5 содержится в немного большей степени, чем x_3.

2. В качестве нечетких можно рассматривать подмножества множества вещественных чисел, состоящих из чисел, приблизительно равных данному вещественному числу a. □

Определим основные теоретико-множественные отношения и операции на нечетких подмножествах. Пусть $A_1 \rightleftharpoons \mu_1$ и $A_2 \rightleftharpoons \mu_2$ — нечеткие подмножества множества M. Говорят, что A_1 *содержится в* A_2 или имеется *включение* A_1 *в* A_2, если $\mu_1(x) \leq \mu_2(x)$ для любого $x \in M$. Нечеткие подмножества A_1 и A_2 называются *равными* или *совпадающими*, если $\mu_1 = \mu_2$.

Нечеткое подмножество A_1 называется *дополнением* нечеткого подмножества A_2, если $\mu_1(x) = 1 - \mu_2(x)$ для любого $x \in M$.

Нечеткое подмножество $B \rightleftharpoons \mu$ называется *пересечением* (*объединением*) нечетких подмножеств A_1 и A_2 и обозначается через $A_1 \wedge A_2$ (соответственно через $A_1 \vee A_2$), если

$$\mu(x) = \min\{\mu_1(x), \mu_2(x)\} \quad (\mu(x) = \max\{\mu_1(x), \mu_2(x)\})$$

для любого $x \in M$.

Непосредственно проверяется, что система

$$\mathfrak{F} = \langle \{\mu \mid \mu : M \to [0,1]\}; \wedge, \vee, \bar{\cdot}, 0, 1 \rangle,$$

состоящая из всех нечетких подмножеств данного множества M с операциями пересечения, объединения, дополнения, константами $0 : M \to \{0\}$ и $1 : M \to \{1\}$, удовлетворяет следующим теоретико-множественным законам: ассоциативности, коммутативности, идемпотентности, дистрибутивности, поглощения, де Моргана, двойного отрицания. Кроме того, выполняются действия с константами: $A \vee 0 = A$, $A \wedge 0 = 0$, $A \vee 1 = 1$, $A \wedge 1 = A$. Таким образом, любая система $\mathfrak{F}$ является дистрибутивной решеткой. Однако в отличие от булевых алгебр в этой решетке не выполняются законы дополнения: соотношения $A \wedge \overline{A} = 0$ и $A \vee \overline{A} = 1$ верны лишь для констант 0 и 1.

В параграфе 6.11* мы определили понятия контактной схемы и функции проводимости между полюсами, соответствующей формуле логики высказываний. Заменяя в этих определениях пропозициональные переменные логики высказываний на пропозициональные переменные (т.е элементарные события) нечеткой логики, получаем понятие *нечеткой контактной схемы* и *нечеткой функции проводимости*. При этом каждому контакту x ставится в соответствие значение $\mu(x) \in [0,1]$, называемое *током контакта*. Затем по индукции определяется *ток μ нечеткой функции проводимости* исходя из соотношений $\mu(x \wedge y) = \min(\mu(x), \mu(y))$ (для последовательного соединения $x \wedge y$) и $\mu(x \vee y) = \max(\mu(x), \mu(y))$ (для параллельного соединения $x \vee y$). *Токами нечеткой контактной схемы* называется совокупность токов нечетких функций проводимости.

П р и м е р 4.1.2. Если ток контакта x равен 0,3, ток контакта y — 0,6, то ток последовательного соединения равен 0,3, а ток параллельного соединения $x \vee y$ — 0,6. □

При рассмотрении предикатных нечетких логик определяется понятие *нечеткого n-местного отношения* $P^{(n)}$ на множествах $A_1, A_2, \ldots, A_n$ в виде функции $\mu_P : A_1 \times A_2 \times \ldots \times A_n \to [0,1]$.

Способы задания нечетких отношений соответствуют общим способам задания функций. Это, например, перечисление всех элементов множества μ_P, если множества $A_1, A_2, \ldots, A_n$ конечны, или аналитическое задание в виде арифметического терма. Бинарные нечеткие отношения $P^{(2)}$ могут задаваться в виде поверхности $\{(x,y,z) \in \mathbb{R}^3 \mid z = \mu_P(x,y)\}$, в виде *матрицы весов* $W = (w_{ij})$, где $w_{ij} = \mu_P(a_1^i, a_2^j)$, $a_1^i \in A_1$, $a_2^j \in A_2$, или в виде взвешенного графа с матрицей весов W.

П р и м е р 4.1.3. Предположим, необходимо построить нечеткое отношение $\mu : A \times B \to [0,1]$, которое описывает упрощенную схему поиска неисправности в автомобиле. С этой целью в качестве множества A рассмотрим множество предпосылок или причин неисправности $\{a_1, a_2, a_3, a_4\}$, в котором a_1 — "неисправность аккумулятора", a_2 — "неисправность карбюратора", a_3 — "низкое качество бензина", a_4 — "неисправность системы зажигания". В качестве множества B определим множество заключений или проявлений неисправности $\{b_1, b_2, b_3\}$, где b_1 — "двигатель не запускается", b_2 — "двигатель работает неустойчиво", b_3 — "двигатель не развивает полной мощности". При этом между каждым элементом множества предпосылок и каждым элементом множества следствий существует некоторая, вообще говоря неоднозначная, причинно-следственная связь.

Функция μ, описывающая степень уверенности в том, что та или иная причина неисправности может привести к тому или иному следствию, определяется исходя из субъективного опыта механика, марки автомобиля, условий его эксплуатации и учета других факторов.

Нечеткое отношение μ может быть записано, например, в виде следующего множества: $\{((a_1, b_1), 1), ((a_1, b_2), 0.1), ((a_1, b_3), 0.2), ((a_2, b_1), 0.8), ((a_2, b_2), 0.9), ((a_2, b_3), 1), ((a_3, b_1), 0.7), ((a_3, b_2), 0.8), ((a_3, b_3), 0.5), ((a_4, b_1), 1), ((a_4, b_2), 0.5), ((a_4, b_3), 0.2)\}$. Матрица весов для нечеткого отношения μ имеет следующий вид:

$$\begin{pmatrix} 1 & 0.1 & 0.2 \\ 0.8 & 0.9 & 1 \\ 0.7 & 0.8 & 0.5 \\ 1 & 0.5 & 0.2 \end{pmatrix}. \quad \square$$

Вышеизложенные определения позволяют естественным образом проинтерпретировать классические логики (исчисления высказываний и предикатов) в виде нечетких логик. Более того, нечеткие логики являются обобщениями классических логик, придающими каждому высказыванию некоторую степень уверенности.

За последние годы теория и практика, связанная с нечеткими логиками, получили весьма заметное развитие. Системы нечеткого вывода позволяют решать задачи автоматического управления, классификации данных, распознавания образов, принятия решений, машинного обучения и многие другие. Эта проблематика исследований тесно связана с целым рядом других научно-прикладных направлений, таких как: нечеткое моделирование, нечеткие экспертные системы, нечеткая ассоциативная память, нечеткие логические контроллеры, нечеткие регуляторы и нечеткие системы.

4. Модальные логики. *Модальная логика* — область логики, в которой наряду с обычными высказываниями рассматриваются *модальные высказывания*, т.е. высказывания, характеризующие степень достоверности суждений.

Различают три типа модальностей, каждый из которых подразделяется на виды:

— модальности общего вида: “необходимо”, “возможно”, “невозможно”, “случайно”;

— модальности, связанные с характеристиками действий и поступками людей в обществе: “обязательно”, “разрешено”, “запрещено”, “безразлично” и др.;

— модальности, являющиеся характеристиками знаний: “доказано”, “не доказано”, “опровергнуто”, “не опровергнуто”, “знает”, “верит”, “убежден”, “сомневается”.

Язык основных пропозициональных модальных логик получается добавлением к алфавиту исчисления высказываний новых одноместных связок (*модальных операторов*) $\Box$ (*необходимо*) и $\Diamond$ (*возможно*). В силу эквивалентности формул $\Diamond\varphi$ и $\neg\Box\neg\varphi$ в качестве исходного берется один модальный оператор, например $\Box$, а другой выводится из аксиом $\vdash \Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$.

Определим модальные исчисления I_0 и T, называемые *исчислениями Фейса-фон Вригта*. *Исчисление* I_0 получается из исчисления высказываний

— введением символа $\Box$;

— добавлением в определение формул фразы “если φ — формула, то $\Box\varphi$ — формула” (при этом формулы, содержащие модальный символ $\Box$, называются *модальностями*);

— введением дополнительной схемы аксиомы $\Box(\varphi \to \psi) \vdash (\Box\varphi \to \Box\psi)$;

— введением дополнительного правила вывода $\dfrac{\Gamma \vdash \varphi}{\Gamma \vdash \Box\varphi}$, называемого *правилом Гёделя*.

Исчисление T получается из исчисления I_0 добавлением схемы аксиом $\Box\varphi \vdash \varphi$.

Следующее *исчисление* S4 образуется за счет добавления к исчислению T аксиомы $\Box\varphi \vdash \Box\Box\varphi$. Если же к исчислению T добавить аксиому $\neg\Box\varphi \vdash \Box\neg\Box\varphi$, то получают *исчисление* S5. Наконец, *исчисление Брауэра* получается добавлением к исчислению T следующей *аксиомы Брауэра*:$\varphi \vdash \Box\Diamond\varphi$.

Можно показать, что все вышеприведенные модальные исчисления

непротиворечивы.

При рассмотрении предикатной модальной логики к модальным аксиомам добавляются аксиомы, описывающие действие модальных операторов на кванторы, например, так называемая *аксиома Баркан*: $\forall x \,\Box\, \varphi \vdash \Box\, \forall x \,\varphi$.

Специфика понятия истинности в модальных логиках позволяет вводить дополнительные аксиомы и правила вывода и изучать их выразительные возможности.

Система модальной логики может быть проинтерпретирована в терминах многозначной логики (простейшая система — как трехзначная: "истина", "ложь", "возможно"). Это обстоятельство, а также возможность применения модальной логики к построению теории "правдоподобных" выводов указывают на ее глубокое родство с вероятностной логикой.

Приведем некоторые семантические интерпретации пропозициональных модальных логик.

Пусть дана формула $\varphi(A_1, \ldots, A_n)$. *$m$-Означиванием формулы* φ (где $m > 0$) называется любая функция

$$v_m(\varphi) : \{0, 1, \ldots, m\}^n \to \{0, 1, \ldots, m\},$$

которая по любым значениям переменных A_i из множества $\{0, 1, \ldots, m\}$ выдает значения для формулы φ снова из множества $\{0, 1, \ldots, m\}$. При этом значению 0 соответствует истина.

Функция $v_m(\cdot)$, которая ставит в соответствие каждой формуле φ исчисления I ее m-означивание $v_m(\varphi)$, называется *m-означиванием исчисления I*.

Формула φ называется *v_m-общезначимой* (обозначается $\models_{v_m} \varphi$), если $v_m(\varphi) \equiv 0$.

Означивание v_m называется *характеристическим*, если выполняются следующие условия:

а) для любой формулы $\varphi(A_1, \ldots, A_n)$ исчисления высказываний функция $v_m(\varphi) \restriction \{0, 1\}^n$ совпадает с истинностной функцией $f_{\neg\varphi}$;

б) класс теорем исчисления I совпадает с классом v_m-общезначимых формул.

