Обновления баз данных при динамических ограничениях целостности *

М.И. Дехтярь

Тверской госуниверситет,кафедра информатики,

Email: Michael.Dekhtyar@tversu.ru

Резюме

В этой статье рассматривается следующая задача выполнения обновлений баз данных (БД): по данной логической программе Ф, которая формализует динамические ограничения целостности (ДОЦ), т.е. ограничения на возможные переходы из одного состояния базы в другое, начальному состоянию БД I и внешнему запросу на обновление Δ, задающему добавляемые, удаляемые и заменяемые данные, нужно минимально изменить I и получить такое результирующее состояние I', для которого выполнено обновление Δ, чтобы пара $<I, I'>$ удовлетворяла всем ограничениям из Ф. Эта задача рассмотрена как для обычных полных (реляционных) БД, так и для частичных БД, в которых отрицательная информация представлена явно. Предложены алгоритмы ее решения в обоих случаях. Для частичных баз алгоритм имеет полиномиальную сложность, а для полных показана полнота задачи в сложностном функциональном классе $FNP//OptP[O(\log n)]$.

Чтобы найти практическое решение рассматриваемой задачи, предложен метод предварительного корректного расширения исходного запроса на обновление и одновременной оптимизации ДОЦ, не зависящий от исходного состояния БД. Для частичных БД построен полиномиальный алгоритм, вычисляющий оператор, дающий оптимальную пару (наибольшее корректное обновление/ наиболее простые ДОЦ). Показано, что для полных БД такого алгоритма, по-видимому, не существует (если $P \neq NP$). Для них предложена некоторая полиномиально вычислимая аппроксимация оптимального расширения. Кроме того, построен полиномиальный алгоритм, вычисляющий оптимальное решение для класса полных БД с позитивными ДОЦ (без отрицаний в телах предложений).

1 Введение

В этой статье мы рассматриваем интересную проблему, которая связана с возможностями баз данных нового поколения интеллектуально выполнять внешние запросы на обновление. Системы управления такими базами данных включают основанные на продукциях подсистемы, непрерывно поддерживающие ограничения целостности.

*Эта работа поддержана Российским Фондом Фундаментальных Исследований (проекты 99-01-00374, 00-01-00254).

Мы изучаем следующую задачу (мы называем ее *проблемой наименьших достаточных изменений*, сокращенно, *проблемой НДИ*): по данной теории Φ, которая формализует ограничения на возможные изменения базы данных в результате одной транзакции, начальному состоянию базы данных I и внешнему запросу на обновление Δ, задающему добавляемые, удаляемые и изменяемые данные, нужно так минимально изменить I на новое состояние I', чтобы выполнить Δ и обеспечить истинность Φ на паре (переходе) $< I, I' >$. Основные результаты данной работы продолжают и расширяют результаты по этой проблеме, полученные нами с соавторами в работах [7, 8, 9, 10, 11]. В отличие от указанных работ, в которых изучались статические ограничения целостности, фильтрующие допустимые состояния БД, здесь рассматривается более широкий класс динамических ограничений целостности (ДОЦ), которые выделяют допустимые транзакции, т.е. переходы из одного состояния БД в другое. Еще одно расширение связано с классом рассматриваемых обновлений. К ним добавлены операторы замены одних фактов другими, не сводимые непосредственно к операторам добавления и удаления.

В современных СУБД под транзакцией понимается неделимая с точки зрения воздействия на БД последовательность операторов манипулирования данными: вставки, удаления, замены. Если СУБД обнаруживает, что ее состояние, полученное в результате выполнения транзакции, не удовлетворяет ограничениям целостности, то она автоматически осуществляет возврат к тому состоянию БД, в котором она находилась перед началом транзакции (обычно этому соответствует выполнение оператора отката ROLLBACK). Такой подход, называемый *"проверкой ограничений целостности"*, прост в реализации, но недостаточно гибок и значительно ограничивает возможности проведения обновлений БД.

Другой подход — " поддержание ограничений целостности" — основан на том, что при нарушении ограничений целостности делается попытка автоматически их восстановить, не возвращаясь в исходное состояние. Процедуры такого восстановления могут программироваться вручную разработчиками приложений или выполняться автоматически средствами, встроенными в СУБД. В частности, наиболее развитые современные СУБД (например, ORACLE, SQL-Server и др.) предоставляют пользователям средства для определения ограничений целостности в виде систем активных правил и программирования автоматического поддержания этих ограничений при обновлениях баз данных (механизм триггеров). При этом становится возможным налагать ограничения не только на состояния БД, но и на сами транзакции, точнее на пары соседних состояний БД. Мы будем называть такие ограничения *динамическими.* Примерами динамических ограничений целостности являются, например, условия: "зарплата не должна уменьшаться", "на должность ведущего научного сотрудника можно назначать лишь того, кто работает в должности старшего научного сотрудника", "принимать заказ на продажу N принтеров следует лишь тогда, когда их текущее количество на складе не меньше $N + 2$" и т.п.

Отметим, что при обеспечении целостности БД с помощью таких механизмов как триггеры часто возникают проблемы, связанные с отсутствием у них ясной декларативной семантики и не всегда обоснованными ограничениями операционной семантики (например, выбором константы, ограничивающей глубину последовательных вызовов триггеров). Кроме того, чтобы гарантировать корректность системы таких правил, необходимо обеспечить выполнение нескольких труднопроверяемых свойств: завершаемости, конфлюэнтности и детерминированности (см. [26]). Интенсивные усилия, предпринятые в последнее время для определения точной семантики активных правил (см. [3, 5, 25, 26]), пока не привели к ее унификации, но позволяют надеяться на преодоление в будущем указанных выше трудностей.

Настоящая работа посвящена изучению возможностей автоматического поддержания динамических ограничений целостности средствами СУБД, т.е. возможностям интеллектуального выполнения обновлений БД. Такая постановка вопроса не нова. В работе [1] содержится обзор более ранних исследований по обновлениям баз данных и знаний. В последующих работах [6, 13, 16, 18] были предложены методы обновлений дедуктивных БД, основанные на абдукции изменяемых фактов. Недавняя работа [21] представляет собой обзор около 20 методов поддержания ограничений целостности и выполнения обновлений, предложенных в последнее время. Большинство из них относятся к обновлениям представлений (views) баз данных. Различия рассмотренных методов связаны как с различными средствами задания схем БД, представлений, ограничений целостности и обновлений, так и с разными механизмами выполнения обновлений. Интересно, что все методы, кроме собственного, авторы обзора [21] считают неполными и на примерах демонстрируют некорректность многих из них. Отметим еще два интересных подхода к обновлениям представлений, не вошедших в упомянутый обзор: теоретико-модельный подход для обновлений дедуктивных баз данных, предложенный в [14], и "ревизия" логических программ [22]. Мы не будем здесь подробно останавливаться на методах обновления представлений баз данных (краткий обзор работ по обновлениям пропозициональных баз знаний и представлений БД можно найти в [8]). Отметим только, что большинство предложенных подходов связано с трансляцией запросов на обновление представления (интенсиональной модели) в запросы на обновление (экстенсионального) состояния БД. Алгоритмы выполнения последних и являются основным предметом данной работы.

Запросы на обновление данных, хранимых в реляционных БД, могут задаваться явно, например, с помощью соответствующего оператора языка SQL или средствами алгоритмических алгебр, обобщающих SQL (см. [1]), либо неявно посредством формул некоторого логического языка (см., например, [2]). При наличии ограничений целостности действие обновления, т.е. реальное изменение состояния БД, может отличаться от задаваемого самим запросом. Приведем небольшой пример, демонстрирующий выполнение обновлений при динамических ограничениях целостности.

Пример 1 *Рассмотрим базу данных “Служащие”, которая содержит сведения о должностях сотрудников и их зарплате. В приведенных ниже ограничениях целостности Ф, представленных в виде логической программы, штрихи у имен предикатов (таблиц) означают, что они относятся к состоянию БД после обновления, а предикаты без штрихов — к текущему состоянию БД. Ограничения $r_1 - r_3$ отражают связь между должностью сотрудника и его зарплатой, r_4 постулирует единственность зарплаты, r_5 — единственность должности, а r_6 утверждает, что сотрудник может стать старшим инженером только, если перед этим он был инженером.*

Ограничения целостности Ф:

$r_1: \textit{должность}'(X, \textit{зав\_отд}) \leftarrow \textit{зарплата}'(X, Y), Y > 8000.$

$r_2: \textit{должность}'(X, \textit{ ст.\_инженер}) \leftarrow \textit{зарплата}'(X, Y), 3000 \leq Y \leq 5000.$

$r_3: \textit{должность}'(X, \textit{инженер}) \leftarrow \textit{зарплата}'(X, Y), Y < 3000.$

$r_4: \neg\textit{зарплата}'(X, Y) \leftarrow \textit{зарплата}'(X, Z), Y \neq Z.$

$r_5: \neg\textit{должность}'(X, Y) \leftarrow \textit{должность}'(X, Z), Y \neq Z.$

$r_6: \textit{должность}(X, \textit{инженер}) \leftarrow \neg\textit{должность}(X, \textit{ст.\_инженер}),$
$\textit{должность}'(X, \textit{ст.\_инженер}).$

Пусть в текущем состоянии I базы данных содержатся факты:
{ зарплата(петр, 2000), должность(петр, инженер),
зарплата(анна, 10000), должность(анна, зав_отд)}

Предположим, что запрос на обновление требует, чтобы зарплата *петра* повысилась до 4000. Это означает, что факт *зарплата(петр, 4000)* должен быть добавлен к состоянию БД. Такое добавление приведет к нарушению ограничений целостности r_2 и r_4. Для их восстановления факт *должность(петр, ст_инженер)* должен быть добавлен к I, а факт *зарплата(петр, 2000)* — из него удален. После этого оказывается нарушенным ограничение r_5. Чтобы оно выполнилось, требуется удалить факт *должность(петр, инженер)* . В результате получится состояние $I' =$ { *зарплата(петр, 4000), должность(петр,ст_инженер), зарплата(анна, 10000), должность(анна, зав_отд)*}, которое удовлетворяет статическим ограничениям $r_1 - r_5$, кроме того, переход от I к I' не нарушает ограничение r_6. Таким образом, все ограничения из Ф выполнены. Однако, если бы требовалось понизить до 4000 зарплату *анне*, т.е. добавить факт *зарплата(анна, 4000)* , то для полученного аналогичным образом состояния $I'' =$ { *зарплата(петр, 2000), должность(петр,инженер), зарпла-*

та(анна, 4000), должность(анна, ст_инженер)} переход от I к I'' не удовлетворял бы динамическому ограничению r_6. Нетрудно понять, что и для всякого другого результирующего состояния $\tilde{I}$ переход от I к $\tilde{I}$ не удовлетворял бы ограничениям Φ, т.е. корректное выполнение обновления в этом случае было бы невозможно.

Отметим также, что I', вообще говоря, не является единственным возможным корректным результатом обновления, увеличивающего зарплату *петра* до 4000. Например, состояния базы

$I_1 = \{$ *зарплата(петр, 4000), должность(петр,ст_инженер)}*, и

$I_2 = \{$ *зарплата(петр, 4000), должность(петр,ст_инженер), зарплата(анна, 10000), должность(анна, зав_отд), должность(иван,инженер)}*

таковы, что переходы к ним от I также удовлетворяют всем ограничениям Φ. Все же естественно предпочесть им в качестве результата обновления состояние I' , поскольку по сравнению с I_1 в I' сохранено больше фактов исходного состояния БД, а по сравнению с I_2 при одинаковых удаленных из I фактах в I' добавлено меньше новых фактов.

Содержательная цель настоящей работы состоит в построении методов поиска такого “ближайшего” к I состояния I'. Чуть более точно наша задача формулируется следующим образом: по заданной логической программе Φ, формализующей динамические ограничения целостности (ДОЦ), исходному состоянию БД I и запросу на обновление Δ, включающему множество Δ^+ добавляемых к I фактов, множество Δ^- фактов, которые нужно удалить из I, и множество Δ^* пар фактов, в которых первый должен быть заменен на второй, требуется найти такое минимальное реальное изменение I' состояния I, что при переходе от I к I' выполнено обновление Δ и не нарушено ни одно из ограничений Φ (т.е. гарантируется, что $\Delta^+ \subseteq I'$, $I' \cap \Delta^- = \emptyset$, для всякой пары $(a, b) \in \Delta^*$, если $a \in I$, то $a \notin I'$ и $b \in I'$, и $< I, I' > \models \Phi$). Как мы уже отмечали, мы называем эту проблему *проблемой наименьших достаточных изменений* (*проблемой НДИ*). Поскольку результирующих состояний, удовлетворяющих указанным условиям, может быть несколько, мы также рассматриваем проблему перечисления всех таких состояний — проблему ПНДИ.

Что касается критерия близости состояний БД, то чаще всего в работах по обновлениям баз данных и знаний использовался критерий, основанный на симметрической разности состояний (см., например, [12, 22, 24]). В [7] было предложено соединить этот критерий с критерием максимального пересечения. Этот комбинированный критерий вначале минимизирует множество удаляемых известных данных, а затем минимизирует добавление новых данных, что лучше соответствует идеологии баз данных. Независимо такой же критерий был использован в работе [20].

Проблемы НДИ и ПНДИ мы рассматриваем для двух классов баз данных. Один из них — это обычные реляционные БД, в которых отрицательная информация представлена неявно: отрицание некоторого факта имеет место, если этот факт отсутствует в текущем состоянии БД. Такие базы мы называем *полными*, поскольку для каждого факта известно истинен он или ложен в данный момент. Другой класс — это БД с явным представлением отрицательной информации. Состояния таких БД содержат как позитивные факты, так и их отрицания. Некоторый факт истинен, если он входит в текущее состояние БД, и ложен, если это состояние содержит его отрицание. Если же ни факт, ни его отрицание не входят в текущее состояние БД, то его значение является неизвестным. Поэтому базы данных из этого класса мы называем *частичными*.

В предыдущих работах [9, 11] была исследована сложность некоторых алгоритмических проблем разрешения, связанных с проблемой НДИ. Даже в простейших случаях они оказываются достаточно сложными. Так, например, если статические ограничения целостности заданы базисной дефинитной логической программой, то проблема включения литерала в результат-решение проблемы НДИ является coNP-полной проблемой для частичных БД и Σ_2^p-полной проблемой для полных БД. Для полных БД даже задача совместности ограничения целостности и обновления оказывается NP-полной. В настоящей работе мы дополняем эти результаты и устанавливаем оценки сложности непосредственно для проблемы поиска НДИ в терминах функциональных сложностных классов. В частности, показано, что эта проблема для полных БД является полной в функциональном классе $\mathrm{FNP//OptP}[O(\log n)]$ (теорема 6), а для частичных БД построен алгоритм ЧНДИ, который решает ее за полиномиальное время (теорема 8). Предложены также алгоритмы для решения проблемы перечисления всех минимальных корректных результатов обновлений (проблемы ПНДИ) и получены оценки их сложности, которые экспоненциально зависят от числа фактов, не попавших в обновление и полиномиально — от размера ДОЦ (теоремы 7, 10). Каждый новый добавляемый или удаляемый факт существенно уменьшает пространство поиска возможных решений и, следовательно, время их поиска. Каждое упрощение ДОЦ — удаление ограничения или литерала из его условия — также упрощает поиск решения, так как уменьшает время на проверку выполнения ДОЦ.

Чтобы найти практическое решение проблемы поиска НДИ, мы расширяем на динамические ограничения целостности метод предварительного корректного расширения исходного запроса на обновление и одновременной оптимизации ДОЦ ([9, 10]). Такая оптимизация обновлений не зависит от исходного состояния БД и может быть проведена на этапе трансляции запроса на обновление. Идея нашего метода состоит в том, что запрос на обновление Δ можно последовательно и корректно расширить, итеративно протаскивая его через ДОЦ Φ. Корректность такого расширения означает, что оно сохраняет неизменным множество допустимых для Φ и Δ переходов $Tr(\Phi, \Delta)$, т.е. таких пар состояний БД, которые удовлетворяют ограничениям Φ и на которых выполнено обновление Δ. Нашей целью является получение после конечного числа итераций максимального корректного расширения Δ' исходного запроса на обновление и наиболее простого представления Φ' ограничений целостности. Использование пары Φ', Δ' вместо Φ, Δ может существенно ускорить алгоритмы поиска НДИ. Результаты, представленные во второй части работы, дают теоретическое обоснование предложенного подхода к оптимизации обновлений на этапе трансляции и имеют ясный практический смысл. Мы описываем операторы, применение которых к исходным ДОЦ и обновлению доставляет оптимальную (по отношению к порядку на парах) пару вида обновление/ДОЦ. Эта пара состоит из максимально расширенного обновлени и минимальных по размерам ДОЦ, эквивалентных исходной паре. Предложен алгоритм, который строит такую оптимальную пару для класса частичных БД за полиномиальное время (теорема 13). К сожалению, существование аналогичного алгоритма для полных БД маловероятно, поскольку предложение 14 (2) показывает, что в таком случае выполнялось бы равенство P=NP. Поэтому в общем случае мы предлагаем для полных БД использовать итерацию операторов прямого и обратного операторов расширения обновлений, которая дает некоторую аппроксимацию оптимальной пары обновление/ДОЦ. Более того, для одного практически интересного класса полных БД, а именно для полных БД, имеющих ДОЦ с позитивными телами правил, оптимальные пары вида (наибольшее корректное обновление/ наиболее простые эквивалентные ДОЦ) все же можно находить за полиномиальное время (теорема 17).

Статья организована следующим образом. Все необходимые предварительные понятия и определения приведены в разделе 2. Вопросы совместности обновлений и ДОЦ рассмотрены в разделе 3. Там же приведено определение операторов консервативных обновлений и установлены их некоторые свойства. Раздел 4 посвящен алгоритмам для поиска консервативных обновлений. В параграфе 4.1 перечислены некоторые результаты о сложности двух проблем разрешения, связанных с проблемой НДИ, из предыдущих работ [9, 10, 11]. В параграфе 4.2 установлена оценка сложности задачи поиска НДИ для полных БД и приведен алгоритм для перечисления всех решений этой задачи. Задачи поиска НДИ в классе частичных БД рассмотрены в параграфе 4.3. В частности, там приведен полиномиальный алгоритм, доставляющий одно из решений этой задачи, и показано, как его можно преобразовать в алгоритм перечисления всех ее решений. В разделе 5 вводятся обобщенные обновления и определяется класс операторов расширения обновлений. В параграфе 5.2 описаны два метода упрощения ДОЦ, а в параграфе 5.3 определены для каждого из видов БД прямой и обратный операторы расширения обновлений. С их помощью в разделе 6 строится вычислимый за полиномиальное время оператор, доставляющий оптимальные пары вида обновление/ДОЦ для частичных БД. Раздел 7 посвящен анализу задачи поиска максимальных расширений для полных БД и построению эффективного алгоритма для решения этой задачи в случае БД с позитивными ДОЦ.

2 Основные понятия и определения

В этой работе мы будем рассматривать как полные (обычные реляционные), так и частичные базы данных (БД). Состояниями полных БД являются множества фактов (наборов записей отношений, строк таблиц). Отрицательная информация представлена в них неявно: отрицание некоторого факта имеет место, если сам этот факт отсутствует в данном состоянии. В частичных БД отрицательная информация представляется явно: некоторый факт a истинен, если он содержится в данном состоянии, и ложен, если в состоянии (явно!) содержится противоположный факт $\neg a$. Иначе a считается неизвестным. Таким образом, состояния частичных БД являются множествами литералов. Полные БД можно считать частным случаем частичных при выполнении "гипотезы о замкнутом мире", согласно которой всякий факт, не входящий в состояние БД является ложным, а его отрицание – истинным. При ее принятии позитивная часть состояния БД однозначно определяет все состояние, поэтому ее можно рассматривать как само состояние. Ясно, что полные БД – это, по-существу, обычные реляционные базы данных.

Мы предполагаем, что читатель знаком с основными понятиями и терминологией логического программирования (см., например, [19]), но для замкнутости изложения приводим все необходимые обозначения и определения.

Язык. Пусть $\mathbf{S}$ – это сигнатура 1-го порядка, включающая множество символов предикатов (имен отношений, таблиц) $\mathbf{Pr}$ и множество констант $\mathbf{C}$ (без других символов функций). Будем считать, что множество предикатов разбито на два подмножества: $\mathbf{Pr} = \mathbf{Prc} \cup \mathbf{Prn}$, где $\mathbf{Prn} = \{p' | p \in \mathbf{Prc}\}$. Содержательно, предикаты из $\mathbf{Prc}$ относятся к текущему состоянию базы данных, а их копии со штрихами из $\mathbf{Prn}$ – к следующему состоянию, получающемуся в результате обновления. Зафиксируем также счетное множество переменных $\mathbf{V}$. Для обозначения констант будем использовать числа и идентификаторы, начинающиеся со строчных букв, например, *петров, ветер, a17*, а идентификаторы переменных будут начинаться с прописных букв, например, *Сотрудники,* X_1, Z. Мы будем считать, что сигнатура $\mathbf{S}$ содержит бесконечное множество констант $\mathbf{C}$, но иногда множество рассматриваемых констант естественно считать конечным, например, таковым является число различных констант в текущем состоянии базы данных, в ограничениях целостности, в обновлении. Назовем *областью* конечное подмножество $\mathbf{D}$ множества $\mathbf{C}$. Для области $\mathbf{D}$ через $\mathbf{A}(\mathbf{S}, \mathbf{D})$ обозначим множество всех атомов над $\mathbf{S}$ с переменными из $\mathbf{V}$ и константами из $\mathbf{D}$. $\mathbf{B}(\mathbf{S},\mathbf{D})$ обозначает множество всех *базисных* (т. е. без переменных) атомов из $\mathbf{A}(\mathbf{S},\mathbf{D})$. Для атома $a \in \mathbf{A}(\mathbf{S},\mathbf{D})$ сам a и его отрицание $\neg a$ являются *литералами*. Пара литералов вида $(a, \neg a)$ или $(\neg a, a)$ называется *контрарной парой*. Для контрарной пары (l_1, l_2) мы будем писать $l_1 = \neg.l_2$ и $l_2 = \neg.l_1$. Множество всех литералов над $\mathbf{V}$ и $\mathbf{D}$ обозначим через $\mathbf{L}(\mathbf{S},\mathbf{D})$. Для каждого $W \subseteq \mathbf{L}(\mathbf{S},\mathbf{D})$ мы обозначаем через $\neg.W$ множество $\{a \mid \neg a \in W\} \cup \{\neg a \mid a \in W\}$. Через $\mathbf{LB}(\mathbf{S},\mathbf{D})$ обозначим множество всех базисных литералов. Таким образом, $\mathbf{LB}(\mathbf{S},\mathbf{D}) = \mathbf{B}(\mathbf{S},\mathbf{D}) \cup \neg.\mathbf{B}(\mathbf{S},\mathbf{D})$. Для произвольного множества литералов A через A^c будем обозначать его подмножество, содержащее все литералы A с предикатами из множества $\mathbf{Prc}$, а через A^n – подмножество, содержащее все литералы со "штрихованными" предикатами из множества $\mathbf{Prn}$. Очевидно, $A = A^c \cup A^n$ и $A^c \cap A^n = \emptyset$. В частности, $\mathbf{LB^c}(\mathbf{S},\mathbf{D})$ – это множество всех базисных литералов, относящихся к текущему состоянию БД, а $\mathbf{LB^n}(\mathbf{S},\mathbf{D})$ – к следующему состоянию БД. Введем также обозначения для снятия и навешивания штрихов. Для "штрихованного" литерала $l \in \mathbf{LB^n}(\mathbf{S},\mathbf{D})$ через $c(l)$ обозначим его версию без штриха и для множества "штрихованных" литералов $W \subseteq \mathbf{L^n}(\mathbf{S},\mathbf{D})$ положим $c(W) = \{c(l) | \; l \in W\}$.Аналогично, для литерала $l \in \mathbf{LB^c}(\mathbf{S},\mathbf{D})$ через $n(l)$ обозначим его версию со штрихом и для множества "нештрихованных" литералов $W \subseteq \mathbf{L^c}(\mathbf{S},\mathbf{D})$ положим $n(W) = \{n(l) | \; l \in W\}$.

Ограничения целостности *Динамические ограничения целостности (ДОЦ)* будут представляться с помощью обобщенных логических программ над $\mathbf{S}$ и $\mathbf{D}$. Каждая такая программа Φ состоит из конечного множества предложений вида

$$r = (l \leftarrow l_1, ..., l_n),$$

где $n \geq 0$ и l, l_i $(i = 1, \ldots, n)$ – литералы из $\mathbf{L}(\mathbf{S},\mathbf{D})$. Предложения Φ будем называть также *ограничениями*. Для предложения r через $head(r)$ обозначим его *голову* l, а через $body(r)$ – его *тело* $l_1, \ldots, l_n$. Мы будем часто рассматривать $body(r)$ как *множество литералов*. Отметим, что негативные литералы могут встречаться как в телах, так и в головах предложений. Предложение с пустым телом $(n = 0)$ называется фактом и записывается как l. (вместо $l \leftarrow .$). Предложение, все литералы которого являются базисными, называется базисным. Базисное предложение r_1 является базисным вариантом предложения r, если оно получено из r с помощью некоторой подстановки констант вместо всех переменных r. При фиксированной области $\mathbf{D}$ мы определим для ДОЦ Φ их базисную развертку над $\mathbf{D}$, $gr_{\mathbf{D}}(\Phi)$, как множество всех базисных вариантов предложений из Φ с константами из $\mathbf{D}$. Через $\mathbf{IC}(\mathbf{S},\mathbf{D})$ обозначим множество всех базисных динамических ограничений целостности в сигнатуре $\mathbf{S}$ с константами из $\mathbf{D}$.

Определим следующий частичный *упрощающий порядок* на $\mathbf{IC}(\mathbf{S},\mathbf{D})$: $\Phi_1 \preceq \Phi_2$, если $\forall r \in \Phi_1 \; \exists r' \in \Phi_2 \; (head(r) = head(r') \;\&\; body(r) \subseteq body(r'))$. Это отношение означает, что Φ_1 может быть получено из Φ_2 с помощью удаления некоторых ограничений из Φ_2 и некоторых литералов из тел оставшихся ограничений. Как обычно, $\Phi_1 \prec \Phi_2$ означает, что $\Phi_1 \preceq \Phi_2$ и $\Phi_1 \neq \Phi_2$.

Совместные состояния БД. Мы будем рассматривать в качестве состояний БД как полные, так и частичные интерпретации ДОЦ над *замкнутыми областями*. Это означает, что всякий раз для рассматриваемой задачи фиксируется некоторая область используемых констант $\mathbf{D}$. *Частичной интерпретацией* или *частичным состоянием БД* над $\mathbf{D}$ является произвольное подмножество литералов из $\mathbf{LB^c}(\mathbf{S}, \mathbf{D})$. Поскольку множество $\mathbf{LB^c}(\mathbf{S}, \mathbf{D})$ конечно, то и всякая интерпретация является конечным множеством. Для такой интерпретации $I \subseteq \mathbf{LB^c}(\mathbf{S}, \mathbf{D})$ мы положим $I^+ = I \cap \mathbf{B^c}(\mathbf{S}, \mathbf{D})$ и $I^- = I \cap \neg.\mathbf{B^c}(\mathbf{S}, \mathbf{D})$. Интерпретация I является *корректной*, если в ней нет ни одной контрарной пары литералов. Содержательно, в корректном частичном состоянии БД I атомы из позитивной части I^+ рассматриваются как истинные, атомы из негативной части $\neg.I^-$ – как ложные, а все остальные атомы – как неизвестные. Этим и объясняется термин "частичное".

Частичная интерпретация I является *полной*, если $I^+ \cup \neg.I^- = \mathbf{B^c}(\mathbf{S}, \mathbf{D})$ и $I^+ \cap \neg.I^- = \emptyset$. Из этого определения следует, что полные интерпретации однозначно определяются их положительными частями, поэтому мы будем идентифицировать их с подмножествами атомов из $\mathbf{B^c}(\mathbf{S}, \mathbf{D})$. Таким образом, как мы уже заметили выше, понятие полной интерпретации соответствует понятию состояния обычной реляционной базы данных. Далее мы вместо терминов "частичная интерпретация" и "полная интерпретация", как правило, используем их синонимы "частичное состояние БД" и "полное состояние БД", соответственно.

