

ОЦЕНКА ЭФФЕКТИВНОСТИ ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЫ ДЛЯ СОЗДАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ САПР КАК СИСТЕМЫ С БАЗОЙ ЗНАНИЙ

Григорьев А.В.

Донецкий национальный технический университет

Describes a knowledge conception method specific character. Consider the teaching processes organization complication estimation questions and inference in knowledge base.

Введение. Ранее автором в работах [1-6] были изложены основные принципы нового метода представлений знаний в базе знаний инструментальной оболочки для построения интеллектуальных САПР – метаэвристической оболочки (МЭО). Данный метод представления знаний имеет ряд специфических особенностей:

- использование контекстно-свободных грамматик как основы представления знаний;
- сужение грамматик до уровня И-ИЛИ-деревьев;
- наличие продукций работы с образцами, рассматриваемыми как КС грамматики;
- наличие продукций, задающих семантические зависимости в КС грамматиках;
- организация обучения, вывода и процесса уточнения задачи синтеза на основе теоретико-множественных операций (ТМО) над КС порождающими грамматиками.

Завершенность работ по созданию алгоритмов выполнения теоретико-множественных операций над КС грамматиками, модели построения интеллектуальных САПР с выбранным способом представления знаний, а так же ряда смежных работ, делают актуальной задачу оценки эффективности работы интеллектуальных САПР с данным методом представления знаний.

Цель данной работы является:

- анализ существующих оценок качества систем с базами данных;
- выбор или построение собственных оценок;
- формирование вывода о качестве баз знаний и определение перспективных работ.

1. Специфика представления знаний в МЭО

Главная специфика представления знаний в МЭО - построение базы знаний, работающий с формально-языковым представлением моделей, т.е. работа с базой знаний текстов, представленных

грамматиками с определенными над ними дополнительными продукциями. Предложена следующая модификация продукционного метода представления знаний:

1) Факты:

- совокупностью фактов, т.е. основой представления знаний являются контекстно-свободные порождающие грамматики, суженные до уровня И-ИЛИ-дерева, описывающие множество возможных текстов-решений (прототипов или технических заданий - ТЗ), определенных в рамках модели САПР решения типичных задач проектирования;

- почему грамматики сужены до И-ИЛИ-дерева: для грамматик, имеющих циклические ссылки в синтаксических правилах вывода, практически невозможно разработать корректный алгоритм ТМО (в общем случае, из-за теоретической невозможности таких ТМО [7]);

- ТМО над грамматиками нужны для обеспечения: вывода, обучения, уточнения постановки задачи;

2) Правила вывода:

- система правил вывода - семантические продукции над ИЛИ-узлами контекстно-свободных порождающих грамматик, суженных до И-ИЛИ-дерева;

- назначение продукции – сузить число возможных комбинаций альтернатив ИЛИ-узлов до семантически верных, описывающих имеющие смысл прототипы;

- модальность правил вывода над множеством фактов (т.е. текстов описания модели) - существует (соответствует требованиям проектировщика) или - не существует (не соответствует требованиям);

- предполагается деление правил вывода на явные и - неявные;

- неявные правила вывода, оформлены в форме списков прототипов, связанных с терминальными и нетерминальными символами грамматик; формируются автоматически при обучении базы знаний; такие грамматики представляют собой модифицированную предикативную (модифицировано-атрибутную) грамматику, принятую в ПРОЛОГе;

- явные правила вывода, оформлены в виде продукции над ИЛИ-узлами контекстно-свободных порождающих грамматик, суженных до И-ИЛИ-дерева, формируются экспертом по знаниям;

3) Процесс вывода в пределах модуля знаний:

- процесс вывода – сужение множества возможных решений, описанных в модуле знаний - И-ИЛИ-дерева, описывающим

множество возможных решений, в соответствии с рядом требований, задаваемых конечным пользователем (проектировщиком);

4) Работа с базами знаний как с моделями предметных областей:

- имеются операции над базой знаний – продукционный вывод, обучение, формирование требуемой базы знаний;

- обучение – объединение грамматик с грамматиками либо отдельными прототипами, с сохранением или без сохранения имеющихся семантических связей (правил вывода), заданных в одной из двух форм – явной и неявной;

- формирование требуемой базы знаний - система обобщенных макро-продукций, использующих теоретико-множественные операции над базами знаний, поддерживающих диалог с пользователем на предмет выявления требуемой ему предметной области.

