1 Глобальные предикаты и языки запросов

Мы будем рассматривать реляционные базы данных.

Под *реляционной* сигнатурой мы будем понимать сигнатуру, в которой нет ненульместных символов операций, другими словами, каждый символ этой сигнатуры либо является символом отношения, либо является константой (нульместным символом операции).

Как и в [CT, §7.1], под схемой баз данных мы будем понимать фиксированную реляционную сигнатуру, но всегда будем рассматривать также обогащенную схему, в которую дополнительно включены константа 0 для обозначения нуля, символ $<$ бинарного отношения для обозначения стандартного линейного упорядочения натуральных чисел, символ $'$ унарной операции перехода к следующему натуральному числу и константа $LAST$ для обозначения наибольшего элемента.

Определение 1.1 (стандартной базы данных). *Под стандартной базой данных мы будем понимать любое состояние этой обогащенной схемы, в котором основное множество является начальным отрезком натуральных чисел, $LAST$ интерпретируется как наибольший элемент основного множества, а $<$ — как стандартное отношение линейного порядка на этом начальном отрезке натуральных чисел. Другими словами, мы рассматриваем некоторую реляционную сигнатуру*

$$\langle P_1, \ldots, P_r,\ c_1, \ldots, c_m \rangle \tag{1}$$

и конечные алгебраические системы $\mathcal{A}$ вида

$$\langle \{0,\ 1, \ldots, LAST\}, P_1^{\mathcal{A}},\ P_2^{\mathcal{A}}, \ldots, P_r^{\mathcal{A}},\ c_1^{\mathcal{A}}, \ldots, c_m^{\mathcal{A}},\ <,\ ',\ 0,\ LAST \rangle,$$

в которых $c_i^{\mathcal{A}}$ является выделенным элементом (значением константы c_i), $P_i^{\mathcal{A}}$ является значением в $\mathcal{A}$ символа P_i и является отношением соответствующей местности на множестве $\{0, 1, \ldots, LAST\}$, $<$ является обычным отношением линейного порядка, а одноместная операция $'$ задаётся обычным образом:

$$0' = 1;\ 1' = 2; \ldots; (LAST - 1)' = LAST;\ LAST' = LAST.$$

При этом для $i = 1,\ 2, \ldots, r$ местности $P_i^{\mathcal{A}}$ и P_i совпадают.

$\mathcal{A}$ называется стандартной базой данных для схемы (1).

Надо заметить, что определение стандартной базы данных отличается от определения арифметической системы из [CT, определение 7.4.2] тем, что, кроме наименьшего элемента и операции, дающей следующий элемент, стандартная база данных содержит также отношение, задающее упорядочение элементов. Понятно, что это упорядочение может быть задано некоторой простой программой, использующей наименьший элемент и операцию получения следующего элемента. Но это упорядочение не может быть задано никакой формулой логики предикатов, использующей только символы наименьшего элемента и операции получения следующего элемента. С другой стороны, существуют формулы логики предикатов сигнатуры, содержащей только символ для отношения порядка, определяющие наибольший элемент, наименьший элемент и следующий элемент для заданного элемента. Действительно, следующим является такой элемент, который больше заданного, но между которым и заданным нет других элементов. Поэтому при наличии отношения линейного порядка мы могли бы и не рассматривать операцию получения следующего элемента и константы, выделяющие наибольший и наименьший элементы, но рассматриваем эту операцию и эти константы только для сокращения обозначений.

Определение 1.2 (глобального предиката). *Пусть Ω — произвольная схема баз данных, $\mathfrak{K}$ — некоторый класс ее стандартных баз данных. $\mathfrak{K}$-глобальным n-местным предикатом называется отображение, которое каждой системе $\mathcal{A} \in \mathfrak{K}$ ставит в соответствие некоторый n-местный предикат, заданный на основном множестве системы $\mathcal{A}$. Мы будем опускать $\mathfrak{K}$, когда ясно, о каком классе систем идет речь.*

Нульместные глобальные предикаты называются булевыми.

Обогащая начальную реляционную сигнатуру дополнительными константами

$$a_1, \dots, a_n$$

и рассматривая вместо первоначального класса $\mathfrak{K}$ класс $\mathfrak{K}_n$ стандартных баз данных обогащенной сигнатуры вида

$$(\mathcal{A},\ b_1, \dots, b_n),$$

где $\mathcal{A} \in \mathfrak{K}$, мы можем свести n-местный $\mathfrak{K}$-глобальный предикат к булеву $\mathfrak{K}_n$-глобальному предикату.

Действительно, n-местному $\mathfrak{K}$-глобальному предикату $\mathfrak{G}$ соответствует такой булев $\mathfrak{K}_n$-глобальный предикат $\mathfrak{G}_n$, что

$$\langle b_1, \dots, b_n \rangle \in \mathfrak{G}(\mathcal{A}) \Longleftrightarrow \mathfrak{G}_n((\mathcal{A},\ b_1, \dots, b_n))$$

для любой $\mathcal{A} \in \mathfrak{K}$.

По этой причине далее мы будем рассматривать только булевы глобальные предикаты.

Каждый булев $\mathfrak{K}$-глобальный предикат определяет подкласс класса $\mathfrak{K}$, составленный из тех и только тех систем из $\mathfrak{K}$, на которых этот глобальный предикат истинен. Этот подкласс мы будем отождествлять с этим булевым глобальным предикатом.

Для задания глобальных предикатов, как уже отмечалось в [СТ, §7.1], используются языки запросов, простейшим из которых является язык логики предикатов. Этот язык действительно является практически используемым. Например, некоторой модификацией этого языка является язык **SQL**, используемый в системе **ORACLE** и во многих других промышленных программных продуктах.

Напомним точные определения $\mathcal{P}$-вычислимого глобального предиката, и в частности, определение кода стандартной базы данных.

Определение 1.3 (кода стандартной базы данных). *Рассмотрим схему баз данных*

$$\Omega = \langle R_1^{(j_1)},\ R_2^{(j_2)}, \dots, R_r^{(j_r)},\ c_1, \dots, c_m \rangle,$$

где $c_1, c_2, \dots, c_m$ — константы, а $R_1, R_2, \dots, R_r$ — символы отношений местности, соответственно, $j_1,\ j_2, \dots, j_r$. Мы считаем, что все элементы Ω упорядочены в приведенном порядке.