Допустимые переходы. Обычно состояния баз данных обновляются в результате выполнени последовательностей команд вставки, удаления и замены данных, называемых транзакциями. На время выполнения одной транзакции данные, как правило, закрыты для доступа и изменения другими транзакциями. В этой работе нас будут интересовать лишь "крайние точки" процесса выполнения транзакции: исходное состояние БД и состояние, получившееся после завершения транзакции, поэтому мы назовем *переходом* любую пару $< I, I' >$ состояний БД. В зависимости от того, какие состояния – частичные или полные – рассматриваются, будем называть переходы *частичными* или *полными*. Переход $< I, I' >$ является *корректным*, если оба входящие в него состояния I и I' корректны. Корректные переходы будут играть роль моделей ДОЦ. Распространим на них покомпонентное отношение включения: если $t_1 =< I_1, I_1' >$ и $t_2 =< I_1, I_1' >$ – два перехода, то $t_1 \subseteq t_2$ означает, что $I_1 \subseteq I_2$ и $I_1' \subseteq I_2'$. Динамические ограничения целостности выделяют среди всех переходов допустимые переходы, т.е. те переходы, на которых они выполнены (являются истинными) следующим образом. Для частичного перехода $<I, I'>$ и базисного литерала l мы скажем, что l истинен (имеет место) на $<I, I'>$ (обозначается $<I, I'>\models l$), если $l \in I$ при $l \in \mathbf{LB^c(S,D)}$ и $c(l) \in I'$ при $l \in \mathbf{LB^n(S,D)}$. Для полного перехода $<I, I'>$ и базисного атома a $<I, I'>\models a$ означает, что $a \in I$ при $a \in \mathbf{B^c(S,D)}$ и $c(a) \in I'$ при $a \in \mathbf{B^n(S,D)}$, а $<I, I'>\models \neg a$ означает, что $a \notin I$ при $a \in \mathbf{B^c(S,D)}$ и $c(a) \notin I'$ при $a \in \mathbf{B^n(S,D)}$.

Базисное динамическое ограничение целостности (предложение) $r = (l \leftarrow l_1, ..., l_n)$ *выполнено на переходе* $<I, I'>$, если $<I, I'>\models l$, когда $<I, I'>\models l_i$ для всех $1 \leq i \leq n$. ДОЦ Φ *выполнены на переходе* $<I, I'>$ (или переход $<I, I'>$ *удовлетворяет* Φ), если каждое ограничение из $gr_{\mathbf{D}}(\Phi)$ выполнено на $<I, I'>$. Это обозначается через $<I, I'>\models \Phi$.

Отметим, что имеется простое взаимно-однозначное соответствие между парами множеств литералов из $\mathbf{LB^c(S,D)}$ (т.е. без штрихов), первое из которых содержит литералы, относящиеся к текущему состоянию БД, а второе — к следующему за ним, и множествами литералов из $\mathbf{LB(S,D)}$. Именно, каждой такой паре $< I_1, I_2 >$ можно сопоставить множество $I_1 \cup n(I_2)$, а множеству $W \subseteq \mathbf{LB(S,D)}$ — пару $< W^c, c(W^n) >$. Очевидно, что эти сопоставления взаимно-обратны. Мы не будем вводить для них специальных обозначений, но иногда для сокращения числа рассматриваемых случаев будем отождествлять пару с соответствующим ей множеством и наоборот. Например, $\{l\}\cup < I_1, I_2 >$ равно $< I_1 \cup \{l\}, I_2 >$, если $l \in \mathbf{LB^c(S,D)}$, и — $< I_1, I_2 \cup \{c(l)\} >$, если $l \in \mathbf{LB^n(S,D)}$. Определенное выше условие истинности литерала l на переходе $<I, I'>$ для частичных БД можно было бы записать в виде $l \in <I, I'>$.

Обновления. Как уже отмечалось, одно из основных свойств базы данных — это ее способность изменяться во времени, т.е. обновлять свое содержимое в соответствии с запросами пользователей. Мы будем рассматривать запросы на обновление БД, состоящие из команд трех видов: *ДОБАВИТЬ(факт), УДАЛИТЬ(факт)* и *ЗАМЕНИТЬ(факт1, факт2)*. Для частичных БД в качестве фактов выступают базисные литералы, а для полных – базисные атомы. Таким образом, *обновление* – это тройка $\Delta = (D^+, D^-, D^*)$, где для частичных БД D^+, D^- — это подмножества $\mathbf{LB^c(S,D)}$, а $D^* \subseteq \mathbf{LB^c(S,D)} \times \mathbf{LB^c(S,D)}$, а для полных БД D^+ и D^- принадлежат $\mathbf{B^c(S,D)}$, а $D^* \subseteq \mathbf{B^c(S,D)} \times \mathbf{B^c(S,D)}$. Содержательно, D^+ – это множество фактов, которые требуется добавить к исходному состоянию БД, D^- – удалить, а пары литералов из D^* являются аргументами команды ЗАМЕНИТЬ.

Мы будем обозначать компоненты D^+, D^- и D^* обновления Δ как Δ^+, Δ^- и Δ^*, соответственно. Для произвольного множества пар литералов X через X_{in} обозначим его проекцию на первую координату, а через X_{out} — на вторую. В частности, $\Delta^*_{in} = \{l \mid \exists l'((l, l') \in \Delta^*\}$, а $\Delta^*_{out} = \{l' \mid \exists l((l, l') \in \Delta^*\}$. Для обоих видов баз данных через

$\mathbf{UP}(\mathbf{S}, \mathbf{D})$ будем обозначать множество всех обновлений в сигнатуре $\mathbf{S}$ с константами из $\mathbf{D}$.

Определение 1 *Обновление* Δ выполнено на переходе $<I, I'>$ *тогда и только тогда, когда* $\Delta^+ \subseteq I'$, $\Delta^- \cap I' = \emptyset$ *и для всякой пары* $(l, l') \in \Delta^*$, *если* $l \in I$, *то* $l \notin I'$, *а* $l' \in I'$.

Обозначим через $UT(\Delta)$ *множество всех переходов, на которых выполнено обновление* Δ.

Обновления Δ_1 *и* Δ_2 *назовем* эквивалентными, *если* $UT(\Delta_1) = UT(\Delta_2)$, *т.е. для любого перехода* $<I, I'>$ *обновление* Δ_1 *выполнено на* $<I, I'>$ *тогда и только тогда, когда* Δ_2 *выполнено на* $<I, I'>$.

Это определение позволяет избежать коллизий, связанных с порядком выполнения добавлений, удалений и замен. Мы будем предполагать, что для всех рассматриваемых обновлений $\Delta = (D^+, D^-, D^*)$ выполняются условия корректности обновлений (КОР):

(1) $D^+ \cap D^- = \emptyset$,

(2) $D^+ \cap D^*_{in} = \emptyset$ и

(3) $D^- \cap D^*_{out} = \emptyset$.

Предложение 1 *Если для обновления* Δ *существует переход, на котором оно выполнено, то для него справедливы условия (КОР).*

Определим по обновлению Δ обновление Δ' следующим образом: $\Delta'^+ = \Delta^+$, $\Delta'^- = \Delta^- \cup \{l \mid \exists l_1((l, l_1) \in \Delta^* \& l_1 \in \Delta^+)\}$, $\Delta'^* = \Delta^* \setminus \{(l, l_1) \in \Delta^* \mid l_1 \in \Delta^+\}$. Тогда имеет место простое

Предложение 2 *Определенное выше обновление* Δ' *эквивалентно обновлению* Δ *и при этом*

(4) $\Delta'^+ \cap \Delta'^*_{out} = \emptyset$.

Ясно, что и проверка условий (1) – (3), и построение эквивалентного обновления, удовлетворяющего условию (4), можно выполнить эффективно за полиномиальное время. Поэтому далее мы будем предполагать, не ограничивая общности, что для рассматриваемых обновлений выполнены условия (1) – (4).

Отметим также, что при выбранном определении выполнения обновлений команда *ЗАМЕНИТЬ*(a, b) добавляет факт b к состоянию БД только, если в нем перед ее выполнением содержался факт a. Поэтому она не эквивалентна паре безусловных команд *УДАЛИТЬ(a), ДОБАВИТЬ(b).*

Определение 2 *Скажем, что переход* $<I, I'>$ совместим *с обновлением* Δ, *если* $I' \cap (D^- \cup \neg.D^+) = \emptyset$ *и для всякой пары* $(l, l') \in \Delta^*$, *если* $l \in I$, *то* $l \notin I'$ *и* $\neg.l' \notin I'$.

Таким образом, совместимость $<I, I'>$ с Δ означает, что заключительное состояние перехода I' можно расширить, добавив новые литералы, до такого состояния I'', что обновление Δ выполнено на переходе $<I, I''>$.

На множестве обновлений $\mathbf{UP}(\mathbf{S}, \mathbf{D})$ определим покомпонентное включение по первым двум компонентам:
$\Delta_1 \sqsubseteq \Delta_2 \Leftrightarrow D_1^+ \subseteq D_2^+$, и $D_1^- \subseteq D_2^-$.

Операторы на ДОЦ и обновлениях. Мы будем рассматривать операторы Γ типа $\mathbf{IC}(\mathbf{S}, \mathbf{D}) \times \mathbf{UP}(\mathbf{S}, \mathbf{D}) \to \mathbf{IC}(\mathbf{S}, \mathbf{D}) \times \mathbf{UP}(\mathbf{S}, \mathbf{D})$. Если $\Gamma(\Phi, \Delta) = (\Phi', \Delta')$, и $\Delta' = (D'^+, D'^-, D'^*)$, то компоненты результата Φ', Δ', D'^+, D'^- и D'^* будем обозначать, соответственно, как $\Gamma(\Phi, \Delta)^{ic}$, $\Gamma(\Phi, \Delta)^{up}$, $\Gamma(\Phi, \Delta)^+, \Gamma(\Phi, \Delta)^-$ и $\Gamma(\Phi, \Delta)^*$. Через Γ^n будем обозначать n-кратную композицию оператора Γ, а через Γ^ω – оператор $\Gamma^\omega(\Phi, \Delta) = \lim\limits_{n \to \infty} \Gamma^n(\Phi, \Delta)$.

Мы будем далее, как правило, опускать $\mathbf{S}$ и $\mathbf{D}$, когда это не будет приводить к неоднозначности, и вместо $\mathbf{A}(\mathbf{S}, \mathbf{D}), \mathbf{L}(\mathbf{S}, \mathbf{D}), \mathbf{B}(\mathbf{S}, \mathbf{D}), \mathbf{LB}(\mathbf{S}, \mathbf{D}), \mathbf{UP}(\mathbf{S}, \mathbf{D})$, $\mathbf{IC}(\mathbf{S}, \mathbf{D})$ будем просто писать $\mathbf{A}$, $\mathbf{L}$, $\mathbf{B}$, $\mathbf{LB}$, $\mathbf{UP}, \mathbf{IC}$.

Сложностные классы. Мы предполагаем, что читателю известны основные понятия, связанные со сложностью вычислений (см., например, [15]). Для полноты изложения напомним некоторые из них. Мы будем в этой работе рассматривать алгоритмические проблемы двух видов. *Проблемы разрешения* представляют собой пары множеств (D, Y), где D — это множество исходных данных, входов проблемы, являющихся, обычно, строками в некотором конечном алфавите, а Y — это подмножество допустимых, распознаваемых, "хороших" данных из D. Алгоритм (детерминированный) корректно решает такую проблему, если он для каждого входа из Y выдает ответ "Да", а для каждого входа из $D \setminus Y$ — ответ "Нет". Недетерминированный алгоритм решает такую проблему, если для всякого входа из Y у него есть путь вычисления, приводящий к ответу "Да" и ни для какого входа из $D \setminus Y$ такого пути нет (предполагается, что на любом пути алгоритм либо останавливается с ответом "Да", либо вовсе не останавливается). Проблемы разрешения мы будем классифицировать по сложности с помощью хорошо известных классов: P — класс проблем, решаемых за полиномиальное время детерминированными алгоритмами, NP — класс проблем, решаемых за полиномиальное время недетерминированными алгоритмами, P^{NP} — класс проблем, решаемых за полиномиальное время с помощью оракулов (подпрограмм) из класса NP (его другое обозначение Δ_2^P) и др. Известно, что эти классы инвариантны относительно конкретной формализации алгоритмов, решающих проблемы (основное ограничение — отсутствие неограниченной распараллеливаемости).

Говорят, что проблема разрешения $\Pi = (D, Y)$ *сводится за полиномиальное время к проблеме* $\Pi' = (D', Y')$, если существует вычислимая за полиномиальное время функция f такая, что для всякого входа $w \in D$ $f(w) \in D'$ и $w \in Y \Leftrightarrow f(w) \in Y'$. Проблема Π называется *трудной для сложностного класса* C (или C-трудной), если к ней сводится за полиномиальное время всякая проблема из класса C. Π является C*-полной*, если она C-трудная и $\Pi \in C$. Например, NP-полной является проблема 3-КНФ, у которой D — это множество булевых формул в конъюнктивной нормальной форме, в которых каждая дизъюнкция содержит не более трех литералов, а Y состоит из выполнимых формул такого вида.

Другой вид алгоритмических проблем — *проблемы поиска* — является обобщением проблем разрешения. Они исследованы меньше проблем разрешения, обзор результатов по таким проблемам можно найти в [17]. Здесь для каждого входа $w \in D$ имеется некоторое (возможно, пустое) множество решений (ответов) S_w. Чтобы решить проблему поиска алгоритм должен вычислить (возможно, недетерминированно) некоторое решение $y \in S_w$, если S_w непусто. В качестве модели алгоритмов для такого вычисления можно, например, рассматривать машины Тьюринга(детерминированные или недетерминированные) с выходной лентой. Если такая машина на входе w останавливается в заключительном состоянии, то содержимое выходной ленты в этот момент должно являться элементом множества S_w. Отметим, что недетерминированная машина Тьюринга вычисляет, вообще говоря, некоторую частичную многозначную функцию. Понятие сводимости обобщается на функции следующим образом (см. [17, 23]). Проблема поиска Π *сводится за полиномиальное время* к проблеме Π', если существуют две вычислимые за полиномиальное время функции f и g такие, что для каждого входа w проблемы Π $f(w)$ является входом для Π', причем $S_w \neq \emptyset \iff S'_{f(w)} \neq \emptyset$, а функция g позволяет по любому решению $y \in S_{f(w)}$ проблемы Π' получить некоторое решение $g(w, y) \in S_w$ проблемы Π. Как и в случае проблем разрешения, проблема поиска Π является *трудной* для класса проблем поиска C, если к ней сводится за полиномиальное время любая проблема из C. Если к тому же Π принадлежит классу C, то она называется C-полной проблемой. Для обозначения функциональных сложностных классов обычно используют приставку "F" перед обозначением соответствующего класса для задач разрешения. Так, FP и FNP — это классы задач поиска, разрешимых за полиномиальное время детерминированными и, соответственно, недетерминированными машинами Тьюринга, а FP^{NP} — это класс проблем поиска, разрешимых за полиномиальное время с оракулами (подпрограммами) из NP. Еще один интересный функциональный сложностной класс $FNP//OptP[O(\log n)]$ был определен в работе [4]. В него входят задачи поиска, для которых за недетерминированное полиномиальное время можно по входу w найти некоторое решение $y \in S_w$, используя дополнитнльную информацию о значении

$opt(w)$ решения соответствующей оптимизационной NP-проблемы, имеющем размер $O(\log|w|)$. Этому классу, например, принадлежит проблема нахождения клики максимального размера в графе. Из определений непосредственно следует, что между перечисленными выше функциональными сложностными классами имеются следующие включения:
$FP \subseteq FNP \subseteq FNP//OptP[O(\log n)] \subseteq FP^{NP}$.

3 Консервативные обновления

3.1 Совместность обновлений и ограничений целостности

В общем случае, запрос на обновление может оказаться несовместным с ограничениями целостности. Поэтому при рассмотрении задачи поиска наименьших достаточных обновлений естественно требовать выполнения условия совместности обновления и ДОЦ, которое мы формализуем следующим образом.

Определение 3 *Для* $\Phi \in \mathbf{IC}$ *и* $\Delta \in \mathbf{UP}$ *обозначим через* $Tr(\Phi, \Delta)$ *множество всех переходов* $<I, I'> \models \Phi$, *на которых выполнено обновление* Δ.
Для фиксированного исходного состояния I *пусть* $Acc_I(\Phi, \Delta) = \{I' \mid < I, I' > \in Tr(\Phi, \Delta)\}$.
Обновление Δ совместно *с ДОЦ* Φ, *если* $Tr(\Phi, \Delta) \neq \emptyset$. *Обновление* Δ совместно с *ДОЦ* Φ для состояния БД I, *если* $Acc_I(\Phi, \Delta) \neq \emptyset$.

Совместность Φ с Δ означает, что $Tr(\Phi, \Delta)$ содержит хотя бы один переход, а совместность Φ с Δ для состояния ДБ I означает, что имеется хотя бы одно состояние I' такое, что переход $<I, I'> \in Tr(\Phi, \Delta)$, т.е. можно так выполнить запрос на обновление Δ, что ДОЦ Φ будут выполнены на соответствующем переходе.

При проверке совместности и при построении корректных переходов важную роль будет играть оператор непосредственной выводимости и его транзитивное замыкание, которые мы сейчас определим.

Пусть $\Phi \in \mathbf{IC}(\mathbf{S}, \mathbf{D})$. Для частичного перехода $<I, I'>$ мы положим

$$cl_\Phi(<I, I'>) = \{l \mid \exists r = (l \leftarrow l_1, ..., l_n) \in \Phi \ \ (\bigwedge_{i=1}^{n} <I, I'> \models l_i \ )\}.$$

Таким образом, $cl_\Phi(< I, I' >)$ — это множество фактов, которые можно вывести с помощью Φ за один шаг из множества литералов $I \cup n(I')$.

Сильный оператор непосредственной выводимости определяется на корректном переходе $<I, I'>$ как

$$T^{\in}_{\Phi}(<I,I'>) = \begin{cases} < cl_{\Phi}(<I,I'>)^c, c(cl_{\Phi}(<I,I'>)^n) >, & \text{если } cl_{\Phi}(<I,I'>) \text{ корректно} \\ < \mathbf{LB^c(S,D)}, \mathbf{LB^c(S,D)} >, & \text{в противном случае} \end{cases}$$

Это определение естественным образом расширяет на переходы обычное определение непосредственной выводимости для состояний. Иногда нам удобно будет рассматривать оператор $T^{\in}_{\Phi}$ в применении к произвольному множеству литералов A. Как мы уже отмечали, всякое такое множество можно рассматривать как переход, первая компонента которого состоит из "нештрихованных" литералов из A, а вторая — из "штрихованных", поэтому мы положим $T^{\in}_{\Phi}(A) = T^{\in}_{\Phi}(< A^c, c(A^n) >)$. Отметим некоторые свойства сильного оператора непосредственной выводимости, очевидным образом вытекающие из его определения.

Предложение 3 *(1) Оператор $T^{\in}_{\Phi}$ является монотонным: если $I_1 \subseteq I_2$ и $I'_1 \subseteq I'_2$, то $T^{\in}_{\Phi}(< I_1, I'_1 >) \subseteq T^{\in}_{\Phi}(< I_2, I'_2 >)$.*

(2) Существует наименьшая неподвижная точка оператора $T^{\in}_{\Phi}$:
$M^{min}_{\Phi} = lfp(T^{\in}_{\Phi}) = \bigcup\limits_{i=0}^{\infty} (T^{\in}_{\Phi}(< \emptyset, \emptyset >))^i$.

(3) Существует алгоритм, вычисляющий M^{min}_{Φ} за время, линейное относительно $|\Phi|$.

Ясно, что если переход M^{min}_{Φ} корректен, то он является минимальной (частичной) моделью Φ. Для произвольного перехода $<I,I'>$ мы положим

$$M^{min}_{\Phi}(<I,I'>) = M^{min}_{\Phi \cup I \cup n(I')}.$$

Таким образом, $M^{min}_{\Phi}(<I,I'>)$ — это множество фактов, которые можно вывести с помощью Φ из множества литералов $I \cup n(I')$.

Используя это определение, для частичных интерпретаций можно определить переход из $Tr(\Phi, \Delta)$ следующим образом. Положим $\Phi(\Delta) = \Phi \cup \{n(l). \mid l \in \Delta^+\} \cup \{n(l_2) \leftarrow l_1 \mid (l_1, l_2) \in \Delta^*\}$. Рассмотрим минимальную неподвижную точку $M^{min}_{\Phi(\Delta)} =< M_1, M_2 >$, которая, фактически, содержит множество фактов, выводимых из литералов D^+ с помощью Φ и предложений $\{n(l_2) \leftarrow l_1 \mid (l_1, l_2) \in \Delta^*\}$. Для состояния БД I положим $\Phi(\Delta, I) = \Phi(\Delta) \cup \{l. \mid l \in I\}$. Определим переход $< J_1, J_2 >= M^{min}_{\Phi(\Delta, I)}$. Следующее утверждение содержит простые условия, при которых $M^{min}_{\Phi(\Delta)} \in Tr(\Phi, \Delta)$, а $J_2 \in Acc_I(\Phi, \Delta)$, что позволяет эффективно проверять свойство совместности.

Предложение 4 *(1) Для частичных БД $M^{min}_{\Phi(\Delta)} \in Tr(\Phi, \Delta)$ тогда и только тогда, когда $M_2 \cap \Delta^- = \emptyset$ и $M_1 \cap M_2 \cap \Delta^*_{in} = \emptyset$.*
(2) $J_2 \in Acc_I(\Phi, \Delta)$ тогда и только тогда, когда $J_1 = I$, $J_2 \cap D^- = \emptyset$ и $J_1 \cap J_2 \cap \Delta^*_{in})\} = \emptyset$.
(3) Существует алгоритм, который конструирует по Φ и Δ переход $< M_1, M_2 >$ за время, линейное от суммарного размера Φ и Δ Для частичных БД совместность Φ с Δ можно проверить за такое же время.

(4) Существует алгоритм, который конструирует по Φ, Δ *и* I *переход* $< J_1, J_2 >$ *за время, линейное от суммарного размера* Φ, Δ *и* I*. Для частичных БД совместность* Φ *с* Δ *для* I *можно проверить за такое же время.*

Для полных БД существование совместного перехода $M^{min}_{\Phi(\Delta)}$ не гарантирует совместности Φ с Δ.

Пример 2 *Рассмотрим ДОЦ* $\Phi = \{r_1 : \neg c' \leftarrow a', b';\ r_2 : \neg b' \leftarrow \neg a', c'\}$ *и обновление* $\Delta = (\{b, c\}, \emptyset, \emptyset)$*. Тогда легко проверить, что* $M^{min}_{\Phi(\Delta)} = \{\{b, c\}, \{b, c\}\}$*. Но не существует полного перехода* $<I, I'>$*, удовлетворяющего* Φ*, на котором выполнено* Δ*. Действительно, если* Δ *выполнено на* $<I, I'>$*, то* $\{b, c\} \subseteq I'$*. Тогда, если* $a \in I'$*, то нарушено ограничение* r_1*, а если* $a \notin I'$*, то нарушено ограничение* r_2*.*

С другой стороны, для полных БД совместность Φ с Δ гарантирует, что частичный переход $M^{min}_{\Phi(\Delta)\cup n(\neg.D^-)}$ не противоречит фактам полных переходов из $Tr(\Phi, \Delta)$.

Лемма 1 (Полные БД) *Пусть* $\Delta = (D^+, D^-, D^*)$ *и ДОЦ* Φ *совместны и пусть* $M^{min}_{\Phi(\Delta)\cup n(\neg.D^-)} = < J, J' >$ *. Тогда для всякого перехода* $<I, I'> \in Tr(\Phi, \Delta)$ *выполнены следующие условия:*

$< J^+, J'^+ > \subseteq <I, I'>$ *и*

$\neg.J^- \cap I = \neg.J'^- \cap I' = \emptyset.$

Доказательство. Пусть $t_0 =< \emptyset, \emptyset >$ и положим для $i \geq 0$ $t_{i+1} =< J_{i+1}, J'_{i+1} >= T^{\in}_{\Phi}(t_i)$. Мы покажем индукцией по i, что для всех $i \geq 0$ справедливы соотношения

$$(*)\quad J_i^+ \subseteq I,\ \ J_i^{'+} \subseteq I' \text{ и } \neg.J_i^- \cap I = \neg.J_i'^- \cap I' = \emptyset.$$

Для $i = 0$ эти условия, очевидно, выполнены.

Предположим теперь, что условия (*) выполнены при $i \leq k$ и проверим их при $i = k + 1$. Пусть l — это произвольный литерал, принадлежащий $t_{k+1} \setminus t_k$. Возможны следующие четыре случая:

(1) $l \in J_{k+1} \setminus J_k$ и $l = a$ — атом;

(2) $l \in J_{k+1} \setminus J_k$ и $l = \neg a$ — отрицание атома;

(3) $l \in J'_{k+1} \setminus J'_k$ и $l = a$ — атом;

(4) $l \in J'_{k+1} \setminus J'_k$ и $l = \neg a$ — отрицание атома.

Тогда из определения t_{k+1} в случаях (1) и (2) существует предложение $r = (l \leftarrow body) \in \Phi(\Delta) \cup n(\neg.D^-)$, для которого $t_k \models body$, а в случаях (3) и (4) — предложение $r = (n(l) \leftarrow body) \in \Phi(\Delta) \cup n(\neg.D^-)$, для которого $t_k \models body$. Тогда условия (*) при $i = k$ гарантируют, что во всех случаях (1) – (4) $<I, I'> \models body$. Для предложения r имеются три возможности:

(а) $r \in \Phi$;

(б) $r = (n(a) \leftarrow b)$ для некоторой пары атомов $(b, a) \in D^*$;

(в) $r = \neg n(a)$. для некоторого $a \in D^-$.
Поскольку $<I, I'>\models \Phi$, то в варианте (а) $<I, I'>\models l$. Это означает, что в случае (1) $l = a$ входит в I, а в случае (3) $l = a$ входит в I'. В случае (2) $<I, I'>\models \neg a$ означает, что $a \notin I$, а в случае (4) из $<I, I'>\models \neg n(a)$ следует $a \notin I'$. В варианте (б) возможен лишь случай (3). Тогда $<I, I'>\models a$ означает, что $a \in I$ и поскольку замена $(a, a') \in D^*$ выполнена на переходе $< I, I' >$, то $a' \in I'$. В варианте (в), который возникает лишь при переходе от t_0 к t_1, возможен лишь случай (4). Поскольку обновление Δ выполнено на переходе $<I, I'>$, то $D^- \cap I' = \emptyset$ и $a \notin I'$. Таким образом, мы показали, что всякий новый атом a, попавший на шаге $(k+1)$ в $< J^+_{k+1}, J'^+_{k+1} >$ содержится в соответствующей компоненте перехода $<I, I'>$, а если в t_{k+1} попадает его отрицание, т.е. $\neg a \in J^-_{k+1}$ или $\neg a \in J'^-_{k+1}$, то a не входит в $< I, I' >$. Следовательно, условия (*) выполнены для t_{k+1}. Отсюда следует, что они также выполнены и для перехода $< J, J' >$, являющегося неподвижной точкой оператора $T^{\in}_{\Phi}$.$\Box$

Следующее утверждение является вариантом леммы 1 для понятия совместности относительно состояния БД.

Лемма 2 (Полные БД) *Пусть* $\Delta = (D^+, D^-, D^*)$ *и ДОЦ* Φ *совместны для состояния* I *и* $M^{min}_{\Phi(\Delta, I)\cup n(\neg.D^-)} =< J, J' >$. *Тогда* $J^+ = I$, $\neg.J^- \cap I = \emptyset$ *и для всякого состояния* $I' \in Acc_I(\Phi, \Delta)$ *выполнены следующие условия:*
$J'^+ \subseteq I'$ *и*
$\neg.J'^- \cap I' = \emptyset$.

