МЕТОДЫ И ТЕХНОЛОГИИ ПРИНЯТИЯ РЕШЕНИЙ

УДК 004.89, 004.832 DOI: 10.18287/2223-9537-2021-11-4-521-532

Интеграция методов поиска в ширину и логического вывода для удовлетворения табличных ограничений

А.А. Зуенко

Институт информатики и математического моделирования – обособленное подразделение Федерального государственного бюджетного учреждения Федерального исследовательского центра «Кольский научный центр Российской академии наук» (ИИММ КНЦ РАН), Апатиты, Мурманской обл., Россия

Аннотация

В рамках технологии программирования в ограничениях широко применяются табличные ограничения: обычные таблицы, *compressed*-таблицы, *smart*-таблицы, сегментированные таблицы и т.п. С их помощью могут быть представлены любые другие виды ограничений, а алгоритмы распространения табличных ограничений (логического вывода на табличных ограничениях) позволяют отфильтровывать «лишние» значения из доменов переменных, обладая при этом низкой вычислительной сложностью. В предыдущих исследованиях автора было предложено *smart*-таблицы подразделять на структуры *C*- и *D*-типов. Общепринятая методология решения задач удовлетворения ограничений заключается в совместном применении методов распространения ограничений и методов поиска в глубину с возвратами. В настоящем исследовании предлагается интегрировать методы поиска в ширину с авторским методом распространения табличных ограничений. *Smart*-таблицы *D*-типа предлагается представлять как соединение нескольких ортогонализованных *smart*-таблиц *C*-типа. Шаг поиска заключается в выборе пары *smart*-таблиц *C*-типа для соединения и последующем распространении ограничений. Для определения порядка соединения ортогонализованных *smart*-таблиц на каждом шаге поиска используется специализированная эвристика, которая обеспечивает сокращение пространства поиска с учётом дальнейших вычислений. При распространении ограничений ускорение процесса вычислений достигается за счёт применения разработанных правил редукции для случая *smart*-таблиц *C*-типа. Разработанный гибридный метод позволяет отыскивать все решения задач удовлетворения ограничений, моделируемых с помощью одной или нескольких *smart*-таблиц *D*-типа, без разложения табличных ограничений в элементарные кортежи.

Ключевые слова: *программирование в ограничениях, smart-таблицы, поиск в ширину, распространение ограничений, задача удовлетворения ограничений.*

Цитирование: *Зуенко, А.А.* Интеграция методов поиска в ширину и логического вывода для удовлетворения табличных ограничений/ А.А. Зуенко // Онтология проектирования. – 2021. – Т.11, №4(42). - С.521-532. – DOI: 10.18287/2223-9537-2021-11-4-521-532.

Введение

Во многих системах искусственного интеллекта (ИИ) применяются методы логического анализа. Они активно развиваются в рамках такого направления ИИ, как программирование в ограничениях, которое основывается на декларативной парадигме представления знаний в форме задачи удовлетворения ограничений [1].

Задача удовлетворения ограничений описывается множеством переменных $X_1, X_2,\ldots, X_n$ и множеством ограничений $C_1, C_2,\ldots, C_m$ [2, 3]. Каждая переменная X_i имеет непустую область определения D_i. Каждое ограничение C_i на некотором подмножестве переменных задаёт допустимые комбинации значений для этого подмножества. Решением задачи является полное

присваивание значений всем переменным $\{X_1 = v_1,\ldots, X_n = v_n\}$, которое удовлетворяет всем ограничениям.

В настоящей работе рассматриваются задачи программирования в ограничениях, где переменные дискретны и определены на конечных множествах значений. Статья посвящена вопросам обработки табличных ограничений [4-14]. Простейшим примером табличных ограничений являются реляционные таблицы. В виде реляционных таблиц, содержащих множество выполняющих подстановок, может быть представлен любой конечный предикат. Такое эксплицитное представление для многих отношений нецелесообразно, поскольку приводит к экспоненциальному росту сложности процедур удовлетворения ограничений. Поэтому предпринимаются попытки разработать более компактное табличное представление качественных зависимостей, в частности были предложены такие виды табличных ограничений как *compressed*-таблицы (*compressed tables*) и *smart* –таблицы (*smart tables*) [8-14].

Изучение прототипов показало, что известные типы табличных ограничений ориентированы на моделирование дизъюнктивных нормальных форм конечных предикатов и недостаточно хорошо подходят для моделирования конъюнктивных нормальных форм логических формул.

Некоторые исследования автора, посвящённые обработке табличных ограничений, описаны в [15, 16], подробный анализ приведён в [17].

В рамках программирования в ограничениях общепринятая методология поиска заключается в совместном применении методов распространения ограничений с методами поиска в глубину с возвратами, при этом используются специализированные эвристики для выбора переменной и её значения, а также стратегии возврата к состоянию, явившемуся причиной возникновения недопустимого присваивания. Особенность методов поиска в глубину с возвратами заключается в том, чтобы пошагово расширять частичное допустимое решение до полного. Если частичное решение недопустимо, то осуществляется возврат и предыдущее частичное решение расширяется в альтернативном направлении.

В настоящей работе предлагается метод, сочетающий стратегию поиска в ширину и авторский метод распространения табличных ограничений. Метод предназначен для решения задач удовлетворения ограничений, представленных с помощью *smart* таблиц *D*-типа.

