УДК 004.89:510.649

А.А. АФОНИН

О ДРЕВОВИДНОЙ ФОРМЕ ПОИСКА ОПРОВЕРЖЕНИЯ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ С ЛОГИЧЕСКИМИ ВОЗМОЖНОСТЯМИ

Abstract. *The paper is devoted to the study of intelligent system possibilities given by the tree-like form of refutation search when using the resolution technique with paramodulation-type rules. Calculation of so-called literal trees intended for the establishment of formula unsatisfiability of first-order classic logic, both with equality and without it, are considered. Results about their correctness and completeness are given.*
Key words: *intelligent system with logical possibilities, tree-like form of inference search, first-order classic logic, calculution, equality, SLD-resolution, paramodulation rule.*

Анотація. *Робота присвячена вивченню можливостей інтелектуальних систем, які надає деревовидна форма пошуку спростування при використанні резолюційної техніки, включаючи правила парамодуляційного типу. Розглядаються числення так званих літеральних дерев, які призначені для встановлення невиконання формул класичної логіки першого порядку як з рівністю, так і без неї. Наводяться результати про їх коректність та повноту.*
Ключові слова: *інтелектуальна система з дедуктивними можливостями, деревовидна форма пошуку спростування, класична логіка першого порядку, числення, рівність, SLD-резолюція, правило парамодуляції.*

Аннотация. *Работа посвящена изучению возможностей интеллектуальных систем, которые предоставляет древовидная форма поиска опровержения в случае использования резолюционной техники, включая правила парамодуляционного типа. Рассматриваются исчисления так называемых литеральных деревьев, которые предназначены для установления невыполнимости формул классической логики первого порядка как с равенством, так и без него. Приводятся результаты об их корректности и полноте.*
Ключевые слова: *интеллектуальная система с дедуктивными возможностями, древовидная форма поиска опровержения, классическая логика первого порядка, исчисление, равенство, SLD-резолюция, правило парамодуляции.*

1. Введение

В качестве логического аппарата интеллектуальных систем с логическими возможностями, как правило, выступает резолюционная техника [1], основанная на линейной форме поиска логического вывода. С момента появления этой техники, в целях повышения эффективности поиска вывода, начали изучаться возможности организации процесса построения вывода в древовидной форме, что привело к появлению SLD-метода [2, 3], широко используемого на практике в различных интеллектуальных системах с логическими возможностями, например, в разнообразных системах логического программирования [4–6].

SLD-метод в своей первоначальной формулировке является полным методом для множеств хорновых дизъюнктов и неполным в общем случае. Поэтому были сделаны попытки получения его полных расширений в случае дизъюнктов произвольной структуры за счет добавления к нему минимальных логических средств. К их числу относятся, например, работы [7, 8], а также [9], в которой на базе секвенциального подхода получено полное расширение SLD-метода в виде литеральных деревьев.

Данная работа продолжает проведенные в [9] исследования по литеральным деревьям с большей ориентацией на их практическое применение, чем [9], что может быть объяснено развитием специальной техники работы с упорядоченными дизъюнктами в случае классического исчисления предикатов как без равенства, так и вместе с ним, которая использует древовидный процесс поиска опровержения.

2. Предварительные сведения

Рассматривается классическая логика первого порядка с равенством, в сигнатуру которой могут входить функциональные символы.

Известно, что установление выводимости любой формулы эквивалентно установлению невыполнимости ее отрицания с последующей элиминацией положительных кванторов в универсальном замыкании отрицания [10]. Следовательно, можно считать, что выполняется исследование на невыполнимость бескванторных формул вместо переменных, в которые могут быть подставлены любые термы. Более того, можно ограничиться только рассмотрением формул вида $(L_{1,1} \vee ... \vee L_{1,r1}) \wedge ... \wedge (L_{n,1} \vee ... \vee L_{n,rrn})$, где $L_{1,1}, ..., L_{n,rrn,k}$ – атомарные формулы или их отрицания. Это следует из того факта, что любая бескванторная формула классической логики первого порядка может быть приведена к конъюнктивной нормальной форме такими преобразованиями, которые сохраняют логическую эквивалентность. Поэтому в дальнейшем будет использоваться следующая система понятий и обозначений.

Атомарная формула или ее отрицание называется литерой. Равенство обозначается символом "="; при этом используется инфиксная запись. Дополнением литеры L называется результат приписывания к L знака отрицания, если L есть атомарная формула, и результат снятия с L знака отрицания в противном случае. Если L – литера, то ее дополнение обозначается $\sim L$, а пара литер L и $\sim L$ называется контрарной парой.