Очевидно, что m-означивание, задаваемое следующими соотношениями, удовлетворяет условию а) определения характеристического означивания:

$$v^*_m(\neg\varphi) = \begin{cases} 0, & \text{если } v^*_m(\varphi) = m, \\ m, & \text{если } v^*_m(\varphi) \neq m; \end{cases}$$

$$v^*_m(\varphi \wedge \psi) = \max(v^*_m(\varphi), v^*_m(\psi));$$

$$v^*_m(\varphi \vee \psi) = \min(v^*_m(\varphi), v^*_m(\psi));$$

$$v^*_m(\varphi \to \psi) = \begin{cases} 0, & \text{если } v^*_m(\varphi) \geq v^*_m(\psi), \\ v^*_m(\psi), & \text{если } v^*_m(\varphi) < v^*_m(\psi); \end{cases}$$

$$v^*_m(\Box\,\varphi) = \begin{cases} 0, & \text{если } v^*_m(\varphi) = 0, \\ m, & \text{если } v^*_m(\varphi) \neq 0; \end{cases}$$

$$v^*_m(\Diamond\,\varphi) = \begin{cases} 0, & \text{если } v^*_m(\varphi) \neq m, \\ m, & \text{если } v^*_m(\varphi) = m. \end{cases}$$

Теорема 4.1.3. 1. *В модальных исчислениях* T, S4 *и* S5 *любая доказуемая формула* v^*_m*-общезначима.*

2. *В исчислениях* T, S4 *и* S5 *нет характеристического означивания.* □

Из приведенной теоремы вытекает, что исчисления T, S4 и S5 весьма существенно отличаются от классической логики и близки к логике интуиционистской.

Среди различных семантических интерпретаций модальных логик важное место занимает *семантика Крипке*.

Моделью Крипке называется система $\langle W, R, G, v\rangle$, где:

— W — фиксированное непустое множество, называемое множеством *возможных миров*;

— R — рефлексивное бинарное отношение на множестве W, для которого условие $(w, w') \in R$ означает, что мир w' *достижим* из мира w;

— $G \in W$ — фиксированный элемент, называемый *действительным миром*;

— v — отображение T*-означивания* или T*-оценивания*, которое любой формуле $\varphi(A_1, \ldots, A_n)$ и любому миру w ставит в соответствие функцию $v(\varphi, w) : \{0,1\}^n \to \{0,1\}$, удовлетворяющую следующим условиям:

$v(\neg\varphi, w) = 1 \Leftrightarrow v(\varphi, w) = 0$;
$v(\varphi \wedge \psi, w) = 1 \Leftrightarrow v(\varphi, w) = v(\psi, w) = 1$;
$v(\varphi \vee \psi, w) = 1 \Leftrightarrow v(\varphi, w) = 1$ или $v(\psi, w) = 1$;
$v(\varphi \to \psi, w) = 1 \Leftrightarrow v(\varphi, w) = 0$ или $v(\psi, w) = 1$;
$v(\Box\,\varphi, w) = 1 \Leftrightarrow v(\varphi, w') = 1$ для любого $w' \in R(\{w\})$;
$v(\Diamond\,\varphi, w) = 1 \Leftrightarrow v(\varphi, w') = 1$ для некоторого $w' \in R(\{w\})$.

По определению *необходимость* в модели Крипке означает истинность во всех возможных достижимых мирах, а *возможность* — истинность хотя бы в одном из достижимых миров. Таким образом, в

моделях Крипке необходимость ведет себя как всеобщность, а возможность — как существование.

T-означивание v называется *B-означиванием* (соответственно S4-*означиванием*, S5-*означиванием*), если отношение R является симметричным (предпорядком, отношением эквивалентности).

Множество формул X называется *I-выполнимым*, где $I \in \{T, B, \mathrm{S4}, \mathrm{S5}\}$, если существует такое I-означивание v, что $v(\varphi, G) = 1$ для любой формулы $\varphi \in X$. Множество формул, не являющееся I-выполнимым, называется *I-невыполнимым*. Формула φ называется *I-опровержимой* (соответственно *I-общезначимой*), если множество $\{\neg\varphi\}$ I-выполнимо (I-невыполнимо).

Теорема 4.1.4. (теорема о непротиворечивости модальных исчислений). *В модальных исчислениях T,* S4 *и* S5 *любая доказуемая формула соответственно T-,* S4- *и* S5-*общезначима.* □

Исчисление I называется *полным по Крипке*, если всякая не выводимая в I формула исчисления I I-опровержима.

Теорема 4.1.5. (теорема о полноте модальных исчислений) *Исчисления T и* S4 *полны по Крипке.* □

Для предикатных модальных исчислений модели Крипке имеют вид $\langle W, R, G, D, \mu, v\rangle$, где $D = \{D_w \mid w \in W\}$, D_w — носитель мира w, μ — интерпретация предикатных символов в D, v — означивание, определяющее истинность формул на элементах $\bigcup\limits_{w\in W} D_w$. При этом для исчислений, содержащих аксиому Баркан, требуется выполнение импликации $(w, w') \in R \Rightarrow D_{w'} \subseteq D_w$.

Символы $\Box$ и $\Diamond$ могут пониматься иначе, чем “необходимость” и “возможность”. Например, допускаются следующие интерпретации:

— $\Box$ — “обязательность”, а $\Diamond$ — “позволение”;

— $\Box$ — “доказуемость”, а $\Diamond$ — “непротиворечивость”;

— $\Box$ — “везде” или “всегда”, а $\Diamond$ — “кое-где” или “иногда”.

Последние модальности называются *пространственно-временны́ми* и рассматриваются в следующем пункте.

5. Временны́е (темпоральные) логики. *Временны́е* или *темпоральные логики* — это модальные логики, которые получаются добавлением к логике высказываний новых символов, отражающих свойства времени.

Рассмотрение реального процесса во времени заставляет отступиться от двузначной логики. Например, между периодом, когда идет дождь,

и периодом, когда дождь прекратился, имеется промежуточное состояние, когда количество капель слишком мало для того, чтобы сказать, что идет дождь, но слишком велико, чтобы утверждать, что дождь уже закончился. Таким образом, появляется третье значение высказывания: "ни истинно, ни ложно".

Временна́я логика Прайора — это логика будущего. Она содержит новый символ F, который называется *символом будущего*, и новый символ G. При этом для любой формулы φ формула $F\,\varphi$ интерпретируется как "будет φ", а формула $G\,\varphi$ читается как "всегда будет φ" и связана с формулой $F\,\varphi$ следующей аксиомой:

$\vdash G\,\varphi \leftrightarrow \neg F\,\neg\varphi$.

Кроме того, к исчислению высказываний добавляются схемы аксиом $F\,(\varphi \vee \psi) \equiv F\,\varphi \vee F\,\psi$, $\quad F\,F\,\varphi \vdash F\,\varphi$ и правила вывода

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash F\,\varphi}, \qquad \frac{\varphi \vdash \psi}{F\,\varphi \vdash F\,\psi}.$$

Возможность и необходимость определяются через символы F и G следующими соотношениями:

$\Diamond\,\varphi = \varphi \vee F\,\varphi$, $\quad \Box\,\varphi = \varphi \wedge G\,\varphi$.

Прайор показал, что полученное модальное исчисление, содержащееся во временно́й логике, сильнее исчисления S4, но слабее S5.

Леммон предложил *минимальную временну́ю логику*, основанную на модальностях P — "было" и "F" — будет. Предложенное им исчисление добавляет к исчислению высказываний аксиомы

$$\neg F\neg(\varphi \to \psi) \vdash (F\,\varphi \to F\,\psi), \quad F\,\neg P\,\neg\varphi \vdash \varphi,$$
$$\neg P\neg(\varphi \to \psi) \vdash (P\,\varphi \to P\,\psi), \quad P\,\neg F\,\neg\varphi \vdash \varphi$$

и правила вывода

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \neg F\,\neg\varphi}, \qquad \frac{\Gamma \vdash \varphi}{\Gamma \vdash \neg P\,\neg\varphi}.$$

Данная логика не делает никаких предположений о природе времени: его бесконечности в прошлом или будущем, непрерывности или неразветвленности.

Во *временно́й логике фон Вригта* к исчислению высказываний добавляется бинарная связка T, позволяющая по формулам φ и ψ строить формулу $(\varphi T \psi)$, которая читается как "сейчас происходит событие φ, а затем, т.е. *в следующий момент времени*, происходит событие ψ".

С помощью формул $(\varphi_1 T(\varphi_2 T(\varphi_3 T \ldots \psi \ldots)))$, в которых формулы $\varphi_1, \ldots, \varphi_n$ являются описаниями состояний, описывается *история*

мира. При этом любая такая формула называется *фрагментом истории*. Термин "история" имеет двойственное значение: он может означать последовательность как самих полных состояний мира, так и их описаний.

К исчислению высказываний добавляются аксиомы
$((\varphi_1 \vee \varphi_2)T(\psi_1 \vee \psi_2)) \equiv (\varphi_1 T \psi_1) \vee (\varphi_1 T \psi_2) \vee (\varphi_2 T \psi_1) \vee (\varphi_2 T \psi_2)$,
$(\varphi T \psi) \wedge (\varphi T \chi) \vdash (\varphi T(\psi \wedge \chi))$,
$\varphi \equiv (\varphi T(\psi \vee \neg\psi))$,
$\vdash \neg(\varphi T(\psi \wedge \neg\psi))$
и правило вывода

$$\frac{\Gamma, \varphi_1 \vdash \varphi_2;\ \Gamma, \psi_1 \vdash \psi_2}{\Gamma, (\varphi_1 T \psi_1) \vdash (\varphi_2 T \psi_2)}.$$

Время в этой темпоральной логике дискретно и линейно упорядочено. Если число полных состояний мира равно 2^n, то число возможных историй в m последующих моментах равно 2^{mn}.

§ 4.2. Предикатные логики

1. Многосортные логики первого порядка. Двусортная логика первого порядка очень похожа на обычную логику первого порядка (логику предикатов), за исключением того, что имеется два сорта переменных.

П р и м е р 4.2.1. 1. Аксиомы линейного пространства можно естественным образом записать, имея один сорт переменных $u, v, w, \ldots$ для векторов, а другой сорт $\alpha, \beta, \gamma, \ldots$ — для скаляров (элементов поля). Таким образом, линейное пространство состоит из тройки $\langle \mathfrak{V}, \mathfrak{F}, \cdot \rangle$, где $\mathfrak{F} = \langle F; +_{\mathfrak{F}}, \cdot_{\mathfrak{F}}, 0, 1 \rangle$ — поле, а $\mathfrak{V} = \langle V; +, 0 \rangle$ — векторная система с операцией сложения векторов и выделенным нулевым вектором, $\cdot$ — умножение на скаляр.

2. При изучении групп нередко рассматривается класс групп $\mathfrak{A} = \langle A; \cdot, e \rangle$, в которых все элементы a имеют *конечный порядок*, т.е. такое наименьшее натуральное число $n > 0$, что $a^n = e$. Условие конечности порядков всех элементов выражается формулой $\forall x\, \exists n \geq 1\ (x^n \approx e)$, которая не является формулой исчисления предикатов. Более того, в силу неограниченности конечных порядков по теореме компактности формулы ИП, описывающей класс групп, у которых все элементы имеют конечные порядки, не существует вовсе. С другой стороны, введение двусортных систем $\langle \mathfrak{A}, \mathfrak{N}, (\cdot)^{\cdot} \rangle$ (где $\mathfrak{N} = \langle \mathbb{N}; \leq \rangle$, $(\cdot)^{\cdot}$ — операция возведения элементов группы $\mathfrak{A}$ в натуральные степени), а также квантора $\forall_1$ для элементов системы $\mathfrak{A}$ и квантора $\exists_2$ для

элементов системы $\mathfrak{N}$ позволяют записать искомую формулу в виде $\forall_1 x \exists_2 n\,((n \geq 1) \wedge (x^n \approx e))$. $\square$

В общем случае *двусортная* или *двуосновная* алгебраическая система $\langle \mathfrak{A}, \mathfrak{B}, \Sigma \rangle$ состоит из двух обычных алгебраических систем $\mathfrak{A}$ и $\mathfrak{B}$, а также некоторых операций и отношений на их объединении, соответствующих символам сигнатуры Σ.