1 Табличные ограничения для представления знаний

В работе [17] предложено рассматривать следующие виды табличных ограничений: обычная реляционная таблица, *compressed* таблица, *basic smart* таблица, *smart* таблица. *Smart* таблицы являются обобщением всех названных типов табличных ограничений. *Smart* таблицы предложено подразделять на структуры *C*- и *D*-типа. *Smart*-таблицы могут в своей схеме содержать не только простые, но и составные атрибуты, значениями которых являются отношения из предопределённого множества. *Smart*-таблицы *C*-типа соответствуют дизъюнктивным нормальным формам логических формул с элементарными одно и двуместными предикатами, а *smart*-таблицы *D*-типа соответствуют конъюнктивным нормальным формам таких формул.

В настоящей работе при записи содержимого *smart*-таблиц используются две фиктивные компоненты: полная компонента (обозначается «*») – это множество, равное домену соответствующего (по месту её расположения в кортеже) атрибута; пустое множество – $\varnothing$.

Smart-таблица *C*-типа записывается в виде матрицы, ограниченной прямыми скобками.

***Пример* 1**. Рассматривается ограничение, которое описывает вариативные условия формирования цены на товар при решении задачи планирования закупок для некоторой компании. На естественном языке это ограничение может быть записано так: либо цена товара 1

составляет 35 долларов, при этом должно быть закуплено больше 15 единиц товара 1 и столько же единиц товара 2, либо цена товара 1 равняется 50 долларам, в противном случае.

На языке логических формул это ограничение можно выразить следующим образом:

$$(X_1 \geq 15) \wedge (X_1 = X_2) \wedge (C = 35) \vee (X_1 < 15) \wedge (C = 50) \vee (X_1 \neq X_2) \wedge (C = 50),$$

где X_1 – количество закупаемого товара 1, $X_1 \in [0, 111]$; X_2 – количество закупаемого товара 2, $X_2 \in [0, 250]$; C – цена товара, $C \in \{35, 50\}$.

Данное ограничение в виде *smart*-таблицы *C*-типа запишется так:

$$T[X_1, X_2, C, X_1X_2] = \begin{bmatrix} \geq 15 & * & \{35\} & = \\ < 15 & * & \{50\} & * \\ * & * & \{50\} & \neq \end{bmatrix}.$$

Каждая строка этой *smart*-таблицы соответствует некоторому дизъюнкту приведённой логической формулы. Данная *smart*-таблица содержит в описании своей схемы составной атрибут X_1X_2, доменом которого является множество бинарных отношений $\{=, \neq\}$.

Как видно из примера 1, с помощью *smart*-таблиц *C*-типа удобно представлять дизъюнктивные нормальные формы (ДНФ) конечных предикатов.

С помощью *smart*-таблиц *D*-типа моделируются коньюнктивные нормальные формы (КНФ) конечных предикатов. Содержимое *smart*-таблицы *D*-типа заключается в перевёрнутые прямые скобки. *Smart*-таблицы *D*-типа позволяют легко вычислять дополнение *smart*-таблиц *C*-типа относительно заданного универсума: требуется взять дополнение каждой компоненты *smart*-таблицы. Например, дополнением *smart*-таблицы $T\ [X_1, X_2, C, X_1X_2]$ является *smart*-таблица *D*-типа:

$$\overline{T}[X_1, X_2, C, X_1X_2] = \left]\begin{matrix} < 15 & \varnothing & \{50\} & \neq \\ \geq 15 & \varnothing & \{35\} & \varnothing \\ \varnothing & \varnothing & \{35\} & = \end{matrix}\right[.$$

Smart-таблицу *D*-типа $\overline{T}$ можно представить в виде:

$$[(X_1 < 15) \wedge (X_1 \neq X_2) \wedge (C = 50)] \vee [(X_1 \geq 15) \wedge (C = 35)] \vee [(X_1 = X_2) \wedge (C = 35)],$$

Поскольку *smart*-таблицы являются «сжатым» представлением многоместных отношений (обычных таблиц), то к ним могут быть применены все операции реляционной алгебры. Стоит подчеркнуть, что при выполнении операций дополнения и соединения не требуется раскладывать *smart*-таблицы в совокупность элементарных кортежей.

Любую *smart*-таблицу *D*-типа можно представить как соединение (обозначено символом $\bowtie$)нескольких диагональных *smart*-таблиц *C*-типа, каждая из которых соответствует некоторой строке исходной *smart*-таблицы *D*-типа. Например:

$$\overline{T}[X_1, X_2, C, X_1X_2] = \left]\begin{matrix} < 15 & \varnothing & \{50\} & \neq \\ \geq 15 & \varnothing & \{35\} & \varnothing \\ \varnothing & \varnothing & \{35\} & = \end{matrix}\right[=$$

$$= \begin{bmatrix} < 15 & * & * & * \\ * & * & \{50\} & * \\ * & * & * & \neq \end{bmatrix} \bowtie \begin{bmatrix} \geq 15 & * & * & * \\ * & * & \{35\} & * \end{bmatrix} \bowtie \begin{bmatrix} * & * & \{35\} & * \\ * & * & * & = \end{bmatrix} = \begin{bmatrix} < 15 & * & \{35\} & * \\ \geq 15 & * & \{50\} & = \\ * & * & \{35\} & \neq \end{bmatrix} \quad (1)$$

Диагональной называется квадратная *smart*-таблица, в которой только на главной диагонали могут находиться нефиктивные компоненты. В приведённом выражении из диагональных *smart*-таблиц *C*-типа исключены строки, содержащие пустые компоненты.

