*MTA SZTAKI Közlemények 36/1987 pp.* 5-29

# ЦИКЛИЧЕСКИЕ И АЦИКЛИЧЕСКИЕ СХЕМЫ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

ГАЛЯ АНГЕЛОВА

Институт математики Болгарской Академии Наук

В этой работе рассматриваются понятия ациклической схемы и циклической схемы реляционной базы данных. Эти понятия введены в [BFMMUY81] с целью дать ответ на вопрос - при каких условиях каждое состояние базы данных, которое является попарно совпадающим (pairwise consistent), является также сплошь совпадающим (join consistent). Этот вопрос поставлен в 1976-ом году, но его ответ дан в [BFMMUY81] в 1981-ом году с введением понятия циклической и ациклической схемы базы данных.

В параграфе 1 описаны основные понятия. В параграфе 3 даны определения попарного и сплошного совпадения. В параграфе 4 рассмотрен алгоритм редукции гиперграфов, который в настоящей работе используется для содержательного определения понятия ацикличности (вместо формального определения ациклического гиперграфа, использованного например в [Fa83] и в [Fa83a]). В параграфе 5 исследованы некоторые свойства гиперграфов в процессе редукции и доказано существование так называемого поглощающего ребра для ациклических гиперграфов.

## 1. Основные понятия

Codd [Cod70] предложил реляционную модель баз данных. В классической модели реляционных баз данных рассматривается множество атрибутов (attributes) U и множество отношений (relations) над атрибутами U. В последные годы введено понятие реляционной схемы. Здесь мы вводим основные понятия, используя [Ull82], [Mai83] и [Дри82].

Пусть дано множество атрибутов

$U = \{A_1, A_2, \ldots, A_n\}$.

Множество U будем называть универсум (universum).

Реляционной схемой (relational scheme) R будем называть конечное множество атрибутов $\{A_{i1}, A_{i2}, \ldots, A_{im}\}$, где $A_{ij} \in U$ для $1 \leq j \leq m$. Реляционные схемы будем обозначать через $R_1, R_2, \ldots, R_k$.

Каждому атрибуту $A_i$ сопоставляется множество значений - так называемый домен (domain). Домен атрибута $A_i$ будем обозначать через $dom(A_i)$.

Пусть R - реляционная схема, $R = \{A_1, A_2, \ldots, A_n\}$. Отношением r над реляционной схемой R будем называть конечное множество упорядоченных n-ок:

$r = \{<a_1 a_2 \ldots a_n> \mid a_i \in dom(A_i),\ 1 \leq i \leq n\}$.

Элементы $<a_1 a_2 \ldots a_n>$ будем называть кортежами отношения r. Если r - отношение над реляционной схемой $R = \{A_1, A_2, \ldots, A_n\}$, будем обозначать этот факт через r(R) или $r(A_1 A_2 \ldots A_n)$.

Пусть U - данное множество атрибутов. Каждое конечное множество реляционных схем $\mathbf{R} = \{R_1, R_2, \ldots, R_k\}$ будем называть схемой базы данных (database scheme) тогда и только тогда, когда $R_i \subset U$ для $1 \leq i \leq k$ и $U = R_1 \cup R_2 \cup \ldots \cup R_k$.

(Здесь предполагается, что $R_i \neq U$ для $1 \leq i \leq k$).

Пусть $\mathbf{R}=\{R_1,R_2,\ldots,R_k\}$ - схема реляционной базы данных. Каждое множество конкретных отношений $\{r_1,r_2,\ldots,r_k\}$ соответственно над реляционными схемами $\{R_1,R_2,\ldots,R_k\}$ будем называть состоянием базы данных (database state). Будем обозначать состояние через $d=\{r_1,r_2,\ldots,r_k\}$.

Пусть $U=\mathbf{R}$, т.е. будем рассматривать U как одинственная реляционная схема в данной схеме базы данных. Тогда каждое конкретное отношение (состояние) I над схемой U будем называть универсальной реляцией (universal relation, universal instance) [MUV84], [AnZ85].

## 2. Гиперграфы, соответствующие схемам реляционных баз данных

Одновременно с введением реляционной модели базы данных началось и исследование разных способов организации данных в отдельных массивах (файлях) [Cod70]. Создавались и соответствующие алгоритмы для определении возможной организации данных при наличии заданных ограничений. Например, алгоритм декомпозиции в 3НФ с сохранением фунциональных зависимостей [Улл80] предлагает способ разложения в "группы" атрибутов U, обеспечивая при этом известные преимущества [Cod72] сохранения данных в отдельных отношениях. Этот алгоритм в качестве формальной процедуры может быть использован в автоматизации проектирования реляционных баз данных. Легко видет, что для этого алгоритма всегда существует "выход", который является единственным с точностью до переименования атрибутов и полученных реляционных схем.

В последнее время рассматривается и другой способ разделения атрибутов U на группы, а именно, путем определения U с помощью предикатов [FMU82]. Рассмотрим следующий пример:

ПРИМЕР 1. [FMU82] Пусть атрибуты U - C(курс), T(преподавател), R(аудитория), H(время), S(студент) и G(оценка). При дефинировании универсальной реляции над этими атрибутами записываем:

{ ctrhsg | t преподает c, курс c собирается в r во время h, s получает g по c } .