Двусортная (или многосортная) логика только внешне сильнее (хотя часто более естественна, чем обычная логика), поскольку любую двусортную систему $\langle \mathfrak{A}, \mathfrak{B}, \Sigma \rangle$ можно превратить в обычную систему $\langle A \cup B; A, B, \ldots \rangle$ с одноместными предикатами A и B для выделения различных сортов элементов. Это сведение позволяет перенести многие результаты логики первого порядка на многосортную логику, предоставляющую известное удобство для работы.

При рассмотрении двусортных систем $\langle \mathfrak{A}, \mathfrak{B}, \Sigma \rangle$ с фиксированной системой $\mathfrak{B}$ получается так называемая $\mathfrak{B}$-*логика*. Например, $\mathbb{R}$-логика удобна при изучении евклидовых пространств, поскольку поле скаляров $\mathbb{R}$ фиксировано. Если система $\mathfrak{B}$ бесконечна, то $\mathfrak{B}$-логика сильнее логики первого порядка. В частности, для $\mathfrak{B}$-логики неверна теорема компактности.

2. Слабая логика второго порядка — это логика, в которой некоторым естественным образом строится понятие *конечного*. Пусть дана некоторая сигнатура Σ, $x, y, z, \ldots$ — переменные, из которых строятся формулы сигнатуры Σ. Расширим сигнатуру Σ до сигнатуры Σ^* введением нового предикатного символа принадлежности $\in$ и добавим к старому списку новые переменные $a, b, c, \ldots$. Таким образом получается двусортный алфавит $\langle \Sigma \cup \{x, y, z, \ldots\}, \{\in\} \cup \{a, b, c, \ldots\} \rangle$. Данную алгебраическую систему $\mathfrak{A} = \langle A; \Sigma \rangle$ расширим до *системы наследственно конечных множеств* $\mathrm{HF}(\mathfrak{A}) = \langle \mathfrak{A}, \mathrm{HF}(A); \in \restriction (A \cup \mathrm{HF}(A)) \rangle$ над $\mathfrak{A}$ в соответствии со следующей схемой:

$$\mathrm{HF}_0(A) = \varnothing,$$

$$\mathrm{HF}_{n+1}(A) = \{X \mid X \text{ — конечное подмножество } A \cup \mathrm{HF}_n(A)\},$$

$$\mathrm{HF}(A) = \bigcup_{n \in \omega} \mathrm{HF}_n(A).$$

Элементы множества $\mathrm{HF}(A)$ называются *допустимыми множествами* над A. Очевидно, что любое натуральное число, а также любая конечная последовательность натуральных чисел являются допустимым множеством над любым множеством A.

В слабой логике второго порядка разрешается использовать формулы сигнатуры Σ^*, в которых переменные $a, b, c, \ldots$ интерпретируются множествами из $\mathrm{HF}(A)$.

Слабая логика второго порядка имеет ту же силу, что и $\langle\omega; \leq\rangle$-логика, но значительно более естественна в алгебраическом контексте, поскольку позволяет непосредственно работать с целыми числами, конечными множествами, конечными последовательностями и т.д.

3. Бесконечные логики. Слабая логика второго порядка позволяет определить понятие конечного в семантике логики. Бесконечные логики, напротив, вводят в синтаксис бесконечные конструкции, подобные бесконечной формуле

$$\forall x\,((x \approx 0) \vee (2x \approx 0) \vee \ldots).$$

Логика $L_{\omega_1\omega}$ допускает дополнительно следующее правило образования: если $\Phi \rightleftharpoons \{\varphi_n \mid n \in \omega\}$ — счетное множество формул, то $\bigwedge \Phi \rightleftharpoons \bigwedge\limits_{n\in\omega} \varphi_n$ (конъюнкция Φ) и $\bigvee \Phi \rightleftharpoons \bigvee\limits_{n\in\omega} \varphi_n$ (дизъюнкция Φ) являются формулами. Обозначение логики $L_{\omega_1\omega}$ объясняется тем, что в ней допустимы счетные $(< \omega_1)$ конъюнкции и дизъюнкции и только конечное число $(< \omega)$ кванторов. Логика $L_{\omega_1\omega}$ позволяет выражать те понятия, которые выразимы в логике первого порядка по модулю счетного количества информации.

П р и м е р 4.2.2. Рассмотрим некоторый тип $p(\overline{x}) \in D(T)$, состоящий из формул счетной сигнатуры Σ. Если $p(\overline{x})$ — главный тип, то он определяется некоторой главной формулой $\varphi(\overline{x})$. Если же тип $p(\overline{x})$ не является главным, то он определяется лишь счетным множеством формул $\Phi(\overline{x}) = \{\varphi_n(\overline{x}) \mid n \in \omega\}$. Переходя к логике $L_{\omega_1\omega}$, получаем, что тип $p(\overline{x})$ определяется уже формулой $\bigwedge \Phi(\overline{x})$. $\square$

4. Логика с новыми кванторами. Рассмотрим исчисление предикатов некоторой сигнатуры Σ. Пусть Q — новый символ. Добавим к правилам образования формул следующее: если φ — формула, то $\mathrm{Q}x\,\varphi$ также является формулой. Существует много различных интерпретаций для Q. Например, можно определить, что $\mathfrak{A} \models \mathrm{Q}x\,\varphi(x)$ тогда и только тогда, когда существует бесконечно много элементов a, для которых $\mathfrak{A} \models \varphi(a)$. Эта логика, обозначаемая через $L(\mathrm{Q}_0)$, эквивалентна слабой логике второго порядка.

Если определить, что $\mathfrak{A} \models \mathrm{Q}x\,\varphi(x)$ тогда и только тогда, когда существует несчетно много элементов a, для которых $\mathfrak{A} \models \varphi(a)$, то по-

лучается логика с квантором "существует несчетно много". В этой логике, обозначаемой через $L(\mathrm{Q})$, справедлива и теорема компактности, и теорема о полноте, но неверна теорема Лёвенгейма — Скулема. При этом понятие "существует несчетно много" обеспечивает математически точную модель для неформального понятия "много". Используя терминологию "много" и "мало" для "несчетности" и "ненесчетности" соответственно, при формализации можно ввести следующие *аксиомы Кейслера*:

1) $\vdash \forall y\, \neg \mathrm{Q}x\, (x \approx y)$ (для любого y существует мало x, для которых $x = y$);

2) $\forall x(\varphi \to \psi), \mathrm{Q}x\, \varphi \vdash \mathrm{Q}x\, \psi$ (если $\varphi \to \psi$ для всех x и имеется много x, удовлетворяющих φ, то много x удовлетворяют ψ);

3) $\mathrm{Q}x\, (\varphi \vee \psi) \vdash \mathrm{Q}x\, \varphi \vee \mathrm{Q}x\, \psi$ (если много x удовлетворяют $\varphi \vee \psi$, то много x удовлетворяют φ или много x удовлетворяют ψ);

4) $\neg \mathrm{Q}x\, \exists y\, \varphi, \forall x\, \neg \mathrm{Q}y\, \varphi \ \vdash\ \neg \mathrm{Q}y\, \exists x\, \varphi$ (если существует мало x, для которых $\exists y\, \varphi(x, y)$, и если для каждого x существует мало y, для которых $\varphi(x, y)$, то существует мало y, для которых $\exists x\, \varphi(x, y)$).

§ 4.3. Предикатные временны́е логики и их приложение к программированию

Временны́е логики используются в программировании для описания и верификации программ. При этом *описание программы* состоит в выражении с помощью языка временно́й логики свойств программы, характеризующих ее правильное вычислительное поведение, а ее *верификация* — в использовании аппарата временно́го исчисления для доказательства того, что данная программа обладает интересующим свойством.

Программы рассматриваются как объекты, выраженные на формальном языке, обладающие определенной информационной и логической структурой и подлежащие исполнению на автоматических устройствах. Исследование программ проводится преимущественно на основе двух моделей вычислений: последовательных программ с памятью, или операторных программ, и рекурсивных программ. При этом обе модели строятся над некоторой абстрактной алгебраической системой $\mathfrak{A}$ конечной сигнатуры Σ.

Определение класса программ слагается из трех частей: схемы программы (синтаксиса), интерпретации и семантики. *Схема программы* — это конструктивный объект, показывающий, как строится программа с использованием сигнатурных символов. *Интерпретация* — это

задание конкретного носителя и сопоставление символам сигнатуры конкретных операций и отношений на носителе. *Семантика* — это способ сопоставления каждой программе результата ее выполнения. Как правило, с программами связывают вычисляемые ими функции. Интерпретация обычно входит в семантику как параметр, поэтому схема программы задает множество программ и вычисляемых ими функций, которое получается при варьировании интерпретаций над некоторым запасом базовых операций.

Схема программы с памятью, или *операторная схема* задается в виде конечного ориентированного *графа переходов*, имеющего обычно одну входную и одну выходную вершины, вершины с одной (*преобразователи*) и двумя (*распознаватели*) исходящими дугами. С помощью символов сигнатуры Σ, включающего константные символы, и счетного множества символов переменных обычным образом строятся множество $T(\Sigma)$ функциональных термов и множество $PT(\Sigma)$ *предикатных термов*, состоящее из термов $\chi_P(t_1, \ldots, t_n)$ (где $t_1, \ldots, t_n \in T(\Sigma)$) от *характеристических функций*

$$\chi_P(\overline{x}) = \begin{cases} 1, & \text{если } \overline{x} \in P, \\ 0, & \text{если } \overline{x} \notin P. \end{cases}$$

Каждому распознавателю сопоставляется некоторый предикатный терм, а преобразователю — *оператор присваивания*, имеющий вид $x := t$, где x — символ переменной, а t — функциональный терм. Конечная совокупность $(x_1, \ldots, x_k)$ всех переменных в схеме образует ее *память*. Интерпретация в дополнение к конкретизации базовых операций предписывает каждой переменной область ее изменения. Для программ с памятью наиболее обычна так называемая *операционная семантика*, состоящая из алгоритма выполнения программы на заданном состоянии памяти. Программа выполняется при движении по графу переходов. При попадании на распознаватель вычисляется предикатный терм и происходит переход по дуге, соответствующей значению характеристической функции. При попадании на преобразователь с оператором $x := t$ вычисляется значение t и присваивается переменной x. Результат выполнения программы — состояние памяти при попадании на выходную вершину.

Схема рекурсивной программы, или *рекурсивная схема*, использует кроме функциональных так называемые *условные термы*, образующие вместе с первыми множество *вычислительных термов*. Условный терм задает вычисление посредством разбора случаев, имеет вид $(\pi|t_1|t_2)$, где π — предикатный, а t_1 и t_2 — вычислительные термы, и

соответствует конструкции условного выражения:

$$\textbf{if } \pi \textbf{ then } t_1 \textbf{ else } t_2.$$

Рекурсивная схема состоит из главного вычислительного терма с входными переменными и конечного набора рекурсивных уравнений вида $f(x_1, \ldots, x_n) = t$, где f — символ определяемой функции, $x_1, \ldots, x_n$ — переменные, t — терм с переменными из множества $\{x_1, \ldots, x_n\}$ и с символами определяемых функций из набора уравнений. При естественных предположениях на интерпретацию базовых операций система уравнений относительно определяемых функций всегда имеет так называемую *наименьшую неподвижную точку* — совокупность функций, удовлетворяющих уравнениям, с графиками, содержащимися в графиках любых других решений уравнений. Подставляя в главный терм вместо символов определяемых функций соответствующие компоненты наименьшей неподвижной точки, получают функциональный терм, задающий некоторую функцию входных переменных, которая и объявляется функцией, вычисляемой рекурсивной программой.