Результирующая *smart*-таблица *C*-типа «сжато» описывает множество всех выполняющих подстановок КНФ, которая представлена исходной *smart*-таблицей *D*-типа. Элементарные подстановки легко получить, рассматривая в отдельности каждую строку *smart*-таблицы *C*-типа. Поскольку рассматриваемая *smart*-таблица является дополнением *smart*-таблицы *T*, то она описывает запрещённые комбинации количества товаров и цен. В частности, из первой строки результирующей *smart*-таблицы *C*-типа следует, что товар 1 не может быть закуплен по цене 35 долларов, если его количество меньше 15 штук.

Иллюстрацией того, как выполняется соединение *smart*-таблиц *C*-типа, может служить соединение второй и третьей таблиц в приведённом выражении (1):

$$\begin{bmatrix} \geq 15 & * & * & * \\ * & * & \{35\} & * \end{bmatrix} \bowtie \begin{bmatrix} * & * & \{35\} & * \\ * & * & * & = \end{bmatrix} = \begin{bmatrix} \geq 15 & * & \{35\} & * \\ \geq 15 & * & * & = \\ * & * & \{35\} & * \\ * & * & \{35\} & = \end{bmatrix}.$$

При соединении двух *smart*-таблиц *C*-типа выполняется соединение каждой строки первого операнда с каждой строкой второго операнда. При соединении строк для компонент, соответствующих общим атрибутам обоих операндов, выполняется операция пересечения множеств.

Так решается задача нахождения множества всех выполняющих подстановок КНФ конечного предиката. В таком виде решение задачи оказывается слишком трудоёмким.

Одним из методов ускорения вычислений может служить ортогонализация, основанная на известном соотношении Порецкого: $A \vee B = A \vee \overline{A}B$ [3].

В частности, справедливы соотношения:

$$\begin{bmatrix} \geq 15 & * & * & * \\ * & * & \{35\} & * \end{bmatrix} = \begin{bmatrix} \geq 15 & * & * & * \\ < 15 & * & \{35\} & * \end{bmatrix}; \begin{bmatrix} * & * & \{35\} & * \\ * & * & * & = \end{bmatrix} = \begin{bmatrix} * & * & \{35\} & * \\ * & * & \{50\} & = \end{bmatrix}.$$

Соединение этих двух ортогонализованных *smart*-таблиц *C*-типа имеет вид:

$$\begin{bmatrix} \geq 15 & * & * & * \\ < 15 & * & \{35\} & * \end{bmatrix} \bowtie \begin{bmatrix} * & * & \{35\} & * \\ * & * & \{50\} & = \end{bmatrix} = \begin{bmatrix} \geq 15 & * & \{35\} & * \\ \geq 15 & * & \{50\} & = \\ < 15 & * & \{35\} & * \end{bmatrix}.$$

В результате такого соединения получено меньшее количество строк результирующей таблицы, чем при их соединении без предварительной ортогонализации.

В общем случае порядок соединения диагональных *smart*-таблиц *C*-типа оказывает существенное влияние на скорость процесса вычислений.

Следует отметить, что:

$$\begin{bmatrix} \geq 15 & * & \{35\} & * \\ \geq 15 & * & * & = \\ * & * & \{35\} & * \\ * & * & \{35\} & = \end{bmatrix} = \begin{bmatrix} \geq 15 & * & \{35\} & * \\ \geq 15 & * & \{50\} & = \\ < 15 & * & \{35\} & * \end{bmatrix}.$$

Данные *smart*-таблицы моделируют одно и то же множество элементарных кортежей и в этом смысле являются эквивалентными: одно и то же множество элементарных кортежей может моделироваться *smart*-таблицами *C*-типа, которые внешне могут выглядеть по-

разному. Две эквивалентные *smart*-таблицы *C*-типа могут быть получены одна из другой в результате эквивалентных преобразований. Часть из этих преобразований лежит в основе предлагаемого метода вывода на *smart*-таблицах *C*-типа (метод распространения для случая *smart*-таблиц *C*-типа), целью которого является редукция пространства поиска за полиномиальное время.

Установлены следующие правила редукции для случая *smart*-таблиц *C*-типа, позволяющие удалять отдельные «лишние» значения из доменов атрибутов, компонент *smart*-таблицы, а также исключать из рассмотрения целые строки и столбцы.

Утверждение 1 (У1). Если все строки *smart*-таблицы *C*-типа пусты, то есть содержат хотя бы по одной пустой компоненте каждая, то *smart*-таблица пуста.

Утверждение 2 (У2). Если все компоненты некоторого атрибута (столбца *smart*-таблицы *C*-типа) являются полными, то данный атрибут можно удалить из *smart*-таблицы *C*-типа (удаляются все компоненты, стоящие в соответствующем столбце), а пара «удаляемый атрибут – его домен» сохраняется в векторе частичного решения.

Утверждение 3 (У3). Если домен некоторого атрибута *smart*-таблицы *C*-типа содержит значения, не встречающиеся в соответствующем столбце, то эти значения удаляются из данного домена.

Утверждение 4 (У4). Если строка *smart*-таблицы *C*-типа содержит хотя бы одну пустую компоненту (строка пуста), то строка удаляется.

Утверждение 5 (У5). Если компонента некоторого атрибута *smart*-таблицы *C*-типа содержит значение, не принадлежащее соответствующему домену, то это значение удаляется из компоненты.

Утверждение 6 (У6). Если в строке *smart*-таблицы *C*-типа усечена компонента, соответствующая простому атрибуту, который формирует некоторый составной атрибут, то в этой строке компонента, соответствующая упомянутому составному атрибуту, должна быть изменена с учётом нового значения компоненты простого атрибута.

Утверждение 7 (У7). Если в строке *smart*-таблицы *C*-типа усечена компонента, соответствующая составному атрибуту, то в этой строке должны быть изменены компоненты соответствующих простых атрибутов с учётом вновь выведенной компоненты.