Обсудим типичные и специфические черты предлагаемого метода представления знаний:

1) Типичные черты:

- И-ИЛИ-дерево с точки зрения логики – обычный сложный предикат (дерево предикатов, дерево вывода, дерево решений);

- альтернативы в ИЛИ-узлах И-ИЛИ-дерева есть отдельные факты, истинные или ложные;

- И-узлы объединяют факты по И;

- правила вывода также объединяют предикаты (ИЛИ-узлы) по форме «если посылка истина, то истинен вывод»;

- т.о., с точки зрения логики, продукционного вывода, ничего необычного нет;

- в обычной логике пользователь сам определяет правила вывода, связывающие факты в соответствии с известной ему семантикой модели мира.

2) Специфические черты

Специфика нашей семантики модели мира:

- имеется явное разделение фактов на два класса: а) признаки, входящие в ТЗ; б) признаки, характеризующие тексты-прототипы; соответственно, может рассматриваться либо два взаимосвязанных И-ИЛИ-дерева, либо одно, но совмещающее факты двух классов;

- признаки, входящие в ТЗ, ориентированы на диалоговый вариант выбора условия;

- наряду с «обычными» рассматриваются «обобщенные» факты, как некоторая имеющая смысл комбинация частных, «обычных» фактов, т.е. терминальных текстов-прототипов; такая комбинация

задает, в зависимости от типа дерева, либо полное ТЗ, либо полное описание объекта, т.е. текст-прототип;

- любой «обычный» факт, входящий в И-ИЛИ-дерево, трактуется либо как принадлежность, либо – как не принадлежность данной особенности текста к определенному прототипу или – к ТЗ; т.о. факт определяет значение свойства прототипа;

- принадлежность признаков прототипам может определяться пользователем-экспертом по знаниям, например, определяя совместные и – несовместные признаки; принадлежность может определяться автоматически при обучении, путем прямого указания принадлежности данных свойств тем или иным прототипам.

Мы работаем с признаками как обычными предикатами, исключение – предикаты организации диалога синтеза (наличие ответов «все нет», «все да», «часть верна»).

Вывод: можно работать с И-ИЛИ-деревом как с обычным деревом вывода и оценивать сложность как сложность поиска решений по дереву предикатов.

Анализ дерева вывода и И-ИЛИ-дерева:

- обычное И-ИЛИ-дерево вывода есть совмещенное пространство обликов систем (ПОС), т.е. условия выбора в И-ИЛИ-дереве, с элевым пространством систем (ЦПС); ЦПС объединяет терминальные вершины, где описаны либо требуемые решения, объединенные по ИЛИ, либо части решений, объединенные по И; такое И-ИЛИ-дерево, где терминальными вершинами являются решения, обычно строят вручную;

- И-ИЛИ-дерево вывода, такое, как в ПРОЛОГЕ [8,9] или как у Донского, Войтицкого [10-12] - есть предикативное И-ИЛИ-дерево условий выбора, на терминальной вершине такого дерева реально нет описания объекта, оно только подразумевается или может затем отдельно строиться; в ПРОЛОГЕ автоматически строится данное И-ИЛИ-дерево решений для поиска в нем цели, но – это автоматическое построение не прототипов, а условий к ним, что бы сравнить их с целью; результат такого построения – истина или ложь по отношению к цели;

- недостатком как обычного, так и предикативного И-ИЛИ-дерева вывода является проблема автоматического построения прототипа по набору признаков, описанных выше в И-ИЛИ-дереве; т.к. форма представления и детальность описания должна быть такая, что бы обеспечить интерпретацию модели в некоторой имитационной системе моделирования, т.е. описание должно быть формальным и

полным; при ручном построении дерева данные требования трудно обеспечить.

Оценка процессов вывода, обучения и уточнения постановки задачи в данном случае эквивалентна оценке сложности выполнения ТМО над КС порождающими грамматиками. Т.о., мы должны оценить сложность алгоритмов ТМО, принятых в МЭО.

2. Краткая характеристика алгоритмов выполнения теоретико-множественных операций, принятых в МЭО

Теоретико-множественные операции над грамматиками:

- на входе – любые две грамматики, соответствующие вышеописанным условиям, заданные своим начальным символом;
- на выходе ряд новых грамматик, содержащих только оригинальные идентификаторы новых синтермов; они предназначены для идентификации составляющих старых синтермов, вошедших как в общую часть, так и в каждую из двух отличных частей.