Доказательство этой леммы аналогично доказательству леммы 1.

Для полных БД проблемы совместности оказываются NP-полными.

Теорема 1 *Для полных БД оба множества*

$$Comp = \{(\Phi, \Delta) \mid \Phi \in \mathbf{IC} \ \& \ \Delta \in \mathbf{UP} \ \& \ (\Delta \text{ совместно с } \Phi)\} \text{ и}$$

$$Comps = \{(\Phi, \Delta, I) \mid \Phi \in \mathbf{IC} \ \& \ \Delta \in \mathbf{UP} \ \& \ (I \subseteq \mathbf{B^c}) \ \& \ (\Delta \text{ совместно с } \Phi) \text{ для } I\}$$

являются NP-полными.

Доказательство. Очевидно, что $Comp \in NP$. Доказательство его NP-трудности также стандартно. Мы покажем, что 3-КНФ$\leq_p Comp$. Пусть $\alpha = \beta_1 \ \wedge ... \wedge \ \beta_m$ — это произвольная 3-КНФ, где $\beta_j = l^j_1 \ \vee \ l^j_2 \ \vee \ l^j_3$ и $l^j_i \in \{x_1, \neg x_1, ..., x_n, \neg x_n\}$ для всех $1 \leq j \leq m$ and $1 \leq i \leq 3$. Пусть $\mathbf{B}^n = \{a', x'_1, y'_1, ..., x'_n, y'_n\}$ Построим по α пару $(\Phi_\alpha, \Delta_\alpha)$. Ограничения целостности Φ_α включают следующие $(2n+m)$ предложений:
$r_i = (\neg a' \leftarrow x'_i, y'_i), \quad i = 1, ..., n,$
$p_i = (\neg a' \leftarrow \neg x'_i, \neg y'_i), \quad i = 1, ..., n,$
$b_j = (\neg a' \leftarrow \tilde{l^j_1}, \tilde{l^j_2}, \tilde{l^j_3}), \quad j = 1, ..., m,$

где $\tilde{l_i^j} = x'_k$ при $l_i^j = x_k$ и $\tilde{l_i^j} = y'_k$ при $l_i^j = \neg x_k$.
Положим $\Delta_\alpha = (\{a\}, \emptyset, \emptyset)$.

Предположим, что $(\Phi_\alpha, \Delta_\alpha) \in Comp$ и $< \emptyset, I' > \in Tr(\Phi_\alpha, \Delta_\alpha)$. Тогда $a \in I'$ и правила r_i и p_i гарантируют, что для любого $1 \leq i \leq n$ либо $x_i \in I'$, либо $y_i \in I'$, но не оба одновременно. Определим подстановку истиностных значений σ: $\sigma(x_i) = true$, если $x_i \in I'$, и $\sigma(x_i) = false$, если $x_i \in I'$. Так как каждое правило b_j $(1 \leq j \leq m)$ истинно на $<I, I'>$, то существует $1 \leq i \leq 3$, такое, что $I' \not\models \tilde{l_i^j}$, т.е. $\tilde{l_i^j} \notin I'$. Но тогда $\sigma(l_i^j) = true$ и, следовательно, $\sigma(\alpha) = true$.

Предположим, что $\alpha \in$3-КНФ и пусть σ — это подстановка истиностных значений такая, что $\sigma(\alpha) = true$. Определим состояние БД $I' = \{x_i \mid \sigma(x_i) = false\} \cup \{y_i \mid \sigma(x_i) = true\} \cup \{a\}$. Теперь можно проверить, что $< \emptyset, I' > \in Tr(\Phi_\alpha, \Delta_\alpha)$. $\square$

3.2 Консервативные операторы обновлений

При обсуждении проблемы консервативных обновлений во введении мы уже отмечали, что естественно требовать, чтобы изменение исходного состояния БД в результате обновления было минимально возможным. Чтобы сделать это утверждение точным, требуется определить некоторый критерий близости состояний БД. Во многих работах, посвященнных обновлениям баз данных и знаний в качестве такого критерия рассматривается минимальность симметрической разности состояний (см., например, [12, 22, 24]). В работе [7] мы с соавторами предложили объединить его с критерием максимальности пересечения результирующего и исходного состояний. Этот критерий исходит из того соображения, что в результате обновления желательно сохранить в БД максимальное количество уже имеющейся в ней информации и уже при этом условии затем добавить минимальную необходимую новую информацию (консерватизм!). Представляется, что такой подход лучше соответствует идеологии баз данных. Мы формализуем его в следующем определении, минимизируя вначале множество удаляемых фактов, а затем множество добавляемых фактов.

Определение 4 *Пусть* I, I_1, I_2 *– три состояния БД. Мы скажем, что* I_1 ближе к I чем I_2, *если*
(i) $I \cap I_2 \subsetneq I \cap I_1$ *или*
(ii) $(I \cap I_2 = I \cap I_1)$ *и* $I_1 \setminus I \subsetneq I_2 \setminus I$.

Пусть I, I_1 *– два состояния БД и* $\mathbf{K}$ *– это некоторый класс состояний БД. Состояние* I_1 *назовем* ближайшим к I относительно $\mathbf{K}$, *если в* $\mathbf{K}$ *нет состояния* I_2, *которое ближе к* I *чем* I_1.

В [8] для случая статических ограничений целостности были определены операторы, которые по исходному состоянию БД и запросу на обновление выдавали результирующее состояние, удовлетворяющее ограничениям целостности, на котором было выполнено требуемое обновление и которое было ближайшим к исходному состоянию. Это определение естественным образом переносится и на случай динамических ограничений целостности.

Определение 5 *Пусть заданные обновление Δ и ДОЦ Φ совместны. Оператор $\Psi_{\Phi,\Delta}$ на множестве состояний БД назывется* консервативным оператором обновления, *если для каждого состояния БД I такого, что Δ совместно с Φ для I, выполнены следующие условия:*

- $\Psi_{\Phi,\Delta}(I) \in Acc_I(\Phi, \Delta)$,
- *состояние БД $\Psi_{\Phi,\Delta}(I)$ является ближайшим к I по отношению к $Acc_I(\Phi, \Delta)$.*

Первый пункт этого определения говорит о том, что запрос на обновление должен быть выполнен, и что получившийся переход $< I, \Psi_{\Phi,\Delta}(I) >$ должен удовлетворять ДОЦ. Второй пункт фиксирует консервативность обновления — выполняются лишь минимально необходимые изменения в начальном состоянии, чтобы результирующее удовлетворяло первому пункту.

Из этого определения можно получить следующее простое утверждение о "полноте" консервативных операторов обновления.

Теорема 2 *(модельная полнота)*
Пусть $\Psi_{\Phi,\Delta}$ является консервативным оператором обновления и пусть I_1, I_2 – такие два состояния БД, что $< I_1, I_2 >\models \Phi$. Тогда существует такое обновление Δ_0, совместное с Φ, что $I_2 = \Psi_{\Phi,\Delta_0}(I_1)$.

Доказательство. Мы положим просто $\Delta_0 = (I_2 \setminus I_1,\ I_1 \setminus I_2, \emptyset)$. Очевидно, что это обновление выполнено на переходе $< I_1, I_2 >$ и, следовательно, $< I_1, I_2 >\in Tr(\Phi, \Delta_0)$. Легко также проверить, что для всякого $I' \in Acc_{I_1}(\Phi, \Delta_0)$ выполнены следующие включения: $I_1 \setminus I_2 = \Delta_0^- \subseteq I_1 \setminus I'$ и $I_2 \setminus I_1 = \Delta_0^+ \subseteq I' \setminus I_1$. Следовательно, I_2 ближе к I_1 чем всякое другое состояние из $Acc_{I_1}(\Phi, \Delta_0)$ и оно является единственным возможным результатом $\Psi_{\Phi,\Delta_0}(I_1)$. $\square$

Из этого факта, в частности, следует, что для ДОЦ Φ и состояния I область значений всякого консервативного оператора обновления $\Psi_{\Phi,\Delta}$ при варьировании Δ совпадает со множеством всех состояний I' таких, что $<I, I'>$ является моделью Φ. Представляется, что для баз данных это свойство весьма естественно, поскольку в них обычно нет критериев для выбора каких-либо специальных подклассов моделей ДОЦ. Подходы же, связанные с ограничениями возможных результатов обновлений какими-то специальными подклассами моделей (например, стабильными или wf-моделями), которые использованы при определении операторов ревизии в [22, 24] или при обновлении внешних представлений (определяемых отношений) в [14] больше подходят для баз знаний.

Обозначим через $\mathbf{MDS}_{\Phi,\Delta}(I)$ множество всех состояний БД $I' \in Acc_I(\Phi,\Delta)$, ближайших к I относительно $Acc_I(\Phi,\Delta)$. В теореме 2 результат оператора консервативного обновлени детерминирован, т. е. $|\mathbf{MDS}_{\Phi,\Delta_0}(I_1)| = 1$. Конечно, в общем случае это не так и операторы консервативного обновления недетерминированы. Следующий пример показывает, что размер множества возможных результатов $\mathbf{MDS}_{\Phi,I}(\Delta)$ может быть экспоненциальным по отношению к размерам его параметров.

Пример 3 *Пусть* $\Phi = \{c \leftarrow a_i, b_i \mid i = 1, ..., n\}$ *и* $I = \{a_1, ..., a_n, b_1, ..., b_n, c\}$. *Предположим, что требуется удалить атом c, т.е.* $\Delta = (\emptyset, \{c\}, \emptyset\ )$. *Тогда нетрудно проверить, что множество* $\mathbf{MDS}_{\Phi,\Delta}(I)$ *состоит из всех состояний БД вида* $\{\alpha_1, ..., \alpha_n \mid \alpha_i \in \{a_i, b_i\},\ i = 1, ..., n\}$. *Следовательно,* $|\mathbf{MDS}_{\Phi,\Delta_0}(I_1)| = 2^n$.

Проблему *НДИ*, обсуждаемую во введении, сейчас можно переформулировать как задачу нахождения некоторого элемента из $\mathbf{MDS}_{\Phi,\Delta}(I)$ по заданным ДОЦ, обновлению и исходному состоянию БД. Другой важной проблемой, связанной с консервативными операторами обновлений, является задача перечисления всех состояний БД из множества $\mathbf{MDS}_{\Phi,\Delta}(I)$. Мы назовем эту проблему "проблемой перечисления наименьших достаточных изменений" (проблемой ПНДИ).

4 Алгоритмы консервативных обновлений

Проблема выполнения консервативных обновлений является, фактически, проблемой вычисления недетерминированной функции, которая по ДОЦ Φ, обновлению Δ и исходному состоянию БД I выдает результирующее состояние I' из множества $\mathbf{MDS}_{\Phi,\Delta}(I)$. В этом разделе мы покажем, что для частичных БД эта проблему можно решить за полиномиальное время, и что для полных БД она является полной в классе $\mathrm{FNP}//\mathrm{OptP}[O(\log n)]$.

Отметим, что поскольку рассматриваемые исходное состояние БД I , ДОЦ Φ и обновление Δ конечны, то и множество фактов из $\mathbf{LB}(\mathbf{S}, \mathbf{D})$, которые могут оказаться в состоянии, получающемся в результате обновления, является конечным (мы считаем, что константы $\mathbf{D}$ из результирующих состояний должны присутствовать в исходном состоянии, ДОЦ или в обновлении). Следующее утверждение позволяет оценить более точно множество литералов, которые могут появляться в решениях проблемы НДИ.

Предложение 5 *Пусть* $Atoms(\Phi, \Delta, I)$ *— это множество всех атомов, входящих (с отрицанием или без) в ДОЦ* Φ*, обновление* Δ *и состояние БД* I*. Тогда, если* $I' \in \mathbf{MDS}_{\Phi,\Delta}(I)$*, то* $I' \subseteq Atoms(\Phi, \Delta, I)$ *для полных БД и* $I' \subseteq Atoms(\Phi, \Delta, I) \cup \neg.Atoms(\Phi, \Delta, I)$ *для частичных БД.*

(2) Для любых ДОЦ Φ*, обновления* Δ *и состояния БД* I*, если* $I' \in \mathbf{MDS}_{\Phi,\Delta}(I)$*, то число литералов в* I' *не превосходит числа атомов в* $Atoms(\Phi, \Delta, I)$ *и, следовательно,* $|I'| \leq (|\Phi| + |\Delta| + |I|)$*.*

Прежде, чем описывать алгоритмы и оценки сложности для функций, приведем для сравнения полученные в работах [9, 11] результаты о сложности двух проблем разрешения, связанных с проблемой консервативных обновлений.

4.1 Сложность проблем разрешения

Мы будем рассматривать в этой работе комбинированную сложность алгоритмических проблем, т.е. сложность будет оцениваться по отношению к размеру входа проблемы, определяемому как сумма размеров всех его параметров:
$N = |\mathbf{D}| + |I| + |\Delta| + |\Phi|$
(здесь $|\ |$ обозначает размер соответствующих множеств литералов и программы в некоторой стандартной кодировке).

Проблемы разрешения (т.е. проблемы вычисления двузначных функций (предикатов)), которые изучались в [9, 11], связаны с вопросом о том, будет ли некоторый факт верен после обновления. Мы называем их *оптимистическое и пессимистическое попадание в результат* (**ОПР** и **ППР**). Они формулируются следующим образом:

ОПР: По заданному обновлению $\Delta \in \mathbf{UP}$, совместному с ДОЦ $\Phi \in \mathbf{IC}$, исходному состоянию I, и литералу $l \in \mathbf{LB^c}$ определить, существует ли состояние БД I_1 такое, что:
(a) $I_1 \in \mathbf{MDS}_{\Phi,\Delta}(I)$ и
(b) $I_1 \models l$.

ППР: отличается тем, что условие (b) должно выполняться *для всех* состояний БД из $\mathbf{MDS}_{\Phi,\Delta}(I)$.

Обозначим через **ОПР** и **ППР** множества всех решений (I, Δ, Φ, l) проблем **ОПР** и **ППР**, соответственно. Легко видеть, что для полных БД эти множества являются взаимно дополнительными (т.е. $(I, \Delta, \Phi, l) \in$ **ППР** $\Longleftrightarrow (I, \Delta, \Phi, \neg.l) \notin$ **ОПР**). Из приведенной ниже теоремы 4 видно, что эта дуальность имеет место и для частичных БД.

Для полных БД в работе [9] были установлены следующие оценки сложности.

Теорема 3 *(Полные БД) [9].*
(1) Если Φ является дефинитной логической программой (т.е. она не содержит отрицаний) и $\Delta^- = \emptyset$, то обе проблемы **ОПР** *и* **ППР** *разрешимы за полиномиальное время.*
(2) Если Φ является дефинитной логической программой, то **ОПР** *является NP-полной проблемой, а* **ППР** *является co-NP-полной проблемой.*
(3) В общем случае **ОПР** *является Σ_2^p-полной, а* **ППР** *является Π_p^2-полной проблемой.*

Для частичных БД обе проблемы являются более простыми.

Теорема 4 *(Частичные БД) [11].*
(1) Если выполнены следующие ограничения на входные параметры :
a) Φ – нормальная логическая программа (нет отрицаний в головах правил),
b) $\Delta^+ \subseteq \mathbf{B}$ и $\Delta^- = \emptyset$ (нет удалений и отрицаний в Δ), и
c) $I \subseteq \mathbf{B}$ (нет отрицаний в I),
то обе проблемы **ОПР** *и* **ППР** *разрешимы за полиномиальное время.*
(2) Если хотя бы одно из ограничений a), b), c) нарушено, то **ОПР** *является NP-полной, а* **ППР** *является $coNP$-полной.*

Отметим, что приведенные выше теоремы доказаны в указанных работах для случая, когда все ограничения статические и в обновлениях нет замен ($\Delta^* = \emptyset$). Но нетрудно видеть, что они останутся справедливыми и в рассматриваемом нами более общем случае. Действительно, следующее простое утверждение показывает, что для заданного исходного состояния БД I замены в обновлении можно устранить, расширив множества Δ^+ и Δ^-.

Предложение 6 *Для произвольных обновления Δ и состояния БД I положим $\Delta_1^- = \Delta^- \cup (I \cap \Delta_{in})$ $\Delta_1^+ = \Delta^+ \cup \{l_1 |$ существует $l \in (I \cap \Delta_{in})$ такое, что $(l, l_1) \in \Delta^*\}$, и $\Delta_1^* = \emptyset$. Тогда $Acc_I(\Phi, \Delta) = Acc_I(\Phi, \Delta_1)$ и, следовательно, $\mathbf{MDS}_{\Phi,\Delta}(I) = \mathbf{MDS}_{\Phi,\Delta_1}(I)$.*

Что же касается рассмотрения динамических ограничений целостности вместо статических, то нижние оценки сложности переносятся автоматически, а верхние можно получить, обобщив естественным образом соответствующие алгоритмы из [9, 11].

4.2 Задачи поиска НДИ для полных БД

В этом параграфе мы вначале рассмотрим для полных БД задачу поиска решения проблемы НДИ: по заданным ДОЦ $\Phi \in \mathbf{IC}$, обновлению Δ и исходному состоянию БД I найти некоторое состояние БД (решение) $I' \in \mathbf{MDS}_{\Phi,\Delta}(I)$.

Определим вспомогательное множество:
$A = \{(I, \Delta^+, \Delta^-) | \exists I' \in Acc_I(\Phi, \Delta)(\ \Delta^+ \subseteq I' \ \&\ \Delta^- \cap I' = \emptyset)\}$
Нетрудно проверить, что это множество принадлежит классу NP. Действительно, для проверки того, что тройка (I, D^+, D^-) входит в A, достаточно угадать некоторое подмножество атомов I' и проверить выполнены ли для него следующие условия:
1) $<I, I'> \models \Phi$;
2) обновление Δ выполнено на $<I, I'>$;
3) $\Delta^+ \subseteq I'$;
4) $\Delta^- \cap I' = \emptyset$.
В случае выполнения всех условий выдать ответ "Да".
Каждое из условий 1–4 очевидным образом можно проверить за полиномиальное время и, следовательно, $A \in$ NP. Из теоремы 1 непосредственно следует, что A является NP-полным множеством.

Следующий алгоритм решает задачу поиска НДИ относительно оракула A из NP.
Алгоритм НДИ1:
Вход: совместные ДОЦ $\Phi \in \mathbf{IC}$ и обновление Δ, полное состояние БД I.
Выход: состояние БД I'.

1. Пусть $I \setminus \Delta^- = \{a_1, \ldots, a_k\}$;
2. $D^+ := \Delta^+$;
3. **FOR** $i = 1$ **TO** k **DO**
4.   **IF** $(I, D^+ \cup \{a_i\}, \emptyset) \in A$
5.   **THEN** $D^+ := D^+ \cup \{a_i\}$
6.   **END_IF**
7. **END_DO**
8. Пусть $H = \mathbf{B} \setminus (I \cup \Delta^+ \cup \Delta^-) = \{b_1, \ldots, b_m\}$;
9. $D^- := \emptyset$;
10. **FOR**$i = 1$ **TO** m **DO**
11.   **IF** $(I, D^+, D^- \cup \{b_i\}, \emptyset) \in A$
12.   **THEN** $D^- := D^- \cup \{b_i\}$
13.   **END_IF**
14. **END_DO**;
15. $I' := D^+ \cup (H \setminus D^-)$;
16. **Output** I'

Заметим, что результат работы алгоритма может зависеть от порядка перечисления атомов в строках 1 и 8.

Предложение 7 *(Полные БД)*
(1) Если алгоритм НДИ1 *вычисляет результат I', то $I' \in \mathbf{MDS}_{\Phi,\Delta}(I)$.*

(2) Для любого состояния $I' \in \mathbf{MDS}_{\Phi,\Delta}(I)$ существуют такие упорядочения атомов в строках 1 и 8 алгоритма НДИ1, *при которых он выдает I' в качестве результата.*
(3) Задача поиска решения проблемы НДИ принадлежит классу FP^{NP}.

Следующее утверждение показывает, что верхнюю оценку сложности поиска НДИ можно несколько уточнить и определяет также нижнюю оценку для этой задачи.

Теорема 5 *Задача поиска решения проблемы НДИ является полной в классе $FNP//OptP[O(\log n)]$.*

Доказательство. Верхняя оценка. Для каждой входной тройки Φ, Δ, I определим множество пар чисел
$B(\Phi, \Delta, I) = \{(k, m) \mid \exists I' \in Acc_I(\Phi, \Delta)(\ |I \cap I'| = k \ \& \ |Atoms(\Phi, \Delta, I) \setminus (I' \cup I)| = m)\}$.
Ясно, что $B(\Phi, \Delta, I) \in NP$. Обозначим через $N(\Phi, \Delta, I)$ мощность множества $Atoms(\Phi, \Delta, I)$. Тогда из предложения 5 следует, что для $(k, m) \in B(\Phi, \Delta, I)$ $k+m \leq N(\Phi, \Delta, I)$. Зафиксируем некоторую нумерацию ν троек натуральных чисел (N, x, y), у которых $N \geq x$ и $N \geq y$, такую, что:
1) $|\nu(N, x, y)| = O(|N|) = O(\log N)$;
2) для всякого N, если $x_1 > x_2$, то для любых y_1, y_2 имеет место $\nu(N, x_1, y_1) > \nu(N, x_2, y_2)$.
В качестве такой функции можно взять, например, $\nu(N, x, y) = 2^{2|N|}N + 2^{|N|}x + y$. Положим $F(\Phi, \Delta, I) = \max\{\nu(N(\Phi, \Delta, I), k, m) \mid (k, m) \in B(\Phi, \Delta, I)\}$. Тогда задача вычисления функции $F(\Phi, \Delta, I)$ является оптимизационной NP-проблемой, причем длина ее решения $|F(\Phi, \Delta, I)| \leq O(\log(|\Phi| + |\Delta| + |I|)$. Кроме того, из определений множества $B(\Phi, \Delta, I)$ и функций $\nu(N, x, y)$ и $F(\Phi, \Delta, I)$ непосредственно следует, что для любых Φ, Δ и I
$F(\Phi, \Delta, I) = t > 0$ тогда и только тогда, когда существует $I' \in \mathbf{MDS}_{\Phi,\Delta}(I)$ такое, что $|I \cap I'| = k, \ |Atoms(\Phi, \Delta, I) \setminus (I' \cup I)| = m$ и $t = \nu(N(\Phi, \Delta, I), k, m)$. Рассмотрим следующий недетерминированный алгоритм.

Алгоритм НДИ2 :

Вход: совместные ДОЦ $\Phi \in \mathbf{IC}$ и обновление Δ, состояние БД I и значение функции $F(\Phi, \Delta, I) = t$.
Выход: состояние БД I'.

1. Восстановить по t такие значения N, k и m, что $\nu(N, k, m) = t$;
2. Недетерминированно угадать $I' \subseteq Atoms(\Phi, \Delta, I)$;
3. **IF** $(|I \cap I'| = k$ **AND** $|Atoms(\Phi, \Delta, I) \setminus (I' \cup I)| = m)$ **AND**
   $<I, I'> \models \Phi$ **AND** обновление Δ выполнено на $<I, I'>$
4. **THEN Output** I'
5. **END_IF**.

Из свойств функции F и выполнения теста в строке 3 следует, что если алгоритм НДИ2 выдает некоторое сотояние БД I', то оно входит в $\mathbf{MDS}_{\Phi,\Delta}(I)$, т.е. является решением проблемы поиска НДИ. Таким образом, эта проблема поиска принадлежит классу $\mathrm{FNP}//\mathrm{OptP}[O(\log n)]$.

Нижняя граница. Для доказательства трудности задачи поиска решения проблемы НДИ для класса $\mathrm{FNP}//\mathrm{OptP}[O(\log n)]$ мы сведем к ней задачу вычисления X-максимальной модели. Она формулируется следующим образом: по заданной КНФ φ и подмножеству ее переменных X вычислить такую X-часть некоторой модели M формулы φ, что $M \cap X$ максимально, т.е. у φ нет такой модели M', что $M \cap X \subset M' \cap X$ (здесь мы идентифицируем модель булевой формулы с множеством истинных в ней переменных). Трудность этой задачи для класса $\mathrm{FNP}//\mathrm{OptP}[O(\log n)]$ установлена в [4]. Пусть $\varphi = \beta_1 \wedge ... \wedge \beta_m$ — это произвольная 3-КНФ над множеством переменных $X_\varphi = \{x_1, \ldots, x_k\}$, являющаяся конъюнкцией m дизъюнктов вида $\beta_j = l_1^j \vee l_2^j \vee l_3^j$, где $l_i^j \in X_\varphi \cup \neg.X_\varphi$ для всех $1 \leq j \leq m$ и $1 \leq i \leq 3$. Положим $\mathbf{B} = \{a\} \cup X_\varphi$. Для каждого $j = 1, ..., m$ включим в ДОЦ Φ_φ ограничение b_j :

$\neg a' \leftarrow \neg.n(l_1^j), \neg.n(l_2^j), \neg.n(l_3^j)$.

Положим $\Delta = (\{a\}, \emptyset, \emptyset)$, а в качестве исходного состояния БД возьмем $I = X$, где $X \subseteq X_\varphi$. Из этих определений непосредственно следует, что для всякого состояния БД $I' \in Acc_I(\Phi_\varphi, \Delta)$ формула φ истинна на модели $I' \setminus \{a\} \subseteq X_\varphi$. Действительно, поскольку $a \in I'$, то тело любого правила b_j $(j = 1, \ldots, m)$ должно быть ложно на I', а это значит, что для некоторого $i \in \{1, 2, 3\}$ $< I, I' > \models n(l_j^i)$, т.е. $I' \models l_j^i$. Но тогда $I' \models \beta_j$ для всех j и $I' \setminus \{a\} \models \varphi$. С другой стороны, если M — произвольная модель φ, то состояние $I' = \{a\} \cup M$, очевидно, входит в $Acc_I(\Phi_\varphi, \Delta)$. Если теперь $I' \in \mathbf{MDS}_{\Phi_\varphi,\Delta}(I)$, то в соответствии с нашим критерием близости в $Acc_I(\Phi_\varphi, \Delta)$ нет состояния I'' более близкого к I чем I', а это означает, что нет такой модели M формулы φ, для которой $(I' \setminus \{a\}) \cap X \subset M \cap X$. Таким образом, по произвольному решению I' задачи поиска НДИ для тройки $(\Phi_\varphi, \Delta, I)$ легко находится решение $(I' \setminus \{a\}) \cap X$ задачи вычисления X-максимальной модели формулы φ. Поскольку конструкция указанной тройки по φ и X и вычисление одного оптимального значения по другому, очевидно, выполняются за полиномиальное время, то задача поиска решения проблемы НДИ является трудной для класса $\mathrm{FNP}//\mathrm{OptP}[O(\log n)]$. $\square$

Отметим, что приведенные выше алгоритмы *НДИ1* и *НДИ2* для поиска решения проблемы НДИ являются недетерминированными. Причем, если первый из них может породить любое решение, то алгоритм *НДИ2*, вообще говоря, порождает лишь некоторые элементы из $\mathbf{MDS}_{\Phi_\varphi,\Delta}(I)$. Конечно, для практики важно иметь детерминированные алгоритмы поиска одного или всех решений проблемы НДИ. Поэтому сейчас мы рассмотрим детерминированное решение проблемы ПНДИ: по заданным ДОЦ Φ, обновлению Δ и исходному состоянию БД I перечислить множество

$\mathbf{MDS}_{\Phi,\Delta}(I)$ *всех* возможных результатов консервативных обновлений.

Для каждого подмножества $X \subseteq \mathbf{B}(\mathbf{S},\mathbf{D})$ зафиксируем некоторый стандартный линейный порядок на его подмножествах (например, “лексикографический”), согласованный с топологическим частичным порядком, задаваемым отношением включения. Свяжем с ним функцию следования $next_X$, перечисляющую подмножества X в этом порядке. Положим $next_X(X) = \bot$ для некоторой специальной константы $\bot$. Следующий алгоритм перечисляет все решения проблемы НДИ (если таковые существуют) для полных БД.