Пусть Π — программа, φ — формула, относящаяся к входным данным, которая должна быть истинна *перед* выполнением программы Π, ψ — формула, которая должна быть истинна после выполнения программы Π. Формула φ называется *предусловием*, а ψ — *постусловием* Π.

Программа Π называется *частично правильной* относительно φ и ψ, если всякий раз, когда предусловие φ истинно перед выполнением Π и Π заканчивает работу, постусловие ψ будет также истинно. В этом случае используется запись

$$\{\varphi\}\Pi\{\psi\}.$$

Программа Π называется *тотально правильной* относительно φ и ψ, если она частично правильна относительно φ и ψ, и обязательно завершает работу, если φ истинна (т.е. исполнение Π обязательно завершается и ψ истинна в любом состоянии памяти компьютера, которое может получиться при выполнении Π; последнее условие на ψ соответствует $\Box\psi$ в модальной логике). В этом случае используется запись

$$\{\varphi\}\Pi\downarrow\{\psi\}.$$

Предусловие φ и постусловие ψ связаны с конкретной задачей, которую требуется решить и для решения которой написана программа Π. Нам требуется доказать, что она правильна.

С целью верификации компьютерных программ создана *темпоральная логика Пнуели*, позволяющая доказывать наличие у данной программы свойств, характеризующих ее правильное вычислительное поведение. При этом понятие внешнего времени, или темпоральности, существенно лишь для верификации программ, связанных с параллельными вычислениями, поскольку в последовательных программах имеются "внутренние часы", а именно само выполнение. Зная метку в последовательной программе, а также значения программных переменных, можно точно определить, в каком месте программы находится процесс вычисления. При обращении же к недетерминированным, параллельным программам, в которых выполнение состоит из перемешанных между собой операций из различных процессов, требуется различать "где" и "когда" и сохранять внешнюю временну́ю шкалу, независимую от выполнения. Таким образом, логика Пнуели предназначена для верификации программ, связанных с параллельными вычислениями.

Временная логика Пнуели строится как логика первого порядка с добавлением одноместного предиката $L(x)$, выделяющего множество меток l, в которых могут находиться вычислительные процессы. Отношение L дает средства для описания контрольного компонента программных состояний.

Кроме того, логика Пнуели содержит дополнительные символы F (когда-нибудь будет), G (всегда будет) и $\aleph$ (в следующий момент будет), следующие аксиомы:
$G\,(\varphi \to \psi) \vdash (G\,\varphi \to G\,\psi)$,
$\aleph\,(\varphi \to \psi) \vdash (\aleph\,\varphi \to \aleph\,\psi)$,
$G\,\varphi \vdash \varphi$,
$G\,\varphi \vdash \aleph G\,\varphi$,
$G\,\varphi \vdash \aleph\,\varphi$,
$\vdash \aleph\,\neg\varphi \leftrightarrow \neg\aleph\,\varphi$,
$\varphi, G\,(\varphi \to \aleph\,\varphi) \vdash G\,\varphi$,
а также правила вывода

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash G\,\varphi}, \qquad \frac{\Gamma, \psi \vdash \varphi;\ \Gamma \vdash \forall x\,(L(x) \wedge \psi \to \aleph\,\varphi)}{\Gamma, \psi \vdash G\,\varphi}.$$

Последнее правило называется *инвариантным правилом*, согласно которому для установления инвариантности (неизменности) свойства φ в данной программе нужно установить, что это свойство имеет место в начале программы и сохраняется после выполнения каждой команды этой программы.

С помощью инвариантного правила доказывается следующее *свойство исключения критических связей*:

$$\varphi \vdash G\,\neg(L(l_1) \wedge \ldots \wedge L(l_n)),$$

где φ — формула, выражающая исходные условия программы (и включающая значения программных переменных, предусловие и исходные метки всех параллельных процессов), l_i — метка критической секции i-го процесса.

Расшифруем понятие *критической секции*. В параллельных вычислениях параллельные процессы могут включать блоки (секции), содержащие команды, критические для кооперирования этих процессов. Иначе говоря, пока один процесс находится в своей критической секции и выполняет команды этой секции, второй процесс не должен входить в свою критическую секцию, поскольку изменения в ячейках памяти, происходящие при выполнении команд идущего первого процесса, могут исказить вычисления, на которые сориентирован второй процесс. Второй процесс должен ждать (и для этого требуется понятие "внешнего времени"), пока содержимое используемых им ячеек не окажется требуемым. Ожидание происходит в силу выполнения специальных команд программы, называемых *семафорными инструкциями*.

П р и м е р 4.3.1. Рассмотрим параллельную программу с двумя критическими секциями КС1 и КС2. Семафорная инструкция — оператор "**wait** ... **then** ..." с семафорной переменной x_1, которой изначально присваивается значение 1:

	процесс 1		процесс 2
	l_0 : **wait** $x_1 > 0$ **then** $x_1 := x_1 - 1$;		m_0 : **wait** $x_1 > 0$ **then** $x_1 := x_1 - 1$;
КС1	$l_1 : t_1 := x_3 * x_4$; $l_2 : x_3 := t_1$; $l_3 : x_1 := x_1 + 1$;	КС2	$m_1 : t_2 := x_3/x_2$; $m_2 : x_2 := t_2$; $m_3 : x_1 := x_1 + 1$;
	l_4 : **end**		m_4 : **end**

Предположим, что при выполнении процесса 1 значение x_1 уменьшилось до 0. Тогда компьютер вынужден остановить процесс 2, поскольку преградой к его выполнению является семафорная инструкция под меткой m_0, для выполнения которой необходимо, чтобы x_1 было больше 0. Следовательно, компьютер начнет выполнять команды с метками l_1, l_2, l_3 и только после этого перейдет к продолжению выполнения процесса 2. Другими словами, пока процесс 1 находится

в своей критической секции КС1, процесс 2 не достигнет своей критической секции КС2. То же самое будет происходить в случае, когда процесс 2 начнет выполняться первым. □

§ 4.4. Алгоритмические логики

Алгоритмические логики создаются с целью описания семантики языков программирования и включают формулы вида $\{\varphi\}S\{\psi\}$, читающиеся как "если до выполнения оператора S было истинно φ, то после его выполнения будет истинно ψ".

Эти логики были изобретены Р. У. Флойдом (1967 г.), Ч. Хоаром (1969 г.) и представителями польской логической школы (А. Сальвиницкий и др. (1970 г.)).

Хоар определил простой язык программирования через логическую систему аксиом и правил вывода для доказательства частичной правильности программ. Им показано, что определение семантики языка не в терминах выполнения программы, а в терминах доказательства ее правильности упрощает процесс построения программы.

На базе работы Хоара проводились исследования в области аксиоматических определений языков программирования. Появилось много работ по аксиоматизации различных конструкций: от оператора присваивания до различных форм циклов, от вызова процедур до сопрограмм. В 1973 г. были сформулированы правила доказательства правильности для большинства конструкций языка Паскаль. В 1975 г. была построена автоматическая система верификации для языка Паскаль, основанная на аксиомах и правилах вывода. В 1979 г. был определен язык программирования Евклид, в проект которого с самого начала была заложена идея аксиоматизации.

В 1976 г. Э. Дейкстра предложил метод доказательства правильности программ. Суть метода заключается в том, чтобы строить программу вместе с доказательством, причем доказательство должно опережать построение программы. Дейкстра определил для простого языка программирования слабейшие предусловия и показал, как их можно использовать в качестве исчисления для вывода программ. Стало ясно, что использование формализма может привести к построению программ более надежным способом.

Опишем принципы построения алгоритмической логики $\mathbf{L}_0$.

Память в $\mathbf{L}_0$ разделена на *ячейки*. Каждая ячейка имеет *идентификатор*, представляющий собой слово из латинских букв и цифр и начинающийся с буквы. Ячейки содержат натуральные числа.

Программа в $\mathbf{L}_0$ состоит из *операторов*. Исходный оператор — *оператор присваивания* $x := t$, где x — идентификатор, а t — терм сигнатуры $\Sigma = \{+^{(2)}, \cdot^{(2)}, \leq^{(2)}\} \cup \{n^{(0)} \mid n \in \omega\}$, в котором в качестве переменных используются идентификаторы.

Пусть φ — формула сигнатуры Σ_0, истинная на состоянии памяти после присваивания, $(\varphi)^x_t$ — формула, истинная до присваивания. Тогда по оператору присваивания строится формула $\{(\varphi)^x_t\}x := t\{\varphi\}$.

П р и м е р 4.4.1. Если формула φ равна $(x < 2)$ и $t = x + 1$, то формула $(\varphi)^x_t$ равна $(x + 1 < 2)$. Следовательно, чтобы после присваивания $x := x + 1$ стало истинным $(x < 2)$, требуется, чтобы до присваивания выполнялось неравенство $x + 1 < 2$, т.е. $\models \{(x + 1 < 2)\}x := x + 1\{(x < 2)\}$. $\square$

Пусть два оператора S_1 и S_2 выполняются один за другим. Тогда *композиции* S_1S_2 операторов S_1 и S_2 соответствует правило вывода

$$\frac{\{\varphi\}S_1\{\psi\};\ \{\chi\}S_2\{\theta\};\ \psi \to \chi}{\{\varphi\}S_1S_2\{\theta\}}.$$

Условный оператор — это конструкция

$$\mathbf{IF}\, \varphi_1 \to S_1 \triangle \varphi_2 \to S_2 \triangle \ldots \triangle \varphi_n \to S_n\, \mathbf{FI},$$

где $\varphi_1, \ldots, \varphi_n$ — бескванторные формулы сигнатуры Σ_0, а S_1, $\ldots, S_n$ — последовательности операторов. При работе условного оператора проверяются формулы φ_i при текущем состоянии памяти. Если ни одна из формул φ_i не истинна, то фиксируется ошибка. Если же некоторые φ_i истинны, то по некоторым приоритетам выбирается одна из них и выполняется соответствующая последовательность операторов S_i.

Если каждая из команд S_i описана в логике $\mathbf{L}_0$ формулой $\psi_i S_i \chi$, то условный оператор описывается формулой

$$\{(\varphi_1 \to \psi_1) \wedge \ldots \wedge (\varphi_n \to \psi_n)\}\mathbf{IF}\, \varphi_1 \to S_1 \triangle \varphi_2 \to S_2 \triangle \ldots$$

$$\ldots \triangle \varphi_n \to S_n\, \mathbf{FI}\, \{\chi\}.$$

Операторам цикла соответствуют конструкции

$$\mathbf{DO}\, \varphi_1 \to S_1 \triangle \ldots \triangle \varphi_n \to S_n\, \mathbf{OUT}\, \psi_1 \to T_1 \triangle \ldots$$

$$\ldots \triangle \psi_m \to T_m\, \mathbf{OD}.$$

Выполняется оператор цикла следующим образом. Проверяются формулы $\varphi_1, \ldots, \varphi_n, \psi_1, \ldots, \psi_m$ при текущем состоянии памяти. Если

ни одна из них не истинна, то фиксируется ошибка. Если же некоторые истинны, то по некоторым приоритетам выбирается одна из них. Если выбрана φ_i, то выполняется соответствующая последовательность операторов S_i и выполнение цикла возобновляется. Если выбрана ψ_j, то выполняется соответствующая последовательность операторов T_j и выполнение цикла завершается.