Будем пользовател тремя неформально заданными отношениями, которые являются предикатами [FMU82] и которые по нашему мнению имеют смысл в реальном мире: "преподает", "собирается в во время", "получает оценку по".

В этом примере универсальная реляция является множеством тех и только тех кортежей, которые проходят через тест, имплицированный каждым предикатом. Следовательно, универсальную реляцию для этого примера можно записать в виде:

$$\{ ctrhsg \mid P1(c,t) \& P2(c,r,h) \& P3(c,s,g) \} , \qquad (1)$$

где P1(c,t) - предикат "t преподает c",

P2(c,r,h) - предикат "c собирается в r во время h" и

P3(c,s,g) - предикат "s получает оценку g по предмету c". □

Отметим, что определение универсальных реляций с помощью предикатов отражает субъективное представление проектанта базы данных о реальном мире и следовательно, вряд ли может быть автоматизированно; ясно, однако, что таким образом также можно обеспечить разделение атрибутов на семантические группы и следовательно некоторую степень нормализации.

Введение гиперграфов соответствует описанию универсальной реляции с помощью предикатов.

ОПРЕДЕЛЕНИЕ 1. [FMU82] Пусть **E**-конечное множество вершин и **N**-конечное множество ребер, при чем ребра являются непустыми множествами вершин. Тогда упорядоченную пару H=(**E**,**N**) будем называть <u>гиперграфом</u>. □

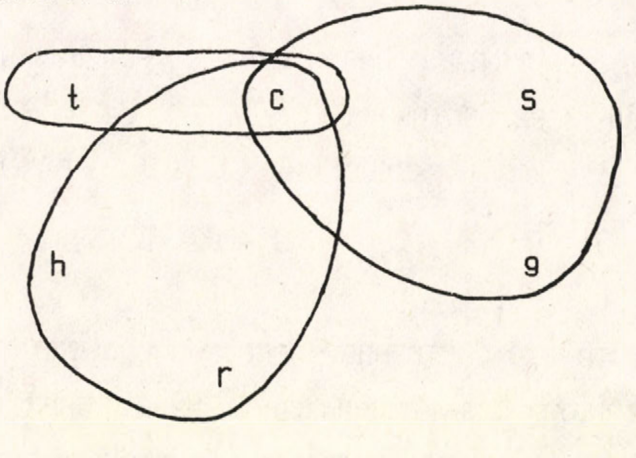
Гиперграф является обобщением обычного графа [Кри78], где каждому ребру соответствуют ровно две вершины.

Каждой универсальной реляции над U будем сопоставлять гиперграф следующим образом:

- каждому атрибуту из U сопоставляется вершина гиперграфа, соответствующий универсальной реляции;

- каждому предикату в определении универсальной реляции над U сопоставляется ребро гиперграфа.

Для универсальной реляции в примере 1 определенной с помощью (1), получаем гиперграф:




с вершинами соответственно c,t,r,h,s,g и ребрами {c,t}, {c,h,r}, {c,s,g}. □

При задании универсальной реляции с помощью предикатов возможно использовать различные множества предикатов. Тогда видно, что одной универсальной реляции можно сопоставлять разные гиперграфы - где каждый гиперграф соответствует отдельного описания универсальной реляции с помощью предикатов.

Пусть H=(**E**,**N**) - гиперграф и пусть **E**=$\{e_1,e_2,\ldots,e_s\}$. Если ясно какие вершины участвуют в ребрах $e_1, e_2, \ldots, e_s$, для удобства иногда будем обозначать H через $H = \{e_1, e_2, \ldots, e_s\}$.

## 3. Попарное и сплошное совпадения

Дадим основные определения.

ОПРЕДЕЛЕНИЕ 2. [Mai83] Состояние базы данных над реляционными схемами $R_1, R_2, \ldots, R_n$ называется <u>попарно совпадающим</u>, если значения одинаковых атрибутов в отношениях над схемами $R_1, R_2, \ldots, R_n$ совпадают. □

ПРИМЕР 2. Пусть дана схема реляционной базы данных **R**$= \{ABC, BCD, AD\}$ и ее состояние

$r_1(ABC) =$

| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |

, $r_2(BCD) =$

| B | C | D |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |

, $r_3(AD) =$

| A | D |
|---|---|
| 0 | 1 |
| 1 | 0 |

.

Покажем, что это состояние - попарно совпадающее.

Каждый атрибут для каждой схемы имеет значения $\{0,1\}$; поэтому значения одинаковых атрибутов в отношениях $r_1(ABC)$, $r_2(BCD)$ и $r_3(AD)$ совпадают.

Из определении 2 вытекает, что это состояние схемы **R**$= \{ABC, BCD, AD\}$ является попарно совпадающим. □

ОПРЕДЕЛЕНИЕ 3. [Mai83] Состояние базы данных над реляционными схемами $R_1, R_2, \ldots, R_n$ называется <u>сплошь совпадающим</u>, если все отношения этого состояния являются проекциями некоторой универсальной реляции с атрибутами $\cup_{i=1}^{n} R_i$. □

ПРИМЕР 3. Рассмотрим состояние из примера 2, отыскивая такое состояние универсальной реляции {ABCD}, что $r_1, r_2$ и $r_3$ являлись проекциями $r_1 = \pi_{ABC}(ABCD)$, $r_2 = \pi_{BCD}(ABCD)$, $r_3 = \pi_{AD}(ABCD)$.