Алгоритм ПНДИ(Φ, Δ, I):

Вход: полное состояние БД I, и обновление $\Delta = (D^+, D^-, D^*)$ совместимое с $\Phi \in \mathbf{IC}$ для I.

Локальные переменные: d^- (заменяемые факты из I),

d^+ (факты, добавляемые в результате замены),

$\tilde{I}$ (результат первоначального обновления),

H_{add} (текущие добавляемые факты),

H_{del} (текущие удаляемые факты),

H^+ (пространство поиска для добавления H_{add}),

H^- (пространство поиска для удаления H_{del}).

Выход: Список TU всех состояний из $\mathbf{MDS}_{\Phi,\Delta}(I)$.

$d^- = I \cap \{l_1 \mid \exists l_2(\ (l_1, l_2) \in D^*\ )\}$;

$d^+ = \{l_2 \mid (\exists l_1 \in d^-)(\ (l_1, l_2) \in D^*\ )\}$;

$\tilde{I} := (I \cup D^+ \cup d^+) \setminus (D^- \cup d^-)$;

$H^- := \tilde{I} \setminus (D^+ \cup d^+)$;   % пространство поиска для H_{del}

$H^+ := \mathbf{B} \setminus (\tilde{I} \cup D^- \cup d^-)$; % пространство поиска для H_{add}

$H_{del} := \emptyset$;

$TU := \emptyset$;

WHILE $H_{del} \neq \bot$ **DO**

    $H_{add} := \emptyset$;

    WHILE $H_{add} \neq \bot$ **DO**

        $I_1 := (\tilde{I} \setminus H_{del}) \cup H_{add}$;

        IF существует $r \in \Phi$ такое, что $< I, I_1 > \not\models r$

        THEN $H_{add} := next_{H^+}(H_{add})$

        ELSE IF нет такого $I' \in TU$ которое ближе к I чем I_1

            THEN $TU := TU \cup \{I_1\}$

            END_IF

        END_IF

    END_DO

    $H_{del} := next_{H^-}(H_{del})$

END_DO

(21) **Output** TU.

Теорема 6 *(Полные БД).*
Алгоритм ПНДИ перечисляет множество $\mathbf{MDS}_{\Phi,\Delta}(I)$ *за время* $O(2^{2(d+b)}N)$, *где* $N = |\Phi| + |\Delta| + |I|$, d *– это размер (число литералов) в* H^-, *а* b *– размер* H^+.

Доказательство. Нетрудно проверить, что для всякого состояни $I' \in Acc_I(\Phi, \Delta)$ существуют $H'_{del} \subseteq H^-$ и $H'_{add} \subseteq H^+$ такие, что $I' = (\tilde{I} \setminus H'_{del}) \cup H'_{add}$. Поскольку $Acc_I(\Phi, \Delta) \neq \emptyset$ и алгоритм ПНДИ(I, Φ, Δ) перебирает все пары (H_{del}, H_{add}), у которых $H_{del} \subseteq H^-$ и $H_{add} \subseteq H^+$, то он в конце концов найдет и проверит (в строке (14)) каждое состояние $I_1 \in Acc_I(\Phi, \Delta)$. если выяснится, что $I_1 \in \mathbf{MDS}_{\Phi,\Delta}(I)$, то I_1 будет включено в TU (строка (15)). Следовательно, $\mathbf{MDS}_{\Phi,\Delta}(I) \subseteq TU$. С другой стороны, порядок в котором перебираются параы (H_{del}, H_{add}), индуцируемый функциями $next_{H^-}$ и $next_{H^+}$, гарантирует, что:

- состояние I_1, первым попавшее в TU в строке (15), принадлежит $\mathbf{MDS}_{\Phi,\Delta}(I)$, и

- если состояние I' добавляется к TU после I'', то I' не может быть ближе к I чем I''.

Поэтому все состояния БД, помещаемые алгоритмом в TU, входят в $\mathbf{MDS}_{\Phi,\Delta}(I)$ и $TU = \mathbf{MDS}_{\Phi,\Delta}(I)$.

Чтобы оценить сложность, отметим, что внешний цикл (строки (8) - (20)) повторяется 2^d раз и на каждой его итерации внутренний цикл (строки (10) - (18)) выполняется не более 2^b раз. Число выходных состояний БД в TU не превышает 2^{d+b}, а размер каждого из них не больше N. Поэтому непосредственная проверка условия в строке (14) требует не более $O(2^{d+b}N)$ шагов. Сложность каждого из операторов в остальных строках не превышает $O(N)$. Суммируя, мы получаем оценку сложности из теоремы. $\square$

4.3 Задачи поиска НДИ для частичных БД

Задачи поиска решения проблемы НДИ для частичных БД оказываются, в целом, проще, чем те же задачи для полных БД. В этом параграфе мы приводим алгоритм, который вычисляет для частичных БД некоторый оператор консервативных обновлений за полиномиальное время. А затем используем его для построения алгоритма перечисления всех решений проблемы НДИ.

Вначале установим некоторые важные свойства выполнимости ДОЦ для частичных БД.

Лемма 3 *В случае частичных БД для любых ДОЦ* $\Phi \in \mathbf{IC}$*:*
(1) если для некоторого перехода $< S, S' >$ *замыкание* $M_{\Phi}^{min}(< S, S' >)$ *корректно,*

то $M_{\Phi}^{min}(< S, S' >) \models \Phi$*;*

(2) если переход $<I, I'>$ *корректен и* $<I, I'>\models \Phi$, *то для любого перехода* $< S, S' >\subseteq<I, I'>$ $\quad M_{\Phi}^{min}(< S, S' >) \subseteq<I, I'>$ *и* $M_{\Phi}^{min}(< S, S' >) \models \Phi$;

(3) если обновление Δ *совместно с ДОЦ* Φ *для состояния БД* I *и* $M_{\Phi(\Delta)}^{min}(< I, \emptyset >) = <I, I'>$, *то* I' *является минимальным состоянием в* $Acc_I(\Phi, \Delta)$;

(4) если $I_1 \in \mathbf{MDS}_{\Phi,\Delta}(I)$, *то* $< I, I_1 >= M_{\Phi(\Delta)}^{min}(< I, I \cap I_1 >)$.

Доказательство. Свойства (1) и (2) непосредственно следуют из определения замыкания $M_{\Phi}^{min}(< S, S' >)$ и того, что в базисном случае вычисление замыкания всегда завершается за конечное число шагов.

Для доказательства (3) заметим, что из совместимости Φ и Δ для I следует, что $Acc_I(\Phi, \Delta) \neq \emptyset$. Пусть I'' — произвольное состояние БД из $Acc_I(\Phi, \Delta)$. Тогда $< I, \emptyset >\subseteq< I, I'' >$ и по пункту (2) $M_{\Phi}^{min}(< I, \emptyset >) =< I, I' >\subseteq< I, I'' >$ и $M_{\Phi}^{min}(< I, \emptyset >) \models \Phi$. Следовательно, $I' \subseteq I''$, т.е. I' является минимальным состоянием в $Acc_I(\Phi, \Delta)$.

Утверждение (4) также следует из (2). Действительно, так как $< I, I \cap I_1 > \subseteq < I, I_1 >$, то $M_{\Phi}^{min}(< I, I \cap I_1 >) =< I, I' >\subseteq< I, I_1 >$ и $< I, I' >\models \Phi$; . Тогда из монотонности оператора замыкания следует, что $I \cap I_1 \subseteq I' \cap I$. А из наибольшей близости I_1 к I получаем, что $I' \cap I = I_1 \cap I$ и $I' \setminus I = I_1 \setminus I$, т.е. $I' = I$. □

Эти свойства оператора замыкания для частичных БД позволяют построить полиномиальный алгоритм для порождения некоторого состояния из $\mathbf{MDS}_{\Phi,\Delta}(I)$. Поскольку проверку совместности Δ с Φ для состояния БД I для частичных БД можно выполнить просто, мы будем рассматривать в следующем алгоритме только совместные входы.

Алгоритм ЧНДИ(Φ, Δ, I):

Вход: частичное состояние БД I, и обновление $\Delta = (D^+, D^-, D^*)$ совместное с $\Phi \in \mathbf{IC}$ для I.

Выход: состояние БД I'.

$d^- = D^- \cup \{l_1 \mid \exists l_2(\ (l_1, l_2) \in D^*\ )\}$;

$d^+ = D^+ \cup \{l_2 \mid (\exists l_1 \in d^-)(\ (l_1, l_2) \in D^*\ )\}$;

$\tilde{I} := I \setminus (d^+ \cup d^-)$;

Зафиксировать некоторое упорядочение $\{l_1, ..., l_k\}$ всех литералов из множества $\tilde{I}$;

$I_2 := I \cap d^+$;

FOR $i = 1$ **TO** k **DO**

$\qquad < I_1', I_2' >:= M_{\Phi(\Delta)}^{min}((\,) < I, I_2 \cup \{l_i\} >)$;

$\qquad$ **IF** $(I_1' = I)$ **AND** $(< I_1', I_2' >$ корректен$)$ **AND** $(I_2' \cap d^- = \emptyset)$

$\qquad$ **THEN** $I_2 := I_2 \cup \{l_i\}$

$\qquad$ **END_IF**

(11) **END_DO**;
(12) $<I, I'>:= M^{min}_{\Phi(\Delta)}(<I, I_2>)$;
(13) **Output** I'.

Теорема 7 *(1) Для всякого частичного состояния БД I, ДОЦ Φ и обновления $\Delta = (D^+, D^- D^*)$, совместного с Φ для I, алгоритм ЧНДИ вычисляет состояние БД $I_2 \in \mathbf{MDS}_{\Phi,\Delta}(I)$ за полиномиальное (фактически, квадратичное) время.*
(2) Для любого состояния $I' \in \mathbf{MDS}_{\Phi,\Delta}(I)$ существует такое упорядочение литералов в строке 4 алгоритма, что результат его работы будет равен I'.

Доказательство. (1) Так как Δ совместно с Φ для I, то $Acc_I(\Phi, \Delta)$ не пусто. Тогда из определения выполнимости обновления следует, что для каждого $I' \in Acc_I(\Phi, \Delta)$ и для определенных в строках 1 и 2 алгоритма множеств d^- и d^+ выполнены соотношения: $d^+ \subseteq I'$ и $d^- \cap I' = \emptyset$. Тогда всякий литерал из $I \cap I'$ либо принадлежит d^+, либо входит во множество $\tilde{I}$, определенное в 3-ей строке алгоритма. Покажем теперь, что инвариантом цикла в строках 6–11 являются следующие условия, выполненные для каждого $i = 0, 1, \ldots, k$ после i-ой итерации цикла:
(а) существует $I'_2 \in Acc_I(\Phi, \Delta)$ такое, что $I_2 \subseteq I'_2 \cap I$ и, (б) если для некоторого $j \leq i$ литерал $l_j \notin I_2$, то нет такого состояния $I' \in Acc_I(\Phi, \Delta)$, для которого $(I_2 \cup \{l_j\}) \subseteq I'$. Перед началом цикла ($i = 0$) условие (а) выполнено по лемме 3 (2), так как $I \cap d^+ \subseteq I'$ для любого $I' \in Acc_I(\Phi, \Delta)$. Условие (б) при $i = 0$ тривиально. Предположим теперь, что оба условия выполнены перед i-ой итерацией цикла, и покажем, что они останутся справедливы и после этой итерации. Действительно, если тест в строке 9 выполнен, то в строке 10 I_2 получает новое значение такое, что I'_2, определенное в строке 7, принадлежит $Acc_I(\Phi, \Delta)$ (по леммам 4 (2) и 3 (2)). Условие (б) выполнено по предположению индукции, т.к. если $l_j \notin I \cap I_2$, то $j \leq i - 1$, поскольку $l_i \in I_2$. Если же тест в строке 9 ложен, то I_2 после i-ой итерации не меняется, и, следовательно, условие (а) остается верным по индукционному предположению. Пусть теперь l_j ($j \leq i$) — такой литерал, что $l_j \notin I_2$. Если $j < i$, то условие (б) выполнено в силу индукционного предположения. Если же $j = i$ и некоторое состояние $I' \in Acc_I(\Phi, \Delta)$ содержит $I_2 \cup \{l_i\}$, то по лемме 3 (2) это означает, что I'_2, определенное в строке 7, также принадлежит $Acc_I(\Phi, \Delta)$ и все условия теста в строке 8 выполнены для перехода $< I'_1, I'_2 >$. Но это противоречит нашему предположению о его нарушении. Следовательно, индукционное условие (б) выполнено и в этом случае.

Рассмотрим теперь состояние I', возвращаемое в строке 13 алгоритма. Оно получает свое значение в строке 12 алгоритма после k-ой итерации цикла 6–11. При этом $< I, I' >= M^{min}_{\Phi(\Delta)}(< I, I_2 >)$. Тогда из инварианта (а) цикла следует, что $I' \in Acc_I(\Phi, \Delta)$, а из инварианта (б) — что $I' \cap I = I_2$. Отсюда по лемме 3 (4) заключаем, что $I' \in \mathbf{MDS}_{\Phi,\Delta}(I)$.

Пусть N — это суммарная длина входных данных алгоритма. Для оценки времени его работы отметим, что цикл в строках 6–11 выполняется не более N раз, оператор замыкания в строке 7 можно вычислить за линейное от N время, операторы в остальных строках еще проще. Поэтому время работы алгоритма *ЧНДИ* не превосходит $O(N^2)$.

(2) Покажем, что при подходящем упорядочении литералов в строке 4 алгоритм *ЧНДИ* может вычислить произвольное состояние $I' \in \mathbf{MDS}_{\Phi,\Delta}(I)$. Пусть множество $I' \cap \tilde{I}$ содержит $r \leq k$ литералов. Зафиксируем в строке 4 такое упорядочение $\{l_1, ..., l_k\}$ литералов из множества $\tilde{I}$, в котором элементы множества $I' \cap \tilde{I}$ идут вначале, т.е. $\{l_1, ..., l_r\} = I' \cap \tilde{I}$. Тогда на первых r итерациях цикла 6–11 все эти литералы попадут в I_2, так как для них будут выполнены все условия теста в строке 7. На следующих итерациях ни один из оставшихся литералов $l_j,\ j > r,$ в I_2 не добавится, так как I' имеет максимальное пересечение с I. Таким образом, в конце работы алгоритма $I_2 = I' \cap I$ и по лемме 3 (4) выдаваемый в строке 13 результат есть I'.

Следствие 8 *Задача поиска решения проблемы НДИ для частичных БД принадлежит классу FP.*

Алгоритм *ЧНДИ* теперь несложно преобразовать в алгоритм, перечисляющий все решения проблемы НДИ. Рассмотрим следующий алгоритм *ПЧНДИ*, вход которого совпадает со входом алгоритма *ЧНДИ*, а выходом является список состояний БД PU, включающий все состояния из $\mathbf{MDS}_{\Phi,\Delta}(I)$. В качестве подпрограммы будем использовать функцию $perm(A, j)$, перечисляющую при $j = 1, 2, \ldots, k!$ в некотором порядке все перестановки $(l_{i_1}, \ldots, l_{i_k})$ произвольного k-элементного множества литералов A. Программа *ПЧНДИ* получается из программы *ЧНДИ* с помощью следующих изменений. Строка 4 заменяется на 4 строки:

(4.1) $PU := \emptyset$;

(4.2) $k := |\tilde{I}|$;

(4.3) **FOR** $j = 1$ **TO** $k!$ **DO**

(4.4) $(\{l_1, ..., l_k\} := perm(\tilde{I}, j)$;

Заключительная строка 13 заменяется на 5 строк:

(13) **IF** $I' \notin PU$ **THEN**

(14) $PU := PU \cup \{I'\}$

(15) **END_IF**

(16) **END_DO**;

(17) **Output** PU.

Таким образом в строке 4.1 инициализируется результирующий список PU. В строке 14 к нему добавляется вычисленное новое состояние из $\mathbf{MDS}_{\Phi,\Delta}(I)$. В строке 4.3 начинается внешний цикл, в котором перебираются (в строке 4.4) все перестановки литералов множества $\tilde{I}$ и для каждой из них в строках 5–11 ищется результирующее состояние I_2'. Из теоремы 7 (2) тогда следует, что в результирующем списке PU окажутся все состояния из $\mathbf{MDS}_{\Phi,\Delta}(I)$. Время работы алгоритма *ПЧНДИ* также легко оценить, исходя из оценки времени алгоритма *ЧНДИ*. Действительно, вставленный внешний цикл в строках 4.3–16 выполняется $k!$ раз, а его тело требует $O(N^2)$ шагов. Следовательно, общее время работы алгоритма *ПЧНДИ* равно $O(k!N^2)$. Таким образом, мы получили следующее утверждение.

Теорема 9 *Для всякого частичного состояния БД I, ДОЦ Φ и обновления Δ, совместного с Φ для I, алгоритм ПЧНДИ корректно перечисляет все состояния БД из $\mathbf{MDS}_{\Phi,\Delta}(I)$ за время $O(k!N^2)$, где $N = |\Phi| + |\Delta| + |I|$, а k — это мощность множества $(I \setminus (d^+ \cup d^-))$.*

5 Операторы расширения обновлений

5.1 Обобщенные обновления и их расширения

Приведенные выше результаты о сложности проблем разрешения и проблем поиска, связанных с задачей корректного выполнения консервативных обновлений (теоремы 3,4,6,9), показывают, что в большинстве случаев эти проблемы весьма сложны, и не оставляют много надежд (например, если $P \neq NP$) на получение для них эффективных алгоритмов. Полиномиальный алгоритм, решающий задачу поиска некоторого решения проблемы НДИ для частичных БД, является здесь приятным исключением. Однако это не исключает возможности существования практических методов, позволяющих в ряде случаев существенно ускорить работу рассмотренных выше алгоритмов. В этом разделе мы распространим на динамические ограничени целостности один такой метод, предложенный в работах [9, 10] для полных и частичных БД при статических ограничениях целостности. Цель этого метода состоит в том, чтобы по возможности уменьшить пространство поиска H^- в алгоритме *ПНДИ* и число $k!$ выполнений главного цикла в алгоритме *ПЧНДИ*. Идея метода состоит в том, чтобы расширить множества Δ^+ и Δ^-, протягивая их через Φ и одновременно упрощая сами ограничения Φ. Подчеркнем, что всякое расширение множеств добавляемых и удаляемых литералов весьма существенно сказывается на времени работы указанных алгоритмов. Из теорем 6,9 следует, что каждый дополнительный литерал во множестве $\Delta^+ \cup \Delta^-$ уменьшает время работы алгоритма поиска *ПНДИ* по крайней мере в

2 раза. Алгоритмы *НДИ1* и *НДИ2* при этом также работают быстрее, т.к. уменьшается число выполнений их основных циклов. Легко увидеть, что при добавлении r новых элементов к $\Delta^+ \cup \Delta^-$ алгоритм *ПЧНДИ* ускорится в $(k-r+1)\dots(k-1)k$ раз. Отметим также, что упрощение ДОЦ Ф за счет удаления из них некоторых ограничений и некоторых литералов из тел правил сокращает время проверки выполнимости ДОЦ на переходах, так как уменьшается число обращений к БД.

Как мы уже отмечали, одно из отличий выполнения обновлений для динамических ограничений целостности по сравнению со статическими состоит в том, что обновление может оказаться неприменимым к некоторому состоянию БД, несмотря на его совместность с ограничениями целостности. Пусть, например, ДОЦ некоторой кадровой базы даннных содержат следующее правило, отражающее непрерывность продвижения по служебной лестнице:

Должность(X,программист) $\leftarrow$ *Должность*$'$(X, *старший_ программист).*

Предположим, что требуется добавить факт *Должность(петров, старший_ программист).* Тогда из указанного правила непосредственно следует, что в состоянии базы, к которому применяется это обновление, должен присутствовать факт *Должность(петров, программист).* Наличие этого факта может включить в работу другие правила, которые приведут к появлению новых фактов, присутствие которых необходимо в исходном состоянии или будет необходимо в результирующем, и т.д. Чтобы единообразно учитывать такие расширения обновлений, мы сейчас обобщим само понятие обновления.

Определение 6 Обобщенное обновление — *это тройка* $\Delta = (D^+, D^-, D^*)$, *где для частичных БД каждое из* D^+, D^- *— это пара подмножеств* $\mathbf{LB^c}$, *а* $D^* \subseteq \mathbf{LB^c} \times \mathbf{LB^c}$, *а для полных БД* D^+ *и* D^- *являются парами подмножеств* $\mathbf{B^c}$ *а* $D^* \subseteq \mathbf{B^c}(\mathbf{S},\mathbf{D}) \times \mathbf{B^c}(\mathbf{S},\mathbf{D})$.

Содержательно, пара $D^+ =< d^+, d'^+ >$ состоит из множества фактов d^+, которые должны присутствовать в исходном состоянии, и множества фактов d'^+, которые требуется добавить к исходному состоянию БД, а пара $D^- =< d^-, d'^- >$ состоит из множества фактов d^-, которых не должно быть в исходном состоянии, и множества фактов d'^-, которые следует из него удалить. Пары литералов из D^* как и прежде являются аргументами команды ЗАМЕНИТЬ.

Мы будем обозначать компоненты D^+, D^- и D^* обновления Δ как $\Delta^+ = < \delta^+, \delta'^+ >$, $\Delta^- =< \delta^-, \delta'^- >$ и Δ^*, соответственно. Для обоих видов баз данных через $\mathbf{GUP}(\mathbf{S},\mathbf{D})$ будем обозначать множество всех обобщенных обновлений в сигнатуре $\mathbf{S}$ с константами из $\mathbf{D}$, опуская, как правило, $\mathbf{S}$ и $\mathbf{D}$. Распространим на обобщенные обновления определение выполнимости на переходе.

Определение 7 *Обобщенное обновление* $\Delta \in \mathbf{GUP}$ выполнено на переходе $< I, I' >$ *тогда и только тогда, когда* $\delta^+ \subseteq I, \delta'^+ \subseteq I'$, $\delta^- \cap I = \emptyset$, $\delta'^- \cap I' = \emptyset$ *и для всякой пары* $(l, l') \in \Delta^*$, *если* $l \in I$, *то* $l \notin I'$, *а* $l' \in I'$.

Обозначим через $UT(\Delta)$ *множество всех переходов, на которых выполнено обновление* Δ.

Обновления Δ_1 *и* Δ_2 *назовем эквивалентными, если* $UT(\Delta_1) = UT(\Delta_2)$.

Расширим естественным образом покомпонентный порядок $\sqsubseteq$ на $\mathbf{GUP(S, D)}$, (напомним, что он определяется по первым двум компонентам обновления Δ^+ и Δ^-), а также сохраним для обобщенных обновлений обозначения классов переходов $Tr(\Phi, \Delta)$ и $Acc_I(\Phi, \Delta)$.

Отметим, что из этих определений непосредственно следует, что "обычное" обновление, которое требует добавить множество фактов D^+, удалить все факты из D^- и произвести замены D^*, эквивалентно обобщенному обновлению Δ с компонентами $\delta^+ = \emptyset$, $\delta'^+ = D^+$, $\delta^- = \emptyset$, $\delta'^- = D^-$ и $\Delta^* = D^*$. Дополнительные компоненты обобщенных обновлений δ^+ и δ^- могут помочь более раннему выявлению невозможности применить обновление к заданному состоянию. В алгоритмы проверки совместности Φ и Δ для состояния БД I или решения проблемы НДИ можно в самом начале добавить тест

(0) **IF** $\delta^+ \not\subseteq I$ **OR** $\delta^- \cap I \neq \emptyset$

THEN Output "Обновление Δ не выполнимо!"

При этом чем больше множества δ^+ и δ^-, тем больше шансов, что тест сработает и дальнейшей проверки не потребуется.

Для обобщенных обновлений множество условий корректности (КОР), определенное в разделе 2 можно несколько уточнить.

Предложение 8 *Для каждого обновления* $\Delta \in \mathbf{GUP}$ *такого, что* $UT(\Delta) \neq \emptyset$, *выполнены следующие условия:*

(1) $\delta^+ \cap \delta^- = \emptyset$ *и* $\delta'^+ \cap \delta'^- = \emptyset$,

(2) $\delta'^+ \cap \Delta^*_{in} = \emptyset$ *и*

(3) $\delta'^- \cap \Delta^*_{out} = \emptyset$.

Кроме того, существует эквивалентное преобразование обновлений $C(\Delta) = \Delta_1$, *для которого справедливы дополнительные условия:*

(4) $\delta_1'^+ \cap \Delta^*_{1,out} = \emptyset$,

(5) $\delta_1^+ \cap \Delta^*_{1,in} = \emptyset$ *и*

(6) $\delta_1^- \cap \Delta^*_{1,in} = \emptyset$.

Доказательство. Утверждения (1)–(3) очевидны. Покажем, как можно вычислть оператор $C(\Delta)$, последовательно устраняя нарушени условий (4)–(6).
Процедура $C(\Delta)$.

(C1) $\Delta_1 := \Delta$;
(C2) **FOR EACH** $l \in \delta_1^{'+} \cap \Delta_{1,out}^*$ **DO**
(C3) $\delta_1^{'-} := \delta_1^{'-} \cup \{l\}$;
(C4) $\Delta_1^* := \Delta_1^* \setminus \{(l_1, l) | (l_1, l) \in \Delta_1^*\}$
(C5) **END_DO**;
(C6) **FOR EACH** $l \in \delta_1^{+} \cap \Delta_{1,in}^*$ **DO**
(C7) $\delta_1^{'-} := \delta_1^{'-} \cup \{l\}$;
(C8) $\delta_1^{'+} := \delta_1^{'-} \cup \{l_1 | (l, l_1) \in \Delta_1^*\}$;
(C9) $\Delta_1^* := \Delta_1^* \setminus \{(l, l_1) | (l, l_1) \in \Delta_1^*\}$
(C10) **END_DO**;
(C11) **FOR EACH** $l \in \delta_1^{-} \cap \Delta_{1,in}^*$ **DO**
(C12) $\Delta_1^* := \Delta_1^* \setminus \{(l, l_1) | (l, l_1) \in \Delta_1^*\}$
(C13) **END_DO**;
(C14) **Output** Δ_1.

Нетрудно проверить, что в результате цикла (C2)–(C5) устраняются все нарушения условия (4), в результате цикла (C6)–(C10) устраняются все нарушения условия (5), а после цикла (C11)–(C13) не остается нарушений условия (6). Кроме того, из определения выполнимости ограничения на переходе следует, что условие $UT(\Delta_1) = UT(\Delta)$ является инвариантом каждого из указанных трех циклов, и следовательно, после завершения процедуры $UT(C(\Delta)) = UT(\Delta)$. $\square$

Ясно, что, как и для обычных обновлений, и проверка условий (1) – (3), и построение эквивалентного обновления $C(\Delta)$, удовлетворяющего условиям (4)–(6), могут быть выполнены эффективно за линейное время. Поэтому далее мы будем предполагать, не ограничивая общности, что для рассматриваемых обновлений выполнены условия (1) – (6).

Ниже мы вводим понятие операторов на множествах ограничений и (обобщенных) обновлений, корректно расширяющих множества добавляемых и удаляемых фактов, а также множества фактов, присутствие которых в исходном состоянии необходимо или запрещено, в соответствии с ДОЦ, и одновременно упрощающих ДОЦ в соответствии с расширенным обновлением. Вначале разобьем пространство $\mathbf{IC} \times \mathbf{GUP}$ на классы эквивалентности следующим образом.

Определение 8 *Пусть* Φ, Φ' *— ДОЦ,* Δ, Δ' *— обощенные обновления. Скажем, что пары* (Φ, Δ) *и* (Φ', Δ') *вляются* эквивалентными *(обозначение:* $(\Phi, \Delta) \equiv_u (\Phi', \Delta')$*),*

если $Tr(\Phi,\Delta) = Tr(\Phi',\Delta')$. *Для любой такой пары* (Φ,Δ) *положим*
$Equ(\Phi,\Delta) = \{(\Phi',\Delta') \mid (\Phi',\Delta') \equiv_u (\Phi,\Delta)\}$.