Результатом так же являются все комбинации объединения общей и двух отличных частей.

Характеристики алгоритма ТМО:

- задача алгоритма – сравнить две грамматики и построить ряд новых грамматик, соответствующих результатам ТМО – объединение, разница, дополнение, пересечение;
- алгоритм выполняет сравнение грамматик как двух множеств слов, т.е. текстов, представленных в сравниваемых грамматиках;
- две грамматики имеют разное назначение – первая это ведущая, и вторая - ведомая;
- ведущая грамматика просматривается по всем возможным ветвям, а – ведомая просматривается многократным проходом вперед и назад с целью поиска совпавших слов или групп слов, заданных синтермами в текущем месте ведущей грамматики;
- при сравнении алгоритм делит исходные синтермы в обеих грамматиках на две части – общая (совпавшая), и - отличная, не совпавшая;
- создает новые оригинальные имена новых синтермов, как совпавших так и не совпавших (амперсенд четный и – амперсенд нечетный);
- обход обеих грамматик выполняется слева - направо и сверху (от корня) – вниз (к ветвям);
- представляет собой многократный проход по первому (ведущему) И-ИЛИ-дереву с целью нахождения соответствующих компонент во втором дереве;

- как для ведомого, так и для ведущего дерева есть прямой и обратный ход алгоритма;

- прямой ход – проверка некоторой ветви дерева;

- обратный – возврат к месту ветвления с параллельным формированием четного и нечетного амперсендов в каждом из пройденных ранее синтермов;

- прямой ход – разворачивание синтермов на компоненты по ходу просмотра;

- обратный ход – сворачивание развернутых ранее синтермов, при автоматическом создании новых синтермов – совпавшей и – не совпавшей части, на которые разбиваются «старые» ИЛИ-синтермы; каждая новая часть содержит подмножество альтернатив из старого ИЛИ-синтерма;

- используются стеки для хранения цепочки разложенных синтермов; знак «&» или «\» в верхушке стека имеет смысл граничного символа разложения;

- синтерм считается не совпавшим, если хотя бы один его компонент не совпал;

- не совпавшие синтермы раскрываются далее, с целью поиска либо совпадающих синтермов, либо с целью деления нижележащего множества цепочек терминальных символов на совпавшую и не совпавшую часть;

Алгоритм является модификацией стандартного алгоритма грамматического разбора LL(1), в котором:

- на входе одна цепочка текста длиной N символов;

- одна читающая головка над лентой входной лентой;

- грамматика – это правила вывода, управляющие движением головки;

- на выходе – ответ «принадлежит ли данная цепочка к множеству синтаксически верных выражений, т.е. правильная она – или ошибочная».

Сложность измеряет:

- число поворотов головки;

- общее число просмотренных символов во входной ленте;

В нашем случае имеется две грамматики, и соответственно:

- две головки;

- две ленты - текущие;

- общее число лент с двух сторон – размер языков.

3. Оценка базы качества базы знаний, построенной на основе предлагаемого метода представления знаний

Все известные меры, которые возможно применить для оценки качества систем с базами знаний могут быть поделены на две группы:

1) Меры оценки механизмов логического вывода и структуры БЗ включают фактуальные оценки и технологические меры; например, фактуальные оценки включают: сложность БЗ, информативность БЗ, надежность вывода решения, достоверность выведенного решения, устойчивость БЗ, быстродействие системы представления и обработки знаний и т.д.

2) Меры оценки критериальных свойств для разработчиков - когнитологов и экспертов: релевантность знаний (их подтвержденность и достаточность), полнота умений (необходимость и достаточность процедур), проверенность знаний (их тестированность и целостность), уровень интеллектуальности (обучаемость, гибкость стратегий рассуждения и интерфейса), наличие мета-знаний и т.д.

Следует отметить громоздкость рассмотренной системы оценки качества БД. Вслед за [10-12] мы будем рассматривать только главные выходные показатели эффективности вывода в базе знаний, задающие эффективность D-алгоритма синтеза логической области дедуктивной выводимости (ЛОДВ), соответствующего по структуре нашему алгоритму, а именно:

- $L(S)$ – длину логического описания ОДВ (число конъюнктов в ДНФ ЛОДВ);

- $M(S)$ – число замещений литералов в процессе работы алгоритма.

