УДК 681.3.06

С.С. ТАНЯНСКИЙ, канд. техн. наук, ***Д.А. РУДЕНКО,*** канд. техн. наук,
В.В. ТУЛУПОВ, канд. техн. наук

СЕМАНТИЧЕСКАЯ ЭКВИВАЛЕНТНОСТЬ СЛАБОСТРУКТУРИРОВАННЫХ БАЗ ДАННЫХ

У статті розглядається клас інформаційних систем, основним компонентом яких є база даних. З урахуванням того, що база даних не розглядається в контексті однієї моделі даних і в загальному випадку може не бути строго структурованої, досліджуються семантичні властивості організації бази даних. Пропонується як семантику бази даних розглядати набір правил визначальних властивостей об'єктів предметної області, у якій функціонує інформаційна система. Описано формальний підхід побудови максимально повної множини правил, на підставі якої робиться висновок про еквівалентність чи ступень подібності баз даних.

In clause the class of information systems is considered, basic component, which the database is. That the database is not examined in a context of one model of the data and generally cannot be strictly structured is investigated semantic properties of organization of a database. It is offered as semantics of a database to consider a set corrected determining property of objects of a subject domain, in which the information system functions. The formalistic approach of construction of maximum complete set of rules is described, on the basis of which is judged equivalence or degree of similarity of databases.

Постановка проблемы. Рассматривая информационную систему (ИС) как совокупность технических и обеспечивающих средств, а также технологических процессов, реализующих такие функции как сбор и хранение информации, поиск и обработку данных, передачу и распределение данных, необходимо найти возможность устанавливать некоторое сходство таких систем. Это необходимо для организации процедур совместного использования информации из разных ИС с различной степенью структурированности данных. Существуют различные типы ИС, но в данной статье будут рассматриваться системы, носителем информации в которых, будут являться базы данных (БД). Под БД будем понимать совокупность данных (возможно со слабой степень структурированности), размещенных на различных запоминающих устройствах и находящихся под управлением специального пакета прикладных программ.

Источниками воздействия на ИС могут быть как пользователи, так и специальные “поставщики” информации. Поступившие в систему сообщения определенным образом аккумулируются в БД, которая представляет синтаксическую модель фрагмента реального мира. Соотнесение элементов с объектами предметной области (ПрО) лежит в основе семантической модели ПрО.

Анализ литературы. Детальный анализ семантических возможностей моделей данных, проведенный в работах Х. Шмидта и Й. Свенсона [1],

У. Кента [2], Б. Лангефорса [3], Д. Маклеода [4] и других авторов, положили начало созданию, так называемых семантических моделей БД, которые позволяют определить новые требования к функционированию ИС и моделированию ПрО в целом.

По отношению к объектам существуют две проблемы: идентификация и адекватное описание объектов [5]. Для детального изучения методов идентификации объектов, их синтаксического и функционального анализа можно обратиться к работам [6, 7]. Дальнейшие выкладки будут направлены на разрешения второй проблемы – адекватного представления объектов и определения их семантических свойств, таким образом, материал, рассматриваемый в статье, является актуальным.

Целью статьи является формальное описание свойств объектов ПрО. Для описания свойств используются предикаты первого порядка, что позволяет абстрагироваться от модели данных ИС и представить БД как множество литералов.

Модель представления данных. Множество хранящихся данных в ИС будем рассматривать как произвольно структурированную БД. Структура БД определяется конечным набором атрибутов, выражающих свойства ПрО, которые ассоциируют некоторое значение из множества допустимых значений данного атрибута с каждым объектом ПрО. Последовательность атрибутов будем называть схемой БД. При этом значения в каждом атрибуте могут изменяться, что влечет изменение состояния БД, а при этом схема остается неизменной.

Для описания БД будем использовать методы логики предикатов. При описании ПрО необходимо проанализировать возможные высказывания, действующие в указанной области, и логические взаимосвязи, существующие между этими высказываниями.