Как мы уже отмечали, порядок $\sqsubseteq$ на **GUP** и порядок $\preceq$ на **IC** имеют понятный вычислительный смысл, связанный с проблемой консервативных обновлений. Они индуцируют естественным образом следующий частичный порядок на каждом классе эквивалентности $Equ(\Phi,\Delta)$:

$$(\Phi_1,\Delta_1) \preceq (\Phi_2,\Delta_2) \iff \Delta_1 \sqsubseteq \Delta_2 \text{ и } \Phi_2 \preceq \Phi_1.$$

Наша основная цель далее состоит в поиске *максимальных элементов* в классе $Equ(\Phi,\Delta)$ относительно порядка $\preceq$, поскольку, как мы уже выяснили выше, использование его вместо пары (Φ,Δ) может обеспечить существенное ускорение алгоритмов обновления БД.

Отметим вначале следующий простой факт.

Предложение 9 *(1) Для любых ДОЦ* Φ,Φ_1 *и* Φ_2 *и обновлений* Δ,Δ_1 *и* Δ_2, *если* $(\Phi_1,\Delta_1) \in Equ(\Phi,\Delta)$ *и* $(\Phi_2,\Delta_2) \in Equ(\Phi,\Delta)$, *то и* $(\Phi_i,\Delta_3) \in Equ(\Phi,\Delta)$, *где* $i \in \{1,2\}$, $\delta_3^+ = \delta_1^+ \cup \delta_2^+$, $\delta_3^{'+} = \delta_1^{'+} \cup \delta_2^{'+}$, $\delta_3^- = \delta_1^- \cup \delta_2^-$, $\delta_3^{'-} = \delta_1^{'-} \cup \delta_2^{'-}$, $\Delta'^* = \Delta_1^* \cup \Delta_2^*$.
(2) В классе $Equ(\Phi,\Delta)$ *имеется одна или несколько пар с абсолютно максимальным обновлением* Δ_{max}, *где*
$\delta_{max}^+ = \bigcup\{\delta_1^+ \mid (\Phi,\Delta_1) \in Equ(\Phi,\Delta)\}$, $\delta_{max}^{'+} = \bigcup\{\delta_1^{'+} \mid (\Phi,\Delta_1) \in Equ(\Phi,\Delta)\}$,
$\delta_{max}^- = \bigcup\{\delta_1^- \mid (\Phi,\Delta_1) \in Equ(\Phi,\Delta)\}$ $\delta_{max}^{'-} = \bigcup\{\delta_1^{'-} \mid (\Phi,\Delta_1) \in Equ(\Phi,\Delta)\}$.

Что касается ограничений целостности, то может оказаться, что два или более $\preceq$-минимальных эквивалентных ДОЦ являются несравнимыми по отношению $\preceq$.

Пример 4 *Пусть* $\Phi_1 = \{r_1 : a' \leftarrow b';\ r_2 : b' \leftarrow a';\ r_3 : c' \leftarrow a'\}$, $\Phi_2 = \{r_1, r_2\} \cup \{r_3' : c' \leftarrow b'\}$ *и пусть требуется добавить факт* c, *т.е.* $\delta^{'+} = \{c\}$, *а остальные компоненты* Δ *пустые.*

Нетрудно проверить, что Φ_1 *и* Φ_2 *являются эквивалентными, но* $\preceq$*-несравнимыми. Так что в классе* $Equ(\Phi,\Delta)\}$ *имеются две несравнимые максимальные пары* (Φ_1,Δ) *и* (Φ_2,Δ).

После этих разъяснений естественно определть расширяющие операторы следующим образом.

Определение 9 *Оператор* $\Gamma : \mathbf{IC} \times \mathbf{UP} \to \mathbf{IC} \times \mathbf{UP}$ *является* оператором расширения обновлений, *если для всех пар* $\Phi \in \mathbf{IC}$ *и* $\Delta \in \mathbf{UP}$ *выполнены следующие условия:*

- $(\Phi,\Delta) \equiv_u \Gamma(\Phi,\Delta)$ *(корректность), и*
- $(\Phi,\Delta) \preceq \Gamma(\Phi,\Delta)$ *(расширение)*

Из этого определения непосредственно следует, что класс операторов расширения обновлений замкнут относительно композиции.

Предложение 10 *Пусть* Γ_1 *и* Γ_2 *— два оператора расширения обновлений. Тогда и оператор* $\Gamma = \Gamma_1 \circ \Gamma_2$ *является оператором расширения обновлений.*

5.2 Упрощение ограничений целостности

Прежде чем переходить к определению конкретных операторов рсширений, определим несколько вспомогательных понятий, связанных с упрощением ДОЦ для заданного обновления. Первое из них связано с литералами, присутствие которых в любом переходе удовлетворяющем заданному обновлению обязательно, и литералами, которые не могут появиться в таком переходе.

Определение 10 *Обновление* Δ вынуждает *множество литералов* A *(обозначение* $\Delta \models A$*) , если для всякого перехода* $<I, I'>$*, на котором выполнено обновление* Δ, $<I, I'> \models \bigwedge_{l \in A} l$.

Обновление Δ запрещает *множество литералов* A *(обозначение* $\Delta \not\models A$*) , если для всякого перехода* $< I, I' >$*, на котором выполнено обновление* Δ, $< I, I' > \not\models \bigwedge_{l \in A} l$.

В следующем утверждении понятия "вынуждения" и "запрещения" конструктивно уточняются для полных и частичных баз данных.

Лемма 4 *(1) Для полных БД*
$\Delta \models A \iff$ *для всякого литерала* $l \in A$ *либо* $l \in (\delta^+ \cup n(\delta^{'+}))$*, либо* $\neg l \in (\delta^- \cup n(\delta^{'-}))$;
$\Delta \not\models A \iff$ *либо в* A *имеется литерал* l *такой, что* $\neg.l \in (\delta^+ \cup n(\delta^{'+}))$ *или* $l \in (\delta^- \cup n(\delta^{'-}))$*, либо для некоторой пары атомов* $(a_1, a_2) \in \Delta^*$ *такой, что* $a_1 \in A$*, хотя бы один из литералов* $n(\neg a_2), n(a_1)$ *также входит в* A.
(2) Для частичных БД
$\Delta \models A \iff A \subseteq (\delta^+ \cup n(\delta^{'+}))$;
$\Delta \not\models A \iff$ *либо в* A *имеется литерал* l *такой, что* $\neg.l \in (\delta^+ \cup n(\delta^{'+}))$ *или* $l \in (\delta^- \cup n(\delta^{'-}))$*, либо для некоторой пары литералов* $(l_1, l_2) \in \Delta^*$ $l_1 \in A$ *и* $n(\neg.l_2) \in A$.

Сейчас мы определим эквивалентное относительно заданного обновления Δ преобразование $res_\Delta(\Phi)$ ДОЦ Φ, которое упрощает программу в соответствии с определенным на ДОЦ порядком $\preceq$. Мы зададим это преобразование с помощью алгоритма *comp_res*, который сначала удаляет из Φ правила, головы которых лежат в Δ^+, затем удаляет правила, тела которых запрещены обновлением Δ, а в конце удаляет из тел литералы из Δ^+ и литералы из Δ^*_{out}, “поддержанные” этими телами. При описании алгоритма мы для упрощения записи используем соглашение о соответствии пары множеств литералов без штрихов $< A, B >$ и множества литералов $A \cup n(B)$, например, $l \in \Delta^+$ является сокращением для $l \in (\delta^+ \cup n(\delta'^+)$.

Алгоритм comp_res для частичных БД
Вход: $\Phi \in \mathbf{IC}$, $\Delta \in \mathbf{GUP}$;
Выход: $\Phi_1 = res_\Delta(\Phi)$.

(1) $\Phi_1 := \Phi$;
(2) Удалить из Φ_1 все предложения r такие, что $head(r) \in \Delta^+$;
(3) Удалить из Φ_1 все предложения r такие, что $body(r) \cap (\neg.\Delta^+ \cup \Delta^- \neq \emptyset$;
(4) Удалить Φ_1 все предложения r такие, что для некоторой пары $(l_1, l_2) \in \Delta^*$
$l_1 \in body(r)$ и $\{\neg.n(l_2), n(l_1)\} \cap body(r) \neq \emptyset$;
(5) **FOR EACH** $r \in \Phi_1$ **DO**
(6) $body(r) := body(r) \setminus (\delta^+ \cup n(\delta'^+))$
(7) **END_DO**;
(8) **FOR EACH** $r \in \Phi_1$ **DO**
(9) $X := \{n(l)|$ для некоторой пары $(l_1, l) \in \Delta^*$ $(l_1 \in body(r))\}$;
(10) $body(r) := body(r) \setminus X$;
(11) **END_DO**;
(12) *Output* Φ_1.

Для полных БД в этом алгоритме следует строку (2) заменить на
(2’) Удалить из Φ_1 все предложения r такие, что $head(r) \in (\Delta^+ \cup \neg.\Delta^-)$;
а вместо оператора в строке (6) поместить строку
(6’) $body(r) := body(r) \setminus (\Delta^+ \cup \neg.\Delta^-)$
Таким образом, $res_\Delta(\Phi)$ состоит из части предложений Φ, из тел которых могут быть удалены некоторые литералы. Это утверждение можно уточнить следующим образом.

Лемма 5 *Пусть $\Phi_1 = res_\Delta(\Phi)$ вляется результатом работы алгоритма comp_res. Тогда Φ_1 состоит из всех таких предложений $l \leftarrow \alpha$, для каждого из которых существует правило $r \in \Phi$ такое, что*
1) $head(r) = l$ и $\alpha \subseteq body(r)$,
2) Δ не вынуждает l,
3) Δ не запрещает $body(r)$,

4) $body(r) \setminus \alpha = \{l \mid (\Delta \models l\} \cup \{l_1 \mid \exists l\ (l \in body(r) \& (l, c(l_1)) \in \Delta^*)\}$.

Хотя предыдущая лемма справедлива для обоих видов баз данных, результат вычисления res может быть разным для полных и частичных БД. .

Пример 5 *Рассмотрим следующие ДОЦ* Φ*, состоящие из 4-х предложений:*
$r_1:\ a' \leftarrow d, b', \neg c';\ \ r_2:\ \neg b' \leftarrow \neg a', d';\ \ r_3:\ c' \leftarrow \neg b', \neg d';\ \ r_4:\ \neg c' \leftarrow e, a', b', d'.$
Пусть обновление $\Delta = (< \{d\}, \{b\} >, < \emptyset, \{c\} >, \{(e, a)\})$.

Тогда, выполнив вариант алгоритма для частичных БД, мы получим ДОЦ $res(\Phi, \Delta)$, включающие три предложения:
$r'_1:\ a' \leftarrow \neg c';\ \ r'_2:\ \neg b' \leftarrow \neg a', d';\ \ r'_4:\ \neg c' \leftarrow e, d'$.

При выполнении алгоритма *comp_ res* для полных БД результат состоит из двух правил:
$r''_1:\ a' \leftarrow;\ \ r''_2:\ \neg b' \leftarrow \neg a', d'.$

Следующее утверждение показывает корректность преобразовани $res_\Delta(\Phi)$, его конфлюэнтность по отношению к обновлениям и эффективную вычислимость.

Лемма 6

(1) Для всех $\Phi \in \mathbf{IC}$ *и* $\Delta \in \mathbf{UP}$ $Tr(\Phi, \Delta) = Tr(res_\Delta(\Phi), \Delta)$.

(2) Предположим, что обновление Δ *является объединением двух обновлений* Δ_1 *и* Δ_2*, т.е.* $\Delta^+ = \Delta_1^+ \cup \Delta_2^+$, $\Delta^- = \Delta_1^- \cup \Delta_2^-$ *и* $\Delta^* = \Delta_1^* \cup \Delta_2^*$*. Тогда для любого* $\Phi \in \mathbf{IC}$ $\quad res_\Delta(\Phi) = res_{\Delta_2}(res_{\Delta_1}(\Phi))$.

(3) Предположим, что ДОЦ Φ *является объединеним двух ДОЦ* $\Phi = \Phi_1 \cup \Phi_2$*. Тогда для любого обновления* $\Delta \in \mathbf{UP}$

$res_\Delta(\Phi) = res_\Delta(\Phi_1) \cup res_\Delta(\Phi)$.

(4) $res(\Phi, \Delta) \preceq \Phi$ *и, следовательно,* $|res(\Phi, \Delta)| \leq |\Phi|$,
где $|\Phi|$ *— это размер* $\Phi \in \mathbf{IC}$ *(т.е. число литералов во всех предложениях* Φ*).*

(5) Алгоритм comp_ res вычисляет $res_\Delta(\Phi)$ *за полиномиальное (квадратичное) время от размера входа.*

Доказательство. (1) Покажем, что условие (Inv): $Tr(\Phi, \Delta) = Tr(\Phi_1, \Delta)$ является инвариантом алгоритма *comp_ res*. Оно, очевидно, выполнено после строки (1) алгоритма. Если Δ выполнено на переходе $< I, I' >$, то и любое предложение $r \in \Phi$ с $head(r) \in \Delta^+$ выполнено на $< I, I' >$. Поэтому удаление таких предложений не изменяет множества $Tr(\Phi_1, \Delta)$ и условие (Inv) справедливо после строки (2). Если предложение $r \in \Phi$ удаляется в строке (3), то существует литерал $l \in body(r) \cap (\neg.\Delta^+ \cup \Delta^-)$. Тогда по лемме 4 (2) $\Delta \not\models body(r)$ и, следовательно, предложение r выполнено на любом переходе из $UT(\Delta)$. Поэтому удаление такого правила также не изменяет множества $Tr(\Phi_1, \Delta)$ и условие (Inv) справедливо после строки (3) алгоритма. Если предложение $r \in \Phi$ удаляется в строке (4), то существует пара литералов $(l_1, l_2) \in \Delta^*$ такая, что $l_1 \in body(r)$ и либо $\neg.n(l_2) \in body(r)$, либо $n(l_1) \in body(r)$. Снова по лемме 4 (2) $\Delta \not\models body(r)$ и предложение r выполнено на любом переходе из $UT(\Delta)$. Поэтому его удаление не изменяет множества $Tr(\Phi_1, \Delta)$ и условие (Inv) справедливо после строки (4) алгоритма. В строках (5)–(7) из тела каждого оставшегося предложени r удаляются литералы из Δ^+. Поскольку для всякого прехода $< I, I' > \in UT(\Delta)$ имеют место включения $\delta^+ \subseteq I$ и $\delta'^+ \subseteq I'$, то $<I, I'> \models body(r) \Leftrightarrow <I, I'> \models body(r) \setminus (\delta^+ \cup n(\delta'^+)$. Поэтому условие (Inv) выполнено и после цикла (5)–(7) алгоритма. В строках (8)–(11) из тела каждого правила r, содержащего пару литералов $(l_1, n(l))$ такую, что $(l_1, l) \in \Delta^*$, удаляется литерал $n(l)$. Заметим, что для всякого прехода $<I, I'> \in UT(\Delta)$, если $l_1 \in I$, то обязательно $l \in I'$. Поэтому $<I, I'> \models body(r) \Leftrightarrow <I, I'> \models body(r) \setminus \{n(l)\}$. Поэтому после каждого такого удаления множество $Tr(\Phi_1, \Delta)$ не изменяется и условие (Inv) выполнено после строки (11) и, следовательно, выполнено и после завершения алгоритма.

Утверждения в пунктах (2), (3) и (4) следуют непосредственно из описания алгоритма.

Пусть $N = |\Delta| + |\Phi|$. Нетрудно понять, что циклы в строках (2),(3) и (5)–(7) можно реализовать за время $O(N)$, а циклы в строках (4) и (8)–(11) можно выполнить за время $O(|\Delta^*|N)$. Отсюда следует оценка времени из пункта (5). □

Как показывает следующий пример, преобразование $res_\Delta(\Phi)$ не обеспечивает максимального упрощения ДОЦ.

Пример 6 *Рассмотрим ДОЦ* Φ*, включающие следующие пять предложений:*
$r_1 : \ a' \leftarrow b, c'; \ \ r_2 : \ d' \leftarrow c'; \ \ r_3 : \ e' \leftarrow b, d'; \ \ r_4 : \ d' \leftarrow b, c'; \ \ r_5 : \ b' \leftarrow b, c', d'.$
Пусть обновление Δ *требует удалить факт* e*, т.е.* $\Delta^- = \{e\}$*.*

Тогда очевидно, что $res_\Delta(\Phi) = \Phi$, т.е. преобразование res не упрощает данные Φ относительно Δ. В то же время легко проверить, что Φ эквивалентнтны относительно Δ гораздо более простым ДОЦ $\Phi' = \{r_2 : \ d' \leftarrow c'; \ \ r_3 : \ e' \leftarrow b, d'; \ \ r_5' : \ b' \leftarrow b, c'\}$. Действительно, r_1 можно удалить, так как, если бы его тело выполнялось на некотором переходе $<I, I'> \in Tr(\Phi, \Delta)$, то по правилам r_2 и r_3 литерал e должен был бы входить в I', но это противоречит обновлению Δ. Предложение r_4 можно удалить, так как оно является частным случаем r_2. Из тела r_5 можно устранить d', так как по правилу r_2 истинность c' на любом переходе, удовлетворяющем Φ, гарантирует и истинность d'.

Чтобы выявить возможность упрощения ДОЦ после res, мы введем сейчас понятие независимого предложения.

Определение 11 *Назовем предложение r* следствием ДОЦ $\Phi \in \mathbf{IC}$, *если всякий переход $< I, I' >$, являющийся моделью Φ (т.е. $< I, I' >\models \Phi$) является также моделью r, т.е. $< I, I' >\models r$. r является* сильным следствием Φ, *если $head(r) \in M_\Phi^{min}(body(r))$. Предложение r является* (слабо) независимым от Φ, *если оно не является (строгим) следствием Φ.*

Непосредственно из этого определения следует

Лемма 7
(1) Для частичных БД r является следствием Φ тогда и только тогда, когда r является сильным следствием Φ.
(2) Для полных БД, если r является сильным следствием Φ, то r является следствием Φ.

Для полных БД отношение "быть сильным следствием" только аппроксимирует отношение "быть следствием". Следующий простой пример показывает, что обращение пункта (2) не имеет места.

Пример 7 *Для полных БД предложение $a \leftarrow$ является следствием ДОЦ $\Phi = \{a \leftarrow b';\ a \leftarrow \neg b'\}$, но не является сильным следствием, так как $a \notin M_\Phi^{min}(<\emptyset, \emptyset>) = \emptyset$ (напомним, что в определении оператора сильной выводимости $T_\Phi^\in$ для применимости некоторого правила требуется, чтобы каждый литерал из его тела содержался (явно!) в исходном переходе).*

Определим теперь подкласс *аккуратных* ДОЦ, в которых нет "лишних" предложений и литералов.

Определение 12 *Пусть $\Phi \in \mathbf{IC}$ и $\Delta \in \mathbf{GUP}$. ДОЦ Φ назовем* аккуратными по отношению к Δ, *если для них выполнены следующие три условия:*
(a) для каждого $r \in \Phi$ $M_{\Phi\setminus\{r\}}^{min}(body(r))$ согласовано с Δ;
(b) каждое предложение $r \in \Phi$ является слабо независимым от $\Phi \setminus \{r\}$;
(c) для каждого предложения $r \in \Phi$ и любого подмножества $\alpha \subset body(r)$ имеет место неравенство $body(r) \setminus M_\Phi^{min}(\alpha) \neq \emptyset$, т.е. в телах предложений нет функциональных зависимостей.

Следующее утверждение показывает, что всякие ДОЦ можно эффективно преобразовать в эквивалентные ДОЦ, которые аккуратны по отношению к заданному обновлению.

Лемма 8 *Существует преобразование $tidy\_IC : \mathbf{IC} \times \mathbf{GUP} \rightarrow \mathbf{IC}$ такое, что*
(1) $tidy\_IC(\Phi, \Delta)$ являются аккуратными ДОЦ по отношению к Δ;
(2) $Tr(tidy\_IC(\Phi, \Delta), \Delta) = Tr(\Phi, \Delta)$;
(3) преобразование $tidy\_IC(\Phi, \Delta)$ вычислимо за квадратичное время.

Доказательство. (i) Мы начнем с алгоритма, вычисляющего указанное в лемме преобразование ДОЦ.

Алгоритм tidy_IC

Вход: $\Phi \in \mathbf{IC}$, $\Delta \in \mathbf{GUP}$;

Выход: Φ_1.

(1) $\Phi_1 := \Phi$;
(2) **FOR EACH** предложения $r \in \Phi_1$ **DO**
(3) **IF** $M^{min}_{\Phi\setminus\{r\}}(body(r))$ не согласовано с Δ
(4) **THEN** $\Phi_1 := \Phi_1 \setminus \{r\}$ **END_IF**
(5) **END_DO**;
(6) **FOR EACH** предложения $r \in \Phi_1$ **DO**
(7) **IF** $head(r) \in M^{min}_{\Phi_1\setminus\{r\}}(body(r))$
(8) **THEN** $\Phi_1 := \Phi_1 \setminus \{r\}$ **END_IF**
(9) **END_DO**;
(10) **FOR EACH** предложения $r \in \Phi_1$ **DO**
(11) **FOR EACH** литерала $l \in body(r)$ **DO**
(12) **IF** $l \in M^{min}_{\Phi_1}(body(r) \setminus \{l\})$
(13) **THEN** $body(r) := body(r) \setminus \{l\}$ **END_IF**
(14) **END_DO END_DO**;
(15) **Output** Φ_1.

Нетрудно заметить, что этот алгоритм в точности следует определению 12: первый цикл в строках (2) - (5) обеспечивает выполнение условия (a), второй цикл в строках (6) - (9) удаляет предложения, являющиеся слабыми следствиями остальных предложений ДОЦ, что обеспечивает выполнение условия (b), а третий цикл в строках (10) - (14) последовательно удаляет из тел предложений все зависимые литералы, чем гарантирует выполнение условия (c). Таким образом, результирующие ДОЦ Φ_1 будут аккуратными по отношению к Δ, что доказывает пункт (1) леммы.

Для доказательства (2) мы покажем, что условие $Tr(\Phi_1, \Delta) = Tr(\Phi, \Delta)$ является инвариантом всех трех циклов алгоритма *tidy_IC*.

Утверждение 1. Если $r \in \Phi$ и $M^{min}_{\Phi\setminus\{r\}}(body(r))$ не согласовано с Δ, то $Tr(\Phi, \Delta) = Tr(\Phi \setminus \{r\}, \Delta)$.

Включение $Tr(\Phi, \Delta) \subseteq Tr(\Phi \setminus \{r\}, \Delta)$ очевидно. Пусть теперь $< I, I' >$ — произвольный переход из $Tr(\Phi \setminus \{r\}, \Delta)$. Предположим, что $< I, I' >\models body(r)$. Тогда по лемме 3(2) $M^{min}_{\Phi\setminus\{r\}}(body(r)) \subseteq< I, I' >$. Отсюда следует, что переход $< I, I' >$ не согласован с Δ, что противоречит выбору $< I, I' >$. Итак, наше предположение неверно и $< I, I' >\not\models body(r)$. Но тогда $< I, I' >\models r$ и $< I, I' >\models \Phi$. Отсюда получаем, что, $< I, I' >\in Tr(\Phi, \Delta)$.

Утверждение 2. Если $r \in \Phi$ и $head(r) \in M^{min}_{\Phi \setminus \{r\}}(body(r))$, то $Tr(\Phi, \Delta) = Tr(\Phi \setminus \{r\}, \Delta)$.

Как и в предыдущем случае, включение $Tr(\Phi, \Delta) \subseteq Tr(\Phi \setminus \{r\}, \Delta)$ тривиально. Пусть теперь $<I, I'>$ — произвольный переход из $Tr(\Phi \setminus \{r\}, \Delta)$. Предположим, что $< I, I' >\models body(r)$. Тогда по лемме 3(2) $M^{min}_{\Phi \setminus \{r\}}(body(r)) \subseteq < I, I' >$. Отсюда получаем, что $< I, I' >\models head(r)$. Но тогда $< I, I' >\models r$ и $< I, I' >\models \Phi$. Следовательно, $<I, I'>\in Tr(\Phi, \Delta)$.

Утверждение 3. Если $r \in \Phi$, $l \in body(r)$ и $l \in M^{min}_{\Phi}(body(r) \setminus \{l\})$ то $Tr(\Phi, \Delta) = Tr((\Phi \setminus \{r\}) \cup \{r'\}, \Delta)$ где $r' = (head(r) \leftarrow (body(r) \setminus \{l\}))$.

В этом случае включение $Tr((\Phi \setminus \{r\}) \cup \{r'\}, \Delta) \subseteq Tr(\Phi, \Delta)$ очевидно. Пусть теперь $<I, I'>$ — это произвольный переход из $Tr(\Phi, \Delta)$. Предположим, что $<I, I'>\models body(r')$. Тогда по лемме 3(2) $M^{min}_{\Phi}(body(r')) \subseteq <I, I'>$. Но тогда $<I, I'>\models l$ и, следовательно, $<I, I'>\models body(r)$. Отсюда получаем, что и $<I, I'>\models head(r)$, поскольку $<I, I'>\models r$. Но тогда $<I, I'>\models r'$ и $<I, I'>\in Tr(\Phi \setminus \{r\}) \cup \{r'\}, \Delta)$.

(3) Легко видеть, что время исполнения каждого из трех циклов алгоритма *tidy_IC* можно ограничить величиной $O((|\Phi| + |\Delta|)^2)$, что дает требуемую общую оценку. □

Нетрудно проверить, что по ДОЦ Φ и обновлению Δ из предыдущего примера 7 алгоритм *tidy_IC* построит указанные там упрощенные ДОЦ Φ'.

5.3 Прямой и обратный операторы расширения обновлений

В этом параграфе мы определим два конкретных оператора расширени обновлений: прямой и обратный. Идея прямого оператора состоит в расширении множества “добавляемых” фактов за счет голов правил, тела которых вынуждаются обновлением. Обратный оператор расширяет множество “удаляемых” фактов за счет литералов, добавление которых к результату обновления приводит к запрещенным литералам (например, когда добавление такого литерала делает истинным тело правила, голова которого является удаляемым фактом).Кроме того, каждый из операторов завершается преобразованием C, определенным в предложении 8, которое устраняет лишние пары из Δ^*.

Частичные базы данных

Прямой оператор F. Пусть $\Phi \in \mathbf{IC}$ и $\Delta \in \mathbf{GUP}$. Тогда $F(\Phi, \Delta) = C(\Delta_1)$, где

$\delta_1^+ = \delta^+ \cup \{l \in \mathbf{LB}^c | \ \exists r \in \Phi \ (l = head(r) \ \& \ \Delta \models body(r))\}$;
$\delta_1'^+ = \delta'^+ \cup \{l \in \mathbf{LB}^c | \ \exists r \in \Phi \ (n(l) = head(r) \ \& \ \Delta \models body(r))\}$;
$\delta_1^- = \delta^- \cup \{\neg.l | \ l \in \delta^+\}$;

$\delta_1'^- \ = \delta'^- \cup \{\neg.l | \ l \in \delta'^+\}$;
$\Delta_1^* \ = \Delta^*$.

Обратный оператор B. Пусть на $\Phi \in \mathbf{IC}$ и $\Delta \in \mathbf{GUP}$. Тогда $B(\Phi, \Delta) = C(\Delta_1)$, где

$\Delta_1^+ \ = \Delta^+$;
$\delta_1^- \ = \delta^- \ \cup \ \{l \in \mathbf{LB}^c | \ M_\Phi^{min}(< \delta^+ \cup \{l\}, \delta'^+ >)) \ \textit{не совместим с} \ \Delta\}$;
$\delta_1'^- \ = \delta'^- \ \cup \ \{l \in \mathbf{LB}^c | \ M_\Phi^{min}(< \delta^+, \delta'^+ \cup \{l\} >)) \ \textit{не совместим с} \ \Delta\}$;
$\Delta_1^* \ = \Delta^*$.

Для полных баз данных определим аналогичные операторы следующим образом.