При этом, $L(S)$ задает пространственную сложностью, $M(S)$ – временную сложностью алгоритма. Временная и пространственная сложность вывода есть функции, зависящие от аргументов, являющимися структурными показателями сложности базы знаний:

- $H(S)$ – высота И-ИЛИ-дерева, т.е. наибольший ярус дерева (ярус целевой вершины полагается нулевым).

- $N(S)$ – число продукций в ЛСП S ;

- $q(S)$ – число нетерминальных литералов в ЛСП S , увеличенное на 1;

- $x(s)$ – число терминальных литералов ЛСП S ;

- k_v - наибольшая степень ИЛИ-вершин И-ИЛИ-дерева, соответствующего ЛСП S ;

- $k_{\&}$ - наибольшая степень И-вершин И-ИЛИ-дерева, соответствующего ЛСП S .

Длина ЛОДВ, построенного D-алгоритмом (ленточная сложность), не превысит

$$L_D(S) \leq k_{\vee}^{k_{\&}} * k_{\vee}^{(q(S)-H(S)+1)*(H(S)-2)} \quad (1)$$

Общее число построенных в процессе работы D-алгоритма конъюнктов не превысит (временная сложность):

$$M_D(S) \leq k_{\vee} + k_{\vee}^{k_{\&}} * k_{\vee}^{\frac{k_{\vee}^{(q(S)-H(S)+1)*(H(S)-1)} - 1}{k_{\vee} - 1}} \quad (2)$$

Т.о., в [10-12] рассматривается типичное предикативное дерево, т.е. дерево условий. Крона – терминальные символы, могущие быть истинными или ложными. Все прочее – множество выражений. Имеются конечные, упорядоченные множества символов: терминальные, целевые (верхние), нетерминальные. Терминальные символы – так же упорядочены. Аксиомы – подмножества упорядоченных терминальных символов, имеющих значение 0 или 1. Правила вывода: если верно некоторое множество выражений над символами (по И), то верно выражение для данного отдельно взятого символа. Теорема – это доказуемая аксиома. Задается как проверка значения 0 или 1 некоторого символа (целевого, нецелевого или терминального). Теоремы и аксиомы – это факты. Доказать теорему – означает показать, что все факты (по И), входящие в терему – истинны.

В грамматической трактовке пространственная сложность – это ленточная сложность, временная сложность – так же временная сложность. В нашем случае - терминальный символ – это часть текста. Значение – 1, если он есть в цепочке, и - если 0, то его просто нет. Т.о., любая цепочка символов потенциально включает все терминальные символы, одни со значением 1 (есть), другие – со значением 0 (нет). Терминальные символы – упорядочены по кроне. Правило вывода – синтерм по И для правого символа (целевого, нетерминального, терминального). Альтернативные правила вывода для одного и того же правого символа – синтерм по ИЛИ. Имя правила вывода – имя синтерма по И. Имя правого символа – имя синтерма по ИЛИ, включает альтернативные правила вывода. Т.о., имеем И-ИЛИ-дерево. Доказать теорему – означает показать, что существует (не существует) данная цепочка терминальных символов. Доказательство может быть прямым – от аксиом к целям, и обратным – от цели к аксиомам. Список «открыт» в D-алгоритме синтеза ЛОДВ – это раскрытие синтерма, сначала по ИЛИ (если он ИЛИ), а затем по И (если он И). Закрытие списка – аналог сворачивания

просмотренного синтерма. Весь D-алгоритм – аналог лексического анализа, когда отыскивается некоторая фраза в грамматике. Если успех – теорема доказана, не успех – теорема не доказана. Совокупность правил – И-ИЛИ-дерево. Нетерминальный символ – подмножество И-ИЛИ-дерева. Если рассматривается нетерминальный символ, то это подмножество слов (имеют место ИЛИ узлы). Вопрос ставится так – есть ли хоть одна фраза из набора, заданного данным И-ИЛИ-деревом, соответствующим данному нетерминальному символу. Можно рассчитывать сложность как число операций над стеками – свернуть и развернуть.