Взаимосвязи между атрибутами будем описывать посредством использования формул. Множество сгенерированных формул будем называть аксиомами или правилами. Правила выражают тот факт, что из определенной комбинации данных в БД можно вывести некоторые другие данные. Семантикой БД будем называть такую интерпретацию множества правил, при котором каждое правило истинно.

Представим информационную компоненту ИС как семейство множеств

$$D = \{D_1, D_2, \ldots, D_n\}, \quad (1)$$

где $D_i = \{d_1, d_2, \ldots, d_m\}$ – множество допустимых значений и

$$R = (A_1, A_2, \ldots, A_n), \quad (2)$$

где $A_1, A_2, \ldots, A_n$ – множество атрибутов.

Соответствие между атрибутами (2) и значениями (1) определим как отображение вида

$$\Psi : R \to D. \quad (3)$$

Отображение (3) устанавливает, какое значение из D соответствует атрибуту из R. Таким образом, в упрощенной форме структурную компоненту ИС, представляющую БД, можно представить как $SDB = \{R, D, \Psi\}$ [8].

Как средства задания структурной компоненты могут использоваться декларативные спецификации, формулы исчисления высказываний или исчисления предикатов первого порядка. Объекты данных, которые удовлетворяют заданным условиям, составляют допустимое состояние БД.

Будем рассматривать БД как набор предикатов. В отличие от арифметических и логических функций, где область значений и область изменений аргументов по типу одна и та же, то есть однородная, у предикатов область значений функции – логическая, а область изменений аргументов – предметная. Таким образом, предикат является неоднородной функцией и может быть использован для моделирования БД.

В логике предикатов элементарным объектом, обладающим истинностным значением, является атомарная формула. Атомарная формула состоит из символического обозначения предиката и термов, выступающих в роли этого предиката. В общем виде, предикат можно представить как

$$p(t_1, t_2, \ldots, t_n), \quad (4)$$

где p – предикат, $t_1, t_2, \ldots, t_n$ – термы.

Число термов определяет размерность предиката, то есть в данном случае предикат p является n-местным. По сути, предикат – это функция, возвращающая булево значение истинно или ложно в зависимости от значений термов.

Аналогично (3) представим одноместный предикат $p(t)$ как отображение

$$\varphi : p \to t. \quad (5)$$

Отображение (5) устанавливает, какое значение t должно соответствовать предикату p, чтобы формула $p(t)$ принимала значение истинно.

Тогда выражение (4) будет соответствовать одноместному предикату, а БД можно описать как множество одноместных предикатов

$$R\,(p_1(t_1), p_2(t_2), \ldots, p_n(t_n)), \quad (6)$$

где предикат $p_i(t_i)$ $(1 \le I \le n)$ принимает значение истинно, если t_i является значением БД и ложно в противном случае.

Зафиксируем некоторый алфавит $\mathfrak{A}$, содержащий константы, переменные и предикаты. Для одноместного предиката p формулу $p(t)$ будем называть позитивным литералом l, а формулу $\neg p(t)$ негативным литералом $\neg l$. Базисный

литерал – это позитивный или негативный литерал, не содержащий переменных. Таким образом, выражение (6) можно записать как

$$R\ (l_1, l_2, \ldots, l_n). \qquad (7)$$

Ограничения целостности (ОЦ) будем выражать множеством правил

$$L = \{l \leftarrow l_1, l_2, \ldots, l_m\}, \qquad (8)$$

где $l, l_1, l_2, \ldots, l_m$ – литералы ($m \geq 1$).

Двуместная логическая связка "$\leftarrow$" представляет собой импликацию и может быть прочитана как выражение "если выполняется $l_1, l_2, \ldots, l_m$, то выполняется и l". Условие выполнимости ОЦ заключается в том, что если все литералы $l_1, l_2, \ldots, l_m$ входят в R, то и l также должен входить в эту R. Если такое условие не выполняется, то возможно нарушение целостности [2].