Утверждение 8 (У8). Если в *smart*-таблице *C*-типа усечён домен простого атрибута, который входит в некоторый составной атрибут, то домен составного атрибута должен быть изменён с учётом нового домена простого атрибута.

Утверждение 9 (У9). Если усечён домен составного атрибута, то должны быть изменены и домены соответствующих простых атрибутов с учётом вновь выведенного домена составного атрибута.

Аналогичные правила редукции для случая *smart*-таблиц *D*-типа приведены в [17].

2 Предлагаемый метод

В задачах удовлетворения ограничений, где требуется нахождение всех решений, представляется целесообразным совместно использовать стратегию поиска в ширину и авторские методы распространения нечисловых ограничений. В настоящем исследовании рассматривается случай, когда задача удовлетворения ограничений представляется в виде единственной или нескольких *smart*-таблиц *D*-типа. Предлагаемый гибридный метод основан на представлении *smart*-таблиц *D*-типа в виде соединения нескольких ортогонализованных *smart*-таблиц *C*-типа. *Ортогонализованной* называется *smart*-таблица *C*-типа, все кортежи (строки) которой попарно ортогональны. Метод позволяет определять такую очерёдность пересечения ортогонализованных *smart*-таблиц, которая бы способствовала сокращению вычислений.

На каждом шаге поиска предлагается выбирать пару *smart*-таблиц *C*-типа таким образом, чтобы максимально сократить область поиска. При выборе *smart*-таблиц для соединения используется следующая эвристика, заимствованная из работы [18]:

$$J(K[\boldsymbol{S}], T[\boldsymbol{R}]) = |K[\boldsymbol{S}]| \times |T[\boldsymbol{R}]| \times |\boldsymbol{S}\backslash\boldsymbol{R}| \times |\boldsymbol{R}\backslash\boldsymbol{S}|,$$

где $K[\boldsymbol{S}]$ – первая из пары выбираемых *smart*-таблиц, $T[\boldsymbol{R}]$ – вторая из пары *smart*-таблиц, выбираемых на текущем шаге поиска, $\boldsymbol{S}$ – схема (набор атрибутов) *smart*-таблицы K, $\boldsymbol{R}$ – схема (набор атрибутов) *smart*-таблицы T. Запись $|K[\boldsymbol{S}]|$ обозначает число (элементарных) кортежей данной *smart*-таблицы *C*-типа.

После соединения пары *smart*-таблиц *C*-типа может оказаться, что в каком-либо из столбцов *smart*-таблицы *C*-типа, содержащей результат соединения, отсутствуют некоторые значения соответствующей переменной. Поэтому, после каждого соединения требуется осуществлять процесс распространения ограничений с применением утверждений У1-У8. При исключении «лишних» значений все *smart*-таблицы по-прежнему остаются ортогонализованными.

Значения функции $J(K[\boldsymbol{S}], T[\boldsymbol{R}])$ могут значительно отличаться в зависимости от трактовки понятия «кортеж *smart*-таблицы *C*-типа». Можно в качестве кортежей *smart*-таблицы *C*-типа рассматривать: а) *smart*-кортежи (строки *smart*-таблицы); б) элементарные кортежи, получающиеся при преобразовании *smart*-таблицы в обычную таблицу.

При рассмотрении альтернативы (б) следует иметь ввиду, что, если диагональная *smart*-таблица *C*-типа ортогонализована, то по её виду легко вычислить мощность множества элементарных кортежей данной *smart*-таблицы без разложения её в обычную реляционную таблицу. Для этого нужно перемножить мощности всех непустых компонент, стоящих в одной строке, а результаты перемножений, полученных для каждой из строк, сложить.

Чтобы оценить, какая из трактовок понятия «кортеж *smart*-таблицы *C*-типа» позволяет лучше осуществлять выбор пары соединяемых *smart*-таблиц на шаге поиска, рассмотрен пример.

***Пример* 2.** Пусть имеется *smart*-таблица, заданная в $\boldsymbol{S} = X_1 \times X_2 \times X_3 \times X_4 = \{1, 2, 3, 4, 5\}$:

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array}\left]\begin{array}{cccc} \{2,3\} & \varnothing & \{1,2,3\} & \varnothing \\ \{2\} & \{1,2,5\} & \{4\} & \varnothing \\ \varnothing & \{1,2,4\} & \{3,4\} & \{3,5\} \\ \{1,2\} & \{3,4,5\} & \varnothing & \{3\} \\ \varnothing & \varnothing & \{1,5\} & \{4\} \\ \varnothing & \{2,3\} & \{1,2,3\} & \{1\} \\ \{2,4,5\} & \varnothing & \{3,5\} & \{3,5\} \\ \{5\} & \varnothing & \{2,3,4\} & \{1,3,4\} \\ \{1,3\} & \varnothing & \{4,5\} & \varnothing \end{array}\right[.$$

Данная *smart*-таблица *D*-типа может быть представлена как соединение следующих *smart*-таблиц *C*-типа:

$$K_1[X_1\ X_3] \bowtie K_2[X_1\ X_2\ X_3] \bowtie K_3[X_2\ X_3\ X_4] \bowtie K_4[X_1\ X_2\ X_4] \bowtie K_5[X_3\ X_4] \bowtie K_6[X_2\ X_3\ X_4] \bowtie$$
$$K_7[X_1\ X_3\ X_4] \bowtie K_8[X_1\ X_3\ X_4] \bowtie K_9[X_1\ X_3],$$