Покажем, что $r_1(ABC)$, $r_2(BCD)$ и $r_3(AD)$ не являются сплошь совпадающими.

Отношение $r_1(ABC)$ содержит кортеж (0,0,0) и $r_2(BCD)$ содержит кортеж (0,0,0). Отсюда вытекает, что универсальная реляция ABCD должна содержать кортеж (0,0,0,0). Следовательно, если $r_1(ABC)$, $r_2(BCD)$ и $r_3(AD)$ являлись бы сплошь совпадающими, то отношение $r_3(AD)$ должно было содержать кортеж (0,0) - что противоречит содержанию $r_3(AD)$.

И так, показано, что $r_1(ABC)$, $r_2(BCD)$ и $r_3(AD)$ не являются сплошь совпадающими.

Рассмотрим другое состояние для схемы базы данных {ABC,BCD,AD}. Пусть

$r_1(ABC) =$

| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |

, $r_2(BCD) =$

| B | C | D |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 1 |

, $r_3'(AD) =$

| A | D |
|---|---|
| 0 | 0 |
| 1 | 1 |

.

Легко видеть, что $r_1(ABC)$, $r_2(BCD)$ и $r_3'(AD)$ являются сплошь совпадающими. Универсальная реляция ABCD, из которой получаются проекции $r_1(ABC)$, $r_2(BCD)$ и $r_3'(AD)$, имеет следующие кортежи:

$r(ABCD) =$

| A | B | C | D |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |

и тогда $r_1(ABC) = \pi_{ABC}(r)$, $r_2(BCD) = \pi_{BCD}(r)$, $r_3'(AD) = \pi_{AB}(r)$. □

Очевидно, сплошное совпадение имплицирует попарное совпаденое;

пример 3 показывает, что обратное неверно.

## 4. Алгоритм редукции гиперграфов

Рассмотрим следующий алгоритм, который называется алгоритмом редукции Грэема (Graham reduction algorithm) [Gra80].

АЛГОРИТМ 1. [FVa84] Алгоритм редукции гиперграфа данной схемы реляционной базы данных.

ВХОД: Гиперграф данной схемы реляционной базы данных записан в виде строк следующим образом:

каждому ребру гиперграфа отводится одна строка записи; при этом одинаковые атрибуты отдельных реляционных схем располагаются всегда один под другим.

ОПЕРАЦИЯ 1. Зачеркнуть имена всех атрибутов, которые появляются только один раз в входной записи;

ОПЕРАЦИЯ 2. Эсли какая-то строка с атрибутами $\{A_1, A_2, \ldots, A_k\}$ целиком содержит другую строку с атрибутами $\{A_{i1}, A_{i2}, \ldots, A_{ip}\}$, т.е. $\{A_{i1}, A_{i2}, \ldots, A_{ip}\} \subset \{A_1, A_2, \ldots, A_k\}$, зачеркнуть строку с атрибутами $\{A_{i1}, A_{i2}, \ldots, A_{ip}\}$.

МЕТОД: Применять операции 1 и 2 в произвольном порядке, сколько раз возможно.

ВЫХОД: Пустая запись - когда после применения операции 1 и 2 все символы входной записи зачеркнуты;

Непустая запись - когда после применения операции 1 и 2 не могут быть зачеркнуты все символы входной записи. □

Операции 1 и 2 алгоритма редукции не добавляют новые символы к входной записи, а только зачеркивают символы с входной записи. Имея ввиду конечности входной записи видно, что алгоритм 1 всегда заканчивает работу.

Легко видеть, что алгоритм 1 работает в полиномиальном периоде времени [FVa84].

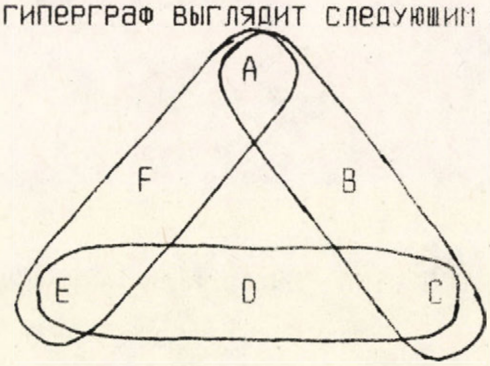
ТЕОРЕМА 1. [BFMY83] Гиперграф является ациклическим тогда и только тогда, когда алгоритм редукции заканчивает работу над этим гиперграфом с пустой записью. □

Теорема 1 позволяет нам ввести следующее определение:

ОПРЕДЕЛЕНИЕ 4. Пусть H - гиперграф, над которым алгоритм 1 заканчивает работу с пустой записью. Тогда H будет называть <u>ациклическим гиперграфом</u>. □

Проиллюстрируем алгоритм редукции на следующем примере.