Пусть $\mathcal{A}$ — некоторая стандартная база данных для схемы Ω, то есть конечное основное множество $|\mathcal{A}|$ вместе с отображением, которое каждому предикатному символу $R_k \in \Omega$ ставит в соответствие j_k -местный предикат

$$R_k^{\mathcal{A}} \subseteq |\mathcal{A}|^{j_k}$$

на множестве $|\mathcal{A}|$, а каждой константе $c_k \in \Omega$ — выделенный элемент $c_k^{\mathcal{A}}$ из $|\mathcal{A}|$.

Как мы договорились, множество $|\mathcal{A}|$ представляет собой начальный отрезок натурального ряда, то есть что, для $n = ||\mathcal{A}||$, $|\mathcal{A}| = \{0,\ 1, \dots, n-1\}$. При этом все элементы $|\mathcal{A}|$ естественным образом упорядочены. Этот порядок на элементах индуцирует и порядок на кортежах элементов, именно, для произвольного $q \in \{0, 1, \dots\}$ и для произвольных

$$\alpha_1, \alpha_2, \dots, \alpha_q, \beta_1, \beta_2, \dots, \beta_q \in |\mathcal{A}|$$

считаем, что

$$\langle \alpha_1, \alpha_2, \dots, \alpha_q \rangle$$

меньше

$$\langle \beta_1, \beta_2, \ldots, \beta_q \rangle,$$

если существует такое $s \le q$, *что*

$$\alpha_1 = \beta_1, \alpha_2 = \beta_2, \ldots, \alpha_{s-1} = \beta_{s-1},$$

но $\alpha_s < \beta_s$. *Пусть*

$$\vartheta_1^q, \vartheta_2^q, \ldots, \vartheta_{n^q}^q$$

перечисляют все кортежи из q *элементов множества*

$$|\mathcal{A}| = \{0,\ 1, \ldots, n-1\},$$

взятые в определенном выше порядке, то есть

$$\vartheta_1^q = \langle 0,\ 0, \ldots, 0 \rangle, \quad \vartheta_2^q = \langle 0,\ 0, \ldots, 0,\ 1 \rangle, \ldots,$$

$$\vartheta_{n^q}^q = \langle n-1,\ n-1, \ldots, n-1 \rangle.$$

Кодом натурального числа $t \in \{0, 1, \ldots\}$ *будем называть слово* $\#(t)$ *в алфавите* $\{0, 1\}$, *представляющее собой двоичную запись числа* t.

Кодом отношения $R_k^{\mathcal{A}} \subseteq |\mathcal{A}|^{j_k}$, *будем называть слово*

$$\#(R_k^{\mathcal{A}}) = \xi_1 \xi_2 \ldots \xi_{n^{j_k}} \in \{0, 1\}^*$$

длины n^{j_k}, *в котором для* $u \le n^{j_k}$

$$\xi_u = \begin{cases} 1, \text{ если } \vartheta_u^{j_k} \in R_k^{\mathcal{A}}; \\ 0, \text{ если } \vartheta_u^{j_k} \notin R_k^{\mathcal{A}}. \end{cases}$$

Наконец, кодом самой стандартной базы данных $\mathcal{A}$ *будем называть слово*

$$\#(\mathcal{A}) \in \{0, 1, \wedge\}^*,$$

где

$$\#(\mathcal{A}) = \wedge \#(LAST) \wedge \#(c_1^{\mathcal{A}}) \wedge \#(c_2^{\mathcal{A}}) \wedge \cdots \wedge \#(c_m^{\mathcal{A}}) \wedge \#(R_1^{\mathcal{A}}) \wedge \cdots \wedge \#(R_r^{\mathcal{A}}) \wedge$$

Отличие от [CT] состоит в том, что в качестве разделителя между кодами отношений и чисел мы теперь используем пробел, а не звёздочку. Ясно, что это различие несущественно.

Напомним теперь определение сложности глобального предиката.

Определение 1.4 (сложности глобального предиката).

Пусть Ω *— некоторая схема баз данных,* $\mathfrak{A}$ *— некоторый класс стандартных баз данных для схемы* Ω*. Пусть* Ξ *— булев* $\mathfrak{A}$*-глобальный предикат. Пусть* $\mathfrak{P}$ *— такой язык в алфавите* $\{0, 1, *\}$*, что*

$$\mathfrak{P} = \{\#(\mathcal{A}) \mid \mathcal{A} \in \Xi\}.$$

Если $\mathcal{C}$ *— некоторый такой класс сложности, что* $\mathfrak{P} \in \mathcal{C}$*, то будем говорить, что* $\mathfrak{A}$*-глобальный предикат* Ξ *имеет сложность* $\mathcal{C}$*.*

Напомним определение классов сложности $PTIME$ и $NPTIME$.

Пусть $f, g : \{0, 1, ..\} \to \{0, 1, ..\}$. Будем говорить, что функция f *асимптотически мажорируется* функцией g (или что g есть *асимптотическая мажоранта* для f), тогда и только тогда, когда

$$(\exists n)(\forall m \geq n) f(m) \leq g(m).$$

Пусть $L \subseteq (\Sigma \setminus \{\wedge\})^*$ — некоторый язык, $u : \{0, 1, \dots\} \to \{0, 1, \dots\}$. Будем говорить, что u есть *верхняя оценка детерминированной (недетерминированной) временной сложности* языка L, если существует такой 1-МТ (соответственно, 1-НМТ) распознаватель M, что M определяет язык L, и функция t_M асимптотически мажорируется функцией u.

Здесь $t_M(n)$ есть время работы M на слове длины не более n в наихудшем останавливающемся случае. Другими словами, для определения $t_M(n)$ мы должны перебрать все останавливающиеся на словах длины не более n пути вычислений M, при которых M не проходит дважды одной и той же конфигурации, и взять максимум по числу тактов.

Класс всевозможных языков в алфавите $(\Sigma \setminus \{\wedge\})$ с верхней оценкой детерминированной временной сложности $f : \{0, 1, \dots\} \to \{0, 1, \dots\}$ обозначается $DTIME(f)$.

Класс всевозможных языков в алфавите $(\Sigma \setminus \{\wedge\})$ с верхней оценкой недетерминированной временной сложности $f : \{0, 1, \dots\} \to \{0, 1, \dots\}$ обозначается $NTIME(f)$.

$$\mathcal{P} = PTIME = \bigcup_{k>0} DTIME(n^k);$$

$$NP = NPTIME = \bigcup_{k>0} NTIME(n^k).$$

Подробное определение 1-МТ и 1-НМТ распознавателей приведено в [CT, §6.1].