где

$$K_1[X_1\ X_3] = \begin{bmatrix} \{2,3\} & * \\ \{1,4,5\} & \{1,2,3\} \end{bmatrix},$$

$$K_2[X_1\,X_2\,X_3] = \begin{bmatrix} \{2\} & * & * \\ \{1,3,4,5\} & \{1,2,5\} & * \\ \{1,3,4,5\} & \{3,4\} & \{4\} \end{bmatrix}, K_3[X_2\,X_3\,X_4] = \begin{bmatrix} \{1,2,4\} & * & * \\ \{3,5\} & \{3,4\} & * \\ \{3,5\} & \{1,2,5\} & \{3,5\} \end{bmatrix},$$

$$K_4[X_1\,X_2\,X_4] = \begin{bmatrix} \{1,2\} & * & * \\ \{3,4,5\} & \{3,4,5\} & * \\ \{3,4,5\} & \{1,2\} & \{3\} \end{bmatrix}, K_5[X_3\,X_4] = \begin{bmatrix} \{1,5\} & * \\ \{2,3,4\} & \{4\} \end{bmatrix},$$

$$K_6[X_2\,X_3\,X_4] = \begin{bmatrix} \{2,3\} & * & * \\ \{1,4,5\} & \{1,2,3\} & * \\ \{1,4,5\} & \{4,5\} & \{1\} \end{bmatrix}, K_7[X_1\,X_3\,X_4] = \begin{bmatrix} \{2,4,5\} & * & * \\ \{1,3\} & \{3,5\} & * \\ \{1,3\} & \{2,4,5\} & \{3,5\} \end{bmatrix},$$

$$K_8[X_1\,X_3\,X_4] = \begin{bmatrix} \{5\} & * & * \\ \{1,2,3,4\} & \{2,3,4\} & * \\ \{1,2,3,4\} & \{1,5\} & \{1,3,4\} \end{bmatrix}, K_9[X_1\,X_3] = \begin{bmatrix} \{1,3\} & * \\ \{2,4,5\} & \{4,5\} \end{bmatrix}.$$

В данном случае произведена ортогонализация, целью которой является подсчёт элементарных кортежей в каждой из перечисленных *smart*-таблиц *C*-типа.

Элементарный шаг поиска заключается в том, что сначала выполняется соединение двух *smart*-таблиц *C*-типа, выбор которых осуществляется согласно приведённой выше эвристике, а затем выполняется процедура распространения ограничений.

Для двух упомянутых интерпретаций понятия «кортеж *smart*-таблицы *C*-типа» значения эвристической функции для каждой пары соединяемых на первом шаге поиска *smart*-таблиц представлены в таблицах 1 и 2.

Таблица 1– Оценка размерности результирующего отношения после выполнения первого шага поиска для варианта с элементарными кортежами

	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9
K_1									
K_2	0								
K_3	4066	9951							
K_4	3838	9393	10807						
K_5	247	2418	0	2626					
K_6	3838	9393	0	10201	0				
K_7	0	9951	11449	10807	0	10807			
K_8	0	10137	11663	11009	0	11009	0		
K_9	0	0	3424	3232	208	3232	0	0	

Таблица 2 – Оценка размерности результирующего отношения после выполнения первого шага поиска для варианта со *smart*-кортежами

	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9
K_1									
K_2	0								
K_3	12	9							
K_4	12	9	9						
K_5	4	12	0	12					
K_6	12	9	0	9	0				
K_7	0	9	9	9	0	9			
K_8	0	9	9	9	0	9	0		
K_9	0	0	12	12	4	12	0	0	

Фактическая размерность результирующего отношения, которое получается при соединении пары *smart*-таблиц, приведена в таблице 3.

Таблица 3 – Фактическая размерность результирующего отношения после соединения пары *smart*-таблиц на первом шаге поиска

	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9
K_1									
K_2	207								
K_3	2056	1644							
K_4	1076	1380	1660						
K_5	141	916	153	1036					
K_6	2404	1444	255	1612	150				
K_7	234	1596	1876	1732	165	1660			
K_8	249	1696	1964	1764	147	1796	279		
K_9	20	180	1384	1312	132	1120	186	204	

Из анализа таблицы 1 и 2 следует, что при обеих интерпретациях понятия «кортеж *smart*-таблицы *C*» значение «ноль» эвристической функции соответствует одним и тем же парам соединяемых *smart*-таблиц. Для вычисления таблицы 1 требуется гораздо больше арифметических операций. Основанием для использования в качестве мощности *smart*-таблицы количества её *smart*-кортежей, а не элементарных кортежей, состоит в том, что минимальному (максимальному) значению в таблице 2 соответствует минимальное (максимальное) значение в таблице 3.

С применением предложенной эвристики выбраны для соединения *smart*-таблицы, имеющие одинаковые схемы: $K_1[X_1\ X_3]$ и $K_9[X_1\ X_3]$. Результат их соединения представлен так:

$$K_1[X_1\ X_3] \bowtie K_9[X_1\ X_3] = \begin{bmatrix} \{2,3\} & * \\ \{1,4,5\} & \{1,2,3\} \end{bmatrix} \bowtie \begin{bmatrix} \{1,3\} & * \\ \{2,4,5\} & \{4,5\} \end{bmatrix} = \begin{bmatrix} \{3\} & * \\ \{2\} & \{4,5\} \\ \{1\} & \{1,2,3\} \end{bmatrix}.$$

Согласно утверждениям У1-У8 можно «сузить» домен переменной X_1 до множества {1, 2, 3}, «просуммировав» значения первого столбца *smart*-таблицы $K_1[X_1\ X_3] \bowtie K_9[X_1\ X_3]$.