Понятия терма и формулы используются в обычном смысле. Формула, являющаяся результатом переименования в ней всех или некоторых переменных, называется ее вариантом. Пустая формула обозначается .

Считаются известными понятия подстановки, унификатора и наиболее общего унификатора (о.н.у.), которые можно найти в [11]. Более того, если E есть выражение, а σ – подстановка, то результат применения σ к E обозначается $E * \sigma$. Для любого множества выражений E_x, $E_x * \sigma$ обозначает множество, содержащее $E * \sigma$ для каждого E из E_x.

Дизъюнкт есть выражение вида $L_1 \vee ... \vee L_k$, где $L_1, ..., L_k$ – литеры. Дизъюнкты отождествляются с упорядоченными мультимножествами составляющих их литер, т.е. дизъюнкты могут содержать повторяющиеся литеры, порядок записи литер в нем является существенным.

Замечание. Представление дизъюнкта в виде упорядоченного мультимножества литер позволяет различать различные вхождения в дизъюнкт одной и той же литеры. Это свойство играет важную роль для последующего материала. Также важным является факт, что каждый дизъюнкт представляет собой "линейное" выражение при чтении его "слева направо", что и определяет линейный порядок на его литерах.

В дальнейшем IS будет обозначать исходное множество (входных) дизъюнктов, то есть то множество, которое исследуется на невыполнимость.

Пусть входной дизъюнкт D вида $L_1 \vee ... \vee L_k$ принадлежит IS. Тогда упорядоченная пара $\langle i, D \rangle$ (или просто i, когда не возникает недоразумения) называется индексом литеры L_i в дизъюнкте D.

Нам потребуются следующие правила вывода.

Правило упорядоченной резолюции OR (имеющее две формы в зависимости от того, является или нет хотя бы одна из его посылок входным дизъюнктом). Пусть упорядоченные дизъюнкты D_1 и D_2 имеют вид $D_1^{'} \vee L$ и $D_2^{'} \vee E$ (или $D_2^{'} \vee E \vee D_2^{''}$ в случае входного дизъюнкта). Если существует наиболее общий унификатор σ множества $\{L, \sim E\}$, то говорится, что по правилу OR из дизъюнктов D_1 и D_2 выводим упорядоченный дизъюнкт $D_1^{'} * \sigma \vee D_2^{'} * \sigma \left(D_1^{'} * \sigma \vee D_2^{'} * \sigma \vee D_2^{''} * \sigma \right)$. При этом литеры результата сохраняют индексы своих предшественников (то есть, если литера k из D_1 (D_2), отличная от $L(E)$, имеет индекс $\langle i, D_1 \rangle$ $(\langle i, D_2 \rangle)$, то $k * \sigma$ имеет в результирующем дизъюнкте индекс $\langle i, D_1 \rangle$ $(\langle i, D_2 \rangle)$).

Замечание. Правило OR представляет собой обычное правило резолюции [11], перенесенное на случай упорядоченных дизъюнктов.

Правило слабой факторизации WF. Пусть имеется упорядоченный дизъюнкт D вида $C_1 \vee L_1 \vee ... \vee C_m \vee L_m \vee C_{m+1} (m > 1)$ и существует наиболее общий унификатор σ множества $\{L_1, ..., L_m\}$, где $L_1, ..., L_m$ – литеры, имеющие один и тот же индекс. Тогда говорится, что по правилу WF из D выводим упорядоченный дизъюнкт $C_1 * \sigma \vee ... \vee C_m * \sigma \vee C_{m+1} * \sigma$. При этом литеры результата сохраняют индексы своих предшественников (то есть, если литера k из D, отличная от $L_1, ..., L_m$, имеет индекс $\langle i, D \rangle$ то $k * \sigma$ имеет индекс $\langle i, D \rangle$) в результирующем дизъюнкте.

Замечание. В случае, когда литеры $L_1, ..., L_m$ не имеют один и тот же индекс, введенное правило слабой факторизации превращается в обыкновенное правило факторизации.

Правильно построенными выражениями исчислений, задаваемых ниже, являются так называемые литеральные деревья [9], под которыми понимаются деревья, растущие "сверху вниз" и узлы которых помечены литерами. Вариантом литерального дерева Tr называется дерево, узлы которого помечены вариантами литер из меток Tr.

Исходное дерево (относительно дизъюнкта $L_1 \vee ... \vee L_k$ из исходного множества IS) состоит только из корня с меткой IS и k его наследников, каждый из которых помечен одной и только одной литерой L_i. Новые деревья строятся в соответствии с правилами вывода так называемых исчислений литеральных деревьев; при этом под выводом в этих исчислениях понимается последовательность литеральных деревьев $Tr_1, ..., Tr_n$, в которой Tr_1 есть исходное дерево и для каждого $j > 1\, Tr_j$ получено по одному правилу вывода из некоторых вариантов предыдущих деревьев.