Грубо говоря, язык принадлежит к классу $PTIME$, если существует такая машина Тьюринга M с входными, рабочими и выходной лентами и такое натуральное число k, что

- M останавливается на любом входе,
- M делает не более n^k шагов, если длина входа не превосходит n,
- M воспринимает вход в том и только в том случае, когда вход представляет собой слово, принадлежащее рассматриваемому языку.

Язык принадлежит к классу $NPTIME$, если существует такая недетерминированная машина Тьюринга M с входными, рабочими и выходной лентами и такое натуральное число k, что

- если M воспринимает вход, то M воспринимает вход, сделав не более n^k шагов, если длина входа не превосходит n,
- M воспринимает вход в том и только в том случае, когда вход представляет собой слово, принадлежащее рассматриваемому языку.

Теперь мы хотим показать, что глобальные предикаты, задаваемые формулами логики предикатов, $\mathcal{P}$-вычислимы.

Итак, пусть Ω — реляционная сигнатура, Ω' — расширенная сигнатура, полученная добавлением символа для отношения линейного порядка к сигнатуре Ω. $L_{\Omega'}$ — язык логики предикатов в сигнатуре Ω', $\varphi \in L_{\Omega'}$ — формула этого языка, а список $x_1, x_2, \ldots, x_n$ задает все свободные переменные формулы φ (мы предполагаем, что множество всех переменных языка $L_{\Omega'}$ линейно упорядочено и перечисляем переменные в этом порядке). Всякая такая формула φ называется *элементарным запросом* (к базе данных со схемой Ω). При этом замкнутые формулы задают булевы элементарные запросы.

Пусть теперь $\mathcal{A}$ — некоторая стандартная база данных для схемы Ω, $|\mathcal{A}|$ — ее основное множество (носитель).

Напомним, что оценкой в системе $\mathcal{A}$ или состоянием системы $\mathcal{A}$ называется произвольное отображение σ множества переменных в носитель системы $\mathcal{A}$.

Ответом на запрос φ для базы данных $\mathcal{A}$ называется n-местный предикат $\varphi_{\mathcal{A}}$, определенный так:

$$\langle a_1, a_2, \ldots, a_n\rangle \in \varphi_{\mathcal{A}} \Leftrightarrow \sigma_{a_1,a_2,\ldots,a_n} \models \varphi$$

для произвольных $a_1, a_2, \ldots, a_n \in |\mathcal{A}|$, где

$$\sigma_{a_1,a_2,\ldots,a_n}(x) = \begin{cases} a_i, & \text{если } x \text{ есть } x_i \text{ и } 0 < i \leq n; \\ a_1 & \text{в остальных случаях.} \end{cases}$$

Другими словами, предикат $\varphi_{\mathcal{A}}$ определяется как множество точек истинности формулы φ в системе $\mathcal{A}$.

Ясно, что формула φ определяет предикат во всякой стандартной базе данных для схемы Ω, то есть глобальный предикат. Каждый глобальный предикат, который можно так задать некоторой формулой логики предикатов, называется *элементарным.*

Теорема 1.1. *Пусть $\mathfrak{A}$ — класс всех стандартных баз данных для схемы Ω и для булевого элементарного запроса ϕ пусть*

$$\mathfrak{P}_\phi = \{\#(\mathcal{A}) \mid \mathcal{A} \in \mathfrak{A}, \mathcal{A} \models \phi\}.$$

Тогда $\mathfrak{P}_\phi \in \mathcal{P} = PTIME$.

Неформальное доказательство. Не ограничивая общности, можно считать, что формула ϕ предваренная. Индукция по числу кванторов. Если кванторов нет, формула ϕ эквивалентна дизъюнкции конъюнкций базисных формул. Базисная формула — это либо атомная формула, либо отрицание атомной формулы. Проверка истинности сводится к проверке истинности нескольких конъюнкций базисных формул. Если хотя бы одна из конъюнкций истинна, то истинна и рассматриваемая формула. Проверка каждой конъюнкции сводится к проверке истинности нескольких базисных формул. Если все эти базисные формулы истинны, то истинна и их конъюнкция. В противном случае эта конъюнкция ложна. Для вычисления истинности $P(d_1, \ldots, d_k)$, где P — символ отношения местности k из схемы базы данных $\mathcal{A}$, надо использовать хранящийся на входной ленте код этой базы данных.

Для определения истинности формулы $(\exists x)\psi$ следует проверить истинность формул $(\psi)^x_j$ для $j = 0, \ldots, LAST$. Если хотя бы одна из этих формул истинна, то истинна и формула $(\exists x)\psi$. В противном случае эта формула ложна. ∎

2 Транзитивное замыкание. Подалгебра

Определение 2.1 (транзитивного отношения). *Бинарное отношение Q на множестве A называется транзитивным, если для любых* *элементов a, b и c множества A из истинности $Q(a,b)$ и $Q(b,c)$ следует истинность $Q(a,c)$.*

Теорема 2.1. *Пересечение любого множества транзитивных на A отношений является транзитивным на A отношением.*

Доказательство. Если для каких-то трех элементов пара первых из них и пара последних из них принадлежат пересечению, то обе эти пары принадлежат каждому отношению и, в силу транзитивности, пара, составленная из первого и последнего элементов, тоже принадлежит каждому отношению и, значит, пересечению всех рассматриваемых отношений. ∎

Всегда существует транзитивное отношение на A, содержащее заданное бинарное отношение Q на A. Действительно, тождественное отношение на A, которое истинно для любой пары элементов A, является транзитивным. Из теоремы следует, что пересечение всех транзитивных отношений, содержащих заданное бинарное отношение, является транзитивным отношением. Это пересечение является наименьшим транзитивным отношением, содержащим заданное отношение. Если заданное отношение само транзитивно, то это пересечение совпадает с заданным отношением.

Определение 2.2 (транзитивного замыкания). *Пересечение всех транзитивных отношений, содержащих заданное бинарное отношение, называется транзитивным замыканием заданного отношения.*

Пусть теперь множество A конечно, а Q является бинарным отношением на A. Тогда следующая конструкция строит транзитивное замыкание отношения Q.

Теорема 2.2 (о транзитивном замыкании). *Пусть R_0 — пусто, а R_{i+1} содержит те и только те пары (a,b), для которых существует такой элемент c множества A, что либо истинно $Q(a,b)$, либо одновременно $R_i(a,c)$ и $Q(c,b)$ истинны.*

Существует такое i, что $R_i = R_{i+1}$. Для этого i отношение R_i является транзитивным замыканием отношения Q.

