

ГЕНЕРАЦИЯ СТРУКТУРНОЙ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ НА ОСНОВЕ АТТРИБУТИВНЫХ ГРАФОВЫХ ИСЧИСЛЕНИЙ

Ю.Г.Степин

Гродненский государственный университет имени Янки Купалы, факультет экономики и управления,
кафедра математического и информационного обеспечения экономических систем
ул. Ожешко д.22, г. Гродно, Республика Беларусь, 230023
+(37529) 3672210; факс: + 375 (152) 73-19-10; e-mail: stepin@grsu.by

Рассматривается подход к построению структурной модели предметной области в терминах теории графовых исчислений. Вводится понятие атрибутивной графовой грамматики. Предлагается алгоритм для генерации структурной модели.

Ключевые слова – атрибутивная графовая грамматика, структурная модель предметной области.

Задача генерации модели предметной области возникает во многих приложениях [1]. Модель предметной области включает в себя интенциональную, экстенциональную и процедурные составляющие [2]. Вычисление ответа на запрос к базе данных (БД) сводится к нахождению ответа в экстенциональной базе данных (ЭБД) с помощью обычных реляционных запросов и применению дедуктивного вывода для генерации запросов к (ЭБД) на основе дедуктивного вывода в интенциональной базе данных. В данной работе рассматриваются две компоненты модели предметной области – структурная модель (описание объекта в терминах элементов, его составляющих и связей между ними) и функциональная модель (описание вычислительных зависимостей между атрибутами предметной области). Каждая из этих моделей может быть описана графами специального вида [3,4].

Предлагается использовать для генерации и модификации графов аппарат графовых грамматик, являющийся развитием алгоритмического подхода к определению графовых грамматик [5]. Это позволяет в рамках одного формального аппарата описывать совместную генерацию этих моделей, что гарантирует их согласование и непротиворечивость. Множество альтернативных вариантов модели задается неявным образом в виде грамматики их порождающей. Язык этой грамматики и соответствует всем допустимым вариантам модели. Требования к конкретному экземпляру модели формулируются в терминах условий правил – подграфов, которые должны содержать результирующая модель и атрибутов, значения которых должны вычисляться. Таким образом, задача генерации структурной модели предметной области сводится к задаче генерации графов языка с заданными подграфами, то есть к задаче продолжимости графа до языка.

В данной работе предлагается алгоритм для решения задачи генерации структуры модели предметной области. Предложен алгоритм и рассмотрены примеры генерации схемы БД и генерации запросов к БД.

Функциональная модель описывается в терминах вычислительных моделей Тьюинга [4]. Вычислительная модель Σ задается тройкой $\Sigma=(A, D, R)$, где A – множество объектов, D – множество зависимостей, R – множество условий. Вычислительная задача Π это тройка $\Pi=(\Sigma, X, Z)$, где $X, Z \subseteq A$. X – исходный набор, Z – искомый набор. Z – множество целевых атрибутов, значения которых необходимо вычислять в данной задаче. Для разрешимости вычислительной задачи необходимо построение графа содержащего гиперпути из X в Z . Множество A включает элементы трех типов: имена объектов и атрибутов предметной области, системные имена (ключи), производные атрибуты – атрибуты, вычисляемые с помощью зависимостей из D .

Атрибутивный ориентированный помеченный граф g над алфавитами Σ_N, Σ_U и множествами значений атрибутов A_N, A_U – это семерка $g=(N, U, F, m_N, m_U, a_N, a_U)$, где N – множество вершин; U – множество дуг; $f_i: U \rightarrow N, i=1,2$ – функции инцидентности; $m_N: N \rightarrow \Sigma_N, m_U: U \rightarrow \Sigma_U$ – функции помечания вершин и дуг, соответственно; $a_N: N \rightarrow 2^A_N, a_U: U \rightarrow 2^A_U$ – функции, приписывающие каждой вершине (дуге) множество значений атрибутов.

Графовые грамматики являются обобщением линейных цепочечных грамматик [5]. Графовая продукция p – это пятерка: $p=(g_l, g_r, \pi, E, \psi)$, где:

g_l, g_r – графы, называемые левой и правой частями продукции; π – предикат применимости; E – преобразование вложения; ψ – функция приписывания значений атрибутов вершинам и дугам модифицируемой части графа.

Результатом применения продукции p к графу g будет граф g' , полученный следующим образом:

в g ищется подграф, изоморфный g_l ; если таковой существует, то проверяется истинность предиката применимости π ; если он истинный то подграф, изоморфный g_l , замещается g_r ; с помощью преобразования вложения E строятся дуги связывающие g_r с неизменившейся частью $g-g_l$; посредством ψ приписываются значения атрибутов вновь построенной части графа g' . Преобразование вложения E строится на основе алгоритмического подхода, позволяющего явно строить дуги между g_r и $g-g_l$.