Полные БД.
Прямой оператор F. Пусть $\Phi \in \mathbf{IC}$ и $\Delta \in \mathbf{GUP}$. Тогда $F(\Phi, \Delta) = C(\Delta_1)$, где

$\delta_1^+ \ = \delta^+ \ \cup \ \{a \in \mathbf{B}^c \mid \exists r \in \Phi \ (a = head(r) \ \& \ \Delta \models body(r))\}$;
$\delta_1'^+ \ = \delta'^+ \ \cup \ \{a \in \mathbf{B}^c \mid \exists r \in \Phi \ (n(a) = head(r) \ \& \ \Delta \models body(r))\}$;
$\delta_1^- \ = \delta^- \cup \ \{a \in \mathbf{B}^c \mid \exists r \in \Phi \ (\neg a = head(r) \ \& \ \Delta \models body(r))\}$;
$\delta_1'^- \ = \delta'^- \cup \ \{a \in \mathbf{B}^c \mid \exists r \in \Phi \ (\neg n(a) = head(r) \ \& \ \Delta \models body(r))\}$;
$\Delta_1^* \ = \Delta^*$.

Обратный оператор B. Пусть на $\Phi \in \mathbf{IC}$ и $\Delta \in \mathbf{GUP}$. Тогда $B(\Phi, \Delta) = C(\Delta_1)$, где

$\delta_1^+ \ = \delta^+ \cup \{a \in \mathbf{B}^c \mid M_\Phi^{min}(< (\delta^+ \cup \neg.\delta^- \cup \{\neg a\}), (\delta'^+ \cup \neg.\delta'^-) >) \ \textit{не совместим с} \ \Delta\}$;
$\delta_1'^+ \ = \delta'^+ \cup \{a \in \mathbf{B}^c \mid M_\Phi^{min}(< (\delta^+ \cup \neg.\delta^-), (\delta'^+ \cup \neg.\delta'^- \cup \{\neg a\}) >) \ \textit{не совместим с} \ \Delta\}$;
$\delta_1^- \ = \delta^- \cup \{a \in \mathbf{B}^c \mid M_\Phi^{min}(< (\delta^+ \cup \neg.\delta^- \cup \{a\}), (\delta'^+ \cup \neg.\delta'^-) >) \ \textit{не совместим с} \ \Delta\}$;
$\delta_1'^- \ = \delta'^- \cup \{a \in \mathbf{B}^c \mid M_\Phi^{min}(< (\delta^+ \cup \neg.\delta^-), (\delta'^+ \cup \neg.\delta'^- \cup \{a\}) >) \ \textit{не совместим с} \ \Delta\}$;
$\Delta_1^* \ = \Delta^*$.

Рассмотрим небольшой пример применения этих операторов.

Пример 8 *Пусть* $\Phi = \{f' \leftarrow e', g; \ b \leftarrow a', e'; \ c' \leftarrow a', \neg d'; \ d' \leftarrow b, c'; \ h \leftarrow g, \neg d'\}$ *а обновление* Δ *имеет следующие компоненты:* $\Delta^+ =< \{g\}, \{e\} >$, $\Delta^- =< \emptyset, \{d\} >$ *и* $\Delta^* = \emptyset$.

Тогда для частичных БД, применяя прямой оператор, получаем $F(\Phi, \Delta)^+ =< \{g\}$, $\{e, f\} >$ и $F(\Phi, \Delta)^- =< \{\neg g\}, \{d, \neg e, \neg f\} >$. Применение обратного опероатора не изменяет обновления Δ. В случае же полных БД получаем для прямого оператора

$F(\Phi,\Delta)^+ =< \{g,h\},\{e,f\} >$ и $F(\Phi,\Delta)^- = \Delta^-$, а для обратного $B(\Phi,\Delta)^+ = \Delta^+$ и $B(\Phi,\Delta)^- =< \emptyset,\{a,d\} >$.

Из определений следует, что оба оператора F и B монотонны на **GUP** по первым двум координатам и антимонотонны по третьей. Они также не меняют множества переходов, на которых выполнено обновление и ДОЦ.

Лемма 9 *Для обоих видов БД и любых пар* $\Phi \in \mathbf{IC}$, $\Delta \in \mathbf{GUP}$ *выполнены следующие равенства:*
(1) $Tr(\Phi,\Delta) = Tr(\Phi, F(\Phi,\Delta))$,
(2) $Tr(\Phi,\Delta) = Tr(\Phi, B(\Phi,\Delta))$.

Доказательство. Поскольку корректность преобразования C следует из предложения 8, то для доказательства леммы нам, фактически, требуется установить четыре равенства $Tr(\Phi,\Delta) = Tr(\Phi,\Delta_1)$ для обновлений Δ_1 из определений операторов F и B. Заметим вначале, что во всех четырех случаях включения вида $Tr(\Phi,\Delta_1) \subseteq Tr(\Phi,\Delta)$ следуют непосредственно из свойства монотонности: $\Delta \sqsubseteq \Delta_1$. Справедливость обратных включений $Tr(\Phi,\Delta) \subseteq Tr(\Phi,\Delta_1)$ для прямых операторов F также достаточно очевидна. Действительно, если $r \in \Phi$ и $\Delta \models body(r)$, то для каждого перехода $< I, I' > \in Tr(\Phi,\Delta)$ имеет место $< I, I' > \models body(r)$ и, следовательно, $< I, I' > \models head(r)$. Таким образом, добавление $head(r)$ к соответствующей компонете Δ^+ не уменьшает $Tr(\Phi,\Delta)$. Для полных БД добавление к Δ^- атомов, отрицания которых вынуждаются Δ, также не уменьшают этот класс переходов, так как ни один из этих атомов не может содержаться в переходе из $Tr(\Phi,\Delta)$.

Рассмотрим теперь обратный оператор B для частичных БД. Предположим, что некоторый переход $< I, I' > \in (TR(\Phi,\Delta) \setminus Tr(\Phi,\Delta_1))$. Это означает, что $< I, I' >$ содержит некоторый литерал $l \in (\Delta_1^- \setminus \Delta^-)$. Тогда из определения Δ_1^- можно заключить, что добавление l в соответствующую компоненту Δ^+ приводит к тому, что переход $M_\Phi^{min}(\Delta^+ \cup \{l\})$ становится не совместимым с Δ. Но так как $l \in < I, I' >$, то $(\Delta^+ \cup \{l\}) \subseteq < I, I' >$ и $M_\Phi^{min}(\Delta^+ \cup \{l\}) \subseteq < I, I' >$ и переход $< I, I' >$ не совместим с Δ, что противоречит выбору этого перехода.

Рассуждения для случая полных БД, в целом, аналогичны. Нужно только заметить, что если для некоторого атома a переход $M_\Phi^{min}(< (\delta^+ \cup \neg.\delta^-), (\delta'^+ \cup \neg.\delta'^- \cup \{\neg a\}) >)$ или $M_\Phi^{min}(< (\delta^+ \cup \neg.\delta^- \cup \{\neg a\}), (\delta'^+ \cup \neg.\delta'^-) >)$ не совместим с Δ, то a обязательно входит в соответствующую компоненту любого перехода из $Tr(\Phi,\Delta)$ и, следовательно, включение a в Δ^+ не уменьшает этот класс переходов. $\square$

Теперь мы, используя преобразования ДОЦ res и операторы F и B на обновлениях, определим прямое и обратное расширения обновлений. Эти определения одинаковы для обоих видов баз данных.

Определим вначале оператор γ_f прямого расширения за один шаг:

$\gamma_f(\Phi,\Delta) \ = \ (res(\Phi,\Delta), F(res(\Phi,\Delta),\Delta))$.

Пусть $\gamma_f^0(\Phi,\Delta) = (\Phi,\Delta)$ и, как обычно, положим для $n \geq 0$ $\gamma_f^{n+1}(\Phi,\Delta) \ = \gamma_f(\gamma_f^n(\Phi,\Delta))$.

Оператор прямого расширения Γ_f определим как предел:

$\Gamma_f(\Phi,\Delta) = \lim\limits_{n\to\infty} \gamma_f^n(\Phi,\Delta)$.

Аналогично, вначале определим оператор γ_b обратного расширения за один шаг:

$\gamma_b(\Phi,\Delta) \ = \ (res(\Phi,\Delta), B(res(\Phi,\Delta),\Delta))$

и его степени: $\gamma_b^0(\Phi,\Delta) = (\Phi,\Delta)$ и $\gamma_b^{n+1}(\Phi,\Delta) \ = \gamma_b(\gamma_b^n(\Phi,\Delta))$.

Оператор обратного расширения Γ_b определим как предел:

$\Gamma_b(\Phi,\Delta) = \lim\limits_{n\to\infty} \gamma_b^n(\Phi,\Delta)$.

Существование пределов в этих определениях обеспечивается свойствами использованных в них исходных операторов F, B и res.

Лемма 10 *Для обоих видов БД — частичных и полных — и для каждой пары* $\Phi \in \mathbf{IC}$ *и* $\Delta \in \mathbf{GUP}$ *справедливы следующие утверждения:*

(1) существует такое $m \geq 0$*, что* $\Gamma_f(\Phi,\Delta) = \gamma_f^m(\Phi,\Delta)$*;*

(2) существует такое $m \geq 0$*, что* $\Gamma_b(\Phi,\Delta) = \gamma_b^m(\Phi,\Delta)$;

(3) выполнены равенства:

$$\Gamma_f(\Phi,\Delta)^{ic} = res(\Phi, \Gamma_f(\Phi,\Delta)^{up}) \quad \text{и} \quad \Gamma_b(\Phi,\Delta)^{ic} = res(\Phi, \Gamma_b(\Phi,\Delta)^{up}).$$

Доказательство. Чтобы установить (1), рассмотрим для каждой пары (Φ,Δ) число $N(\Phi,\Delta) = |\Phi| + |\mathbf{LB}^n| - |\delta'^{+}| - |\delta'^{-}|$. Ясно, что для рассматриваемых нами обновлений $N \geq 0$. Из леммы 6(4) и из определения γ_f следует, что для каждого n, если $\gamma_f^{n+1}(\Phi,\Delta) \neq \gamma_f^n(\Phi,\Delta)$, то $N(\gamma_f^{n+1}(\Phi,\Delta)) < N(\gamma_f^n(\Phi,\Delta))$. Тогда для некоторого $m \leq N(\Phi,\Delta)$ обязательно выполнится равенство $N(\gamma_f^{m+1}(\Phi,\Delta)) = N(\gamma_f^m(\Phi,\Delta))$ и для этого m $\Gamma_f(\Phi,\Delta) = \gamma_f^m(\Phi,\Delta)$. Аналогичное рассуждение устанавливает справедливость (2).

Пункт (3) следует из пунктов (1) и (2). Рассмотрим, например, второе из его равенств. Из (2) следует, что для некоторого $m \geq 1$ $\Gamma_b(\Phi,\Delta)^{ic} = \gamma_b^m(\Phi,\delta)^{ic}$. Обозначим для каждого $i \geq 0$ через $(\Phi^{(i)},\Delta^{(i)})$ пару $\gamma_b^i(\Phi,\Delta)$. Тогда нам нужно доказать, что

$$\Phi^{(m)} = res(\Phi,\Delta^{(m)}). \qquad (*)$$

Докажем вначале индукцией по $i \geq 1$ равенство

$$\Phi^{(i)} = res(\Phi,\Delta^{(i-1)}). \qquad (**)$$

Базис. При $i = 1$ имеем $\Phi^{(1)} = res(\Phi,\Delta) = res(\Phi,\Delta^{(0)})$.

Индукционный шаг. Предположим, что для некоторого $i \geq 1$ $\Phi^{(i)} = res(\Phi,\Delta^{(i-1)})$. Тогда из определения γ_b^{i+1}, индукционного предположени и леммы 6(2) можно вывести следующие равенства:

$\Phi^{(i+1)} = res(\Phi^{(i)},\Delta^{(i)}) = res(res(\Phi,\Delta^{(i-1)}),\Delta^{(i)}) = res(\Phi,\Delta^{(i)})$.

Следовательно, (**) выполнено для любого $i \geq 1$. Чтобы вывести (*), заметим, что

$\gamma_b^m(\Phi, \Delta) = \gamma_b^{m+1}(\Phi, \Delta)$ и, следовательно, $\Phi^{(m)} = \Phi^{(m+1)}$. Теперь, применяя (**) для $i = m+1$, мы получаем, что $\Phi^{(m)} = res(\Phi, \Delta^{(m)})$. □

Для частичных БД действие оператора Γ_f на обновлениях очень близоко к замыканию $T^{\in}_{\Phi(\Delta)}$.

Лемма 11 *Пусть* $\Phi \in \mathbf{IC}$ *и* $\Delta \in \mathbf{GUP}$ *— совместные ДОЦ и обновление и пусть* $\Gamma_f(\Phi, \Delta) = (\Phi_1, \Delta_1)$, *а* $M^{min}_{\Phi(\Delta)} = (M_1, M_2)$. *Тогда имеют место следующие равенства:*

(1) $< \delta_1^+, \delta_1^{'+} >=< M_1, M_2 >$,

(2) $< \delta_1^-, \delta_1^{'-} >=< \neg.M_1, \delta^{'-} \cup \neg.M_2 >$.

Доказательство. (1) Поскольку Φ и Δ совместны, то из предложения 4 (1) следует, что переход $M^{min}_{\Phi(\Delta)}$ совместен. Пусть для $i \geq 0$ обновление $\gamma_f^i(\Phi, \Delta)^{up} =< < \delta_{i1}^+, \delta_{i1}^{'+} >, < \delta_{i1}^-, \delta_{i1}^{'-} >, \Delta_i^* >$. Тогда утверждение (1) следует из равенств $(T^{\in}_{\Phi(\Delta)})^i =< \delta_{i1}^+, \delta_{i1}^{'+} >$, которые можно доказать для всех $i \geq 0$ стандартными рассуждениями по индукции.

Пункт (2) непосредственно следует из (1). □

Покажем теперь, что определенные выше операторы Γ_f и Γ_b удовлетворяют условиям определения 9.

Теорема 10

Для обоих видов баз данных операторы Γ_f *и* Γ_b *являются операторами расширения обновлений.*

Доказательство. Приведем доказательства для полных БД. Доказательства дл частичных БД аналогичны.

Из определений операторов γ_f и γ_b следует, что они монотонны на обновлениях. После этого индукцией по n легко показать, что для каждого n γ_f^n и γ_b^n также монотонны на обновлениях. Тогда по лемме 10 (1), (2) операторы Γ_f и Γ_b также монотонны на обновлениях. Из лемм 10 (3) и 6 (4) следует, что $\Gamma_f(\Phi, \Delta)^{ic} = res(\Phi, \Gamma_f(\Phi, \Delta)^{up}) \preceq \Phi$ и
$\Gamma_b(\Phi, \Delta)^{ic} = res(\Phi, \Gamma_b(\Phi, \Delta)^{up}) \preceq \Phi$. Следовательно, $(\Phi, \Delta) \preceq \Gamma_f(\Phi, \Delta)$ и $(\Phi, \Delta) \preceq \Gamma_b(\Phi, \Delta)$.

Чтобы доказать, что Γ_f не изменяет множество переходов мы установим равенство $Tr(\Phi, \Delta) = Tr(\gamma_f(\Phi, \Delta))$. Из лемм 6 (1), 9 и определения γ_f следует, что:
$Tr(\Phi, \Delta) = Tr(res(\Phi, \Delta), \Delta) = Tr(res(\Phi, \Delta), F(res(\Phi, \Delta), \Delta) = Tr(\gamma_f(\Phi, \Delta))$.
Тогда стандартными рассуждениями по индукции можно установить, что
$Tr(\Phi, \Delta) = Tr(\Gamma_f(\Phi, \Delta))$. Доказательство аналогичного равенства $Tr(\Phi, \Delta) = Tr(\Gamma_b(\Phi, \Delta))$ для оператора B можно провести так же, как и для F. Таким образом, Γ_f и Γ_b являются операторами расширения обновлений.□

Приведем алгоритмы, вычисляющие операторы Γ_f и Γ_b для частичных БД.

Алгоритм FP_expand

Вход: обновление $\Delta \in \mathbf{GUP}$, совместное с ДОЦ $\Phi \in \mathbf{IC}$.

Локальные переменные: Φ_1, Δ_1, Δ^+_{new} и их компоненты.

Выход: (Φ_1, Δ_1).

(1) $\Phi_1 := \Phi;$
(2) $\Delta^-_1 :=< \delta^- \cup \neg.\delta^+, \delta'^- \cup \neg.\delta'^+ >;$
(3) $\Delta_1 := (\Delta^+, \Delta^-_1, \Delta^*);$
(4) $\Delta_{new} := \Delta_1;$
(5) **LOOP**
(6) $\quad\Phi_1 := comp\_res(\Phi_1, \Delta_{new});$
(7) $\quad\delta^+_{new} := \{l \in \mathbf{LB}^c |\ \exists r \in \Phi_1\ (l = head(r)\ \&\ \Delta_1 \models body(r))\};$
(8) $\quad\delta'^+_{new} := \{l \in \mathbf{LB}^c |\ \exists r \in \Phi_1\ (n(l) = head(r)\ \&\ \Delta_1 \models body(r))\};$
(9) $\quad\Delta^-_{new} :=< \neg.\delta^+_{new}, \neg.\delta'^+_{new} >;$
(10) $\quad\Delta_{new} := (< \delta^+_{new}, \delta'^+_{new} >, \Delta^-_{new}, \emptyset);$
(11) $\quad\Delta^+_1 :=< \delta^+_1 \cup \delta^+_{new}, \delta'^+_1 \cup \delta'^+_{new} >;$
(12) $\quad\Delta^-_1 :=< \delta^-_1 \cup \delta^-_{new}, \delta'^-_1 \cup \delta'^-_{new} >;$
(13) $\quad\Delta_1 := (\Delta^+_1, \Delta^-_1, \Delta^*_1);$
(14) $\quad\Delta_1 := C(\Delta_1);$
(15) **UNTIL** $(< \delta^+_{new}, \delta'^+_{new} >=< \emptyset, \emptyset >);$
(16) *Output* (Φ_1, Δ_1).

Сделаем несколько замечаний к алгоритму *FP_expand.* Во-первых, отметим, что результат упрощения ДОЦ в строке (6) совпадает с $comp\_res(\Phi_1, \Delta_1)$. Действительно, при первом исполнении это следует из того, что $\Delta_{new} = \Delta_1$ (см. строку (4)). А перед каждым из следующих исполнений выполнено условие $\Delta_1 = \Delta'_1 \cup \Delta_{new}$, где Δ_1 — новое значение обновления, а Δ'_1 его значение на предыдущей итерации цикла . Тогда $comp\_res(\Phi_1, \Delta_1) = comp\_res(\Phi_1, \Delta_{new})$ по лемме 6 (2). Таким образом, в строках тела цикла (6)–(14) вычисляется оператор $\gamma_f(\Phi, \Delta) = (res(\Phi, \Delta), F(res(\Phi, \Delta), \Delta))$, а с учетом условия выхода из цикла получаем, что результат алгоритма $(\Phi_1, \Delta_1) = \Gamma_f(\Phi, \Delta)$. Второе замечание связано со сложностью алгоритма. Обозначим через N размер входных данных. Тогда количество литералов в каждом из участвующих в алгоритме множеств $\leq N$. Вычисление $comp\_res(\Phi_1, \Delta_{new})$ в строке (2) во всех случаях, кроме первого, происходит при $\Delta^*_{new} = \emptyset$ и поэтому может быть проведено за линейное время. Тогда суммарное время необходимое для выполнения этой строки не превосходит $O(N^2)$. Каждый литерал может появиться в множествах δ^+_{new} и δ'^+_{new} в строках (7) и (8) не более одного раза, поэтому общее время, необходимое для их выполнения (при подходящих списковых структурах данных) не превосходит $O(N)$.

Каждое из из изменений, производимых оператором $C(\Delta_1)$ в строке (14) приводит к уменьшению количества замен в Δ_1^*. Поэтому общее время на выполнение этой строки не превосходит $O(N^2)$. Остальные строки связаны со стандартными операциями над множествами и также могут быть выполнены за время $O(N)$ каждая. Таким образом, мы доказали

Предложение 11 *Алгоритм FP_expand вычисляет оператор* Γ_f *за квадратичное время.*

Следующая программа вычисляет обратный оператор Γ_b для частичных баз данных.

Алгоритм BP_expand .
Вход: обновление $\Delta \in \mathbf{GUP}$, совместное с ДОЦ $\Phi \in \mathbf{IC}$.
Локальные переменные: Φ_1, Δ_1, Δ^+_{new} и их компоненты.
Выход: (Φ_1, Δ_1).

(1) $\Phi_1 := \Phi$;
(2) $\Delta_1 := \Delta$;
(3) $\Delta_{new} := \Delta_1$;
(4) **LOOP**
(5) $\Phi_1 := comp\_res(\Phi_1, \Delta_{new})$;
(6) $< \delta^-_{new}, \delta'^-_{new} :=< \emptyset, \emptyset >$;
(7) **FOR ALL** $l \in \mathbf{LB}^c \setminus (\delta^+ \cup \delta^-)$ **DO**
(8) **IF** переход $M^{min}_\Phi(< \delta^+ \cup \{l\}, \delta'^+ >)$) *не совместим с* $\Delta\}$;
(9) **THEN** $\delta^-_{new} := \delta^-_{new} \cup \{l\}$
(10) **END_IF**
(11) **END_DO**;
(12) **FOR ALL** $l \in \mathbf{LB}^c \setminus (\delta'^+ \cup \delta'^-)$ **DO**
(13) **IF** переход $M^{min}_\Phi(< \delta^+, \delta'^+ \cup \{l\} >)$) *не совместим с* $\Delta\}$;
(14) **THEN** $\delta'^-_{new} := \delta'^-_{new} \cup \{l\}$
(15) **END_IF**
(16) **END_DO**;
(17) $\Delta^-_1 :=< \delta^-_1 \cup \delta^-_{new}, \delta'^-_1 \cup \delta'^-_{new} >$;
(18) $\Delta_1 := (\Delta^+_1, \Delta^-_1, \Delta^*_1)$
(19) $\Delta_1 := C(\Delta_1)$;
(20) **UNTIL** $(< \delta^-_{new}, \delta'^-_{new} >=< \emptyset, \emptyset >)$;
(21) *Output* (Φ_1, Δ_1).

Как и в предыдущем случае, нетрудно показать, что алгоритм *BP_expand* реализует обратное преобразование. На каждой итерации основного цикла в строках

(4)–(20) увеличивается одно из множеств δ_1^- или $\delta_1'^-$. Поэтому число таких итераций $\leq N$. Каждый из внутренних циклов (7)–(11) и (12)–(16) исполняется не более N раз, а каждое их исполнение, требующее вычисления одного замыкания, можно выполнить за время $O(N)$. Как и в алгоритме *FP_ expand* общее время выполнения операторов в строках (5) и (19) не превосходит $O(N^2)$, а в строках (6), (17) и (18) — $O(N)$. Таким образом, время выполнения одной итерации основного цикла не превосходит $O(N^2)$. Отсюда выводим справедливость следующего утверждения.

Предложение 12 *Алгоритм BP_ expand вычисляет оператор* Γ_b *для частичных БД за время* $O(N^3)$.

Отметим, что используя более сложные списковые структуры данных, можно реализовать вычисление Γ_b за время $O(N^2)$.

Для полных БД в алгоритмы вычисления Γ_f и Γ_b требуется внести некоторые изменения. В частности, алгоритм, *FT_ expand*, вычисляющий Γ_f для полных БД, получается из алгоритма *FP_ expand* заменой литерала l в строках (7) и (8) на атом a и, соответственно,$\mathbf{LB}^c$ — на $\mathbf{B}^c$, а также строки (9) на следующие три строки:

(9) $\delta^-_{new} := \{a \in \mathbf{B}^c |\ \exists r \in \Phi_1\ (\neg a = head(r)\ \&\ \Delta_1 \models body(r))\}$;

(9') $\delta'^-_{new} := \{a \in \mathbf{B}^c |\ \exists r \in \Phi_1\ (\neg n(a) = head(r)\ \&\ \Delta_1 \models body(r))\}$;

(9") $\Delta^-_{new} :=< \delta^-_{new}, \delta'^-_{new} >$;

Кроме того, нужно уточнить условие выхода из цикла в строке (15):

(15) **UNTIL** $(< \delta^+_{new}, \delta'^+_{new} >=< \emptyset, \emptyset >)$ **AND** $(< \delta^-_{new}, \delta'^-_{new} >=< \emptyset, \emptyset >)$;

Алгоритм *BT_ expand*, вычисляющий обратный оператор Γ_b для полных баз данных, получается из алгоритма *BP_ expand* заменой литерала l в строках (7) и (12) на атом a и, соответственно, $\mathbf{LB}^c$ — на $\mathbf{B}^c$, а также добавлением двух внутренних циклов после строки (17) для вычисления новых значений δ^+ и δ'^+. После этого вторая половина алгоритма выглядит следующим образом.

(18) $< \delta^+_{new}, \delta'^+_{new} :=< \emptyset, \emptyset >$;
(19) **FOR ALL** $a \in \mathbf{B}^c \setminus (\delta^+ \cup \delta^-)$ **DO**
(20) **IF** переход $M_\Phi^{min}(< \delta^+ \cup \{\neg a\}, \delta'^+ >))$ *не совместим с* $\Delta\}$;
(21) **THEN** $\delta^+_{new} := \delta^+_{new} \cup \{a\}$
(22) **END_IF**
(23) **END_DO**;
(24) **FOR ALL** $a \in \mathbf{B}^c \setminus (\delta'^+ \cup \delta'^-)$ **DO**
(25) **IF** переход $M_\Phi^{min}(< \delta^+, \delta'^+ \cup \{\neg a\} >))$ *не совместим с* $\Delta\}$;
(26) **THEN** $\delta'^+_{new} := \delta'^+_{new} \cup \{a\}$
(27) **END_IF**

END_DO;

$\Delta_1^+ :=< \delta_1^+ \cup \delta_{new}^+, \delta_1'^+ \cup \delta_{new}'^+ >$

$\Delta_1 := (\Delta_1^+, \Delta_1^-, \Delta_1^*)$

$\Delta_1 := C(\Delta_1)$;

UNTIL $(< \delta_{new}^-, \delta_{new}'^- >=< \emptyset, \emptyset >)$ **AND** $(< \delta_{new}^+, \delta_{new}'^+ >=< \emptyset, \emptyset >)$;

Output (Φ_1, Δ_1).

Нетрудно проверить, что для этих алгоритмов справедливы утверждения, аналогичные предложениям 11 и 12.

Предложение 13 *(Полные БД)*
(1) Алгоритм FT_expand вычисляет оператор Γ_f *за квадратичное время.*
(2) Алгоритм BT_expand вычисляет оператор Γ_b *за кубическое время.*

6 Максимальное расширение обновлений для частичных БД

В этом разделе мы установим, что композиция операторов $(\Gamma_f \circ \Gamma_b)$ для частичных БД о беспечивает максимальное расширение обновления, а после дополнительного превращения ДОЦ в аккуратные с помощью преобразования *tidy_IC* можно добиться также максимального упрощения ДОЦ.

Определение 13 *Оператор* Γ_o *на парах* $(\Phi, \Delta) \in \mathbf{IC} \times \mathbf{GUP}$ *зададим следующими соотношениями:*

$\Gamma_o(\Phi, \Delta)^{up} = \Gamma_f \circ \Gamma_b(\Phi, \Delta)^{up}$,

$\Gamma_o(\Phi, \Delta)^{ic} = tidy\_IC(\Gamma_f \circ \Gamma_b(\Phi, \Delta))$.

Легко видеть, что так определенный оператор Γ_o является оператором расширения обновлений.

Следующая теорема показывает, что оператор Γ_o доставляет максимальную пару из множества $Eq(\Phi, \Delta)$.