Доказательство. Индукцией по j докажем, что $R_j \subseteq R_{j+1}$.

Это верно для $j = 0$, так как R_0 является пустым.

Если $R_j \subseteq R_{j+1}$, то $R_{j+1} \subseteq R_{j+2}$.

Так как $R_j \subseteq R_{j+1}$ для любого j, то последовательность

$$R_0,\ R_1, \ldots, R_j, \ldots$$

является неубывающей последовательностью подмножеств конечного множества A^2. По этой причине существует такое i, что $R_i = R_{i+1}$. Для этого i и любого j выполняется равенство

$$R_i = R_{i+j}.$$

Покажем, что R_i является транзитивным замыканием отношения Q.

Ясно, что R_j содержится в транзитивном замыкании Q для любого j.

Действительно, это верно для $j = 0$ в силу пустоты R_0. Пусть T является транзитивным замыканием отношения Q и R_j содержится в T. Покажем, что R_{j+1} тоже содержится в T. Действительно, $R_{j+1}(a, b)$ эквивалентно существованию такого c, что либо $Q(a, b)$, либо одновременно $R_j(a, c)$ и $Q(c, b)$. Если $Q(a, b)$, то $T(a, b)$. Пусть выполняется вторая возможность. По индукционному предположению и определению транзитивного замыкания, из этого следует, что $T(a, c)$ и $T(c, b)$. Из транзитивности T и этого следует $T(a, b)$.

Покажем, что $T \subseteq R_i$. По определению, T является пересечением всех транзитивных отношений, содержащих Q. Остается заметить, что R_i является транзитивным отношением.

Пусть $R_i(a, d)$ и $R_j(d, b)$. Индукцией по j докажем, что тогда $R_i(a, b)$.

При $j = 0$ утверждение очевидно, так как R_0 пусто. Пусть утверждение верно для некоторого j и любых $a,\ d,\ b$.

$R_{j+1}(d, b)$ означает, что либо $Q(d, b)$, либо одновременно $R_j(d, c)$ и $Q(c, b)$ для некоторого c. В первом случае получаем $R_{i+1}(a, b)$ и $R_i(a, b)$. Во втором случае получаем $R_i(a, c)$ и $Q(c, b)$, что влечет $R_{i+1}(a, b)$. Но $R_{i+1} = R_i$. Значит, $R_i(a, b)$.

Итак, $R_i = T$. ∎

Будем теперь рассматривать схему базы данных

$$\langle P,\ c_1,\ c_2 \rangle,$$

где P является символом бинарного отношения, а $c_1,\ c_2$ — константами. Базы данных для этой схемы являются упорядоченными графами с двумя выделенными элементами. Для такого графа $\mathcal{G}$ через $T^{\mathcal{G}}$ обозначим транзитивное замыкание отношения, которое соответствует символу P в $\mathcal{G}$.

Рассмотрим глобальный предикат $\mathfrak{T}$ на упорядоченных графах с двумя выделенными элементами, истинный на таком графе $\mathcal{G}$ тогда и только тогда, когда $T^{\mathcal{G}}(c_1^{\mathcal{G}}, c_2^{\mathcal{G}})$ истинно.

Теорема 2.3 (полиномиальность транзитивного замыкания).

$$\mathfrak{T} \in \mathcal{P} = PTIME.$$

Неформальное доказательство. При построении транзитивного замыкания фактически делается не более квадрата числа элементов шагов. На каждом шаге добавляются все такие пары (a, b), для которых существует такой элемент c, что пара (a, c) уже получена на предыдущем шаге, а пара (c, b) лежит в $P^{\mathcal{G}}$. Имея код отношения, вычисленного на предыдущем шаге, и код $P^{\mathcal{G}}$, за полиномиальное время делаем следующий шаг. ∎

Пусть f является операцией местности n на множестве A. С этой операцией можно связать $(n+1)$-местное отношение P_f на этом же множестве A следующим образом:

для произвольного набора $a_1, \ldots, a_n, b$ элементов множества A истинность $P_f(a_1, \ldots, a_n, b)$ равносильна истинности равенства

$$f(a_1, \ldots, a_n) = b.$$

Разумеется, не всякое $(n+1)$-отношение P на A задает некоторую n-местную операцию на A. Необходимыми и достаточными условиями для этого являются:

- для любых $a_1, \ldots, a_n, b, c$ из $P(a_1, \ldots, a_n, b)$ и $P(a_1, \ldots, a_n, c)$ следует, что $b = c$;

- для любых $a_1, \ldots, a_n$ существует такой элемент b в A, для которого выполняется $P(a_1, \ldots, a_n, b)$.

Мы будем говорить, что отношение *представляет операцию*, если оно удовлетворяет указанным необходимым и достаточным условиям.

Так как эти условия можно задать формулами логики предикатов, то с точки зрения изучения выразительных возможностей можно не рассматривать вообще сигнатур, содержащих символы операций, заменяя их символами отношений и рассматривая только такие интерпретации этих добавленных символов отношений, для которых указанные формулы истинны. Однако с других точек зрения наличие символов операций существенно меняет дело.

Если наоборот сигнатура не содержит вообще символов отношений, то алгебраическая система такой сигнатуры называется *алгеброй*.

Подмножество алгебраической системы *замкнуто относительно некоторой операции*, определенной на основном множестве этой системы, если из того, что какие-то аргументы лежат в этом подмножестве, следует, что и значение рассматриваемой операции на этом наборе аргументов лежит в этом подмножестве. В частности, для константы это означает, что рассматриваемое подмножество содержит значение этой константы.

Операции, соответствующие сигнатурным символам операций в рассматриваемой алгебраической системе, называются *основными операциями* этой алгебраической системы. При этом элементы, соответствующие сигнатурным константам, называются также выделенными элементами этой алгебраической системы.

Определение 2.3 (подалгебры, подсистемы). *Непустое подмножество алгебры называется подалгеброй, если оно замкнуто относительно всех основных операций этой алгебры. Непустое подмножество алгебраической системы называется подсистемой, если оно замкнуто относительно всех основных операций этой системы.*

Если сигнатура не содержит символов операций, всякое непустое подмножество является подсистемой. В общем же случае всегда существует наименьшая подсистема, содержащая заданное непустое подмножество основного множества рассматриваемой алгебраической системы. Эта подсистема определяется как пересечение всех подсистем, содержащих рассматриваемое подмножество, и называется *подсистемой, порожденной этим подмножеством.* Для понимания корректности этого определения достаточно доказать следующую теорему.