На содержательном уровне множество R представляет собой объекты ПрО, а L – свойства, которым эти объекты должны удовлетворять.

Основное условие правильности функционирования БД состоит в том, чтобы БД и ОЦ были совместны. Совместность заключается в отсутствии в R одного и того же позитивного и негативного литерала.

Правила, которые определяют допустимые значения, задают семантику БД.

В дальнейшем исходное состояние БД будем обозначать через R, а состояние, отражающее семантику БД через S. Например, если $R = \{a, b\}$, а $L = \{\neg b \leftarrow a\}$, то R и L совместны, а семантика $S = \{a, b, \neg b\}$ несовместна.

Под модификацией БД будем понимать операцию добавления или удаления литерала, при выполнении которой БД остается совместной. Добавление литерала означает, что l должен присутствовать в семантике модифицированной БД, а удаление – что l не должен входить в семантику модифицированной БД.

Логические следствия правил. Некоторые правила выполняются во всех состояниях, в которых выполнены правила из L, будем называть такие правила следствиями. Обозначим через L^* все следствия правил L или замыкание множества L.

Теория L-правил основывается на том, что в некотором множестве R между L-правилами существуют семантические закономерности, с помощью которых можно выводить одни правила из других, то есть делать выводы о выполнении одних правил на основании знаний, что для множества R выполняются другие правила.

Обозначим через l_i множество литералов получаемых в результате применения правил (8), а $\{l_j\}_i$ – литералы определяющие l_i. Правила (8) перепишем в виде

$$L = \{l_i \leftarrow \{l_j\}_i\}, (1 \le i \le n, 1 \le j \le m). \quad (9)$$

Будем говорить, что множество S удовлетворяет правилам (9), если все элементы l_i входят в S, то есть $l_i \subseteq S$. Рассмотрим два экстремальных вида правил:

$$\varnothing \leftarrow \{l_j\}_I; \quad (10)$$

$$l_i \leftarrow \varnothing. \quad (11)$$

Правило (10) тривиально удовлетворяет любому L. Правило (11) удовлетворяет такому S, в котором все элементы $l_i \in S$. В дальнейшем такие правила рассматриваться не будут.

Для множества S в любой момент существует некоторое множество правил L, которым это множество удовлетворяет. Пусть задано два множества S_1 и S_2 и пусть L удовлетворяет S_1 и не удовлетворяет S_2. Необходимо выявить все допустимые правила из L (в обозначении L') удовлетворяющие S_1 и S_2 (или показать отсутствие такого набора правил).

Чтобы найти L', необходимы семантические знания о S_1. Эти знания определяются множеством L, так как правила являются первичными по отношению к БД и по существу задают ограничения на объекты БД. Замыкание правил L^*, применимых к множеству R конечно, так как существует конечное число подмножеств множества R. Таким образом, всегда можно найти все правила L, которые удовлетворяют S_1, перебрав все возможные правила.

Однако такой подход требует больших временных затрат. Если известны некоторые правила $L \in L'$, то можно вывести остальные правила. Множество правил L влечет за собой правило $l_i \leftarrow \{l_j\}_i$ если все объекты из S, удовлетворяющие всем правилам из L, также удовлетворяют правилам $l_i \leftarrow \{l_j\}_i$. Вывод правил – это процедура устанавливающая, что если S удовлетворяет определенным правилам, то оно должно удовлетворять и некоторым другим правилам не входящим в L. Определим множество правил вывода.

1. Рефлексивность. Если $l \in R$, то $l \leftarrow l$.
2. Аддитивность. Если $l_1 \leftarrow l$ и $l_2 \leftarrow l$, то $l_1, l_2 \leftarrow l$.
3. Транзитивность. Если $l \leftarrow l_1$ и $l_1 \leftarrow l_2$, то $l \leftarrow l_2$.
4. Пополнение. Если $l \leftarrow l_1$, то $l \leftarrow l_1, l_2$.
5. Псевдотранзитивность. Если $l_1 \leftarrow l_2$, $l \leftarrow l_1, l_3$, то $l \leftarrow l_2, l_3$.
6. Проективность. Если $l_1, l_2 \leftarrow l$, то $l_1 \leftarrow l$ и $l_2 \leftarrow l$.