Если каждая из команд S_i описана в логике $\mathbf{L}_0$ формулой $\{\varphi_i'\}S_i\{\chi_i\}$, а T_j — формулой $\{\psi_j'\}T_j\{\theta\}$, то однократное выполнение цикла описывается формулой

$$\{(\varphi_1 \to \varphi_1') \wedge \ldots \wedge (\varphi_n \to \varphi_n') \wedge (\psi_1 \to \psi_1') \wedge \ldots \wedge (\psi_1 \to \psi_1') \wedge$$
$$(\varphi_1 \vee \ldots \vee \varphi_n \vee \psi_1 \vee \ldots \vee \psi_m)\}\mathbf{IF}\ \varphi_1 \to S_1 \triangle \ldots \triangle \varphi_n \to S_n \triangle$$
$$\psi_1 \to T_1 \triangle \ldots \triangle \psi_m \to T_m\, \mathbf{FI}\{\theta\}.$$

Двукратное выполнение цикла соответствует формуле

$$\{(\varphi_1 \to \varphi_1') \wedge \ldots \wedge (\varphi_n \to \varphi_n') \wedge (\psi_1 \to \psi_1') \wedge \ldots \wedge (\psi_1 \to \psi_1') \wedge$$
$$(\varphi_1 \vee \ldots \vee \varphi_n \vee \psi_1 \vee \ldots \vee \psi_m)\}\mathbf{IF}\ \varphi_1 \to S_1 \triangle \ldots \triangle \varphi_n \to S_n \triangle$$
$$\psi_1 \to T_1 \triangle \ldots \triangle \psi_m \to T_m\, \mathbf{FI}\{\chi_1 \vee \ldots \vee \chi_n\}$$
$$\{(\varphi_1 \to \varphi_1') \wedge \ldots \wedge (\varphi_n \to \varphi_n') \wedge (\psi_1 \to \psi_1') \wedge \ldots \wedge (\psi_1 \to \psi_1') \wedge$$
$$(\varphi_1 \vee \ldots \vee \varphi_n \vee \psi_1 \vee \ldots \vee \psi_m) \wedge (\chi_1 \vee \ldots \vee \chi_n)\}\mathbf{IF}\ \varphi_1 \to S_1 \triangle \ldots$$
$$\ldots \triangle \varphi_n \to S_n \triangle \psi_1 \to T_1 \triangle \ldots \triangle \psi_m \to T_m\, \mathbf{FI}\{\theta\}$$

и т.д. до бесконечности. Обозначив через $\mathbf{DO}^k\{\theta\}$ условие правильности k шагов цикла, получаем, что корректность цикла равносильна истинности формулы $\bigwedge\limits_{k\in\omega\setminus\{0\}} \mathbf{DO}^k\{\theta\}$. Но эта формула имеет бесконечную длину и не является формулой логики $\mathbf{L}_0$. Появление таких формул, относящихся к логике $L_{\omega_1\omega}$, порождает серьезные проблемы для алгоритмических логик.

Опишем *алгоритмическую логику Хоара*, которая является основой для логики выводов правильных программ и допускает интерпретации в терминах программных конструкций. Следующие аксиомы, называемые *аксиомами Хоара* или *правилами верификации*, определяют предусловия как достаточные условия, гарантирующие, что исполнение соответствующего оператора при успешном завершении приведет к желательным постусловиям.

A1. $\{(\varphi)_t^x\}x := t\{\varphi\}$ (*аксиома присваивания*).

A2. $\{\varphi\}S\{\psi\} \wedge (\psi \to \chi) \to \{\varphi\}S\{\chi\}$ (*аксиома ослабления постусловия*).

A3. $\{\varphi\}S\{\psi\} \wedge (\chi \to \varphi) \to \{\chi\}S\{\psi\}$ (*аксиома усиления предусловия*).

A4. $\{\varphi\}S_1\{\psi\} \wedge \{\psi\}S_2\{\chi\} \to \{\varphi\}S_1S_2\{\chi\}$ (*аксиома композиции*).

A5. $\{\varphi \wedge \psi\}S\{\chi\} \wedge (\varphi \wedge \neg\psi \to \chi) \to \{\varphi\}\mathbf{if}\,\psi\,\mathbf{then}\,S\,\{\chi\}$ (*аксиома условного оператора*).

A6. $(\{\varphi \wedge \psi\}S_1\{\chi\}) \wedge (\{\varphi \wedge \neg\psi\}S_2\{\chi\}) \to$
$\{\varphi\}\mathbf{if}\,\psi\,\mathbf{then}\,S_1\,\mathbf{else}\,S_2\,\{\chi\}$ (*аксиома альтернативного оператора*).

A7. $\{\varphi \wedge \psi\}S\{\varphi\} \to \{\varphi\}\mathbf{repeat}\,S\,\mathbf{until}\,\psi\{\varphi \wedge \neg\psi\}$ (*аксиома оператора цикла* **until** (до)).

A8. $\{\varphi \wedge \psi\}S\{\varphi\} \to \{\varphi\}\mathbf{while}\,\psi\,\mathbf{do}\,S\{\varphi \wedge \neg\psi\}$ (*аксиома оператора цикла* **while** (пока)).

В аксиомах A7 и A8 утверждается, что формула φ истинна перед выполнением и после выполнения каждого шага цикла. Эта формула называется *инвариантной формулой* или *инвариантом цикла*.

Аксиомы A1 — A8 можно использовать для проверки согласованности передачи данных от оператора к оператору, для анализа структурных свойств текстов программ, для установления условий окончания цикла. Кроме того, аксиомы можно использовать для анализа результатов выполнения программы.

П р и м е р 4.4.2. Рассмотрим задачу нахождения частного q и остатка r от деления n на m.

Входные данные:

n, m — натуральные числа, где $m > 0$.

Выходные данные:

q, r — натуральные числа.

Описание программы S:

задать(n, m)
$r := n$; $q := 0$;
while $m \le r$ **do**
begin
$r := r - m$; $q := q + 1$
end;
выдать(q, r).

Сформулируем предусловие φ:

$$m > 0.$$

Сформулируем постусловие ψ:

$$(r < m) \wedge (n \approx m \cdot q + r).$$

Требуется доказать, что

$$\models \{\varphi\}S\{\psi\}$$

или

$$\models \{m > 0\}S\{(r < m) \wedge (n \approx m \cdot q + r)\}$$

Доказательство.

є	Формулы	Аксиомы и правила вывода
1	$\varphi \to (n \approx n + m \cdot 0)$	в формальной арифметике
2	$\{(n \approx n + m \cdot 0)\}r := n\{(n \approx r + m \cdot 0)\}$	аксиома A1
3	$\{(n \approx n + m \cdot 0)\}q := 0\{(n \approx r + m \cdot q)\}$	аксиома A1
4	$\{\varphi\}r := n\{(n \approx r + m \cdot 0)\}$	A3 к пунктам 1 и 2
5	$\{\varphi\}r := n; q := 0\{(n \approx r + m \cdot q)\}$	A4 к пунктам 3 и 4
6	$(n \approx r + m \cdot q) \wedge (m \leq r) \to$ $(n \approx (r - m) + m \cdot (q + 1))$	арифметика
7	$\{(n \approx (r - m) + m \cdot (q + 1))\}r := r - m$ $\{(n \approx r + m \cdot (q + 1))\}$	аксиома A1
8	$\{(n \approx r + m \cdot (q + 1))\}q := q + 1$ $\{(n \approx r + m \cdot q)\}$	аксиома A1
9	$\{(n \approx (r - m) + m \cdot (q + 1))\}r := r - m;$ $q := q + 1\{(n \approx r + m \cdot q)\}$	A4 к пунктам 7 и 8
10	$\{(n \approx r + m \cdot q) \wedge (m \leq r)\}r := r - m;$ $q := q + 1\{(n \approx r + m \cdot q)\}$	A2 к пунктам 6 и 9
11	$\{(n \approx r + m \cdot q) \wedge (m \leq r)\}$**while** $m \leq r$ **do** **begin** $r := r - m; q := q + 1$ **end** $\{(n \approx r + m \cdot q) \wedge \neg(m \leq r)\}$	A8 к пункту 10
12	$\{\varphi\}S\{(n \approx r + m \cdot q) \wedge \neg(m \leq r)\}$	A4 к пунктам 5 и 11

В силу того что система $\langle \mathbb{N}; \leq \rangle$ является вполне упорядоченным множеством и на каждом шаге значение r уменьшается на положительную величину r, найдется такое значение r, для которого не будет выполняться условие $m \leq r$, и циклический процесс завершится.

Таким образом, установлено, что

$$\{\varphi\}S \downarrow \{\psi\},$$

т.е. программа S является тотально правильной. □

§ 4.5. Задачи и упражнения

1. Доказать, что
   (а) все выводимые в ИИВ формулы выводимы в ИВ;
   (б) формулы $(\neg\neg\varphi \to \varphi)$ и $\varphi \vee \neg\varphi$ не выводимы в ИИВ;
   (в) если в ИИВ доказуемо $\Gamma, \varphi \vdash \psi$, то в ИИВ доказуемо $\Gamma \vdash (\varphi \to \psi)$.

2. Показать, что система функций, состоящая из функции $x \vee y$ и всех функций $e_{ij}(x) = \begin{cases} j, & \text{если } x = i, \\ 0, & \text{если } x \neq i, \end{cases}$ является полной в k-значной логике.

3. Доказать, что для любых событий φ, ψ, $\varphi_1, \ldots, \varphi_n$ справедливы следующие соотношения:

(а) $P(\varphi \wedge \psi) \leq P(\varphi)$;

(б) $P(\varphi \wedge \psi) \geq P(\varphi) + P(\psi) - 1$;

(в) $P(\varphi_1 \vee \ldots \vee \varphi_n) \leq P(\varphi_1) + \ldots + P(\varphi_n)$;

(г) если события $\varphi_1, \ldots, \varphi_n$ попарно несовместны, то $P(\varphi_1 \vee \ldots \vee \varphi_n) = P(\varphi_1) + \ldots + P(\varphi_n)$.

4. Для нечетких подмножеств $A = \{(x, 0.3), (y, 0.9), (z, 1)\}$ и $B = \{(x, 0.7), (y, 0), (z, 0.1)\}$ найти $\overline{A}$, $\overline{B}$, $A \wedge \overline{A}$, $B \vee \overline{B}$, $A \wedge \overline{B}$, $\overline{A} \vee B$.

5. Доказать следующие секвенции:

   (а) $\varphi \vdash \Diamond \varphi$;

   (б) $\Diamond \exists x \neg\varphi \vdash \exists x \neg\Box \varphi$.

6. Представить алгебру матриц в виде многосортной алгебраической системы, для которой соответствующей формулой многосортной логики выразимо множество матриц, имеющих конечный порядок.

7. Написать программу разложения натурального числа на простые сомножители и доказать ее тотальную правильность с помощью алгоритмической логики Хоара.

Варианты контрольной работы

Условия задач

1. Из данной совокупности секвенций выбрать доказуемые, построить их доказательства, для недоказуемых показать их недоказуемость с помощью: а) алгоритма Квайна, б) алгоритма редукции, в) метода резолюций. Среди доказательств недоказуемости выбрать оптимальное в каждом конкретном случае.

2. Найти формулу исчисления предикатов истинную на алгебраической системе $\mathcal{A}$ и ложную на системе $\mathcal{B}$.