Вырожденное дерево, т.е. дерево, не содержащее ни одной вершины, обозначается Δ.

3. Исчисление литеральных деревьев для классической логики без равенства

Построим специальное исчисление литеральных деревьев IT для классической логики первого порядка без равенства.

Правило входного резолюционного расширения IR**.** Пусть IS есть исходное множество дизъюнктов, содержащее входной дизъюнкт D вида $L_1 \vee ... \vee L_j \vee ... \vee L_k$, где $L_1, ..., L_k$ – литеры $(k > 0)$. Пусть Tr есть вариант некоторого литерального дерева, который не имеет общих переменных с D и E – его наиболее правый лист, отличный от . Предположим, что существует н.о.у. σ множества $\{L, \sim E\}$, и при $k > 1$ дерево Tr' построено из Tr посредством добавления к выделенному листу с $E(k-1)$-го наследника с метками $L_1, ..., L_{i-1}, L_{i+1}, ..., L_k$, соответственно, а в случае $k = 1$ дерево Tr' построено из Tr посредством добавления к выделенному листу с меткой E одного наследника с в качестве его метки. Тогда говорится, что по правилу IR из Tr и D выводится дерево $Tr' * \sigma$.

Правило контрарного закрытия CC**.** Пусть Tr есть некоторое литеральное дерево и Br – его наиболее правая ветвь с листом, имеющим метку L, отличную от . Предположим, что для некоторого узла с меткой E из ветви Br существует н.о.у. σ множества $\{L, \sim E\}$, и дерево Tr' построено из Tr посредством добавления к выделенному листу с меткой E одного наследника с в качестве его метки. Тогда говорится, что по правилу CC из Tr и D выводится дерево $Tr' * \sigma$.

Правило вычеркивания цепочки CD**.** Пусть Tr есть некоторое литеральное дерево и Br – его ветвь с листом Lf, имеющим метку . Пусть Ch обозначает такую максимальную цепочку ветви Br, которая при ее просмотре "сверху вниз" заканчивается листом Lf и каждый узел которой, кроме Br, содержит только одного наследника. Если Tr' обозначает результат вычеркивания из Tr всей цепочки Ch, то говорится, что Tr' получено из Tr по правилу CD.

В отношении правила CD будет предполагаться, что оно всегда применяется сразу после любого применения как IR, так и CC, ко всем ветвям с листьями, имеющими метку . Следовательно, мы можем считать, что в процессе построения выводов в исчислении IT порождаются только литеральные деревья, метки всех листьев которого отличны от . В частности, может быть выведено вырожденное дерево Δ.

Относительно исчисления IT имеет место следующий результат.

Предложение 1. Пусть IS обозначает конечное множество дизъюнктов и C – такой дизъюнкт из IS, что множество $IS/\{C\}$ выполнимо. Множество IS не выполнимо тогда и только тогда, когда в исчислении IT существует вывод вырожденного дерева Δ относительно C с IS в качестве исходного множества.

Доказательство (схема). Требуемое утверждение можно получить на основании результатов из работы [9], принимая во внимание, что любое дерево литеральных секвенций, выводимое в исчислении IS [9], может быть легко трансформировано в литеральное дерево, выводимое в IT посредством только правил IR и CC и, быть может, содержащее в листьях, и наоборот. Применяя же правило CD к трансформированному таким способом "образу" любого дерева доказательства в смысле работы [9], получаем выводимость вырожденного дерева Δ в исчислении IT и только его. Q.E.D.

Частным случаем предложения 1 является полнота SLD-метода [9].

Следствие 1. Пусть IS обозначает конечное множество хорновых дизъюнктов и C – такой положительный дизъюнкт (т.е. не содержащий литер со знаком отрицания) из IS, что множество $IS/\{C\}$ выполнимо. Множество IS не выполнимо тогда и только тогда, когда в исчислении IT без правила CC существует вывод вырожденного дерева Δ относительно C с IS в качестве исходного множества.

Доказательство. В случае выполнения условий предложения 2, при поиске вывода вырожденного дерева, могут порождаться только литеры без знака отрицания. Это означает, что правило CC никогда не будет применяться. Далее остается применить предложение 1. Q.E.D.

Заметим, следствие 1 влечет полноту входной резолюции в ее обычной формулировке [11].