Теорема 11 *(Частичные БД). Пусть для совместных обновления* Δ *и ДОЦ* $\Phi \in \mathbf{IC}$ *пара* (Φ_1, Δ_1) *является результатом выполнения оператора* $\Gamma_o(\Phi, \Delta)$. *Тогда для любой пары* $(\Phi_2, \Delta_2) \in Equ(\Phi, \Delta)$ *выполнены следующие условия:*
(i) $\Delta_2^+ \subseteq \Delta_1^+$; $\Delta_2^- \subseteq \Delta_1^-$;
(ii) $\neg(\Phi_2 \prec \Phi_1)$.

Доказательству теоремы предпошлем следующую техническую лемму, устанавливающую некоторые важные свойства оператора Γ_o.

Лемма 12 *В условиях теоремы имеют место следующие утверждения.*
(1) $<\delta_1^+, \delta_1^{'+}> = M^{min}_{\Phi(\Delta)}$.
(2) Для каждого предложения $r \in \Phi_1$ $body(r) \neq \emptyset$.
(3) $(\delta_1^+ \cup n(\delta_1^{'+})) \cap \{l \in \mathbf{LB} | l$ *входит в* $\Phi_1\} = \emptyset$.
(4) Для всякого множества литералов I *и любого подмножества предложений* $\Phi' \subseteq \Phi_1$ *имеет место равенство:*

$$M^{min}_{\Phi'}(\Delta_1^+ \cup I) = \Delta_1^+ \cup M^{min}_{\Phi'}(I).$$

(5) Для каждого предложения $r \in \Phi_1$ *существует переход* $<I, I'> \in (Tr(\Phi_1 \setminus \{r\}, \Delta) \setminus Tr(\Phi_1, \Delta))$.
(6) Для каждого предложения $r \in \Phi_1$ *и каждого непустого подмножества* $\alpha \subseteq body(r)$ *существует переход* $<I, I'> \in Tr(\Phi_1, \Delta)$ *такой, что предоложение* $r' = (head(r) \leftarrow body(r) \setminus \alpha)$ *неверно на этом переходе, т.е.* $<I, I'> \not\models r'$.

Доказательство. (1) Поскольку оператор B для частичных интерпретаций не изменяет Δ_1^+, то требуемое утверждение непосредственно следует из леммы 11 (1).
(2) Из определения следует, что пара (Φ_1, Δ_1) является неподвижной точкой оператора γ_f, т.е. $\gamma_f(\Phi_1, \Delta_1) = (res(\Phi_1, \Delta_1), F(res(\Phi_1, \Delta_1), \Delta_1) = (\Phi_1, \Delta_1)$. Следовательно, $\Phi_1 = res(\Phi_1, \Delta_1)$ и $\Delta_1 = F(\Phi_1, \Delta_1)$. Если бы ДОЦ Φ_1 содержали предложение-факт $r = (l \leftarrow .)$, то в случае $l \in \Delta_1^+$ нарушалось бы первое из этих двух равенств, а в случае $l \notin \Delta_1^+$ — второе.
Пункт (3) также имеет место из-за свойства неподвижной точки: $res(\Phi_1, \Delta_1) = \Phi_1$.
(4) Из (3) следует, что никакое правило из Φ' не может сработать при вычислении $M^{min}_{\Phi'}(D_1^+ \cup I)$ из-за того, что в его тело входит какой-либо литерал из Δ_1^+.
(5) Пусть $\Phi' = \Phi_1 \setminus \{r\}$ и $<I, I'> = M^{min}_{\Phi'}(\Delta_1^+ \cup body(r))$. Тогда $<I, I'>$ — совместный переход и $<I, I'> \models \Phi'$. Из (4) следует, что $<I, I'> = D_1^+ \cup M^{min}_{\Phi'}(body(r))$. Так как ДОЦ Φ_1 аккуратны относительно Δ_1, то по условию (a) определения 12 переход $M^{min}_{\Phi'}(body(r))$ согласован с Δ_1. Это значит, что $\Delta_1^+ \subseteq <I, I'>$ и $\delta_1^- \cap I = n(\delta_1^{'-}) \cap I' = \emptyset$. Следовательно, обновление Δ_1 выполнено на $<I, I'>$ и $<I, I'> \in Tr(\Phi', \Delta_1)$. С другой стороны, по условию (b) определения 12 $head(r) \notin M^{min}_{\Phi'}(body(r))$. Тогда $head(r) \notin <I, I'>$ и, поскольку $body(r) \subseteq I$, то $<I, I'> \not\models r$, т.е. $I \notin Tr(\Phi, \Delta)$.
(6) Пусть $\Phi' = \Phi_1 \setminus \{r\}$ и $<I, I'> = M^{min}_{\Phi_1}(\Delta_1^+ \cup body(r'))$. Тогда $<I, I'> \models \Phi_1$. Из пункта (4) следует, что $<I, I'> = \Delta_1^+ \cup M^{min}_{\Phi_1}(body(r'))$. Из аккуратности Φ_1 относительно Δ_1 можно вывести (по свойству (c) определения 12), что $\alpha \not\subseteq M^{min}_{\Phi_1}(body(r'))$. Следовательно, предложение r в процессе вычисления $M^{min}_{\Phi_1}(body(r'))$ никогда не срабатывает. Тогда получаем, что $M^{min}_{\Phi_1}(body(r')) = M^{min}_{\Phi'}(body(r'))$. Из свойства (a)

определения 12 следует, что $M^{min}_{\Phi'}(body(r))$ согласован с Δ_1. Значит обновление Δ_1 выполнено на $< I, I' >$ и $< I, I' > \in Tr(\Phi_1, \Delta_1)$. С другой стороны, по условию (b) определения 12 $head(r) \notin M^{min}_{\Phi'}(body(r))$. Тогда из монотонности замыкания следует, что $head(r) \notin M^{min}_{\Phi'}(body(r')) = M^{min}_{\Phi_1}(body(r'))$. Мы получили, что $head(r) \notin < I, I' >$, но $body(r') \subseteq < I, I' >$. Следовательно, $< I, I' > \not\models r'$. □

Доказательство теоремы. (i) Так как по лемме 12 (1) $\Delta_1^+ = M^{\Phi}_{\Delta}$ и $M^{\Phi}_{\Delta} \in Tr(\Phi, \Delta) = Tr(\Phi_2, \Delta_2)$, то мы получаем, что $\Delta_2^+ \subseteq M^{\Phi}_{\Delta} = \Delta_1^+$. Чтобы установить второе включение, предположим, что существует литерал $l \in \Delta_2^- \setminus \Delta_1^-$. При этом либо $l \in \mathbf{LB^c}$ и $l \in \delta_2^- \setminus \delta_1^-$, либо $l \in \mathbf{LB^n}$ и $l \in \delta_2^{'-} \setminus \delta_1^{'-}$. Пусть имеет место второй случай (первый рассматривается аналогично). Рассмотрим переход $< J, J' >= M^{min}_{\Phi_1}(\Delta_1^+ \cup \{l\})$. Очевидно, что в этом случае $c(l) \in J'$. Если переход $< J, J' >$ не совместим с Δ_1, то литерал l в результате применения оператора B попадет в $\delta^{'-}$. Но тогда $\gamma_b(\Phi_1, \Delta_1) \neq (\Phi_1, \Delta_1)$, что противоречит тому, что (Φ_1, Δ_1) является неподвижной точкой γ_b. Если переход $< J, J' >$ совместим с Δ_1, то поскольку $\Delta_1^+ \sqsubseteq < J, J' >$, на нем выполняется обновление Δ_1. Кроме того, из определени $< J, J' >$ следует, что $< J, J' > \models \Phi_1$ и, следовательно, $< J, J' > \in Tr(\Phi_1, \Delta_1)$. Но так как $c(l) \in J'$, то обновление Δ_2 не выполнено на $< J, J' >$ и $< J, J' > \notin Tr(\Phi_2, \Delta_2)$, что противоречит условию $(\Phi_2, \Delta_2) \in Equ(\Phi, \Delta)$.

(ii) Предположим, что $\Phi_2 \prec \Phi_1$. Тогда по определению отношения $\preceq$ для каждого предложения $r_2 \in \Phi_2$ существует предложение $r_1 \in \Phi_1$, у которого $head(r_1) = head(r_2)$ и $body(r_1) = body(r_2) \cup \alpha$ для некоторого $\alpha \subseteq \mathbf{LB}$. Если $\Phi_2 \neq \Phi_1$, то либо

(1) существуют два предложения $r' \in \Phi_2$ и $r \in \Phi_1$ такие, что $head(r) = head(r')$ и $body(r) = body(r') \cup \alpha$ для некоторого непустого множества удаленных из $body(r')$ литералов α, либо

(2) в Φ_1 имеется такое предложение r_1, что для каждого $r_2 \in \Phi_2$, если $head(r_1) = head(r_2)$, то $body(r_2) \not\subseteq body(r_1)$.

В случае (1) по лемме 12 (6) существует такой переход $< I, I' > \in Tr(\Phi_1, \Delta_1)$, что $< I, I' > \not\models r'$. Тогда $< I, I' > \not\models \Phi_2$, что противоречит условию $(\Phi_2, \Delta_2) \in Equ(\Phi_1, \Delta_1)$. Таким образом условие (1) не выполнено ни для какой пары r, r'. Тогда в случае (2) $\Phi_2 \subseteq \Phi_1 \setminus \{r^1\}$. Теперь по лемме 12 (5) существует переход $< I, I' > \in Tr(\Phi_1 \setminus \{r\}, \Delta_1) \setminus Tr(\Phi_1, \Delta_1)$. Но $Tr(\Phi_1 \setminus \{r\}, \Delta_1) \subseteq Tr(\Phi_2, \Delta_2)$ и, следовательно, $Tr(\Phi_1, \Delta_1) \neq Tr(\Phi_2, \Delta_2)$, что снова приводит к противоречию с условием $(\Phi_2, \Delta_2) \in Equ(\Phi_1, \Delta_1)$. Таким образом мы показали, что неравенство $\Phi_2 \prec \Phi_1$ невозможно. □

Из этой теоремы легко вывести, что итерируя композицию операторов Γ_f и Γ_b, мы не усиливаем выдаваемые результаты.

Следствие 12 *В случае частичных БД для всякого $n \geq 1$ имеем $(\Gamma_f \circ \Gamma_b)^n = \Gamma_f \circ \Gamma_b$.*

Комбинируя уже имеющиеся у нас алгоритмы для вычисления операторов Γ_f, Γ_b и приведения ДОЦ к аккуратному виду, мы получим алгоритм, вычисляющий максимальное расширение обновлений за полиномиальное время.

Алгоритм Opt_expand .
Вход: обновление $\Delta \in \mathbf{GUP}$, совместное с ДОЦ $\Phi \in \mathbf{IC}$.
Выход: (Φ_1, Δ_1).
(1) $(\Phi_1, \Delta_1) := Fp\_expand(\Phi, \Delta)$;
(2) $(\Phi_1, \Delta_1) := Bp\_expand(\Phi_1, \Delta_1)$;
(3) $\Phi_1 := tidy\_IC(\Phi_1, \Delta_1)$;
(4) $Output(\Phi_1, \Delta_1)$.

Суммируя оценки времени использованных в строках 1–3 алгоритмов из предложений 11, 12 и леммы 8, мы получаем следующий центральный результат этого раздела.

Теорема 13
Для частичных БД алгоритм Opt_expand *по заданным совместным обновлению* Δ *и ДОЦ* $\Phi \in \mathbf{IC}$ *вычисляет максимальный элемент* (Δ_{max}, Φ_1) *в классе эквивалентности* $Equ(\Phi, \Delta)$ *за полиномиальное время.*

Отметим, что алгоритм Opt_expand можно реализовать за время $O(N^2)$, где N — размер входа.

Рассмотрим небольшой пример, иллюстрирующий работу алгоритма Opt_expand.

Пример 9 *Зафиксируем множество атомов* $\mathbf{B} = \{a, b, c, d, e, f, \ g, h, k\}$, *и определим ДОЦ* Φ *следующим образом:*

$$\Phi = \begin{cases} r_1: & \neg d' & \leftarrow e, g', h' \\ r_2: & d' & \leftarrow \neg b' \\ r_3: & e & \leftarrow a', d' \\ r_4: & f' & \leftarrow a', e', \neg g \\ r_5: & g' & \leftarrow c', k' \\ r_6: & \neg g & \leftarrow d', e' \\ r_7: & h' & \leftarrow \neg b', e \\ r_8: & k & \leftarrow c', e \end{cases}$$

Пусть нам нужно выполнить обновление $\Delta = (\{a, \neg b\}, \emptyset, \{(e, h)\})$ *для некоторого состояния БД* I.

Посмотрим, насколько сможет упростить задачу выполнени этого обновления предварительное его расширение и упрощение ограничений целостности с помощью алгоритма Opt_expand.

На первом этапе вычисляется прямой оператор $(\Phi_1, \Delta_1) = Fp\_expand(\Phi, \Delta)$. На 1-ой итерации основного цикла этого алгоритма в $\delta_1^{'+}$ попадает d (правило r_2), на 2-ой — в δ_1^{+} попадает e (правило r_3), и при вычислении в строке 14 $C(\Delta_1)$ в $\delta_1^{'+}$ попадает h. В результате после этого этапа получается новое обновление Δ_1, в котором $\delta_1^{+} = \{e\}$, $\delta_1^{'+} = \{a, \neg b, d, h\}$, $\delta_1^{-} = \{\neg e\}$, $\delta_1^{'-} = \{\neg a, b, \neg d, \neg h\}$, а $\Delta_1^{*} = \emptyset$.

При этом ДОЦ упрощаются до

$$\Phi_1 = \begin{cases} r_1: & \neg d' & \leftarrow g' \\ r_4: & f' & \leftarrow e', \neg g \\ r_5: & g' & \leftarrow e', k \\ r_6: & \neg g & \leftarrow e' \\ r_8: & k & \leftarrow c' \end{cases}$$

а втором этапе алгоритм Bp_extend обнаруживает, что добавление каждого из литералов g и c к $\delta_1^{'+}$ приводит к несовместности с Δ_1. Поэтому эти литералы добавляются к множеству $\delta_1^{'-}$, которое становится равно $\{\neg a, b, c, \neg d, g, \neg h\}$. Остальные компоненты обновления не меняются, а ДОЦ упрощаются до следующих

$$\Phi_1 = \begin{cases} r_4: & f' & \leftarrow e', \neg g \\ r_6: & \neg g & \leftarrow e' \end{cases}$$

Наконец, на третьем этапе алгоритм $tidy\_IC$ устраняет функциональную зависимость в правиле r_4 и выдает "минимальные" ДОЦ

$$\Phi_1 = \begin{cases} r_4: & f' & \leftarrow e' \\ r_6: & \neg g & \leftarrow e' \end{cases}$$

Таким образом, исходное обновление расширилось до нового обновлени Δ_1, в котором $\delta_1^{+} = \{e\}$, $\delta_1^{'+} = \{a, \neg b, d, h\}$, $\delta_1^{-} = \{\neg e\}$, $\delta_1^{'-} = \{\neg a, b, c, \neg d, g, \neg h\}$, и $\Delta_1^{*} = \emptyset$. А ДОЦ сократились до указанных двух правил. Конечно, гораздо проще находить минимально отличающееся от I состояние БД I' такое, что переход $< I, I' > \in Tr(\Phi_1, \Delta_1)$, чем делать это непосредственно для исходных Φ и Δ.

7 Максимальные расширения для полных БД

Как мы установили, для частичных БД поиск максимального расширения обновления можно провести за полиномиальное время. Для полных БД максимальное расширение обновления можно просто представить в терминах соответствующих булевых формул, но его поиск является существенно более сложной задачей.

Напомним, что мы называем обновление полным, если все входящие в него литералы являются атомами, т.е. не имеют отрицаний. Свяжем с ДОЦ Φ и полным обновлением Δ булевы формулы $\Phi^{(b)}$ и $\Delta^{(b)}$ следующим образом. Для предложения $r \in \Phi$ пусть $r^{(b)} = (\bigwedge_{l \in body(r)} l \rightarrow head(r))$. Положим теперь $\Phi^{(b)} = \bigwedge_{r \in \Phi} r^{(b)}$. Для обобщенного обновления $\Delta = (<\delta^+, \delta'^+>, <\delta^-, \delta'^->, \Delta^*)$ положим
$\Delta^{(b)} = \bigwedge_{a \in \delta^+} a \wedge \bigwedge_{a \in \delta'^+} n(a) \wedge \bigwedge_{a \in \delta^-} \neg a \wedge \bigwedge_{a \in \delta'^-} \neg n(a) \wedge \bigwedge_{(a,b) \in \Delta^*} (a \rightarrow (n(b) \wedge \neg n(a)))$.

Предложение 14 *(Полные БД)*
(1) Пусть для ДОЦ Φ и полного обновления Δ обновление Δ_{max} является максимальным эквивалентным обновлением. Тогда
$\delta^+_{max} \cup n(\delta'^+_{max}) = \{a \in \mathbf{B} | \Phi^{(b)} \wedge \Delta^{(b)} \models a\}$ *и*
$\delta^-_{max} \cup n(\delta'^-_{max}) = \{a \in \mathbf{B} | \Phi^{(b)} \wedge \Delta^{(b)} \models \neg a\}$
(здесь $\models$ означает обычное логическое следование.
(2) Функция $f(\Phi, \Delta) = \Delta_{max}$ вычислима за полиномиальное время тогда и только тогда, когда $P = NP$.

Набросок доказательства. Утверждение (1) является, по-существу, прямой переформулировкой определения Δ_{max} на языке пропозициональной логики. Если $P = NP$, то $P = coNP$ и логическое следование можно проверить за полиномиальное время. Тогда, используя (1), можно вычислить Δ_{max} за полиномиальное время. Обратно, если функция f вычислима за полиномиальное время, то, используя ее, можно получить алгоритм проверки выполнимости булевых формул за полиномиальное время. Действительно, пусть α — это произвольная 3-КНФ. Построим по ней ДОЦ Φ_α и обновление Δ_α как в доказательстве теоремы 1. Вычислим $\Delta_1 = f(\Phi, \Delta)$. Тогда легко проверить, что α выполнима $\iff a \notin \Delta_1^-$.

Еще одно отличие между частичными и полными БД можно обнаружить, рассматривая композицию прямого и обратного операторов $\Gamma_f \circ \Gamma_b$. Как было отмечено в следствии 12, итерация этого оператора в случае частичных БД не увеличивает его силу. Для полных БД имеется строгая иерархия операторов $(\Gamma_f \circ \Gamma_b)^n$ по n.

Теорема 14 *В случае полных БД для каждого $n \geq 1$ существуют такие совместные ДОЦ Φ и обновление Δ, что $(\Gamma_f \circ \Gamma_b)^{n+1}(\Phi, \Delta) \neq (\Gamma_f \circ \Gamma_b)^n(\Phi, \Delta)$.*

Доказательство. Фактически, эта иерархия проявляется уже на уровне статических ограничений целостности. Рассмотрим следующий набор ограничений целостности Φ, состоящий из $2(n+1)$ предложения:
$r_1: \ a'_1 \leftarrow b'_0$

$s_1 : \neg b'_0 \leftarrow a'_1, \neg b'_1$

. . .

$r_{i+1} :\ a'_{i+1} \leftarrow b'_i$

$s_{i+1} :\ \neg a'_i \leftarrow a'_{i+1}, \neg b'_{i+1}$

. . .

$r_{n+1} :\ a'_{n+1} \leftarrow b'_n$

$s_{n+1} :\ \neg a'_n \leftarrow a'_{n+1}, \neg b'_{n+1}$.

Пусть $\Delta = (\{b_0\}, \emptyset, \emptyset)$. Тогда непосредственными вычислениями нетрудно проверить, что для всякого $1 \leq i \leq n+1$

$(\Gamma_f \circ \Gamma_b)^i(\Phi, \Delta) = (\Phi_i, \Delta_i)$,

где $\Phi_i = \{r_{i+1}, s_{i+1}, ..., r_{n+1}, s_{n+1}\}$ и $\Delta_i = (\{b_0, a_1, b_1, ..., a_i, b_i\}, \emptyset, \emptyset)$.

Следовательно,

$(\Gamma_f \circ \Gamma_b)^{n+1}(\Phi, \Delta) =$

$(\emptyset, (\{b_0, a_1, b_1, ..., a_{n+1}, b_{n+1}\}, \emptyset, \emptyset))$

$\neq (\Gamma_f \circ \Gamma_b(\Phi, \Delta))^n =$

$(\{r_{n+1}, s_{n+1}\}, (\{b_0, a_1, b_1, ..., a_n, b_n\}, \emptyset, \emptyset))$.

□

Предложение 14 показывает, что, по-видимому, не существует алгоритма, вычисляющего Δ_{max} для полных БД за полиномиальное время. Оператор $\Gamma = (\Gamma_f \circ \Gamma_b)^\omega$, дополненный преобразованием ДОЦ *tidy_IC* позволяет получить за полиномиальное время более или менее хорошую аппроксимацию Δ_{max} в общем случае. Ниже мы выделяем один интересный подкласс ограничений целостности, для которого можно так расширить определенный в предыдущем разделе оператор Γ_o, что он будет строить максимальные расширени обновлений для полных БД. Это класс ДОЦ, которые зависят только от присутствия фактов в состояниях БД, т.е. ДОЦ с предложениями, в телах которых все литералы позитивны. Положим $\mathbf{IC}_p = \{\Phi \in \mathbf{IC} | (\forall r \in \Phi)(\ body(r) \subseteq \mathbf{B})\}$.

Оказывается, между максимальными элементами $Equ(\Phi, \Delta)$ для частичных и полных БД с ДОЦ из $\mathbf{IC}_p$ имеется тесная связь. Чтобы различать классы частичных и полных состояний БД мы будем ниже использовать верхний индекс p для частичных и t для полных БД.

Следующая теорема связывает максимальные обновления для частичных и полных БД.

Теорема 15 *Пусть полное обновление* Δ *совместно с ДОЦ* $\Phi \in \mathbf{IC}_p$. *Тогда*
(1) если Δ_p *— максимальное обновление в* $Equ^p(\Phi, \Delta)$, *то в классе* $Equ^t(\Phi, \Delta)$ *максимальным будет обновление* Δ_t, *у которого* $\Delta_t^+ = \neg.\Delta_p^- \cap \mathbf{B}$ *и* $\Delta_t^- = \Delta_p^- \cap \mathbf{B}$;
(2) если Δ_t *— максимальное обновление в* $Equ^t(\Phi, \Delta)$, *то в классе* $Equ^p(\Phi, \Delta)$ *мак-*

симальным будет обновление Δ_p, у которого $\Delta_p^+ = \Delta_t^+ \cup \{\neg a | \exists r \in \Phi(head(r) = \neg a \ \& \ body(r) \subseteq D_t^+)\}$ и $D_p^- = \neg.D_t^+ \cup D_t^-$.

Доказательство теоремы опирается на следующую лемму о связи $Tr^p(\Phi, \Delta)$ и $Tr^t(\Phi, \Delta)$.

Лемма 13 *Пусть полное обновление Δ совместно с ДОЦ $\Phi \in \mathbf{IC}_p$. Тогда*
(1) если $< I_p, I_p' > \in Tr^p(\Phi, \Delta)$, то $< I_p^+, I_p'^+ > \in Tr^t(\Phi, \Delta)$;
(2) если $< I_t, I_t' > \in Tr^t(\Phi, \Delta)$, то $< I_p, I_p' > \in Tr^t(\Phi, \Delta)$, где
$< I_p, I_p' > = I_t \cup n(I_t') \cup \{\neg a | a \in (\mathbf{B} \setminus (I_t \cup n(I_t')))\}$.

Доказательство леммы. (1) Так как $< I_p, I_p' > \in Tr^t(\Phi, \Delta)$, то $\Delta^+ \subseteq < I_p, I_p' >$. И поскольку в Δ^+ нет отрицательных литералов, то $\Delta^+ \subseteq < I_p^+, I_p'^+ >$. Ясно также, что $< I_p^+, I_p'^+ > \cap \Delta^- \subseteq < I_p, I' > \cap \Delta^- = \emptyset$. Таким образом, обновление Δ выполнено в $< I_p^+, I_p'^+ >$. Предположим теперь, что некоторое предложение $r \in \Phi$ ложно на полном переходе $< I_p^+, I_p'^+ >$. Тогда $body(r) \subseteq I_p^+ \cup I_p'^+ \subseteq I_p \cup I_p{}'$ и $< I_p^+, I_p'^+ > \not\models head(r)$. Если при этом $head(r)$ — атом, то он обязательно входит в $I_p^+ \cup I_p'^+$ и тогда $< I_p^+, I_p'^+ > \models r$. Если же $head(r) = \neg a$ для некоторого $a \in \mathbf{B}$, то $a \notin I_p \cup I_p'$ и $a \notin I_p^+ \cup I_p'^+$. Снова получаем, что $< I_p^+, I_p'^+ > \models r$. Следовательно, наше предположение было ошибочным и $< I_p^+, I_p'^+ > \models \Phi$. Таким образом, $< I_p^+, I_p'^+ > \in Tr^t(\Phi, \Delta)$.
(2) Из определения $< I_p, I_p' >$ следует, что Δ выполнено на $< I_p, I_p' >$. Предположим, что некоторое предложение $r \in \Phi$ ложно на частичном переходе $< I_p, I_p' >$. Тогда $body(r) \subseteq I_p^+ \cup I_p'^+ = I_t \cup I_t'$. Если $head(r)$ — атом, то он входит в $I_t \cup I_t'$, а значит — и в $I_p \cup I_p'$. Следовательно, $< I_p, I_p' > \models r$. Если же $head(r) = \neg a$ для некоторого $a \in \mathbf{B}$, то $a \notin I_t \cup I_t'$ и по определению $< I_p, I_p' >$ литерал $\neg a \in < I_p, I_p' >$. Снова мы получаем, что $< I_p, I_p' > \models r$. Следовательно, наше предположение неверно и $< I_p, I_p' > \models \Phi$. Таким образом, $< I_p, I_p' > \in Tr^p(\Phi, \Delta)$. $\square$

Доказательство теоремы. (1) Покажем вначале, что Δ_t^+ входит во все переходы из $Tr^t(\Phi, \Delta)$. Если для некоторого атома a его отрицание $\neg a \in \Delta_p^-$, то для всякого перехода $< I, I' > \in Tr^p(\Phi, \Delta)$ $\neg a \notin < I, I' >$. Предположим теперь, что в $Tr^t(\Phi, \Delta)$ имеется такой переход $< I, I' >$, что $a \notin < I, I' >$. Тогда по лемме 13 (2) существует переход $< I_p, I_p' > \in Tr^p(\Phi, \Delta)$ такой, что $\neg a \in < I_p, I_p' >$. Поэтому наше предположение неверно и a входит во все полные переходы из $Tr^t(\Phi, \Delta)$.

Для доказательства максимальности Δ_t^+ предположим от противного, что есть атом $a \notin \Delta_t^+$, который входит в каждый переход из $Tr^t(\Phi, \Delta)$. Тогда a также входит в каждый переход из $Tr^p(\Phi, \Delta)$ (иначе по лемме 13 (1) в $Tr^t(\Phi, \Delta)$ также нашелся бы полный переход без a). Но тогда никакой переход из $Tr^p(\Phi, \Delta)$ не содержит $\neg a$ и $\neg a \in \Delta_p^-$, поскольку Δ_p^- является максимальным. Из определения Δ_t^+ тогда заключаем, что $a \in \Delta_t^+$, что противоречит нашему предположению. Следовательно,

Δ_t^+ является максимальным.

Покажем теперь, что множество удаляемых фактов Δ_t^- определено правильно. Пусть $<I, I'>$ — произвольный переход из $Tr^t(\Phi, \Delta)$. Если найдется атом $a \in (<I, I'> \cap \Delta_t^-)$, то по лемме 13 (2) найдется частичный переход $< I_p, I_p' > \in Tr^p(\Phi, \Delta)$ такой, что $a \in < I_p, I_p' >$. Но это противоречит выбору Δ_p^-. Чтобы доказать максимальность Δ_t^- предположим, что атом a не входит ни в один переход из $Tr^t(\Phi, \Delta)$. Тогда по лемме 13 (1) a также не входит ни в какой частичный переход из $Tr^p(\Phi, \Delta)$ и, поскольку Δ_p^- максимально, то $a \in \Delta_p^-$. Тогда из определения Δ_t^- следует, что $a \in \Delta_t^-$. Следовательно, Δ_t^- максимально.