Теперь следует произвести «настройку» остальных *smart*-таблиц *C*-типа (K_1 – K_8) на новый домен переменной X_1. На текущем шаге поиска получаются следующие домены переменных и набор *smart*-таблиц.

Домены: X_1 – {1, 2, 3}, X_2 – {1, 2, 3, 4, 5}, X_3 – {1, 2, 3, 4, 5}, X_4 – {1, 2, 3, 4, 5}.

Ограничения:

$$K_1[X_1\ X_3] \bowtie K_9[X_1\ X_3] = \begin{bmatrix} \{1,3\} & \{1,2,3\} \\ \{2,3\} & \{4,5\} \end{bmatrix},$$

$$K_2[X_1\ X_2\ X_3] = \begin{bmatrix} \{2\} & * & * \\ \{1,3\} & \{1,2,5\} & * \\ \{1,3\} & \{3,4\} & \{4\} \end{bmatrix}, K_3[X_2\ X_3\ X_4] = \begin{bmatrix} \{1,2\} & * & * \\ \{3\} & \{3,4\} & * \\ \{3\} & \{1,2,5\} & \{3,5\} \end{bmatrix},$$

$$K_4[X_1\ X_2\ X_4] = \begin{bmatrix} \{1,2\} & * & * \\ \{3\} & \{3,4,5\} & * \\ \{3\} & \{1,2\} & \{3\} \end{bmatrix}, K_5[X_3\ X_4] = \begin{bmatrix} \{1,\} & * \\ \{2,3\} & \{4\} \end{bmatrix},$$

$$K_6[X_2\ X_3\ X_4] = \begin{bmatrix} \{2,3\} & * & * \\ \{1\} & \{1,2,3\} & * \\ \{1\} & \{4,5\} & \{1\} \end{bmatrix},$$

$$K_7[X_1\ X_3\ X_4] = \begin{bmatrix} \{2\} & * & * \\ \{1,3\} & \{3,5\} & * \\ \{1,3\} & \{2,4,5\} & \{3,5\} \end{bmatrix},$$

$$K_8[X_1\ X_3\ X_4] = \begin{bmatrix} \{1,2,3\} & \{2,3,4\} & * \\ \{1,2,3\} & \{1,5\} & \{1,3,4\} \end{bmatrix}.$$

В *smart*-таблицах K_2-K_7 были откорректированы первые столбцы, в *smart*-таблице K_8 удалена первая строка. Дальнейшие шаги процесса поиска выполняются по аналогии.

Заключение

В работе предложен поиск всех решений задачи удовлетворения ограничений, представленной с использованием *smart*-таблиц *D*-типа, и сводится к определению порядка соединения нескольких ортогонализованных *smart*-таблиц *C*-типа. Использована эвристика для выбора на каждом шаге поиска пары соединяемых *smart*-таблиц *C*-типа. Рассмотрены два варианта использования данной эвристики. Показано, что при определении размерности *smart*-таблицы целесообразно использовать количество её *smart*-кортежей, а не количество элементарных кортежей отношения. Предложены правила редукции для случая *smart*-таблиц *C*-типа.

В случае алгоритмов поиска в глубину с возвратами их временная сложность оценивается как произведение доменов всех переменных. Временная сложность предложенного метода оценивается как произведение, в котором каждый множитель равен количеству непустых компонент соответствующей строки исходной *smart*-таблицы *D*-типа. Предложенный метод даёт значительный выигрыш при большом количестве переменных, высокой размерности их доменов и сравнительно малом количестве строк *smart*-таблицы *D*-типа.

Разработанный метод позволяет эффективно находить все решения задач удовлетворения ограничений без разложения табличных ограничений в элементарные кортежи и легко адаптируется для решения оптимизационных задач. Предложенный метод ориентирован на применение в интеллектуальных системах, предназначенных для слабо формализованных предметных областей.

Благодарности

Исследование выполнено при финансовой поддержке РФФИ в рамках научных проектов № 20-07-00708-а, 19-07-00359-а.

Список источников

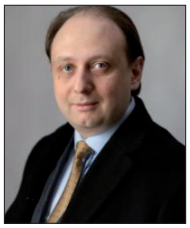
[1] ***Russel, S.*** Artificial Intelligence: A Modern Approach. 3rd edition / S. Russel, P. Norvig. – Prentice Hall, 2010. – 1132 p.

[2] ***Mackworth, A.*** Consistency in networks of relations / A. Mackworth // Artificial Intelligence. – 1977. – 8(1). – P. 99-118. DOI: 10.1016/0004-3702(77)90007-8.

[3] ***Bartak, R.*** Constraint Programming: In Pursuit of the Holy Grail / R. Bartak // Proceedings of the Week of Doctoral Students (WDS99), Part IV. – Prague: MatFyzPress, 1999. P.555–564.