ПРИМЕР 4. Пусть дана схема реляционной базы данных {ABC}, {CDE}, {AEF}. Ее гиперграф выглядит следующим образом:




(2)

Для применения алгоритма редукции нужно записать этот гиперграф как следует:

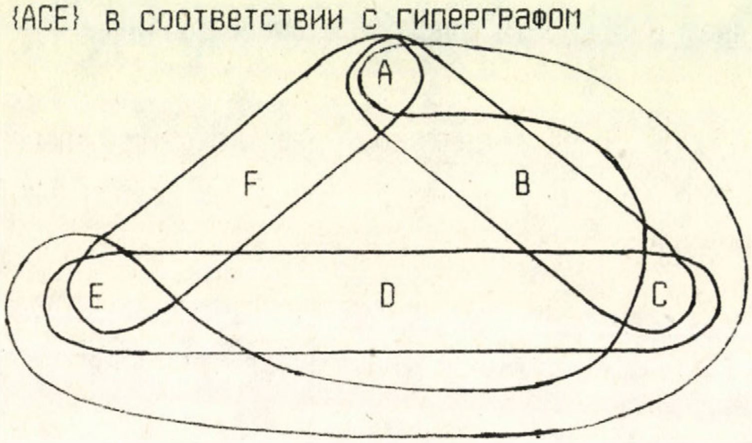
| | | | | | |
|---|---|---|---|---|---|
| A | B | C | | | |
| | | C | D | E | |
| A | | | | E | F |

Над этой записью применяем операцию 1 из алгоритма редукции и получаем

| | | |
|---|---|---|
| A | C | |
| | C | E |
| A | | E |

Эта запись является выходом алгоритма редукции, потому что дальнейшее применение как операции 1, так и операции 2 алгоритма 1 невозможно. Следовательно, гиперграф (2) является циклическим. □

ПРИМЕР 5. Пусть дана схема реляционной базы данных {ABC}, {CDE}, {AEF}, {ACE} в соответствии с гиперграфом




(3)

Покажем, что гиперграф (3) является ациклическим.

Гиперграф (3) можно представить в следующем виде:

| | | | | | |
|---|---|---|---|---|---|
| A | B | C | | | |
| | | C | D | E | |
| A | | | | E | F |
| A | | C | | E | |

(3′)

Применяем операцию 1 алгоритма редукции над записью (3′) и получаем

| | | |
|---|---|---|
| A | C | |
| | C | E |
| A | | E |
| A | C | E |

(4)

Применяем три раза операцию 2 из алгоритма 1 соответственно для строк

{A,C} ⊂ {A,C,E}

$\{C,E\} \subset \{A,C,E\}$ ,

$\{A,E\} \subset \{A,C,E\}$ ,

и получаем строку

$$A \quad C \quad E \tag{4'}$$

Применяем операцию 1 из алгоритма 1 над записью (4′) и получаем пустую запись. Следовательно, гиперграф (3) является ациклическим.

□

Отметим, что гиперграф (3), являясь ациклическим, содержит циклическое подмножество (2). Для обычных графов это не верно [Кри78].

Оказывается, что алгоритм редукции дает нам необходимое и достаточное условие для ответа на вопрос "Когда каждое состояние базы данных, которое является попарно совпадающим, является так же и сплошь совпадающим?". Следующие теоремы дают ответ на этот вопрос.

ТЕОРЕМА 2. [BFMY83] Пусть схема реляционной базы данных является ациклической. Каждое ее состояние является попарно совпадающим тогда и только тогда, когда это состояние - сплошь совпадающее.

□

Следовательно, для ациклических схем баз данных, попарное совпадение эквивалентно сплошном совпадению.

ТЕОРЕМА 3. [BFMY83] Если схема реляционной базы данных является циклической, тогда существует ее состояние, которое является попарно совпадающим, но не является сплошь совпадающим. □

Примеры 2 и 3 показывают такое состояние для схемы {ABC},{BCD},{AD} (которая является циклической).

СЛЕДСТВИЕ 1. [BFMY83] Из теоремы 2 и 3 вытекает, что попарное совпадение эквивалентно сплошном совпадению тогда и только тогда, когда схема базы данных является ациклической. □

Формальное определение понятия ациклического гиперграфа дано в [BFMMUY81, BFMY83, Fa983] и тоже в [MaUl84].

## 5. Некоторые замечания о ацикличности

Докажем некоторые свойства ациклических схем.

Будем использовать запись гиперграфов в виде строк, которая представляет собой вход в алгоритме редукции.

ОПРЕДЕЛЕНИЕ 5. Изолированной вершиной гиперграфа H называется такая вершина, которая участвует только в одном ребре гиперграфа H. □

Изолированные вершины участвуют также только в одной строке записи гиперграфа H в виде строк и отстраняются от этой записи путем применения операции 1 из алгоритма редукции.

ОПРЕДЕЛЕНИЕ 6. Пусть $H = \{e_1, e_2, \ldots, e_p\}$ - гиперграф с ребрами соответственно $e_1, e_2, \ldots, e_p$. Тогда вершины, которые участвуют в хотя бы двух ребрах гиперграфа H, будем называть связанными вершинами. □

Если $H = \{e_1, e_2, \ldots, e_p\}$ - гиперграф, то через $e_1, e_2, \ldots, e_p$ будем обозначать и ребра гиперграфа H, и строки записи для гиперграфа H в виде строк. Будем обозначать множество всех связанных вершин в данной строке $e_i$ через $\mathrm{con}(e_i)$. Следовательно, для каждой строки e (для каждого ребра e) гиперграфа H, мы разделяем вершины строки (ребра) в двух непересекающихся множествах: множество con(e) и множество изолированных вершин.