Очевидно, что $L \subseteq L^*$ и что $L^{**} = L^*$. Два множества S_1 и S_2 логически эквивалентны (в обозначении $S_1 \equiv S_2$), если $L_1^* = L_2^*$. Но построение L^*

соответствует перебору всех подмножеств множества L, что занимает экспоненциальное время [9].

Одним из способов уменьшить время проверки вхождения литерала в S – построить замыкание множества R относительно правил L. Замыканием множества R называется такое множества литералов R^*, для которых $l^* \leftarrow l \in L^*$ т.е. не существует ни одного литерала из R, который бы зависел от l и не принадлежал l^*. Алгоритм вычисления замыкания R имеет вид.

Вход: R, $L = \{l_i \leftarrow \{l_j\}_i\}$. Выход: R^*.

Алгоритм. Будем использовать дополнительную переменную M для сохранения множества литералов. Пусть $M := R$. Последовательно пересматривая правые части правил l_i, проверяем условие $l_i \subseteq R$. Если условие $\{l_j\}_i \subseteq R$ выполняется, то модифицируем $M := R \cup \{l_j\}_i$, исключаем $l_i \leftarrow \{l_j\}_i$ из L и продолжаем пересматривать правые части правил начиная с первого правила в модифицированном множестве L. Если не найдено ни одного правила с правой часть $\{l_j\}_i$, для которой $\{l_j\}_i \subseteq R$, алгоритм закончен.

Для того чтобы убедиться в эквивалентности семантик двух множеств R_1 и R_2 достаточно построить замыкание для одного из них и проверить вхождения в замыкание каждого элемента второго множества. Таким образом, условие $S_1 \equiv S_2$ справедливо, если $R_1 \subseteq R_2^*$ и $R_2 \subseteq R_1^*$.

Выводы. Временная сложность рассмотренных алгоритмов вычисления замыканий зависит от размера входного множества L. Таким образом, меньшее множество правил гарантирует более быстрое исполнение этих алгоритмов.

Анализ требований к информационным системам показывает, что, как правило, L содержит достаточно большое число правил, которые значительно замедляют работу алгоритма. Такая ситуация дает повод для разработки более эффективных методов определения эквивалентных БД. В частности, выделить набор базисных правил, исключив тривиальные и избыточные правила и таким образом уменьшить размер L.

Список литературы: 1. *Schmid H.A., Swenson J.R.* On the semantics of the relation model // In: Proc. of ACM SIGMOD Int. Conf. Management of Data. – 1975. – P. 211 – 223. **2.** *Kent W.* Consequences of assuming a universal relation. – ACM Trans. on Database Systems. – 1981. – V. 3. – P. 3 – 17. **3.** *Langefors B.* Information systems // Information Processing 74. – Amsterdam: North-Holland, 1974. – P. 937 – 945. **4.** McLeod D. The semantic data model. – MIT Press, 1979. **5.** *Цаленко М.Ш.* Моделирование семантики в базах данных. – М.: Наука, 1989. – 288 с. **6.** *Kent W.* Limitations on record-based information models. – ACM Trans. on Database Systems. – 1979. – V. 4. – P. 107 – 131. **7.** *Langefors B.* Infological models and information user views // Inform. Systems. – 1980. – V. 5 – P. 17 – 32. **8.** *Буслік М.М.* Оптимальні зображення реляційних баз даних. – К.: ІСДО, 1993. – 84 с. **9.** *Схрейвер А.А.* Теория линейного и целочисленного программирования. – Т. 1. – М.: Мир, 1991. – 360 с.

Поступила в редакцию 08.04.2005