3. Построить доказательство формулы в исчислении предикатов.

4. Установить, выполнима ли следующая формула и если выполнима, то построить модель этой формулы.

5. Привести к пренексной и клазуальной нормальной формам формулу.

6. Методом резолюций проверить, противоречиво ли множество предложений. Если множество непротиворечиво, то построить модель этого множества.

7. Построить машину Тьюринга для вычисления функции.

8. Доказать примитивную рекурсивность функции из задачи 7, выражая ее через простейшие с помощью операторов суперпозиции и примитивной рекурсии.

Вариант 1

1. а) $(x \to y) \wedge (y \to x),\ (y \to z) \wedge (z \to y) \vdash (x \to z) \wedge (z \to x)$;
   б) $\vdash \neg(x \wedge \neg x)$;
   в) $(x \to y) \to x \vdash y \to x$;
   г) $x \vee y \vdash \neg x \to y$.
2. $\mathcal{A} = \langle N; \leq \rangle,\ \mathcal{B} = \langle N; < \rangle$.
3. $\forall x(P(x) \to Q(x)) \vee \exists x(P(x) \wedge \neg Q(x))$.
4. $\exists x \exists y(\neg(x = y) \wedge P(x, y) \wedge \exists z R(x, y, z) \wedge \exists u \neg R(x, y, u))$.
5. $\neg(\exists x \forall y P(x, y) \vee \exists x \forall y Q(x, y)) \wedge \neg \forall x \exists y R(x, y)$.
6. $\phi_1 = \exists x(P_1(x) \wedge P_2(x))$,
   $\phi_2 = \forall x \forall y(P_1(x) \to (P_2(y) \to \neg P_3(x, y)))$,
   $\phi_3 = \exists x \forall y \neg(P_1(x) \to (P_2(y) \wedge \neg P_3(x, y)))$.
7. $f(x) = x \dot{-} 4$.

Вариант 2

1. а) $y \to \neg x \vdash \neg y \vee \neg x$;
   б) $(x \to \neg y) \wedge (\neg y \to x) \vdash x \vee y$;
   в) $x \vdash (x \wedge y) \vee (x \wedge \neg y \wedge z) \vee (x \wedge \neg y \wedge \neg z)$;
   г) $x, y, \neg z \vdash x \vee y \vee z$.
2. $\mathcal{A} = \langle R; \leq \rangle$, $\mathcal{B} = \langle N; \leq \rangle$.
3. $(\forall x P(x) \to \forall x Q(x)) \vee (\forall x P(x) \wedge \exists x \neg Q(x))$.
4. $\forall x \neg P(x,x) \wedge \forall x, y (P(x,y) \to \neg P(y,x)) \wedge \forall x \forall y \forall z \exists u (P(u,x) \wedge P(u,y) \wedge P(u,z))$.
5. $\forall x (\exists y P(x,y) \to \forall y Q(x,y)) \wedge \neg \exists x (\forall y P(x,y) \to \exists y Q(y,y))$.
6. $\phi_1 = \neg \forall x \forall y (\neg P_3(f(x), x) \to \neg (\neg P_2(y,x) \to P_3(f(x), y)))$,
   $\phi_2 = \forall x (f(x) = f(f(x)))$,
   $\phi_3 = \exists x \forall y \forall z \exists v \forall u \neg (\neg P_1(f(x), z) \to \neg (P_2(v,u) \to (P_3(f(v), y) \vee P_1(y, v))))$.
7. $f(x) = 3x$.

Вариант 3

1. а) $x \wedge y \wedge \neg z \vdash x \vee y \vee z$;
   б) $x \vee y \vdash (x \to y) \vee y$;
   в) $x \to y \vdash (x \to (y \to z)) \to (x \to z)$;
   г) $\neg x, \neg y \vdash ((x \to y) \to z) \vee ((x \to y) \to \neg z)$.
2. $\mathcal{A} = \langle R; \leq \rangle$, $\mathcal{B} = \langle Z; \leq \rangle$.
3. $\forall x \forall y\, P(x, y) \to \forall y \forall x\, P(x, y)$.
4. $\forall x \neg P(x, x) \wedge \forall x, y (P(x, y) \to P(y, x)) \wedge \forall x \forall y \forall z (P(x, y) \wedge P(y, z) \to \neg P(x, z))$.
5. $\exists x P(x, y) \wedge \exists x ((\forall y P(x, y) \to \neg \forall y Q(x, y)) \wedge \exists y R(x, y))$.
6. $\phi_1 = \forall x \neg (P_5(x) \wedge (P_2(x) \vee P_4(x)))$,
   $\phi_2 = \forall x \exists y (((P_1(x) \to P_2(x)) \vee (P_3(x, y) \wedge P_4(y)))$,
   $\phi_3 = \exists x \forall y (P_1(x) \wedge ((P_5(x) \vee P_3(x, y) \to (P_5(x) \wedge P_6(y)))))$.
7. $f(x, y) = x + y + 2$.

Вариант 4

1. а) $y, \neg y, x \vdash (x \vee y) \wedge z$;
   б) $x \to y, y \to z \vdash x \to z$;
   в) $x \wedge \neg y, x \wedge y \vdash x \to (y \to x)$;
   г) $(x \wedge y) \vee z \vdash (x \vee z) \wedge x$.
2. $\mathcal{A} = \langle Z; \leq \rangle$, $\mathcal{B} = \langle N; \leq \rangle$.
3. $\exists x \exists y\, P(x, y) \to \exists y \exists x\, P(x, y)$.
4. $\exists x \forall y\, P(x, y) \wedge \neg \forall x \exists x\, P(x, y)$.
5. $\forall x(\exists y P(x, y) \to Q(x, y)) \to (\neg \forall x \exists y Q(x, y) \wedge \exists x R(x, y))$.
6. $\phi_1 = \exists x \forall y \exists z \forall u(P_3(y, x) \to (P_1(y, u) \vee P_2(x, z)))$,
   $\phi_2 = \exists x \forall y \exists z \forall u(P_4(y) \vee (P_3(y, u) \wedge (P_1(x, z) \to P_2(y, u))))$.
7. $f(x, y) = x \cdot y + 1$.

Вариант 5

1. а) $\vdash x \to \neg\neg x$;
   б) $\vdash ((x \to y) \to y) \to x$;
   в) $x, y \vee z, u \vdash (x \wedge y) \to ((z \vee x) \wedge u)$;
   г) $x, u \vee y \vdash (z \to x) \vee (\neg z \to u)$.
2. $\mathcal{A} = \langle R; \cdot \rangle$, $\mathcal{B} = \langle N; \cdot \rangle$.
3. $\forall x \forall y\, P(x, y) \to \forall x\, P(x, x)$.
4. $\forall x \forall y \exists z (P(x, z) \wedge P(z, y)\ \&\ \neg(x = z) \wedge \neg(z = y))$.
5. $\forall x \exists y \neg(P(x, y) \to \neg Q(x, y)) \vee \neg \exists x \forall y\, R(x, y)$.
6. $\phi_1 = \exists x \forall y \forall z \exists u \forall v (\neg P_2(y, x) \wedge \neg(f_2(y, z) = f_3(u, v)))$,
   $\phi_2 = \forall x \forall y \forall z (P_1(f_1(x), x, y) \to (f_2(x, y) = f_3(z, x)))$,
   $\phi_3 = \forall x \forall y \forall z \forall u (P_1(x, y, z) \vee (\neg P_2(f_1(x), u) \to (f_2(y, z) = f_3(u, v))))$.
7. $f(x) = \begin{cases} x - 1, & \text{при } x \neq 0 \\ 0, & \text{при } x = 0. \end{cases}$

Вариант 6

1. а) $x \vee y \vdash (x \wedge y) \to (x \vee z)$;
   б) $x, y \vee u, z \vee y \vdash (y \to x) \vee (u \to y)$;
   в) $\vdash (x \to y) \vee (y \to x)$;
   г) $x \to y, y \to z \vdash (x \wedge z) \to y$.
2. $\mathcal{A} = \langle N; + \rangle$, $\mathcal{B} = \langle Z; + \rangle$.
3. $\exists x\, P(x, x) \to \exists x \exists y\, P(x, y)$.
4. $\neg \forall x \forall y (x = y) \wedge \forall x \forall y (P(x, y) \vee P(y, x))$.
5. $\neg \forall x \exists y\, (P(x, y) \wedge \neg \exists x \forall y Q(x, y)$.
6. $\phi_1 = \exists x \forall y \exists z \forall u \neg (P_3(y, u) \to P_1(x, x, z))$,
   $\phi_2 = \exists x \forall y \forall z \forall u \forall v ((\neg P_1(y, z, u) \wedge P_3(y, z)) \to P_2(y, x, u, v))$.
7. $f(x) = 2x + 1$.

Вариант 7

1. а) $x \to y \vdash \neg y \to \neg x$;
   б) $x \to y \vdash (x \vee z) \to (y \vee z)$;
   в) $x \to (y \to z) \vdash (x \wedge y) \to z$;
   г) $x \wedge y, y \vee z, z \vee \neg u \vdash (x \to u) \vee (z \to y)$.
2. $\mathcal{A} = \langle N; \cdot \rangle$, $\mathcal{B} = \langle Z; \cdot \rangle$.
3. $\exists x\, P(x) \to Q(x)) \leftrightarrow (\forall x\, P(x) \to \exists x Q(x))$.
4. $\exists x \exists y (\neg(x = y) \wedge \forall x \forall y \forall z (P(x, y) \wedge P(y, z) \to P(x, z))$.
5. $\forall x\, P(x, y) \to \neg \exists y \exists x\, Q(x, y)$.
6. $\phi_1 = \forall x \forall y \exists z \neg P(z, x, y)$,
   $\phi_2 = \forall x \forall y \forall z \forall u \forall v \forall w ((P(x, y, u) \wedge P(x, v, w)) \to (P(y, z, v) \vee P(u, z, w)))$,
   $\phi_3 = \forall x \exists z \forall y \neg (P(x, f(x, y), y) \vee P(z, x, z))$.
7. $f(x) = x + 6$.

Вариант 8

1. а) $x \to y \vdash (z \to x) \to (z \to y)$;
   б) $x \to y \vdash (x \wedge z) \to (y \wedge z)$;
   в) $x \vee y, (y \vee \neg z) \vee u \vdash (z \to x) \wedge (x \vee u)$;
   г) $x \vdash y \to x$.
2. $\mathcal{A} = \langle R; + \rangle$, $\mathcal{B} = \langle N; \cdot \rangle$.
3. $(P(x) \to \exists y Q(y)) \leftrightarrow \exists y (P(x) \to Q(y))$.
4. $\exists x \exists y\, P(x, y) \wedge \forall x \forall y (P(x, y) \wedge P(y, x) \to \neg(x = y))$.
5. $\forall x\, (\exists y\, P(x, y) \vee \forall y Q(x, y)) \wedge \exists x \neg (\exists y P(x, y) \vee \forall y Q(x, y))$.
6. $\phi_1 = \forall y (P_1(y) \to (P_2(y) \vee \exists x P_3(x, y)))$,
   $\phi_2 = \neg(\forall x((\exists y (P_1(y) \wedge P_3(x, y)) \to \exists z (P_2(z) \wedge P_3(x, z)))))$.
7. $f(x) = 2x$.