Отметим, что если запись $\{e_1, e_2, \ldots, e_s\}$ содержит строки, состоящие только из изолированных вершин, то тогда после применении операции 1 алгоритма редукции в полученной записи участвуют и пустие строки. Далее мы не будем отмечать существования пустых строк, если это не нужно.

ОПРЕДЕЛЕНИЕ 7. Будем говорить, что ребро $e_1 = \{A_1, A_2, \ldots, A_n\}$ гиперграфа H <u>поглощает</u> ребро $e_2 = \{A'_1, A'_2, \ldots, A'_k\}$ этого гиперграфа, если

$$\mathrm{con}(\{A'_1, A'_2, \ldots, A'_k\}) \subset \mathrm{con}(\{A_1, A_2, \ldots, A_n\}). \quad \square$$

Будем обозначать этот факт через "$\succ$", т.е. $e_1 \succ e_2$.

Очевидно реляция "$\succ$" является реляцией частичной упорядоченности.

Пусть для двух строк $\{A_1, A_2, \ldots, A_n\}$ и $\{A'_1, A'_2, \ldots, A'_k\}$ выполнено $\{A_1, A_2, \ldots, A_n\} \succ \{A'_1, A'_2, \ldots, A'_k\}$. Тогда в процессе применения алгоритма редукции можно применять операцию 1 над изолированными вершинами в $\{A_1, A_2, \ldots, A_n\}$ и в $\{A'_1, A'_2, \ldots, A'_k\}$, а потом и операцию 2 над этими строками. При этом строка, содержащая вершин con $\{A'_1, A'_2, \ldots, A'_k\}$, исчезает с входной записи. Таким образом, каждая пара строк $e_1 \succ e_2$ может быть редуцирована до получения строки $\mathrm{con}(e_1)$.

ПРИМЕР 6. Рассмотрим изолированные и связанные вершины гиперграфа (3) в примере 5.

Вершины B,D и F являются изолированными для этого гиперграфа. Все остальные вершины этого гиперграфа - т.е. вершины A,C и E, являются связанными. При этом выполнено: $con(\{A,B,C\})=\{A,C\}$, $con(\{C,D,E\})=\{C,E\}$, $con(\{A,E,F\})=\{A,E\}$ и $con(\{A,C,E\})=\{A,C,E\}$. □

Определения 5 и 6 введены для записи гиперграфа, которая является входом в алгоритм редукции (потому что ясно, что некоторые связанные вершини гиперграфа могут превратиться в изолированными после применения операции 2 алгоритма редукции гиперграфов). Обобщим определения 5 и 6 для каждого шага алгоритма 1.

ОПРЕДЕЛЕНИЕ 8. Пусть дана входная запись гиперграфа H соответственно со строками $\{e_1,e_2,\dots,e_s\}$. Запись $\{e'_1,e'_2,\dots,e'_k\}$ будем называть <u>редукцией</u> записи $\{e_1,e_2,\dots,e_s\}$, если запись $\{e'_1,e'_2,\dots,e'_k\}$ получена из записи $\{e_1,e_2,\dots,e_s\}$ путем применения операции 1 возможное число раз и одного применения операции 2 алгоритма редукции гиперграфов. □

Пусть операция 2 алгоритма редукции применяется j раз над входной записю. Полученную записю будем называт редукцией входной записи на j-том уровне.

ОПРЕДЕЛЕНИЕ 9. Пусть дана запись гиперграфа $H=\{e_{i1},e_{i2},\dots,e_{im}\}$, которая является редукцией входной записи на j-том уровне. Все вершины в записи $\{e_{i1},e_{i2},\dots,e_{im}\}$, которые участвуют только в одном ребре множества $\{e_{i1},e_{i2},\dots,e_{im}\}$, будем называть <u>изолированными вершинами (на j-том уровне редукции)</u>; остальные вершины будем называть <u>связанными вершинами (на j-том уровне редукции)</u>. □

ПРИМЕР 7. Рассмотрим изолированные и связанные вершины на разных уровнях редукции для гиперграфа (4).

Рассмотрим запись (4) из примера 5:

| | | |
|---|---|---|
| A | C | |
| | C | E |
| A | | E |
| A | C | E |

На первом уровне редукции этой записи можно получить запись

| | | |
|---|---|---|
| | C | E |
| A | | E |
| A | C | E |

В этой записи все вершины связанны на первом уровне редукции.

На втором уровне редукции можно получит запись

| | | | |
|---|---|---|---|
| A | | E | (5) |
| A | C | E | |

В этой записи вершина C является изолированной на втором уровне редукции, а вершины A и E являются связанными на втором уровне редукции. Применяя еще раз операцию 1 над вершиной C в (5) и операции 2 над вершинами {A,E}, получаем

| | | |
|---|---|---|
| A | | E |

В этой записи все вершины изолированы на третом уровне редукции. □

Будем говорить об изолированных и связанных вершинах, подразумевая, где это возможно, соответствующий уровень редукции.