Тогда рассмотренный выше подход может трактоваться как оценка простейшего случая ТМО над контекстно-свободными (КС) порождающими грамматиками, а именно - пересечение отдельно взятого текста с порождающей КС грамматикой. Оценки сложности для случая сравнения двух грамматик можно построить, исходя из следующих соображений. В наиболее сложном случае, когда пересечение двух грамматик пусто, процесс проверки можно трактовать как проверку всех возможных текстов из одной грамматики со второй грамматикой. Введем обозначения:

N_1 – число текстов в первой грамматике;

N_2 – число текстов во второй грамматике;

$K(N_2)$ – сложность проверки одного текста из первой грамматики со второй грамматикой;

$K(N_1, N_2)$ - суммарная сложность проверки первой грамматики со второй грамматикой;

Тогда возможно дать следующую приблизительную оценку сложности сравнения двух грамматик:

$$K(N_1, N_2) \geq N_1 * K(N_2).$$

Тут $K(N_2)$ предполагается эквивалентной оценке сложности вывода из работ Донского и Войтицкого, т.е. соответствует соотношениям (1) и (2).

Заключение. В работе получены оценки качества баз знаний, построенных на основе предложенного метода представления знаний, заданные временной и ленточной сложностью, выраженных оценками (1) и (2). Следует отметить, что оценка (1) как и оценка (2) экспоненциально зависит от максимальной степени вершин типа «И» и высоты И-ИЛИ-графа, а оценка (2) еще экспоненциально зависит и от числа нетерминальных символов.

Т.о., сложность вывода в предлагаемом методе представления знаний является экспоненциальным, что является малоприменимым для

практического применения в условиях больших баз знаний, что является недостатком данного метода.

Однако соотношения (1) и (2) могут рассматриваться как источник рекомендаций по ограничению структурных параметров логической системы продукций в целях построения базы знаний, допускающей эффективное решение задачи синтеза вывода данного факта. Следовательно, актуальной является задача построения средств и методов снижения сложности вывода в условиях предлагаемого метода представления знаний.

Литература

1. Григорьев А.В. Теоретико-множественные операции над грамматиками как механизм работы со знаниями в интеллектуальных САПР. Вісник Східноукраїнського національного університету імені Володимира Даля, N 2(48). Луганск, ВУТУ, 2002. С. 186-194.

2. Григорьев А.В. Алгоритм выполнения теоретико-множественных операций над грамматиками в среде специализированной оболочки для создания интеллектуальных САПР. Наукові праці національного технічного університета. Серія «Проблеми моделювання і автоматизації проектування динамічних систем» (МАП -2002). Випуск 52: Донецьк: ДонНТУ, 2002. - С.83-93.

3. Григорьев А.В., Каспаров А.А. И/ИЛИ-дерево как средство абстрактного представления знаний. Наукові праці національного технічного університета. Серія «Інформатика, кібернетика та обчислювальна техніка». Випуск 39: Донецьк: ДонНТУ, 2002. - С.36-42.

4. Григорьев А.В. Семиотическая модель базы знаний САПР. Научные труды Донецкого государственного технического университета Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 10: - Донецьк, ДонГТУ, 1999. - С. 30-37.

5. Григорьев А.В. Принципы организации вывода решений в базе знаний инструментальной оболочки для создания интеллектуальных САПР. // Практика і перспективи розвитку інституційного партнерства». Вісник ДонГТУ – ТРТУ. Донецьк: РВА ДонНТУ, 2003 – С.96-106.

6. Григорьев А.В. Специфика выполнения теоретико-множественных операций над контекстно-свободными грамматиками в условиях различных форм дополнительных семантических правил в семиотической модели интеллектуальных САПР. Научные труды

ДонНТУ. Серия «Проблемы моделирования и автоматизации проектирования динамических систем». Выпуск 5 (116). Донецк: ДонНТУ, 2006. – С.91-104.

7. Математическая теория формальных языков / А.Е. Пентус, М.П. Пентус. – М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2006. – 247 с.

8. Братко И. Программирование на языке Пролог для искусственного интеллекта. М.: Мир, 1990. – 560 с.

9. Стобо Д.Ж. Язык программирования Пролог. М.: Радио и связь, 1993. – 368 с.

10. Е.И. Войтицкий. Исследование алгоритмов логического вывода в продукционных системах и оценка их сложности. КДС – 96. Ялта, 1996. С. 163-172.

11. Донской В.И. Логические продукционные системы: анализ и синтез // Техническая кибернетика и системный анализ, 1994, N 4, С. 11-22.

12. Donskoi V.I. Psevdo-boolean optimization with a disjunctive constraint // Comp. Maths Phys., Vol.34, No. 2, pp. 389-398, 1994.

Получено 12.05.2007.