Теорема 2.4. *Пересечение любого множества подсистем, содержащих заданное непустое подмножество рассматриваемой алгебраической системы, является подсистемой этой системы.*

Доказательство. Если какой-то набор элементов принадлежат пересечению, то этот набор принадлежат каждой подсистеме. Значит, значение некоторой основной операции на этом наборе принадлежит каждой подсистеме и потому принадлежит пересечению всех рассматриваемых подсистем. ∎

Пусть теперь множество A конечно, а $\mathcal{A}$ является алгебраической системой, у которой A является основным множеством. Пусть X является непустым подмножеством множества A. Тогда следующая конструкция строит подсистему, порожденную X в $\mathcal{A}$.

Теорема 2.5 (о порожденной подсистеме). *Пусть R_0 — пусто, а R_{i+1} содержит те и только те элементы x из A, которые либо входят в X, либо являются выделенными элементами в $\mathcal{A}$, либо для которых существуют n, такая n-местная основная операция f в $\mathcal{A}$ и такие элементы $a_1, \ldots, a_n$ из R_i, что $x = f(a_1, \ldots, a_n)$.*

Существует такое i, что $R_i = R_{i+1}$. Для этого i множество R_i является подсистемой, порожденной X в $\mathcal{A}$.

Доказательство. Индукцией по j докажем, что $R_j \subseteq R_{j+1}$.

Это верно для $j = 0$, так как R_0 является пустым.

Если $R_j \subseteq R_{j+1}$, то $R_{j+1} \subseteq R_{j+2}$.

Так как $R_j \subseteq R_{j+1}$ для любого j, то последовательность

$$R_0,\ R_1, \ldots, R_j, \ldots$$

является неубывающей последовательностью подмножеств конечного множества A. По этой причине существует такое i, что $R_i = R_{i+1}$. Для этого i и любого j выполняется равенство

$$R_i = R_{i+j}.$$

Покажем, что R_i является подсистемой, порожденной X в $\mathcal{A}$.

Ясно, что R_j содержится в этой подсистеме для любого j.

Действительно, это верно для $j = 0$ в силу пустоты R_0. Пусть T является подсистемой, порожденной X, и R_j содержится в T. Покажем, что R_{j+1} тоже содержится в T. Действительно, $R_{j+1}(a)$ эквивалентно существованию такого n и таких $a_1, \ldots, a_n$ в R_j, что либо a является выделенным элементом или элементом X, либо $a = f(a_1, \ldots, a_n)$ для некоторой основной n-местной операции f. Если выполняется первая возможность, $a \in T$ по определению подсистемы, порожденной X. Пусть выполняется вторая возможность. По индукционному предположению и определению подсистемы, $a \in T$.

Покажем, что $T \subseteq R_i$. По определению, T является пересечением всех подсистем, содержащих X. Остается заметить, что R_i является подсистемой.

R_i содержит все выделенные элементы $\mathcal{A}$. Пусть $a_1, \ldots, a_n$ содержатся в R_i и f является n-местной основной операцией.

Тогда $f(a_1, \ldots, a_n)$ лежит в $R_{i+1} = R_i$.

Итак, $R_i = T$. ∎

Рассмотрим глобальный предикат $\mathfrak{T}_m$ на стандартных базах данных для схемы, полученной из какой-то сигнатуры заменой символов ненульместных операций на символы представляющих их отношений и обогащенной $(m+1)$-ой дополнительной константой. Интерпретации этих дополнительных констант будем называть дополнительными выделенными элементами. Каждой такой базе данных соответствует конечная алгебраическая система первоначальной сигнатуры, полученная заменой представляющих отношений на представляемые ими операции и отбрасыванием упорядочения, первого и последнего элементов, операции получения следующего элемента и дополнительных выделенных элементов. Этот глобальный предикат истинен на такой базе данных тогда и только тогда, когда в соответствующей алгебраической системе первоначальной сигнатуры последний дополнительный выделенный элемент принадлежит подсистеме, порожденной остальными дополнительными выделенными элементами.

Теорема 2.6 (полиномиальность порожденной подсистемы).

$$\mathfrak{T}_m \in \mathcal{P} = PTIME.$$

Неформальное доказательство. При построении порожденной подсистемы фактически делается не более числа элементов шагов. На каждом шаге добавляются все такие элементы, которые получаются как значения основных операций на наборах аргументов, принадлежащих подмножеству, полученному на предыдущем шаге. Имея код подмножества, вычисленного на предыдущем шаге, и код стандартной базы данных, за полиномиальное время делаем следующий шаг. ∎

3 Игры Эренфойхта

Для доказательства невыразимости некоторых глобальных предикатов в языке логики предикатов нам понадобится некоторое достаточное условие неразличимости двух алгебраических систем заданной формулой логики предикатов. Для формулировки этого условия используют так называемые игры Эренфойхта.

Рассмотрим реляционную сигнатуру Ω, состоящую из констант

$$c_1,\ c_2, \ldots, c_m$$

и символов отношений. Рассмотрим две алгебраических системы $\mathcal{A}$ и $\mathcal{B}$ этой сигнатуры. Через $c_i^{\mathcal{A}}$ и $c_i^{\mathcal{B}}$ будем обозначать значения константы c_i в системах $\mathcal{A}$ и $\mathcal{B}$ соответственно.

Определение 3.1 (локального мономорфизма $\mathcal{A}$ в $\mathcal{B}$). *Пусть $a_1, \ldots, a_n$ взяты из $\mathcal{A}$ и $b_1, \ldots, b_n$ взяты из $\mathcal{B}$. Пусть τ одно-однозначно отображает*

$$\{a_1, \ldots, a_n,\ c_1^{\mathcal{A}}, \ldots, c_m^{\mathcal{A}}\}$$

на

$$\{b_1, \ldots, b_n,\ c_1^{\mathcal{B}}, \ldots, c_m^{\mathcal{B}}\}$$

так, что

$$\tau(a_1) = b_1, \ldots, \tau(a_n) = b_n,\ \tau(c_1^{\mathcal{A}}) = c_1^{\mathcal{B}}, \ldots, \tau(c_m^{\mathcal{A}}) = c_m^{\mathcal{B}}.$$

Это отображение τ называется локальным мономорфизмом $\mathcal{A}$ в $\mathcal{B}$, если для каждого символа отношения P из Ω и каждых $d_1, \ldots, d_k$ из

$$\{a_1, \ldots, a_n,\ c_1^{\mathcal{A}}, \ldots, c_m^{\mathcal{A}}\}$$

выполняется $P^{\mathcal{A}}(d_1, \ldots, d_k)$ в $\mathcal{A}$ тогда и только тогда, когда выполняется $P^{\mathcal{B}}(\tau(d_1), \ldots, \tau(d_k))$ в $\mathcal{B}$, где k есть число аргументных мест P.