Легко видеть, что если данная схема базы данных является ациклической, то выполнено одно из следующих утверждений:

а) выход алгоритма редукции (т.е. пустая запись) получается путем применения только операции 1 алгоритма редукции. В этом случае все вершины гиперграфа (все атрибуты схемы) являются изолированными и никакой атрибут не участвует хотя бы два раза в отдельных реляционных схемах, т.е. для всех ребер e гиперграфа H, $\mathrm{con}(e)=\varnothing$. Такие базы данных содержат только семантически несвязанные данные [Ап981], сгруппированные в отдельные несвязанные отношения, не имеющие общих атрибутов. Таких схем баз данных мы рассматривать не будем.

б) выход алгоритма редукции получается путем применения как операцию 1 алгоритма редукции, так и операцию 2 этого алгоритма. В этом случае схема базы данных содержит реляционные схемы, которые имеют общие атрибуты между собой и поэтому являются семантически связанными [Ап981].

ОПРЕДЕЛЕНИЕ 2.10. Пусть $H = \{e_1, e_2, \ldots, e_p\}$ - данный гиперграф. H будем называт <u>связанным гиперграфом</u>, если для каждой пары ребер $(e_i, e_j)$, $1 \leq i \leq p$, $1 \leq j \leq p$, $i \neq j$, существует цепочка различных ребер

$e_i = e'_1, e'_2, \ldots, e'_s = e_j$ такие, что для $1 \leq r \leq s$

1) $e'_r \neq e_i$, если $r \neq 1$;

2) $e'_r \neq e_j$, если $r \neq s$;

3) $e'_r \cap e'_{r+1} \neq \varnothing$ если $r \neq s$;

4) $e'_r \in H$. □

ЛЕММА 1. Если H - связанный гиперграф, то на всех уровнях алгоритма редукции над H получается запись, соответствующая некоторому связанному гиперграфу.

ДОКАЗАТЕЛЬСТВО. Пусть n - соответствующий уровень редукции. Лемму докажем с помощью индукции по отношению к n.

n=1. Покажем, что полученная запись на первом уровне редукции соответствует некоторому связанному гиперграфу H'. Пусть

$H=\{e_1,e_2,\ldots,e_p\}$, а $H'=\{e'_1,e'_2,\ldots,e'_p\}$. Докажем, что $H'$ - связанный гиперграф. На первом уровне редукции имеем $e'_i=\mathrm{con}(e_i)$, $1\le i\le p-1$. Строка $e_p\in H$ зачеркнута с помощью операции 2 алгоритма редукции - она поглощена некоторой строкой $e_r\in H$.

Пусть $e'_i,e'_j$ - ребра из $H'$. Покажем, что они связанны.

Пусть $e_i\in H$, $e_j\in H$ связанны при помощи цепочки

$$e_i=e_{i1},e_{i2},\ldots,e_{is}=e_j.$$

Если $e_{ik}\neq e_p$ для $k=1,2,\ldots,s$, то тогда $e'_i$ и $e'_j$ связанны при помощи цепочки

$$e'_i=e'_{i1},e'_{i2},\ldots,e'_{is}=e'_j.$$

Если $e_{ik}=e_p$ для $k=m$, то тогда $e'_i$ и $e'_j$ связанны при помощи цепочки

$$e'_i=e'_{i1},\ldots,e'_{im-1},e'_r,e'_{im+1},\ldots,e'_{is}=e'_j.$$

Следовательно, полученная на первом уровне редукции запись соответствует некоторому связанному гиперграфу.

Допустим, что на $n$-том уравне редукции полученная запись соответствует некоторому связанному гиперграфу.

Видно, что на $n+1$-ом уровне редукции (анналогично, как на первом уравне редукции) получается запись соответствующая некоторому связанному гиперграфу. □

ЛЕММА 2. Гиперграф $H$ является ациклическим тогда и только тогда, когда на всех уровнях редукции гиперграфа $H$ получается запись, соответствующая некоторому ациклическому гиперграфу.

ДОКАЗАТЕЛЬСТВО. Следует непосредственно из теоремы 1 и из определения 4. □

Отметим, что в процессе редукции ациклического гиперграфа пустую запись можно получить только с помощью операции 1. Последная строка, над которой в этом случае применяется операция 1, поглотила все остальные строки на последном уровне редукции данного гиперграфа.

ПРИМЕР 8. Покажем, что поглощающая строка на последном уровне редукции ациклического гиперграфа зависит от порядка применения операции 2 алгоритма редукции.

Рассмотрим гиперграф

| | | | |
|---|---|---|---|
| A | B | | |
| | B | C | |
| | | C | D |

Если применить операцию 2 над парой {A,B} и {B,C}, где con{A,B}$\subset$con{B,C}, получим в редуцированной на первом уровне записи строку {B,C}; затем если применить операцию 2 над парой {B,C} и {C,D}, где con({B,C})$\subset$con({C,D}), то тогда поглощающая строка этого гиперграфа будет {C,D}. Но если применить операцию 2 на первом уровне редукции над парой con{C,D}$\subset$con{B,C}, то тогда поглощающая строка на втором уровне редукции может быть {B,C}.

□

Лемма 2 и пример 8 показывают, что поглощающая строка всегда существует для ациклических гиперграфов, но какая она конкретно - это зависит от выбора порядка применения операцию 2 алгоритма редукции гиперграфов.