(2) Покажем, что Δ_p^+ — это максимальное множество добавляемых литералов для $Tr^p(\Phi, \Delta)$. Отметим, что из позитивности условий во всех предложениях Φ следует, что $M_\Delta^\Phi = \Delta_t^+ \cup \{\neg a | \exists r \in \Phi(head(r) = \neg a \ \& \ body(r) \subseteq \Delta_t^+\})$. Тогда максимальность Δ_p^+ следует из леммы 11 и теоремы 11.

Установим теперь корректность определения Δ_p^-. Пусть $<I, I'> \in Tr^p(\Phi, \Delta)$. Тогда по лемме 13 (1) $< I, I' > \cap \Delta_t^- = \emptyset$. Если некоторый атом a входит в Δ_t^+, то по той же лемме он входит во все частичные переходы из $Tr^p(\Phi, \Delta)$ и, следовательно, его отрицание $\neg a$ не входит ни в один из них. Таким образом, для любого $<I, I'> \in Tr^p(\Phi, \Delta)$ имеем, что $\Delta_p^- \cap <I, I'> = \emptyset$.

Предположим теперь, что некоторый литерал l не входит ни в один переход из $Tr^p(\Phi, \Delta)$. Если $l \in \mathbf{B}$, то $l \in \Delta_t^-$. В противном случае, нашелся бы такой переход $< I, I' > \in Tr^t(\Phi, \Delta)$, что $l \in < I, I' >$, и по лемме 13 (2) существовал бы переход $< I_p, I_p' > \in Tr^p(\Phi, \Delta)$ такой, что $l \in < I_p, I_p' >$. Если же $l = \neg a$ для некотрого $a \in \mathbf{B}$, то нет такого полного перехода $< I, I' > \in Tr^t(\Phi, \Delta)$, для которого $a \notin < I, I' >$. Иначе снова по лемме 13 (2) нашелся бы переход $< I_p, I_p' > \in Tr^p(\Phi, \Delta)$ такой, что $\neg a \in < I_p, I_p' >$. Следовательно, $a \in \Delta_t^+$ и $\neg a \in \Delta_p^-$. Таким образом мы установили, что Δ_p^- является максимальным множеством удаляемых литералов для $Tr^p(\Phi, \Delta)$. $\Box$

Для ограничений целостности связь между частичными и полными БД не такая прямая. Может оказаться, что самые простые ДОЦ для частичных БД позволяют дальнейшие упрощения, если рассматривать их для полных БД.

Пример 10 *Пусть* $\Phi_1 = \{\neg a' \leftarrow b', c';\ c' \leftarrow a'\}$, *а* $\Phi_2 = \{\neg a' \leftarrow b';\ c' \leftarrow a'\}$.

Тогда $\Phi_2 \prec \Phi_1$ и для $\Delta = (\emptyset, \emptyset, \emptyset)$ переход $< \emptyset, \{b\} > \in Tr^p(\Phi_1, \Delta) \setminus Tr^p(\Phi_2, \Delta)$. Но для полных БД $Tr^t(\Phi_1, \Delta) = Tr^t(\Phi_2, \Delta) = \{<I, I'> | I \subseteq \mathbf{B}^c \ \& \ I' \in \{\emptyset, \{b\}, \{c\}, \{a, c\}, \{b, c\}\}$. Таким образом, для полных БД Φ_1 можно упростить до Φ_2.

Чтобы добиться максимального упрощения ДОЦ для полных БД, мы добавим два дополнительных условия к определению аккуратных ДОЦ. Эти условия относятся к предложениям с отрицательными заключениями.

Определение 14 *Пусть* $\Phi \in \mathbf{IC}_p$ *и* $\Delta \in \mathbf{GUP}$. Φ *являются* п-аккуратными ДОЦ *относительно* Δ, *если они удовлетворяют условиям (a) - (c) определения 12 и (d) для каждого предложения* $r \in \Phi$, *заключение которого* $head(r)$ *является негативным литералом, переход* $M_\Phi^{min}(body(r) \cup \neg.head(r))$ *согласован с* Δ;
(e) для каждого предложения $r \in \Phi$, *заключение которого* $head(r)$ *является негативным литералом, и любого подмножества* $\alpha \subset body(r)$ *разность* $body(r) \setminus M_\Phi^{min}(\alpha \cup \neg.head(r))$, *непуста, т.е. во множестве* $(body(r) \cup \neg.head(r))$ *нет функциональных зависимостей.*

Алгоритм tot_tidy_IC, вычисляющий для заданных ДОЦ Φ эквивалентные п-аккуратные ДОЦ Φ_1 получается из алгоритма $tidy\_IC$ добавлением следующих двух циклов после строки (14).

(15) **FOR EACH** $r \in \Phi_1$ **DO**
(16) **IF** $(head(r) \in \neg.\mathbf{B})$ **AND**
$(M_{\Phi_1}^{min}(body(r) \cup \neg.head(r))$ не согласован с $\Delta)$
(17) **THEN** $\Phi_1 := \Phi_1 \setminus \{r\}$ **END_IF**
(18) **END_DO**;
(19) **FOR EACH** предложения $r \in \Phi_1$ **DO**
(20) **IF** $head(r) \in \neg.\mathbf{B}$
(21) **THEN FOR EACH** $b \in body(r)$ **DO**
(22) **IF** $b \in M_{\Phi_1}^{min}((body(r) \setminus \{b\}) \cup \{\neg.head(r)\})$ **THEN** $body(r) := body(r) \setminus \{b\}$
(23) **END_IF**
(24) **END_DO**
(25) **END_IF**
(26) **END_DO**;

Следующая лемма устанавливает корректность преобразования на первом цикле в строках (15) - (18).

Лемма 14 *Пусть* Δ *— полное обновление,* $\Phi \in \mathbf{IC}_p$, $a \in \mathbf{B}$ *и* $r \in \Phi$ *— это такое предложение, что* $head(r) = \neg a$ *и* $M_\Phi^{min}(body(r)) \cup \{a\})$ *не согласовано с* Δ. *Тогда* $Tr^t(\Phi, \Delta) = Tr^t(\Phi \setminus \{r\}, \Delta)$.

Доказательство. Включение $Tr^t(\Phi, \Delta) \subseteq Tr^t(\Phi \setminus \{r\}, \Delta)$ очевидно. Предположим теперь, что имеется переход $<I, I'> \in Tr^t(\Phi \setminus \{r\}, \Delta) \setminus Tr^t(\Phi, \Delta)$. Это означает, что $<I, I'> \not\models r$. В таком случае, $body(r) \subseteq <I, I'>$ и $a \in <I, I'>$. Тогда $M_{\Phi \setminus \{r\}}^{min+}(body(r) \cup \{a\}) = M_\Phi^{min+}(body(r) \cup \{a\}) \subseteq <I, I'>$. Но в этом случае переход $<I, I'>$ не согласован с Δ, что противоречит условию выбора $<I, I'> \in (Tr^t(\Phi \setminus \{r\}, \Delta)$. $\Box$

Следующая лемма устанавливает корректность преобразования на втором цикле в строках (19) - (26).

Лемма 15 *Пусть* $\Phi \in \mathbf{IC}_p$, $a, b \in \mathbf{B}$, *а* $r \in \Phi$ *— это такое предложение, у котрого* $head(r) = \neg a$, $body(r) = body' \cup \{b\}$ *и* $b \in M^{min}_{\Phi}(body') \cup \{a\})$. *Тогда для всякого полного перехода* $<I, I'>$ *имеет место соотношение*
$<I, I'> \models \Phi \;\Leftrightarrow\; <I, I'> \models (\Phi \setminus \{r\}) \cup \{r' :\ \neg a \leftarrow body'\}$.

Доказательство. Проверим эквивалентность r и r'. Так как $body' \subset body(r)$, то из $<I, I'> \models r'$ следует, что $<I, I'> \models r$.

Предположим теперь, что $<I, I'> \models r$. Если $a \notin <I, I'>$, то, очевидно, $<I, I'> \models r'$. Если же $a \in <I, I'>$ и $<I, I'> \not\models r'$, то $body' \subseteq <I, I'>$ и $b \in M^{min}_{\Phi}(body') \cup \{a\}) \subseteq <I, I'>$. Но в таком случае $body(r) \subseteq <I, I'>$ и $<I, I'> \not\models r$, что противоречит нашему предположению Следовательно, $<I, I'> \models r'$. $\square$

Из двух предыдущих лемм и определения алгоритма tot_tidy_IC мы получаем следующий аналог леммы 8.

Лемма 16 *Существует преобразование* $tot\_tidy\_IC : \mathbf{IC}_p \times \mathbf{UP} \rightarrow \mathbf{IC}_p$ *такое, что*
(i) $Tr^t(tidy\_IC(\Phi, \Delta), \Delta) = Tr^t(\Phi, \Delta)$;
(ii) $tot\_tidy\_IC(\Phi, \Delta)$ *являются n-аккуратными ДОЦ по отношению к* Δ;
(iii) преобразование $tot\_tidy\_IC(\Phi, \Delta)$ *вычислимо за полиномиальное (фактически, квадратичное) время.*

Теперь мы можем описать алгоритм tot_opt_expand, вычисляющий оператор расширения обновлений Γ_{to}, который оптимален для полных БД с ограничениями целостности из класса $\mathbf{IC}_p$.

Алгоритм tot_opt_expand:
Вход: $\Phi \in \mathbf{IC}_p$, $\Delta \in \mathbf{GUP}$.
Выход: (Φ_1, Δ_1).

(1) $(\Phi_1, \Delta_1) := fp\_expand(\Phi, \Delta)$;
(2) $(\Phi_1, \Delta_1) := bp\_expand(\Phi_1, \Delta_1)$;
(3) $\Delta_1^+ := \neg.\Delta_1^- \cap \mathbf{B}$; $\Delta_1^- := \Delta_1^- \cap \mathbf{B}$;
(4) $\Phi_1 := tot\_tidy\_IC(\Phi_1, \Delta_1)$;
(5) $Output\ (\Phi_1, \Delta_1)$.

Таким образом, мы определяем $\Gamma_{to}(\Phi, \Delta)$ как результат (Φ_1, Δ_1) выполнения алгоритма tot_opt_expand на входе (Φ, Δ). Установим оптимальность этого оператора.

Теорема 16 *Зафиксируем произвольное полное обновление* Δ, *совместное с ДОЦ* $\Phi \in \mathbf{IC}_p$ *и обозначим через* (Φ_1, Δ_1) *результат применения оператора* $\Gamma_{to}(\Phi, \Delta)$.

Тогда для любой другой эквивалентной пары $(\Phi_2, \Delta_2) \in Equ^t(\Phi, \Delta)$ *выполнены следующие условия:*
(i) $\Delta_2^+ \subseteq \Delta_1^+, \quad \Delta_2^- \subseteq \Delta_1^-$;
(ii) $\neg(\Phi_2 \prec \Phi_1)$.

Доказательство. Пункт (i) следует из определени Γ_{to} и теоремы 15.
(ii) Предположим, что $\Phi_2 \prec \Phi_1$. Так же, как в доказательстве пункта (ii) теоремы 11, нужно рассмотреть два случая.

В первом из них имеются такие предложения $r' \in \Phi_2$ и $r \in \Phi_1$, что $head(r) = head(r')$ и $body(r) = body(r') \cup \alpha$ для некоторого непустого множества литералов α.

Если $head(r) = a \in \mathbf{B}$, то по лемме 12 (6) существует переход $< I, I' >\in Tr^p(\Phi_1, \Delta_1)$ такой, что $<I, I'>\not\models r'$. Тогда по лемме 13 (1) $< I^+, I'^+ >\in Tr^t(\Phi_1, \Delta_1)$ и также $< I^+, I'^+ >\not\models r'$. Но тогда $< I^+, I'^+ >\not\models \Phi_2$, что противоречит тому, что $(\Phi_2, \Delta_2) \in Equ(\Phi_1, \Delta_1)$.

Предположим теперь, что $head(r) = \neg a$ для некоторого $a \in \mathbf{B}$. Определим переходы $< I, I' >= M^{min}_{\Phi_1}(body(r') \cup \{a\})$ и $< I_1, I_1' >= M^{\Phi}_{\Delta} \cup < I, I' >$. Так как Φ_1 являются п-аккуратными относительно Δ, то из условия (d) определения 14 следует, что $< I, I' >$ согласован с Δ. Тогда обновление Δ выполнено на $< I_1, I_1' >$ и также на $< I_1^+, I_1'^+ >$. Из условия (e) определения 14 следует, что $\alpha \not\subseteq < I_1^+, I_1'^+ >$. Следовательно, $< I_1^+, I_1'^+ >\models r$ и $< I_1^+, I_1'^+ >\not\models r'$, что противоречит тому, что $(\Phi_2, \Delta_2) \in Equ^t(\Phi, \Delta)$.

Итак, мы можем далее предполагать, что ни одна пара предложений $r \in \Phi_1, r' \in \Phi_2$ не удовлетворяет условию первого случая. Тогда единственной причиной неравенства $\Phi_2 \prec \Phi_1$ может быть существование такого предложения $r \in \Phi_1$, что $\Phi_2 \preceq \Phi_1 \setminus \{r\}$. Если $head(r) \in \mathbf{B}$, то снова, используя лемму 12 (5) и лемму 13 (1), можно найти полный переход $<I, I'>$, принадлежащий разности $Tr(\Phi_2, \Delta_2) \setminus Tr(\Phi_1, \Delta_1)$.

Если же $head(r) = \neg a$ для некоторого $a \in \mathbf{B}$, то мы рассмотрим переход $<I, I'>= M^{min}_{(\Phi_1 \setminus \{r\})}(body(r) \cup \{a\})$. Он является (частичной) моделью $\Phi_1 \setminus \{r\}$, совместной с Δ_1 и, значит, — с Δ_2. При этом предложение r ложно на $< I, I' >$. Определим тогда еще один переход $< I_1, I_1' >= M^{\Phi}_{\Delta} \cup < I, I' >$ и рассмотрим его позитивную часть $< I_1^+, I_1'^+ >$. Из этого определения и леммы 13 легко вывести, что $< I_1^+, I_1'^+ >\in Tr^t(\Phi_2, \Delta_2) \setminus Tr^t(\Phi_1, \Delta_1)$, что противоречит тому, что $(\Phi_2, \Delta_2) \in Equ(\Phi, \Delta)$. Таким образом, мы установили, что неравенство $\Phi_2 \prec \Phi_1$ не выполнено ни для каких ДОЦ Φ_2, эквивалентных Φ относительно Δ. $\square$

Объединяя теоремы 13 и 16, мы получаем следующий результат об оптимальном операторе расширения обновлений для класса ДОЦ с положительными условиями и о сложности его вычисления.

Теорема 17 *Для класса полных БД с ДОЦ из* $\mathbf{IC}_p$ *алгоритм* tot_opt_expand *по* *заданным совместным полному обновлению Δ и $\Phi \in \mathbf{IC}_p$ вычисляет максимальный элемент (Δ_{max}, Φ_1) из $Equ^t(\Phi, \Delta)$ за полиномиальное (фактически, квадратичное) время.*

Закончим этот раздел небольшим примером применения оператора максимального обновления для полных БД с позитивными ДОЦ.

Пример 11 Рассмотрим фрагмент кадровой БД со следующими ДОЦ Φ*:*

r_1 : *зарплата'(1000)* $\leftarrow$ *отдел'(пс), должность'(программист);*

r_2 : $\neg$ *должность'(программист)* $\leftarrow$ *отдел'(пс), зарплата'(300), образование(выс) ;*

r_3 : $\neg$ *зарплата'(1000)*$\leftarrow$ *зарплата'(300);*

r_4 : $\neg$ *отдел'(пс)* $\leftarrow$ *образование(сред);*

r_5 : *образование(выс)*$\leftarrow$ *должность'(программист);*

r_6 : *должность'(техник)* $\leftarrow$ *отдел'(пс), зарплата(500);*

r_7 : $\neg$ *зарплата(500)* $\leftarrow$ *должность'(техник),отдел(торговый).*

Пусть исходное (полное) состояние БД $I = \{$*зарплата(300), образование(выс)*$\}$. Предположим, что нужно добавить новые факты о зачислении сотрудника в отдел программных средств (пс) на должность программиста, т.е. $\Delta^+ = \{$ *отдел(пс), должность(программист)* $\}$, а $\Delta^- = \Delta^* = \emptyset$. Тогда алгоритм *tot_opt_expand* вначале вычислит максимальное частичное расширение обновления Δ, применив последовательно прямое и обратное преобразования *fp_expand* и *bp_expand*. После первого из них

$\delta_1^+ = \{$ *образование(выс)*$\}$,

$\delta_1'^+ = \{$ *отдел(пс), должность(программист), зарплата(1000),*$\neg$*зарплата(300)* $\}$,

$\delta_1^- = \{\neg$ *образование(выс)*$\}$,

$\delta_1'^- = \{\neg$ *отдел(пс)*$, \neg$ *должность(программист)*$, \neg$ *зарплата(1000)* $\}$,

а ограничения целостности упрощаются до Φ_1:

r_2' : $\neg$ должность'(программист) $\leftarrow$ зарплата'(300) ;

r_3 : $\neg$ зарплата'(1000)$\leftarrow$ зарплата'(300);

r_4 : $\neg$ отдел'(пс) $\leftarrow$ образование(сред);

r_6' : должность'(техник) $\leftarrow$ зарплата(500);

r_7 : $\neg$ зарплата(500) $\leftarrow$ должность'(техник),отдел(торговый).

Обратное преобразование увеличит множество запрещенных фактов. После него

$\delta_1^- = \{\neg$ *образование(выс), образование(сред)*$\}$,

$\delta_1'^- = \{\neg$ *отдел(пс)*$, \neg$ *должность(программист)*$, \neg$ *зарплата(1000),зарплата(300)* $\}$.

Из ограничений целостности будут удалены r_2', r_3 и r_4 и мы получим Φ_1:
r_6' : должность'(техник) $\leftarrow$ зарплата(500);
r_7 : $\neg$ зарплата(500) $\leftarrow$ должность'(техник),отдел(торговый).
Затем в обновлении будут оставлены лишь позитивные литералы (строка 3) и будет определено максимальное полное обновление Δ_1:
$\delta_1^+ = \{$ *образование(выс)*$\}$,
$\delta_1'^+ = \{$ *отдел(пс), должность(программист), зарплата(1000)*$\}$,
$\delta_1^- = \{$ *образование(сред)*$\}$,
$\delta_1'^- = \{$*зарплата(300)* $\}$.
И, наконец, процедура получения п-аккуратных ДОЦ обнаружит в предложении r_7 зависимость b от a и c и упростит его. Таким образом, результирующие ДОЦ Φ_1 состоят из двух простых предложений:
r_6' : должность'(техник) $\leftarrow$ зарплата(500);
r_7' : $\neg$ зарплата(500) $\leftarrow$ отдел(торговый).
В результате применения полученного обновления Δ_1 мы получим новое состояние $I' = \{$ *образование(выс), отдел(пс), должность(программист), зарплата(1000)*$\}$.
Нетрудно проверить, что результирующие ДОЦ Φ_1 выполнены на переходе $< I, I' >$. Хотя в нашем заключительном применении обновления Δ_1 множества δ_1^+ и δ_1^-, относящиеся к исходному состоянию, не участвовали явно, информация, заключенная в них, также может оказаться полезной. Если бы, например, в исходном состоянии I не было бы факта *образование(выс)* или присутсвовал факт *образование(сред),* то можно было бы сразу заключить, что предложенное обновление Δ к нему применить нельзя и выдать соответствующее сообщение об этом пользователю.

8 Заключение

Мы рассмотрели проблему выполнения обновлений полных и частичных БД с динамическими ограничениями целостности, т.е. ограничениями на выполнямые переходы из одного состояния БД в другое, и обобщили и уточнили результаты, полученные ранее нами с соавторами в работах [7, 8, 9] для статических ограничений целостности. В качестве языка задания ДОЦ использовались обобщенные логические программы. Указанная проблема была формализована как задача поиска необходимых достаточных изменений (НДИ) исходного состояния БД и как задача перечисления всех вариантов необходимых достаточных изменений (ПНДИ). Были предложены алгоритмы решения этих задач и получены оценки их сложности. Для частичных баз алгоритм, построенный для решения задачи НДИ, имеет полиномиальную сложность, что позволяет существенно оптимизировать алгоритм решения ПНДИ. Для полных БД задача НДИ оказывается существенно сложнее: она полнота сложностном функциональном классе $\mathrm{FNP}//\mathrm{OptP}[O(\log n)]$.

Для поиска практического решения рассматриваемой проблемы в этом случае предложено выполнение обновления разбить на два этапа. На первом этапе — трансляции запроса — исходный запрос на обновление максимально расширяется с использованием информации о ДОЦ, и одновременно происходит оптимизация (упрощение) ДОЦ. Этот этап не зависит от исходного состояния БД. На втором этапе происходит выполнение расширенного запроса относительно упрощенных ДОЦ для заданного исходного состояния БД. Отмечено, что каждый новый факт в обновлении и каждое упрощение ДОЦ дают значительное сокращение времени выполнения обновления на втором этапе в алгоритмах, решающих задачи НДИ и ПНДИ.

Основные результаты об оптимизации обновлений показывают, что

- для частичных БД оптимальные пары вида (наибольшее корректное обновление/ наиболее простые эквивалентные ДОЦ) всегда существуют и могут быть найдены за полиномиальное время;

- для полных БД в общем случае существование эффективного алгоритма для поиска таких оптимальных пар маловероятно, т.к. влечет выполнение равенства $P = NP$;

- для полных БД, имеющих ДОЦ с позитивными телами правил, оптимальные пары (наибольшее корректное обновление/ наиболее простые эквивалентные ДОЦ) все же можно находить за полиномиальное время.

Фактически по оптимальным парам (наибольшее корректное обновление/ наиболее простые эквивалентные ДОЦ) для частичных БД можно строить корректные пары для полных БД, которые, вообще говоря, не будут оптимальными, но достаточно подходящими их аппроксимациями.

Отметим также, что предложенный подход к оптимизации обновлений можно использовать для создания эффективных интерактивных процедур выполнения обновлений, когда максимально расширенное обновление непосредственно применяется к исходному состоянию БД, а возникающие при этом конфликты предлагается разрешить самому пользователю, после чего полученная от него информация используется для нового расширения обновления и т.д.

Таким образом, результаты работы показывают, что в СУБД и СУБЗ можно практически реализовать более "интеллектуальный " подход к выполнению обновлений, чем используемый сейчас, основанный на автоматическом откате любой транзакции, нарушающей ограничение целостности. Представляет также интерес соединение примененного здесь подхода с различными методами выполнения обновлений не только экстенсиональных состояний БД, но и интенсиональных состояний, задаваемых представлениями баз данных (см. [21]).

Благодарности

Автор благодарен А.Я. Диковскому за многолетнюю совместную работу по обновлениям баз данных и поддержку при подготовке данной статьи. Автор также признателен С.М. Дудакову за полезные обсуждения результатов о сложности обновлений.

Список литературы

[1] Abiteboul, S.:Updates a new Frontier. In: *Proc. of the Second International Conference on the Theory of Databases, ICDT'88.* LNCS **326** (1988) 1-18.

[2] Bonner, A.J., Kifer, M.: An Overview of Transaction Logic. *Theoretical Computer Science,* **133**(2) (1994) 205-265.

[3] Ceri S., Widom J.: Deriving incremental production rules for deductive data. Information Systems, vol. **19**, no. 6 (1994) 467-490.

[4] Chen Z.-Z., Toda S.: The complexity of selecting maximal solutions.*Information and Computation*, 119, no. 2, 231-239.

[5] U. Dayal, E. Hanson, and J. Widom : Active database systems. In: W. Kim (ed.) *Modern Database Systems, Addison Wesley* (1995) 436-456.

[6] Decker H.: An extension of SLD by abduction and integrity maintenance for view updating in deductive databases. In: *Proc. of the 1996 International Conference on Logic Programming.* MIT Press, (1996) 157-169.

[7] Dekhtyar, M., Dikovsky, A., Spyratos, N.: On Conservative Enforced Updates. In: Dix, J., Furbach, U., Nerode, A., editors: *Proceedings of 4th International Conference, LPNMR'97.* Dagstuhl Castle, Germany, LNCS **1265** (1997) 244-257.

[8] Дехтярь М.И., Диковский А.Я., Спиратос Н.: Восстановление ограничений целостности за счет наименьших достаточных изменений. Программирование, N 2 (1998) 27 - 37.

[9] Dekhtyar, M., Dikovsky, A., Dudakov, S., Spyratos, N.: Monotone Expansion of Updates in Logical Databases. In: *Proc. of 5th International Conference LPNMR'99.* LNAI **1730** (1999) 132-147.

[10] Dekhtyar, M., Dikovsky, A., Dudakov, S., Spyratos, N.: Maximal Expansions of Database Updates. In *Foundations of Information and Knowledge Systems, FoIKS 2000.* LNCS **1762** (2000) 72-87.

[11] Dekhtyar, M., Dikovsky, A., Dudakov, S. : On Complexity of Updates Through Integrity Constraints. In :*Proc. of the First Int. Conf. on Computational Logic (CL 2000)*, LNAI **1861** (2000) 867-881.

[12] Eiter, T., Gottlob, G.: On the complexity of propositional knowledge base revision, updates, and counterfactuals. Artificial Intelligence **57** (1992) 227-270.

[13] Eshghi, K., Kowalski, R. A.: Abduction Compared with Negation by Failure. In: *Proc. of the 1989 International Conference on Logic Programming.* (1989)

[14] Fernandez, J.A., Grant, J., Minker, J.: Model-Theoretic Approach to View Updates in Deductive Databases. *Journ. of Automated Reasoning,* **17**(1996) 171-197.

[15] Гэри М., Джонсон Д.: Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.

[16] Guessoum A., Lloyd J.W.: Updating knowledge bases. New Generation Computing, **8** (1990) 71-89.

[17] Jenner B., Torán J.: The complexity of obtaining solutions for problems in NP and NL. In:L.A. Hemaspaandra, A.L. Selman eds.: *Complexity theory retrospective II.* N-Y, Springer-Verlag, (1997) 155-178.

[18] Kakas A.C., Mancarella P.: Database updates through abduction. IN: *Proc. 16th VLBD Conference.* (1990) 650-661.

[19] Lloyd, J.W., Foundations of Logic Programming. Second, Extended Edition. Springer (1993)

[20] Lobo, J., Trajcevski, G., Minimal and consistent evolution of knowledge bases. Journ. of Applied Non-Classical Logics, **7**, no. 1-2 (1997) 117-146.

[21] Mayol E., Teniente E.: A survey of current methods for integrity constraint maintainance and view updates. In: *Proc. of Advances in Conceptual Modeling: ER '99 Workshops on Evolution and Change in Data Management, Reverse Engineering in Information Systems, and the World Wide Web and Conceptual Modeling.* LNCS **1727**, Springer (1999) 62-73.

[22] Marek, V.W., Truszcińsky, M.: Revision programming, database updates and integrity constraints. In: *International Conference on Data Base theory, ICDT*. LNCS **893** (1995) 368-382.

[23] Papadimitriou C.H.:Computational Complexity. Addison-Wesley (1994).

[24] Przymusinski, T.C., Turner, H.: Update by Means of Inference Rules. In: V.W.Marek, A.Nerode, M.Truszczyński, editors, *Logic Programming and Nonmonotonic Reasoning.* Proc. of the Third Int. Conf. LPNMR'95, Lexington, KY, USA (1995) 166-174.

[25] Zaniolo C.: A unified semantics for active and deductive databases. In: *Proc. of 1st International Workshop on Rules in Database Systems, RIDS'93.* Springer (1993) 271-287.

[26] Zaniolo C.,Cheri S.,Faloutsos C., Snodgrass R.T., Subrahmanian V.S.,Zicari R.: Advanced database systems. Morgan Kaufmann Publishers, Inc., San Francisco (1997).