Кроме продукции p будем использовать условные правила $q: q=(g_l, \pi)$, где g_l и π как и в p . Условные правила служат для определения структурных свойств графов.

Графовая грамматика GG это шестерка $GG = (\Sigma_N, \Sigma_U, A_N, A_U, g_0, P)$, где $\Sigma_N, \Sigma_U, A_N, A_U$ алфавиты меток и атрибутов вершин и дуг, соответственно, g_0 – начальный граф, аксиома грамматики, P – множество продукций. Языком графовой грамматики $L(GG)$ является множество терминальнопомеченных графов, выводимых из некоторого g_0 .

В зависимости от вида левой и правой части продукции и структуры преобразования вложения различают различные типы графовых грамматик [5]. Практически все они являются NP-полными. Автором рассмотрен и исследован специальный класс линейных графовых грамматик, имеющих полиномиальную сложность вывода.

Структурные свойства генерируемой модели описываются с помощью совокупности условных правил Q_z . Задача генерации модели задается парой (GG, Q_z) .

Задача проектирования структурной модели предметной области задается парой (Π, G_Q) , где $\Pi = (\Sigma, X, Z)$, $\Sigma = (A, D, R)$, $G_Q = (GG, Q_z)$. $GG = (\Sigma_N, \Sigma_U, A_N, A_U, d_0, P)$, причем $A \subseteq A_N$, D – совокупность графовых продукций специального вида, генерирующих подграфы, соответствующие вычисляемым атрибутам, R – описывается с помощью условных продукций и предикатов применимости в обычных продукциях. Таким образом, вычислительная модель также описывается в терминах графовой грамматики.

Решение задачи G_Q ищется в два этапа:

На первом этапе строится псевдообратная грамматика GG^{-1} (задача построения обратной грамматики для графовой грамматики с произвольной структурой преобразования вложения в общем виде неразрешима), с помощью которой строится дерево возможных планов построения целевого графа, т.е. определяется множество возможных путей

$$P^* = (p_{i1}, p_{i2}, \dots, p_{in}) \text{ таких, что}$$

$$a) p_{i1} \text{ применимо к } g_0 \quad g_0 \xrightarrow{p_{i1}} g_{i1};$$

$$b) \forall j = 2, \dots, n \quad p_{ij} \text{ применимо к } g_{j-1};$$

$$g_{j-1} \xrightarrow{p_{ij}} g_j \text{ и } Q_z(g_n).$$

Причем доказано, что решение задачи возможно тогда и только тогда, когда дерево путей P существует.

Для линейных графовых грамматик доказана конечность построения дерева P , предложен алгоритм его построения, полиномиально зависящий от величины l , где

$$l = \max_{q \in Q} \{ |N(q)| \}$$

($|N(q)|$ – число вершин графа g_i условного правила q).

На втором этапе строится множество всех допустимых решений задачи – это множество будем обозначать L_z .

$$L_z = \left\{ g \left| Q_z(g), \exists P' \subset P^*, g_0 \xrightarrow{P'} g \right. \right\},$$

где $\xrightarrow{P'}$ последовательное применение продукций из последовательности P' .

Множество L_z является, вообще говоря, неограниченным, поэтому в качестве решения выбираются элементы, удовлетворяющие дополнительным условиям.

Для удобства программной реализации предлагается использовать программные Графовые грамматики [5], которые содержат дополнительный механизм явного управления очередностью применения продукций грамматики.

ЛИТЕРАТУРА

- [1] Тальхейм, Б. Достижения и проблемы концептуального моделирования / Б. Тальхейм // Материалы IX Международной конференции «Интеллектуальные системы и компьютерные науки». Москва. – 2006. – Т 1. – С. 245-264.
- [2] Вагин, В.Н. Делюкция и обобщение в системах принятия решений./ В.Н. Вагин. – М.: Наука, 1988. – 256 с.
- [3] Анкудинов, Г.И. Синтез структуры сложных объектов / Г.И. Анкудинов. – Л.: Ленинградский ун-т, 1986. – 260с.
- [4] Тыгу, Э.Х. Концептуальное программирование/ Э.Х. Тыгу. – М.: Наука, 1984. – 256 с.
- [5] Nagl, M. Graph-Grammatik-Theorie. Anwerclungen. Implementierung/ M.Nagl. – Wiesbaden: Vieweg Verlag, 1979, 347 s.
- [6] Bunke, H. Attributed Programmed Graph Grammars and Their Application to Schematic Diagram Interpretation/ Bunke H. //IEEE Trans. Pattern. Anal. And Mach Intell. – 1982. – V 4, №6. – S. 574-582.