Определение 3.2 (n-игры Эренфойхта на $\mathcal{A}$ и $\mathcal{B}$). *В n-игре Эренфойхта участвуют два игрока. Первый называется Разрушителем, а второй — Повторителем. Игра состоит из n ходов. На каждом ходе Разрушитель выбирает один элемент в одной из этих систем, а Повторитель после этого выбирает один элемент в другой из этих систем. Пусть в результате игры в системах $\mathcal{A}$ и $\mathcal{B}$ выбраны соответственно элементы $a_1, \ldots, a_n$ и $b_1, \ldots, b_n$. Пусть τ одно-однозначно отображает*

$$\{a_1, \ldots, a_n,\ c_1^{\mathcal{A}}, \ldots, c_m^{\mathcal{A}}\}$$

на

$$\{b_1, \ldots, b_n,\ c_1^{\mathcal{B}}, \ldots, c_m^{\mathcal{B}}\}$$

так, что

$$\tau(a_1) = b_1, \ldots, \tau(a_n) = b_n,\ \tau(c_1^{\mathcal{A}}) = c_1^{\mathcal{B}}, \ldots, \tau(c_m^{\mathcal{A}}) = c_m^{\mathcal{B}}.$$

Повторитель выиграл, если это отображение τ является локальным мономорфизмом. В противном случае выиграл Разрушитель. Повторитель имеет выигрышную стратегию в n-игре Эренфойхта на $\mathcal{A}$ и $\mathcal{B}$, если он может выиграть при любой игре Разрушителя.

Теорема 3.1 (об n-игре Эренфойхта). *Если Повторитель имеет выигрышную стратегию в n-игре Эренфойхта на $\mathcal{A}$ и $\mathcal{B}$, то $\mathcal{A}$ и $\mathcal{B}$ не различаются замкнутыми предваренными формулами, содержащими не более n кванторов.*

Доказательство. Индукцией по n.

При $n = 0$, если Повторитель имеет выигрышную стратегию в 0-игре, то τ является локальным мономорфизмом. Это означает, что рассматриваемые системы не различаются замкнутыми атомными формулами (такая формула имеет вид $P(d_1, \ldots, d_k)$ для какого-то P из Ω и каких-то $d_1, \ldots, d_k$ из $\{c_1, \ldots, c_m\}$, где k обозначает число аргументных мест у P, либо имеет вид $c_i = c_j$), значит, $\mathcal{A}$ и $\mathcal{B}$ не различаются и замкнутыми бескванторными формулами.

Пусть теорема верна для n. Докажем её для $n+1$. Пусть Повторитель имеет выигрышную стратегию в $(n+1)$-игре на $\mathcal{A}$ и $\mathcal{B}$. Замкнутая предваренная формула Ψ с $n+1$ квантором, неразличимость относительно которой систем $\mathcal{A}$ и $\mathcal{B}$ требуется доказывать, для некоторого переменного x и некоторой предваренной формулы Φ с n кванторами либо сама имеет вид $(\exists x)\Phi$, либо такой вид эквивалентен отрицанию этой формулы Ψ. Так как неразличимость относительно отрицания совпадает с неразличимостью относительно самой формулы, то мы будем предполагать, что такой вид имеет сама формула. Будем предполагать, что Ψ истинна в $\mathcal{A}$, и докажем, что эта формула истинна и в $\mathcal{B}$. Обратный случай сводится к этому простым переименованием $\mathcal{A}$ и $\mathcal{B}$.

Так как $(\exists x)\Phi$ истинна в $\mathcal{A}$, то существует такой элемент a_0 в $\mathcal{A}$, что Φ истинна для каждого такого состояния σ системы $\mathcal{A}$, для которого $\sigma(x) = a_0$. Пусть константа c_0 не входит в Ω. Это означает, что $(\Phi)^x_{c_0}$ истинна в обогащении $(\mathcal{A},\ a_0)$ системы $\mathcal{A}$, в котором c_0 интерпретируется как a_0. $(\Phi)^x_{c_0}$ получается из Φ заменой свободных вхождений x на c_0.

Пусть Разрушитель выбирает на первом шаге игры элемент a_0 в $\mathcal{A}$. Используя свою выигрышную стратегию, Повторитель может так выбрать элемент b_0 в $\mathcal{B}$, что дальше Повторитель имеет выигрышную стратегию в n-игре на $(\mathcal{A},\ a_0)$ и $(\mathcal{B},\ b_0)$.

По индукционному предположению, последние две системы не различаются замкнутыми предваренными формулами с n кванторами. Значит, $(\Phi)^x_{c_0}$ истинна в $(\mathcal{B},\ b_0)$. Это означает, что Ψ истинна в $\mathcal{B}$. ∎

Справедлива и обратная теорема о том, что если $\mathcal{A}$ и $\mathcal{B}$ не различаются замкнутыми предваренными формулами, содержащими не более n кванторов, то Повторитель имеет выигрышную стратегию в n-игре

Эренфойхта на $\mathcal{A}$ и $\mathcal{B}$. Но нам эта обратная теорема не понадобится, и мы не будем её доказывать.

В качестве примера использования игр Эренфойхта докажем, что чётность носителя не выражается никакой замкнутой формулой логики предикатов в классе конечных линейных порядков.

Предварительно договоримся о терминологии. 0 обозначает минимальный элемент в рассматриваемом конечном линейном порядке, $[c,d] = \{x \mid c \leqslant x \leqslant d\}$, $|c-d|-1$ обозначает число отличных от c и d элементов множества $[c,d]$ в этом линейном порядке, $c+1$ обозначает элемент, следующий сразу за c в этом линейном порядке, $c-1$ обозначает элемент, расположенный непосредственно перед c в этом линейном порядке, $c+(n+1)$ есть $(c+n)+1$, $c-(n+1)$ есть $(c-n)-1$, наконец, n обозначает $0+n$. Например, $|(c+1)-c| = 1$.

В конечном линейном порядке каждый элемент однозначно представляется некоторым натуральным числом. Элемент чётен, если чётным является представляющее этот элемент натуральное число.