Вариант 9

1. а) $\neg x \to \neg y \vdash y \to x$;
   б) $\neg(x \wedge \neg y) \vdash \neg x \vee y$;
   в) $x \vee y, x \vee z, x \vee \neg z \vee u \vdash u \vee y$;
   г) $x \to (y \vee z) \vdash (x \wedge \neg y) \to z$.
2. $\mathcal{A} = \langle N; + \rangle$, $\mathcal{B} = \langle Z; + \rangle$.
3. $(\forall x\, P(x) \to Q(y)) \leftrightarrow \exists x(P(x) \to Q(y))$.
4. $\exists x \forall y(P(x,y) \to \forall z \neg R(x,y,z)) \wedge \exists x \exists y\, P(x,y)$.
5. $\forall x(\neg(\exists y P(x,y) \to \forall y Q(x,y)) \vee (\exists y P(x,y) \to \forall y Q(x,y)))$.
6. $\phi_1 = \forall x(\exists y(P_1(x,y) \wedge P_2(y)) \to \exists y(P_3(y) \wedge P_4(x,y)))$,
   $\phi_2 = \neg(\neg\exists x P_3(x) \to \forall x \forall y P_1(x,y) \to \neg P_2(y)))$.
7. $f(x) = 2x + 2$.

Вариант 10

1. а) $x \to (\neg z \vee y) \vdash (x \wedge z) \to y$;
   б) $x \vee y, z \vee \neg y \vee u \vdash (x \wedge y) \vee (u \to z)$;
   в) $x \to y, y \to \neg x \vdash$;
   г) $(x \wedge y) \to (x \vee u) \vdash u \to (x \vee y)$.
2. $\mathcal{A} = \langle R; \cdot \rangle$, $\mathcal{B} = \langle Z; \cdot \rangle$.
3. $(\exists x\, P(x) \to Q(y)) \leftrightarrow \forall x (P(x) \to Q(y))$.
4. $\forall x\, P(x, x) \wedge \forall x \exists y (P(x, y) \wedge \neg(x = y))$.
5. $(\neg\exists x \exists y P(x, y) \vee \exists x \forall y Q(x, y)) \wedge \neg\forall x \exists y R(x, y)$.
6. $\phi_1 = \forall x ((P_1(x) \wedge \neg P_2(x)) \to \exists y (P_3(x, y) \wedge P_4(y)))$,
   $\phi_2 = \exists x (P_5(x) \wedge P_4(x) \wedge \forall y (P_3(x, y) \to P_5(y)))$,
   $\phi_3 = \forall x (P_5(x) \to \neg P_2(x))$.
7. $f(x) = x + 5$.

Вариант 11

1. а) $x \vee y, x \to y, y \to z \vdash u \vee z$;
   б) $\vdash (x \wedge y) \to (x \vee y)$;
   в) $x \vee y, \neg x \vee z, u \wedge (x \to z) \vdash (u \wedge y) \to z$;
   г) $y \vdash (y \to x) \to x$.
2. $\mathcal{A} = \langle C; + \rangle$, $\mathcal{B} = \langle N; + \rangle$.
3. $(P(x) \to \forall y Q(y)) \leftrightarrow \forall y (P(x) \to Q(y))$.
4. $\exists x \neg (f(x) = x) \wedge \exists x (f(f(x)) = x)$.
5. $\neg((\exists x \exists y P(x, y) \wedge \exists x \forall y Q(x, y))) \vee \exists x \exists y R(x, y)$.
6. $\phi_1 = \forall x \forall y \exists z ((P_1(x, y) \to P_2(x, y)) \wedge P_3(x, y, z))$,
   $\phi_2 = \forall x \forall y (P_2(x, y) \to P_3(x, y))$,
   $\phi_3 = \exists x \exists y \exists z (P_1(x, y) \wedge \neg P_3(x, y, z))$.
7. $f(x, y) = x + y + 1$.

Вариант 12

1. а) $x \to (y \to z) \vdash (x \to y) \to z$;
   б) $x \to y \vdash (x \vee z) \to (y \vee z)$;
   в) $x \wedge (y \vee z), \neg y \vee z, x \vee \neg z \vee y \vdash y$;
   г) $x \to y \vdash \neg(x \wedge \neg y)$.
2. $\mathcal{A} = \langle C; + \rangle$, $\mathcal{B} = \langle Z; + \rangle$.
3. $(\forall x P(x) \vee \forall x Q(x)) \to \forall x (P(x) \vee Q(x))$.
4. $\forall x (P(x) \to Q(x)) \wedge \exists x \exists y (\neg(x = y) \wedge \neg P(x) \wedge \neg Q(x) \wedge \neg P(y) \wedge \neg Q(y))$.
5. $\exists x (\exists y P(x, y) \vee \forall y Q(x, y)) \to \exists y \forall y (P(x, y) \wedge Q(x, y))$.
6. $\phi_1 = \exists x \forall y \exists z (P_1(x, y) \wedge \neg P_2(x, z) \wedge P_3(x, y, z))$,
   $\phi_2 = \forall x \forall y (P_1(x, y) \to P_2(x, y))$,
   $\phi_3 = \forall x \exists y \forall z (\neg P_3(x, y, z) \wedge \neg P_1(x, y))$.
7. $f(x, y) = \begin{cases} x - y, & \text{при } x \geq y \\ 0, & \text{при } x < y. \end{cases}$

Вариант 13

1. а) $\vdash (x \to z) \vee (z \to x)$;
   б) $y, x \vee u \vdash (z \to y) \vee (\neg z \to y)$;
   в) $\vdash \neg\neg x \to x$;
   г) $x \vee y, x \vee \neg y \vee u, y \vee u \vdash \neg y \vee u$.
2. $\mathcal{A} = \langle C; + \rangle$, $\mathcal{B} = \langle R; + \rangle$.
3. $\exists x(P(x) \wedge Q(x)) \to (\exists x P(x) \wedge \exists x Q(x))$.
4. $\forall x(\neg P(x, x) \wedge \neg P(x, s(x)) \wedge P(x, s(s(x))))$.
5. $\exists x \forall y P(x, y) \to \neg(\forall x \neg(\forall y P(x, y) \wedge \exists y Q(x, y)))$.
6. $\phi_1 = \forall x \exists y(f(x) = f(f(y)) \wedge P_1(x, y))$,
   $\phi_2 = \exists x \forall y \forall z(P_2(x, y) \wedge P_3(x, y, z))$,
   $\phi_3 = \exists x \forall y \forall z(P_1(x, y) \to P_3(x, y, z))$.
7. $f(x, y) = x \cdot y + 2$.

Вариант 14

1. а) $x, \neg x, x \vee u \vdash x \to u$;
   б) $x \to u, u \to z \vdash x \to z$;
   в) $y \vee x \vee u, \neg x \vee \neg y \vee u, \neg u \vee x \vdash u \vee y$;
   г) $\vdash (x \to y) \to ((y \to u) \to (x \to u))$.
2. $\mathcal{A} = \langle C; \cdot \rangle$, $\mathcal{B} = \langle N; \cdot \rangle$.
3. $(P(x) \vee \forall y Q(y)) \leftrightarrow \forall y (P(x) \vee Q(y))$.
4. $\exists x \forall y (P(x, y) \to (\neg P(y, x) \to (P(x, x) \leftrightarrow P(y, y))))$.
5. $\neg(\forall x \neg(\forall y P(x, y) \wedge \exists y Q(x, y))) \to \exists y \forall x R(x, y)$.
6. $\phi_1 = \forall x (f(x) = g(f(x)) \wedge \neg P_1(x, x))$,
   $\phi_2 = \exists x \neg(g(x) = g(g(x)) \wedge \forall x \exists y (\neg P_1(x, y) \wedge P_2(x, y))$,
   $\phi_3 = \forall x \forall y ((P_1(x, y) \to \neg P_1(y, x)) \wedge \exists z P_1(x, z))$.
7. $f(x, y) = x!$, где $0! = 1$.

Вариант 15

1. а) $x, y, y \vee u, u \vee x \vee \neg z \vdash u \vee z$;
   б) $\vdash (x \to ((y \vee x) \to z)) \to (y \to z)$;
   в) $x \vee y \vdash (z \wedge x) \to (y \wedge z)$;
   г) $\neg x, \neg y \vdash (x \to z) \to (y \to z)$.
2. $\mathcal{A} = \langle C; \cdot \rangle$, $\mathcal{B} = \langle Z; \cdot \rangle$.
3. $(P(x) \wedge \exists y Q(y)) \leftrightarrow \exists y (P(x) \wedge Q(y))$.
4. $\exists x \forall y \exists z ((P(y, z) \to P(x, z)) \to (P(x, x) \to P(y, x)))$.
5. $\forall x \exists y P(x, y) \wedge \forall x (\exists y Q(x, y) \wedge \neg(\exists y R(x, y) \vee \exists y P(x, y)))$.
6. $\phi_1 = \exists x \exists y \forall z ((P_1(x, y) \to \neg P_2(y, z)) \wedge (\neg P_2(y, z) \to P_1(x, y)))$,
   $\phi_2 = \forall x \exists y (P_1(x, y) \wedge \neg P_2(y, x))$,
   $\phi_3 = \neg \exists x \exists y (\neg P_1(x, y) \wedge P_2(x, y))$.
7. $f(x) = 4x$.

Вариант 16

1. а) $\vdash x \vee (\neg x \to x)$;

   б) $x \vee \neg y \vdash \neg x \to \neg y$;

   в) $x \wedge y \wedge z \vdash (x \vee y) \to (y \vee z)$;

   г) $x \wedge \neg x \vdash y \to \neg y$.

2. $\mathcal{A} = \langle C; \cdot \rangle$, $\mathcal{B} = \langle R; \cdot \rangle$.

3. $(P(x) \vee \exists y Q(y)) \leftrightarrow \exists y (P(x) \vee Q(y))$.

4. $\exists x P(x,x) \wedge (\forall x \forall y \forall z (P(x,x) \wedge (P(x,z) \to P(x,y) \vee P(y,z))) \to \exists y \forall z P(y,z))$.

5. $\forall x \forall y P(x,y) \vee \forall x (\exists y Q(x,y) \vee (\neg \exists y R(x,y) \wedge \neg \exists y P(x,y)))$.

6. $\phi_1 = \exists x \forall y \exists z \forall u (P_1(x) \wedge ((P_2(x,y) \to \neg P_3(y,z)) \vee \neg (P_1(z) \to P_3(z,u))))$,
   $\phi_2 = \forall x (P_1(x,y) \to \exists y (P_2(x,y) \wedge \neg P_3(x,y)))$.