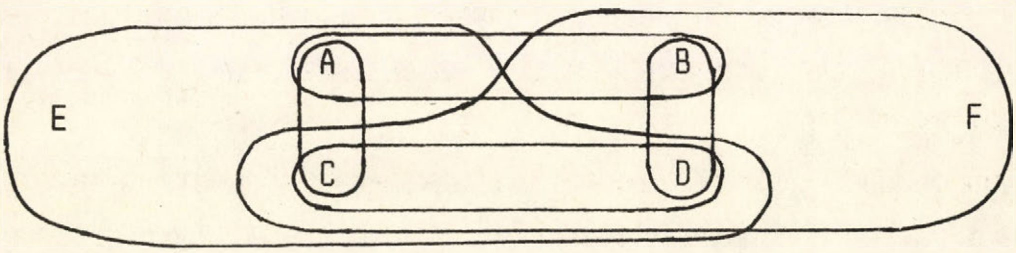
ОПРЕДЕЛЕНИЕ 11. [Fa983a] Гиперграф H называется <u>A-ациклическим</u>, если алгоритм редукции закончивает работу с пустой записю над этим гиперграфом. □

ОПРЕДЕЛЕНИЕ 12. Пусть $H=\{e_1,e_2,\ldots,e_s\}$ - A-ациклический гиперграф. Циклической компонентой в H будем называть такую совокупность ребер H′, H′⊂H, для которой алгоритм редукции не заканчивает работу с пустой записью над входом H′. □

Как уже видно из леммы 2, для каждого A-ациклического гиперграфа H существует ребро, которое поглощает все остальные ребра при применении алгоритма редукции над H.

ОПРЕДЕЛЕНИЕ 13. Пусть $H=\{e_1,e_2,\dots,e_s\}$ - гиперграф. Пусть его ребро $e_i$ поглощает ребро $e_j$ путем одного применения операцию 2 алгоритма редукции. Тогда $e_i$ будем называть непосредстенно поглощающим для ребра $e_j$. □

ПРИМЕР 9. Лемма 2.2 показывает, что для циклических компонентов в A-ациклическом гиперграфе всегда существует поглощающее ребро на некотором уровне редукции. Но как видно из следующей схемы базы данных, непосредственное поглощающее ребро может не быть единственным для циклических компонентов в A-ациклических гиперграфах.




Этот гиперграф - A-ациклический. Здесь циклическая компонента состоит из ребер {A,C},{A,B},{B,D} и {C,D}. Ребро {A,C,D,E} - непосредственно поглощающее ребро для {A,C} и {C,D}, а ребро {A,B,D,F} - непосредственно поглощающее ребро для {A,B} и {B,D}. □

ОПРЕДЕЛЕНИЕ 14. [Fa983a] Пусть H - A-ациклический гиперграф, который не содержит циклических компонент. Тогда будем называть H В-ациклическим гиперграфом. Все гиперграфы, которые не являются В-ациклическими, будем называть В-циклическими. □

ТЕОРЕМА 4. [Fa983a] Гиперграф H является В-циклическим тогда и только тогда, когда в нем существует цепочка

$(S_1, x_1, S_2, x_2, \ldots, S_m, x_m, S_{m+1})$

такая, что

1) $x_1, x_2, \ldots, x_m$ – разные вершины гиперграфа;

2) $S_1, S_2, \ldots, S_m$ – разные ребра гиперграфа, а $S_{m+1} = S_1$;

3) $m \geq 3$, т.е. цепочка содержит по крайней мере три ребра;

4) $x_i \in S_i \cap S_{i+1}$ $(1 \leq i \leq m)$ и никакому другому $S_j$. □

ЛЕММА 3. Алгоритм редукции заканчивает работу с пустой записью над каждым подмножеством данного гиперграфа H тогда и только тогда, когда H является В-ациклическим.

ДОКАЗАТЕЛЬСТВО. а) Достаточность. Пусть H является В-ациклическим, т.е. по теореме 4 в нем нет цепочки циклически связанных ребер. Следовательно, в каждом $H'$, $H' \subset H$, тоже нет такой цепочки и по теореме 4 каждое множество $H'$ является тоже В-ациклическим. Тогда алгоритм редукции заканчивает работу с пустой записью над $H'$.

б) Необходимость. Пусть алгоритм 1 заканчивает работу с пустой записью над каждым подмножеством данного гиперграфа H. Допустим, что H не является В-ациклическим. Тогда по теореме 4 в нем существует цепочка $(S_1, x_1, S_2, x_2, \ldots, S_m, x_m, S_{m+1})$

Покажем, что над этой цепочкой алгоритм редукции не может заканчивает работу с пустой записью.

Применяем операцию 1 над изолиранными вершинами из $S_1, S_2, \ldots, S_m$.

В полученную запись участвуют m строки:

$$con(S_1)=X_m \cup X_1$$

$$con(S_2)=X_1 \cup X_2$$

$$- - -$$

$$con(S_m)=X_{m-1} \cup X_m$$

Так как $X_i \neq X_j$ при $i \neq j$, $1 \leq i \leq m$, $1 \leq j \leq m$, то нельзя применить операцию 2 алгоритма редукции над полученной записю. Видно, что невозможно применять и операцию 1 алгоритма редукции - так как полученная запись состоит только из связанных вершин.