Пусть $Order$ обозначает класс всех конечных алгебраических систем сигнатуры $\langle < \rangle$, которые являются линейными порядками. Другими словами, $Order$ состоит из всех алгебраических систем сигнатуры $\langle < \rangle$, в которых $<$ интерпретируется транзитивным, тотальным и антисимметричным бинарным отношением.

Лемма 3.2. *Не существует такой $\langle < \rangle$-формулы $\phi(x)$, что для любой $\mathcal{A} \in Order$ и любого $a \in |\mathcal{A}|$ выполняется $(A,a) \models \phi$ тогда и только тогда, когда a чётно.*

Короче, чётность не выражается никакой формулой логики предикатов в классе $Order$.

Доказательство немедленно следует из следующей леммы.

Лемма 3.3. *Пусть $\mathcal{A}, \mathcal{B} \in Order$ и пусть $\bar{a} = a_1, a_2, \ldots, a_k$, $\bar{b} = b_1, b_2, \ldots, b_k$ являются соответственно последовательностями элементов линейных порядков $\mathcal{A}$ и $\mathcal{B}$, а элементы этих последовательностей образуют подмножества A' и B'. Пусть ещё эти подмножества включают наименьшие и наибольшие элементы этих линейных порядков $\mathcal{A}$ и $\mathcal{B}$.*

Для любых $i, j \leq k$ пусть $a_i < a_j$ тогда и только тогда, когда $b_i < b_j$, и пусть расстояния $|a_i - a_j|$ и $|b_i - b_j|$ либо равны, либо оба больше, чем 2^n.

Тогда Повторитель имеет выигрышную стратегию в n-игре Эренфойхта на $(\mathcal{A}, \bar{a})$, $(\mathcal{B}, \bar{b})$.

Доказательство. . Очевидно для $n = 0$.

Пусть $n > 0$. Докажем, что Повторитель имеет выигрышную стратегию в n-игре Эренфойхта на

$$(\mathcal{A}, \bar{a}), (\mathcal{B}, \bar{b}).$$

Пусть $a \in |\mathcal{A}|$ — ход Разрушителя (случай, когда Разрушитель выбирает элемент в $\mathcal{B}$, рассматривается аналогично). Если существуют такие $j \leq k$ и $r \in [-2^{n-1}, 2^{n-1}]$, что $a = a_j + r$, стратегия Повторителя состоит в выборе $b = b_j + r$. Иначе пусть a_j будет таким наибольшим элементом в $\bar{a}$, что $a_j < a$. В этом случае стратегия Повторителя состоит в выборе $b = b_j + 2^{n-1} + 1$.

Легко проверить, что условия леммы выполнены для $\bar{a}, a, \bar{b}, b$ и $n - 1$. ∎

Применяя лемму 3.3, допустим, что чётность выражается предварённой формулой с n кванторами с одной свободной переменной. Рассмотрим такой конечный линейный порядок $\mathcal{A} \in Order$, что $m = |\mathcal{A}| > 2^{n+1} + 3$. По лемме 3.3, Повторитель имеет выигрышную стратегию в n-игре Эренфойхта на $(\mathcal{A}, 2^n + 1, 0, m - 1)$ и $(\mathcal{A}, 2^n + 2, 0, m - 1)$. Но рассматриваемая формула различает эти линейные порядки. Это противоречит теореме 3.1.

4 Неполнота FO для $PTIME$

Целью настоящего параграфа является доказательство теоремы о том, что транзитивное замыкание не выразимо в языке FO логики предикатов. Для краткости мы будем формулы логики предикатов называть FO формулами.

Рассмотрим для положительного r две конечные стандартные базы данных $\mathcal{A}_r$ и $\mathcal{B}_r$ для схемы, содержащей символ P бинарного отношения и две константы c_0 и c_1.

Основное множество обеих баз данных представляет собой множество

$$A_r \;=\; B_r \;=\; \{\langle i, j\rangle \mid i, j \leqslant 2^r\}.$$

Таким образом, каждая из баз данных содержит ровно $(2^r + 1)(2^r + 1)$ элементов.

Формально мы нарушаем при этом условие, что основным множеством стандартной базы данных должен быть начальный отрезок натуральных чисел. Однако это условие нужно только для того, чтобы стандартным образом определить наименьший, наибольший элементы и линейный порядок. Если это сделать отдельно, то условие становиться несущественным.

Интерпретацией c_0 в обеих базах данных является $a_0 = b_0 = \langle 2^{r-1}, 0\rangle$. Интерпретацией c_1 в $\mathcal{A}_r$ является $a_1 = \langle 2^{r-1}, 2^r\rangle$, а интерпретацией c_1 в $\mathcal{B}_r$ является $b_1 = \langle 2^{r-1} - 1, 2^r\rangle$. Линейный порядок в обеих базах данных определяется одинаково и задаётся лексикографически:

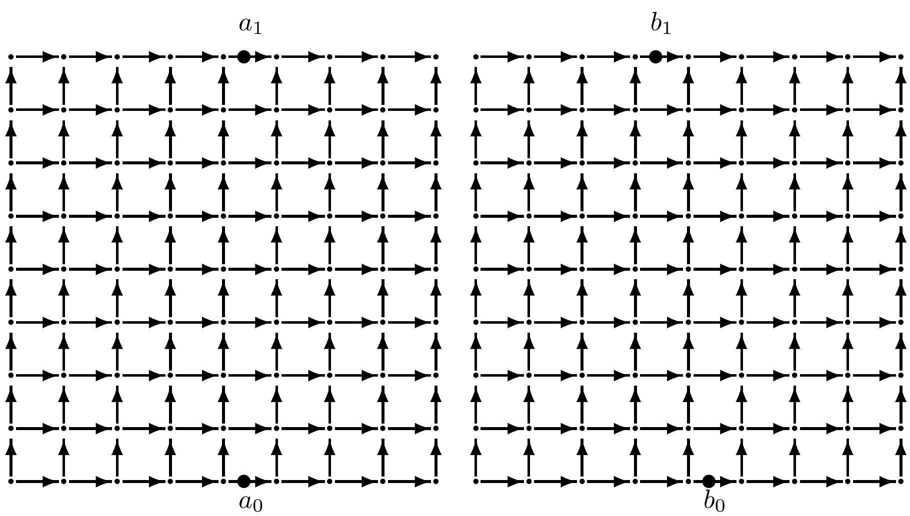
$$\langle i, j\rangle < \langle u, v\rangle \iff (i < u \text{ или } (i = u \text{ и } j < v)).$$

Это означает, что точка, лежащая выше, больше, а из точек, лежащих на одном уровне, больше та, которая правее. Отношение P определяется в обеих базах данных одинаково:

$$P^{\mathcal{A}}(\langle i, j\rangle, \langle u, v\rangle) \Leftrightarrow P^{\mathcal{B}}(\langle i, j\rangle, \langle u, v\rangle) \Leftrightarrow \\ ((i = u \text{ и } v = j + 1) \text{ или } (u = i + 1 \text{ и } v = j)).$$

Это означает, что вторая точка либо лежит в той же строке, но в следующем столбце, либо лежит в том же столбце, но в следующей строке.