[4] ***Charlier, B.*** Automatic Synthesis of Smart Table Constraints by Abstraction of Table Constraints / B. Charlier, M. Khong, C. Lecoutre, Y. Deville // Proceedings of IJCAI 2017. 2017. P. 681-687. DOI: https://doi.org/10.24963/ijcai.2017/95.
[5] ***Audemard, G.*** Segmented Tables: An Efficient Modeling Tool for Constraint Reasoning / G. Audemard, C. Lecoutre, M. Maamar // Proceedings of ECAI 2020. 2020. P.315-322.
[6] ***Yap, R.*** Generalized Arc Consistency Algorithms for Table Constraints: A Summary of Algorithmic Ideas / R. Yap, W. Wang // Proceedings of AAAI 2020. 2020. P.13590-13597. DOI: https://doi.org/10.1609/aaai.v34i09.7086.
[7] ***Perez, G.*** Improving GAC-4 for table and MDD constraints / G. Perez, J.C. Regin // CP 2014. LNCS. 2014. 8656. P.606-621. DOI: http://dx.doi.org/10.1007/978-3-319-10428-7_44.
[8] ***Verhaeghe, H.*** Extending compact-table to negative and short tables / H. Verhaeghe, C. Lecoutre, P. Schaus // Proceedings of AAAI 17. 2017. P.3951-3957. DOI: https://dl.acm.org/doi/abs/10.5555/3298023.3298142.
[9] ***Ingmar, L.*** Making Compact-Table Compact / L. Ingmar, C. Schulte // Proceedings of CP 2018, Lecture Notes in Computer Science. 2018. Vol.11008. P.210-218. DOI: https://doi.org/10.1007/978-3-319-98334-9_14.
[10] ***Cheng, K.*** An MDD-based generalized arc consistency algorithm for positive and negative table constraints and some global constraints / K. Cheng, R. Yap // Constraints. 2010. 15(2). P.265-304. DOI: http://dx.doi.org/10.1007/s10601-009-9087-y.
[11] ***Jefferson, C.*** Extending simple tabular reduction with short supports / C. Jefferson, P. Nightingale // Proceedings of IJCAI 2013, 2013. P.573-579.
[12] ***Mairy, J.*** The Smart Table Constraint / J. Mairy, Y. Deville, C. Lecoutre // In: Michel, L. (eds.) Integration of AI and OR Techniques in Constraint Programming. CPAIOR 2015. Lecture Notes in Computer Science. – Springer. Cham, 2015. Vol.9075. P.271-287.
[13] ***Verhaeghe, H.*** Extending Compact-Table to Basic Smart Tables. Principles and Practice of Constraint Programming / H. Verhaeghe, C. Lecoutre, Y. Deville, P. Schaus // CP 2017, Lecture Notes in Computer Science. 2017. 10416. P.297-307. DOI: http://dx.doi.org/10.1007/978-3-319-66158-2_19.
[14] ***Schneider, A.*** PW-CT: Extending Compact-Table to Enforce Pairwise Consistency on Table Constraints / A. Schneider, B. Choueiry // CP 2018, Lecture Notes in Computer Science. 2018. 11008. P.345-361. DOI: https://doi.org/10.1007/978-3-319-98334-9_23.
[15] ***Zuenko, A.*** Development of n-tuple algebra for logical analysis of databases with the use of two-place predicates / A. Zuenko, A. Fridman // Journal of Computer and Systems Sciences International. 2009. Vol.48(2). P.254-261. DOI: http://dx.doi.org/10.1134/S1064230709020099.
[16] ***Zuenko, A.*** Local Search in Solution of Constraint Satisfaction Problems Represented by Non-Numerical Matrices / A. Zuenko // Proceedings of the 2nd International Conference on Computer Science and Application Engineering (CSAE '18) 2018. 45. DOI: 10.1145/3207677.3277959.
[17] ***Зуенко А.А.*** Компактное представление ограничений на основе новой интерпретации понятия «кортеж многоместного отношения» / А.А. Зуенко // Онтология проектирования. 2020. Т.10. №4(38). С.503-515. DOI: 10.18287/2223-9537-2020-10-4-503-515.
[18] ***Møller, G.*** On the Technology of Array-Based Logic. / G. Møller // Ph. D. thesis. – 1995. – http://www.arraytechnology.com/documents/lic.pdf.</bib>.

Сведения об авторе



Зуенко Александр Анатольевич, 1983 г.р., кандидат технических наук, ведущий научный сотрудник Института информатики и математического моделирования – обособленного подразделения ФИЦ «Кольский научный центр Российской академии наук». Области научных интересов: программирование в ограничениях; моделирование слабо формализованных предметных областей. Author ID (RSCI): 528493; Author ID (Scopus): 26536974000; Researcher ID (WoS): E-7944-2017. *zuenko@iimm.ru.*

Поступила в редакцию 08.12.2021, после рецензирования 17.12.2021. Принята к публикации 22.12.2021.

Breadth First Search and Inference Methods Integration to satisfy table constraints

A.A. Zuenko

Institute for Informatics and Mathematical Modeling – Subdivision of the Federal Research Centre "Kola Science Centre of the Russian Academy of Sciences", Apapity, Russia

Abstract

Within the Constraint Programming technology, so-called table constraints such as typical tables, compressed tables, smart tables, segmented tables, etc, are widely used. They can be used to represent any other types of constraints, and algorithms of the table constraint propagation (logical inference on constraints) allow eliminating a lot of "redundant" values from the domains of variables, while having low computational complexity. In the previous studies, the author proposed to divide smart tables into structures of *C*- and *D*-types. The generally accepted methodology for solving constraint satisfaction problems is the combined application of constraint propagation methods and backtracking depth-first search methods. In the study, it is proposed to integrate breadth-first search methods and author`s method of table constraint propagation. D-type smart tables are proposed to be represented as a join of several orthogonalized C-type smart tables. The search step is to select a pair of C-type smart tables to be joined and then propagate the restrictions. To determine the order of joining orthogonalized smart tables at each step of the search, a specialized heuristic is used, which reduces the search space, taking into account further calculations. When the restrictions are extended, the acceleration of the computation process is achieved by applying the developed reduction rules for the case of C-type smart tables. The developed hybrid method allows one to find all solutions to the problems of satisfying constraints modeled using one or several D-type smart tables, without decomposing tabular constraints into elementary tuples.