Следовательно, над цепочки $(S_1, x_1, S_2, x_2, \ldots, S_m, x_m, S_{m+1})$ алгоритм редукции не может закончить работу с пустой записю, что является противоречием. □

## 6. Заключение

В этой работе рассмотрено понятие ациклической и циклической схемы реляционной базы данных. Введено понятие поглощающей строки и показано, что для ациклических гиперграфов всегда существуют поглощающие строки. Они определяются выбором порядка операции из алгоритма редукции гиперграфов.

Реляционные базы данных с ациклическими гиперграфами представляют собой важный класс баз данных, потому что для них можно легко решать некоторые вопросы сохранения и обработки информации (см. [BFMU83], [Fag83], [Fag83a], [Yan81] ). В [MaUl84] показано, что с помощью ациклических гиперграфов можно искать эффективные алгоритми оптимизации некоторых запросов в реляционных базах данных. В заключении можно сказать, что интерпретация схемы реляционной базы данных в качестве гиперграфа позволила исследовать разных связей между атрибутами универсума и таким образом ввести впервые классификацию схем реляционных баз данных.

## Л И Т Е Р А Т У Р А

[Ang81] Angelova, G. The Use of Natural Language as a Query Language in a Relational Data Base. Proceedings of the Fourth Int. Seminar on Data Base Management Systems, Schwerin, GDR, December 1981, pp.226-238.

[AnZ85] Angelov, Zh. Towards a Universal Relation View. Proceedings of the Eigth International Seminar on Data Base Management Systems, Piestany, CSSR, September 1985, pp. 9-17.

[BFMY83] Beeri,C., Fagin,R., Maier,D. and Yannakakis, M. On the Desirability of Acyclic Database Schemes. Journal of the ACM, Vol. 30, No.3, July 1983, pp.479-513.

[BFMMUY81] Beeri, C., Fagin, R., Maier, D., Mendelzon, A., Ullman, J. and Yannakakis, M. Properties of Acyclic Database Schemes. Proceedings of the Thirteenth Annual ACM Symposium on the Theory of Computing (1981), pp. 355-362.

[Cod70] Codd, E.F. A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, Vol. 13, No. 6, June 1970, pp.377-387.

[Cod72] Codd, E.F. Further Normalization of the Data Base Relational Model. In Data Base Systems, Courant Computer Science Symposia, Vol. 6, Prentice-Hall, Englewood Cliffs, N.J., 1972, pp.33-64.

[FMU82] Fagin, R., Mendelzon, A. and Ullman, J. A Symplified Universal Relation Assumption and its Properties. ACM Transactions on Data Base Systems, Vol.7, No.3, September 1982, pp.nnn-mmm.

[Fag83] Fagin, R. Degrees of Acyclicity for Hypergraphs and Relational Data Base Schemes. Journal of the ACM, Vol. 30, No.3, July 1983, pp.514-550.

[Fag83a] Fagin, R. Acyclic Data Base Schemes of Various Degrees - a Painless Introduction. Lecture Notes in Computer Science, G. Ausiello and M. Protassi (eds.), Vol.159, 1983, pp.65-89.

[FVa84] Fagin, R. and Vardi, M. The Theory of Data Dependencies - a Survey. IBM Research Report No. 4321, 1984.

[Gra80] Graham, M. On the Universal Relation. In: A Panache of DBMS Ideas III, D. Tsichritzis (ed.), Technical Report CSRG-111, April, 1980, University of Toronto.

[Mai83] Maier, D. The Theory of Relational Databases. Computer Science Press, Rockville, Maryland, 1983.

[MUV84] Maier, D., Ullman, J. and Vardi, M. On the Foundations of the Universal Relation Model. ACM Transactions on Data Base Systems, Vol.9, No.2, June 1984, pp.283-308.

[MaUl84] Maier, D. and Ullman, J. Connections in acyclic hypergraphs. Theoretical Computer Science 32(1984), North-Holland, pp.185-199.

[Ull82] Ullman, J. Principles of Database Systems. Computer Science Press, 1982.

[Yan81] Yannakakis, M. Algorithms for Acyclic Database Schemes. Proceedings 1981 Very Large Data Bases Conference, pp. 82-94.

[Дри82] Дрибас, В.П. Реляционные модели баз данных. Минск, Издательство БГУ им. В.И.Ленина, 1982.

[Кри78] Кристофидес, Н. Теория графов. Москва, "Мир", 1978.

Cyclic and acyclic relational database schemes

G. Angelova

Summary

The paper discusses the acyclicity (and cyclicity) of relational database schemes. The concepts of pairwise consistency and join consistency are described. The Graham reduction algorithm for determining acyclicity is presented. The concept of so called "absorbent" hyperedge is introduced. It is shown that acyclic hypergraphs always contain at least one absorbent hyperedge.

Ciklikus és aciklikus relációs adat-bázis sémák

G. Angelova

Összefoglaló

A cikk a relációs adatbázis sémák aciklikusságát /ill. ciklikusságát/ vizsgálja. Ismerteti a páronkénti ill. együttes konzisztencia fogalmát és az aciklikusság megállapitására szolgáló Graham-féle redukciós algoritmust. Bevezeti az u.n. "elnyelő" /"absorbent"/ hiper-él fogalmát és megmutatja, hogy az aciklikus hiper-gráfok tartalmaznak "elnyelő" hiper-élt.