7. $f(x) = \begin{cases} 2x, & \text{при } x > 2 \\ x, & \text{при } x \leq 2. \end{cases}$

Вариант 17

1. а) $\neg x, \neg y \vdash ((y \to x) \to z) \vee ((\neg z \to y) \to x)$;
   б) $x \vee \neg y \vee u, y \vee \neg u, x \vee z \vdash x \vee y$;
   в) $x, y, z \vdash (x \to y) \vee (y \to z)$;
   г) $x \vee y \vdash (x \to z) \vee (y \to z)$.
2. $\mathcal{A} = \langle R; \cdot \rangle,\ \mathcal{B} = \langle Q; \cdot \rangle$.
3. $(P(x) \wedge \forall y Q(y)) \leftrightarrow \forall y (P(x) \wedge Q(y))$.
4. $\exists x \exists y \exists z \neg((x \cdot y) \cdot z = x \cdot (y \cdot z))$.
5. $\neg(\forall x \forall y P(x, y) \to \forall x \forall y Q(x, y)) \wedge \forall x \forall y P(x, y)$.
6. $\phi_1 = \exists x (\exists y P_1(x, y) \wedge \forall y (P_1(x, y) \to \neg P_2(y, x)))$,
   $\phi_2 = \forall x (\exists y P_1(x, y) \wedge \exists z P_2(x, z))$,
   $\phi_3 = \forall x \exists y \forall z \exists u (P_2(x, y) \wedge P_1(y, z) \wedge P_2(z, u))$.
7. $f(x) = \begin{cases} x - 3, & \text{при } x > 3 \\ 0, & \text{при } x \leq 3. \end{cases}$

Вариант 18

1. а) $x \vee y \vee z \vdash (x \vee y) \to (y \to z)$;
   б) $\vdash (x \wedge y) \to (y \to (z \to y))$;
   в) $x \vee y \vee u, \neg x \vee u \vee \neg y, \neg u \vee y \vdash u \to x$;
   г) $y \to x, x \to \neg z \vdash y \to (x \to \neg z)$.
2. $\mathcal{A} = \langle N; \cdot \rangle$, $\mathcal{B} = \langle Q; \cdot \rangle$.
3. $(\exists x P(x) \vee \exists x Q(x)) \leftrightarrow \exists x (P(x) \vee Q(x))$.
4. $\forall x \neg P(x,x) \wedge \forall x \forall y (P(x,y) \to \neg P(y,x)) \wedge \forall x \forall y \exists z (P(z,x) \wedge P(z,y))$.
5. $\forall x \neg (\forall y P(x,y) \vee \exists y Q(x,y)) \vee \neg \forall x \exists y Q(x,y)$.
6. $\phi_1 = \forall x (P_1(x) \to \neg P_2(x))$,
   $\phi_2 = \forall x \forall y (P_1(x) \to \exists z P_3(x,y,z))$,
   $\phi_3 = \exists x \forall y \forall z \exists u ((P_1(x) \vee P_2(y)) \wedge P_3(x,y,z) \wedge \neg P_3(x,y,u))$.
7. $f(x) = 7x$.

Вариант 19

1. а) $(x \wedge y) \vee z \vdash (x \wedge \neg z) \to (y \vee z)$;
   б) $\vdash x \wedge (y \vee z) \to ((x \wedge y) \to (y \wedge z))$;
   в) $x \vee y \vee z, \neg x \vee \neg z, y \vee \neg x \vdash u \vee x \vee z$;
   г) $(x \to y) \to z \vdash x \to (y \to z)$.
2. $\mathcal{A} = \langle Z; \cdot \rangle$, $\mathcal{B} = \langle Q; \cdot \rangle$.
3. $(\forall x P(x) \wedge \forall x Q(x)) \leftrightarrow \forall x (P(x) \wedge Q(x))$.
4. $\forall x P(x, x, x) \wedge \exists x \exists y (\neg(x = y) \wedge P(x, y, x)) \wedge \neg \forall x \forall y \forall z P(x, y, z)$.
5. $\neg \forall x (\exists y P(x, y) \wedge \exists y Q(x, y)) \wedge \forall x \neg \exists y Q(x, y)$.
6. $\phi_1 = \neg \forall x \exists y (P_1(x, f(x)) \wedge (P_2(x, y) \to \neg P_1(f(x), y)))$,
   $\phi_2 = \exists x \forall y \exists z (\neg(f(x) = f(f(x))) \wedge (P_1(x, y) \to \neg P(y, z)))$,
   $\phi_3 = \forall x (\exists y P_1(x, y) \to \exists z \neg P_2(x, z))$.
7. $f(x, y) = \min\{x, y\}$.

Вариант 20

1. а) $x, \neg y, \neg x \vee y \vdash x \to y$;
   б) $x \vee u \vee \neg z, \neg x \vee z, \neg u \vee y \vdash x \to y$;
   в) $\vdash \neg(x \wedge \neg x)$;
   г) $((x \to y) \to z) \to x \vdash y \vee x \vee z$.
2. $\mathcal{A} = \langle Q; + \rangle$, $\mathcal{B} = \langle Z; + \rangle$.
3. $\neg\exists x P(x) \leftrightarrow \forall x \neg P(x)$.
4. $\forall x \forall y (x \cdot s(y) = s(x \cdot y)) \wedge \exists x \exists y \neg (x \cdot y = y \cdot x)$.
5. $\neg((\exists x \forall y P(x, y) \vee \exists x \exists y Q(x, y)) \vee \exists x \exists y R(x, y))$.
6. $\phi_1 = \exists x \forall y \forall z \neg \phi_1(x, y, z)$,
   $\phi_2 = \forall x \forall y \exists z (P_2(x) \wedge P_1(x, y, z) \wedge (\neg P_2(y) \to P_1(y, x, z)))$,
   $\phi_3 = \forall x \exists y \forall z \exists u (P_1(x, y, z) \wedge \neg P_1(x, y, u))$.
7. $f(x) = 2x + 3$.

Вариант 21

1. а) $x \vee z \vdash \neg x \to z$;
   б) $y, z, \neg u \vdash y \vee z \vee u$;
   в) $y \wedge \neg x, y \wedge x \vdash y \to (x \to y)$;
   г) $(x \wedge z) \vee y \vdash (x \vee y) \wedge x$.
2. $\mathcal{A} = \langle Q; + \rangle$, $\mathcal{B} = \langle N; + \rangle$.
3. $\neg \forall x P(x) \leftrightarrow \exists x \neg P(x)$.
4. $\exists x \exists y \neg (x = y) \wedge \forall x \forall y \exists z (f_1(z) = x \wedge f_2(z) = y)$.
5. $\neg((\exists x \forall y P(x, y) \vee \exists x \exists y (x, y)) \vee \exists x \forall y R(x, y)$.
6. $\phi_1 = \exists x \forall y \forall z \exists u (P(x, y) \wedge \neg (f_1(x, z) = f_2(x, u)))$,
   $\phi_2 = \forall x \forall y \forall z (P(x, y) \to (f_1(x, y) = f_2(x, z)))$,
   $\phi_3 = \forall x \forall y (P(f_1(x, y), y) \to \neg P(f_2(x, y), y))$.
7. $f(x) = 3x + 1$.

Вариант 22

1. а) $y, z \vee u, x \vdash (y \wedge z) \to ((u \vee y) \wedge z)$;
   б) $y \to x \vdash (y \wedge u) \to (x \wedge u)$;
   в) $y \to z, z \to \neg y \vdash$;
   г) $z \vdash (z \to y) \to y$.
2. $\mathcal{A} = \langle N; - \rangle,\ \mathcal{B} = \langle Z; - \rangle$.
3. $\forall x P(x) \leftrightarrow \neg \exists x \neg P(x)$.
4. $\forall x \exists y \exists z (R(y, x) \wedge R(z, x) \wedge \neg (y = z)$.
5. $\forall x \neg (\exists y P(x, y) \vee \forall y Q(x, y) \vee \exists y R(x, y)) \wedge \neg \forall x \forall y R(x, y)$.
6. $\phi_1 = \exists x \forall y \exists z \exists u \neg (P_1(x, y) \wedge \neg (P_2(x, z, u))$,
   $\phi_2 = \exists x \forall y \forall z \forall u (\neg P_1(y, z) \to (P_2(x, y, z) \wedge \neg P_2(z, u, x)))$,
   $\phi_3 = \forall x \forall y (P_1(x, y) \to \neg P_1(y, x))$.
7. $f(x) = 2x + 4$.

Вариант 23

1. а) $\neg(y \wedge \neg z) \vdash \neg y \vee z$;
   б) $x \vee y, x \vee z, x \vee \neg z \vee u \vdash u \vee y$;
   в) $\vdash (y \wedge z) \to (y \vee z)$;
   г) $x \vdash z \to x$.
2. $\mathcal{A} = \langle Q; - \rangle$, $\mathcal{B} = \langle N; - \rangle$.
3. $\exists x P(x) \leftrightarrow \neg \forall x \neg P(x)$.
4. $\forall x \exists y \exists z \exists u (P(x,y) \wedge P(x,z) \wedge P(x,u) \wedge R(y,z,u)) \wedge \forall x \neg R(x,x,x)$.
5. $(\exists x \neg \forall y P(x,y) \wedge \exists x \forall y Q(x,y)) \wedge \exists x \forall y (P(x,y) \to R(x,y))$.
6. $\phi_1 = \forall x (P_1(x) \vee \neg(P_2(x) \wedge P_3(x)))$,
   $\phi_2 = \exists y \forall x (P_4(x,y) \wedge (P_4(y,x) \to \neg(P_1(x) \vee \neg P_2(x))))$,
   $\phi_3 = \forall x (P_3(x) \to \neg \exists y P_4(x,y))$.
7. $f(x) = \mathrm{sg}x = \begin{cases} 0, \text{ если } x = 0 \\ 1, \text{ если } x > 0. \end{cases}$

Вариант 24

1. а) $y \to (z \to u) \vdash (y \to z) \to u$;
   б) $\vdash \neg\neg y \to y$;
   в) $y \vee x \vee u, \neg x \vee \neg y \vee u, \neg u \vee x \vdash u \vee y$;
   г) $x \vee \neg z \vdash \neg y \to \neg z$.
2. $\mathcal{A} = \langle N; \leq \rangle$, $\mathcal{B} = \langle N; P(x,y) \rangle$, где $P(x,y)$ означает, что y делится на x.
3. $\exists x \forall y P(x,y) \to \forall y \exists x P(x,y)$.
4. $\forall x \exists y \exists z \exists u \exists v (R(x,y,z) \wedge R(x,u,v)) \wedge \neg \exists x R(x,x,x)$.
5. $\neg \exists x (\forall y P(x,y) \vee \exists y Q(x,y) \vee \forall x (\exists y P(x,y) \to \forall y Q(x,y)))$.
6. $\phi_1 = \forall x \exists y \exists z \forall u (P_1(x,y) \wedge (P_1(x,y) \to P_2(x,z,u)))$,
   $\phi_2 = \exists x \forall y \forall z (P(x,y,z) \wedge \exists u \exists v (P_2(x,y,z) \to (P_1(x,u) \wedge P_1(v,x))))$,
   $\phi_3 = \forall x \forall y \exists z (P_2(z,x,y) \wedge \neg P_1(z,x))$.
7. $f(x,y) = \begin{cases} x - y, & \text{при } x > y \\ 0, & \text{при } x \leq y. \end{cases}$

Вариант 25

1. а) $y \to x, x \to \neg u \vdash y \to (x \to \neg u)$;
   б) $\vdash (y \wedge z) \to (z \to (x \to z))$;
   в) $y \wedge z \wedge u \vdash (y \vee z) \to (z \vee u)$;
   г) $x \wedge \neg y, x \wedge y \vdash z \to (x \to y)$.
2. $\mathcal{A} = \langle N; \leq \rangle$, $\mathcal{B} = \langle N; P(x,y) \rangle$, где $P(x,y)$ означает, что x и y взаимно просты.
3. $\exists x(P(x) \to Q(x)) \vee (\forall x P(x) \wedge \forall x \neg Q(x))$.
4. $\forall x \neg R(x,x,x) \wedge \exists x \exists y \exists z \exists u \exists v (R(x,y,z) \wedge R(x,u,v) \wedge \neg \exists w R(z,w,v))$.
5. $\neg(\exists x \forall y P(x,y) \to \forall x(\exists y Q(x,y) \to \exists y R(x,y)))$.
6. $\phi_1 = \exists x \forall y \forall z \forall u (P_1(x,y) \wedge (P_2(y,z,x) \wedge \neg P_3(u,x)))$,
   $\phi_2 = \forall x \forall y (P_1(x,y) \to (P_3(x,y) \vee \exists z P_2(x,y,z)))$,
   $\phi_3 = \exists x \exists y (P_1(x,y) \wedge \forall z P_2(x,y,z))$.
7. $f(x,y) = \max(x,y)$.