Key words: *constraint programming, smart tables, breadth first search, constraint propagation, constraint satisfaction problem.*

Citation: *Zuenko AA.* Breadth First Search and Inference Methods Integration to satisfy table constraints [In Russian]. *Ontology of designing*. 2021; 11(4): 521-532. DOI: 10.18287/2223-9537-2021-11-4-521-532.

Acknowledgment: This work was supported by the Russian Foundation for Basic Research (RFBR), project numbers 20-07-00708-a, 19-07-00359-a.

List of tables

Table 1 – The estimation of the size of the resulting relation after the first step of the search for the case of elementary tuples
Table 2 – The estimation of the size of the resulting relation after the first step of the search for the case of smart tuples
Table 3 - The actual size of the resulting relation after joining a pair of smart tables at the first step of the search

References

[1] ***Russel S, Norvig P.*** Artificial Intelligence: A Modern Approach. 3rd ed. Prentice Hall, 2010. 1132 p.
[2] ***Mackworth A.*** Consistency in networks of relations. Artificial Intelligence. 1977; 8(1): 99-118. DOI: 10.1016/0004-3702(77)90007-8.
[3] ***Bartak R.*** Constraint Programming: In Pursuit of the Holy Grail. Proc. of the Week of Doctoral Students (WDS99), Part IV. Prague: MatFyzPress, 1999: 555–564.
[4] ***Charlier B, Khong M, Lecoutre C, Deville Y.*** Automatic Synthesis of Smart Table Constraints by Abstraction of Table Constraints. Proc. of IJCAI 2017, 2017: 681-687. DOI: https://doi.org/10.24963/ijcai.2017/95.
[5] ***Audemard G, Lecoutre C, Maamar M.*** Segmented Tables: An Efficient Modeling Tool for Constraint Reasoning. Proc. of the ECAI 2020, 2020: 315-322.
[6] ***Yap R, Wang W.*** Generalized Arc Consistency Algorithms for Table Constraints: A Summary of Algorithmic Ideas. Proc. of the AAAI 2020, 2020: 13590-13597. DOI: https://doi.org/10.1609/aaai.v34i09.7086.
[7] ***Perez G, Regin JC.*** Improving GAC-4 for table and MDD constraints. CP 2014. LNCS, 2014; 8656: 606-621. DOI: http://dx.doi.org/10.1007/978-3-319-10428-7_44.

[8] ***Verhaeghe H, Lecoutre C, Schaus P.*** Extending compact-table to negative and short tables. Proc. of AAAI 17, 2017: 3951-3957. DOI: https://dl.acm.org/doi/abs/10.5555/3298023.3298142.
[9] ***Ingmar L, Schulte C.*** Making Compact-Table Compact. CP 2018, LNCS, 2018; 11008: 210-218. DOI: https://doi.org/10.1007/978-3-319-98334-9_14.
[10] ***Cheng K, Yap R.*** An MDD-based generalized arc consistency algorithm for positive and negative table constraints and some global constraints. Constraints, 2010; 15(2): 265-304. DOI: http://dx.doi.org/10.1007/s10601-009-9087-y.
[11] ***Jefferson C, Nightingale P.*** Extending simple tabular reduction with short supports. Proc. of IJCAI 2013. 2013: 573-579.
[12] ***Mairy J, Deville Y, Lecoutre C.*** The Smart Table Constraint. Integration of AI and OR Techniques in Constraint Programming. CPAIOR 2015. LNCS, 2015; 9075: 271-287.
[13] ***Verhaeghe H, Lecoutre C, Deville Y, Schaus P.*** Extending Compact-Table to Basic Smart Tables. Principles and Practice of Constraint Programming. CP 2017. LNCS, 2017; 10416: 297-307. DOI: http://dx.doi.org/10.1007/978-3-319-66158-2_19.
[14] ***Schneider A, Choueiry B.*** PW-CT: Extending Compact-Table to Enforce Pairwise Consistency on Table Constraints. CP 2018. LNCS, 2018; 11008: 345-361. DOI: https://doi.org/10.1007/978-3-319-98334-9_23.
[15] ***Zuenko A, Fridman A.*** Development of n-tuple algebra for logical analysis of databases with the use of two-place predicates. Journal of Computer and Systems Sciences International. 2009; 48(2): 254-261. DOI: http://dx.doi.org/10.1134/S1064230709020099.
[16] ***Zuenko A***. Local Search in Solution of Constraint Satisfaction Problems Represented by Non-Numerical Matrices. Proc. of the 2nd Int. Conf. on Computer Science and Application Engineering (CSAE '18), 2018: 45. DOI: 10.1145/3207677.3277959.
[17] ***Zuenko A.A.*** Compact representation of constraints based on a new interpretation of the concept “tuple of a multi-place relation” [In Russian]. *Ontology of designing*. 2020; 10(4): 503-515. DOI: 10.18287/2223-9537-2020-10-4-503-515.
[18] ***Møller G.*** On the Technology of Array-Based Logic. Ph. D. thesis, 1995; http://www.arraytechnology.com/documents/lic.pdf.

About the author

Alexander Anatolievich Zuenko, (b. 1983) PhD, a leading researcher of Institute for Informatics and Mathematical Modeling – Subdivision of the Federal Research Centre “Kola Science Centre of the Russian Academy of Sciences”. Research interests: constraint programming; poorly formalized subject domains modeling. Author ID (RSCI): 528493; Author ID (Scopus): 26536974000; Researcher ID (WoS): E-7944-2017. *zuenko@iimm.ru*.

Received December 08, 2021. Revised December 17, 2021. Accepted December 22, 2021.