Для $r = 3$ обе системы приведены на следующем рисунке:



Система $\mathcal{A}_3$ **Система $\mathcal{B}_3$**

Очевидно, что $\mathfrak{T}(\mathcal{A}_r)$ истинно, но $\mathfrak{T}(\mathcal{B}_r)$ ложно для любого положительного r.

Теорема 4.1 (неэлементарность транзитивного замыкания).
Глобальный предикат $\mathfrak{T}$ не является элементарным.

Доказательство. От противного. Пусть этот глобальный предикат задается FO формулой ϕ. Тогда $\mathcal{A}_r \models \phi$ и $\mathcal{B}_r \models \neg\phi$ для любого положительного r.

Можно считать, что формула ϕ является предваренной и содержит s кванторов [CT, теоремы 3.24 и 3.28]. Мы получим противоречие, доказав, что $\mathcal{A}_{2s}$ и $\mathcal{B}_{2s}$ на самом деле не различаются никакой предваренной формулой, содержащей не более s кванторов. По теореме 3.1, для этого достаточно доказать, что Повторитель имеет выигрышную стратегию в s-игре Эренфойхта на $\mathcal{A}_{2s}$ и $\mathcal{B}_{2s}$.

Опишем эту стратегию. Расстоянием между $\langle\alpha,\gamma\rangle$ и $\langle\beta,\delta\rangle$ равно

$$|\alpha-\beta|+|\gamma-\delta|.$$

Мы будем использовать на каждом шаге i игры два числа — радиус $\rho(i)$ для этого шага и сдвиг $\sigma(i)$ для выбираемых на этом шаге элементов. Мы полагаем: $\rho(i)=2^{s-i}$. На шаге i выбираются элементы $a_{i+1}=\langle\alpha_{i+1},\gamma_{i+1}\rangle$ из $\mathcal{A}_{2s}$ и $b_{i+1}=\langle\beta_{i+1},\delta_{i+1}\rangle$ из $\mathcal{B}_{2s}$. Элементы a_i и b_i для $i=0,1,\ldots,s,s+1$ называются соответствующими друг другу. Стратегия состоит в том, что $\gamma_{i+1}=\delta_{i+1}$, а $\alpha_{i+1}=\beta_{i+1}+\sigma(i)$. Таким образом стратегия сводится к определению $\sigma(i)$. Сразу скажем, что сдвиг на каждом шаге равен либо 0, либо 1. Сдвиг равен 1 в том и только том случае, когда выбранный на этом шаге Разрушителем элемент находится на расстоянии, не превышающем радиуса для этого шага, от какого-то такого ранее выбранного элемента этой же базы данных, сдвиг которого тоже равен 1. При этом сдвиг a_0 и b_0 равен 0, а сдвиг a_1 и b_1 равен 1. Эти элементы и только они выбраны до начала игры.

Для корректности этой стратегии надо понять, что если на каком-то шаге в $\mathcal{A}_{2s}$ выбирается элемент первого столбца, то сдвиг на этом шаге равен 0. В самом деле, пусть до шага i каждый выбранный элемент, для которого сдвиг равен 1, находится на расстоянии, превышающем $2^{2s-i}-1$, от любого элемента первого столбца. Тогда после шага i (перед шагом $i+1$) каждый элемент, для которого сдвиг равен 1, находится на расстоянии, превышающем

$$2^{2s-i}-1-2^{s-i}>2^{2s-1}-2^{2s-i-1}-1=2^{2s-i-1}-1.$$

Это объясняется тем, что каждый элемент, сдвиг которого равен 1, отстоит на расстоянии, не превышающем 2^{s-i}, от какого-то ранее выбранного элемента, сдвиг которого тоже равен 1.

Лемма 4.2. *Если выбранные после какого-то шага игры элементы одной из рассматриваемых баз данных находятся на расстоянии, не превышающем радиуса для этого шага, то соответствующие элементы другой из рассматриваемых баз данных находятся на таком же расстоянии.*

Если один из выбранных после какого-то шага игры элементов одной из рассматриваемых баз данных меньше другого выбранного после этого шага элемента, то соответствующий первому элементу в другой из рассматриваемых баз данных тоже меньше элемента, соответствующего второму из рассматриваемых выбранных элементов в другой из рассматриваемых баз данных.

Доказательство. Индукцией по номеру шага игры. Надо заметить, что выбранные элементы, расстояние между которыми не превосходит радиуса, имеют одинаковый сдвиг. ∎

Из леммы сразу следует, что отображение, переводящее выбранные элементы в им соответствующие, является локальным мономорфизмом. ∎

Определение 4.1 (полноты класса глобальных предикатов). *Пусть Ω — некоторая реляционная сигнатура, $\mathfrak{A}$ — некоторый класс стандартных баз данных для схемы Ω, $\mathcal{C}$ — некоторый класс сложности.*

Пусть $\mathfrak{B}$ — некоторый такой класс $\mathfrak{A}$-глобальных предикатов, что каждый $\mathfrak{A}$-глобальный предикат $\mathfrak{P} \in \mathfrak{B}$ имеет сложность $\mathcal{C}$. Будем говорить, что $\mathfrak{B}$ является $\mathcal{C}$-полным для $\mathfrak{A}$, если для всякого $\mathfrak{A}$-глобального предиката $\Xi : \mathcal{A} \longmapsto \Xi^{\mathcal{A}}$, имеющего сложность $\mathcal{C}$, найдется такой $\mathfrak{P} \in \mathfrak{B}$, что для всякой системы $\mathcal{A} \in \mathfrak{A}$ предикаты $\Xi^{\mathcal{A}}$ и $\mathfrak{P}^{\mathcal{A}}$ совпадают.

Из теорем 1.1, 2.3, и 4.1 следует

Теорема 4.3 (неполнота элементарных запросов для $PTIME$). *Класс элементарных глобальных предикатов лежит в $PTIME$, но не является $PTIME$-полным.* ∎