4. Исчисление литеральных деревьев для классической логики с равенством

При разработке машинно-ориентированных методов поиска вывода в классической логике с равенством, как правило, используются различные модификации правила резолюции, отражающего стандартные способы обращения с равенством в обычных математических текстах. (Для обозначения равенства используется символ "="). В этой связи ниже демонстрируются (и исследуются) некоторые возможности встраивания парамодуляционной техники в исчисление литеральных деревьев. Предполагается, что читатель знаком с правилом резолюции, обозначаемым P в его формулировке из [11].

Для формулировки результатов о корректности и полноте предлагаемых расширений используется следующее понятие E-выполнимости множества формул.

Множество S дизъюнктов называется E-не выполнимым тогда и только тогда, когда является не выполнимым каждое множество $S \bigcup Eq$, где Eq – минимальное множество, обладающее следующими свойствами для всевозможных термов t, t', t'', построенных из констант и функциональных символов, входящих в дизъюнкты из S (если формулы из S не содержат ни одной константы, то считается, что специальная константа c_0 участвует в построении t, t', t''):

(1) $t = t \in Eq$,

(2) если $t' = t \in Eq$, то $t' = t \in Eq$ и

(3) если $t' = t \in Eq$ и $t' = t'' \in Eq$, то $t' = t, t = t' \in Eq$.

Также напомним, что выражение вида $f(x_1, ..., x_k) = f(x_1, ..., x_k)$, где f – k-арный функциональный символ и $(x_1, ..., x_k)$ – переменные, называется аксиомой функциональной рефлексивности.

Если S – множество дизъюнктов, то $AFR(S)$ обозначает множество всех аксиом функциональной рефлексивности для функциональных символов из S.

Специфической чертой предлагаемых парамодуляционных расширений исчисления литеральных деревьев является то, что при формулировке правил парамодуляционного типа во внимание принимаются "направления" парамодуляции. Это позволяет ввести четыре парамодуляционных правила, разбиение которых на две группы по три правила в каждой дает возможные минимальные парамодуляционные расширения, полные в общем случае.

Правило парамодуляционного расширения P_{in}. Пусть IS есть исходное множество дизъюнктов, Tr – литеральное дерево и L – метка наиболее правого листа из Tr, отличная от . Предположим, что IS содержит такой дизъюнкт P, что правило парамодуляции P применимо из C в L, где L рассматривается как однолитерный дизъюнкт, C – вариант D и C не имеет общих переменных с литерами из Tr. Предположим, что при этом применении P порождаются о.н.у. σ и парамодулянт $L_1 \vee ... \vee L_k$. Тогда говорится, что дерево, являющееся результатом добавления в дереве $Tr * \sigma$ к листу с $L * \sigma$ k наследников с метками $L_1, ..., L_k$ в порядке их чтения "слева направо", выводится по правилу P_{in} из Tr относительно D.

Правило парамодуляционного расширения P_{out}. Пусть IS есть исходное множество дизъюнктов, Tr – литеральное дерево и L – метка наиболее правого листа из Tr, отличная от . Предположим, что IS содержит такой дизъюнкт D, что правило парамодуляции P применимо из L в C, где L рассматривается как однолитерный дизъюнкт, C – вариант D, и C не имеет общих переменных с литерами из Tr. Предположим, что при этом применении P порождаются о.н.у. σ и парамодулянт $L_1 \vee ... \vee L_k$. Тогда говорится, что дерево, являющееся результатом добавления в дереве $Tr * \sigma$ к листу с $L * \sigma$ k наследников с метками $L_1, ..., L_k$ в порядке их чтения "слева направо", выводится по правилу P_{in} из Tr относительно D.

Правило парамодуляционного расширения P_{down}. Пусть Tr есть литеральное дерево и L – метка наиболее правого узла из Tr, отличная от . Предположим, что в ветви с листом L существует такой узел с меткой E, что правило парамодуляции P применимо из E в L, где E и L рассматриваются как однолитерные дизъюнкты. Предположим, что при этом применении P порождаются о.н.у. σ и парамодулянт L'. Тогда говорится, что дерево, являющееся результатом добавления в дереве $Tr*\sigma$ к листу с $L*\sigma$ одного наследника с меткой, выводится по правилу P_{out} из Tr относительно E.

Правило парамодуляционного расширения P_{up}**.** Пусть Tr есть литеральное дерево и L – метка наиболее правого узла из Tr, отличная от . Предположим, что в ветви с листом L существует такой узел с меткой E, что правило парамодуляции P применимо из L в E, где E и L рассматриваются как однолитерные дизъюнкты. Предположим, что при этом применении *P* порождаются о.н.у. σ и парамодулянт L'. Тогда говорится, что дерево, являющееся результатом добавления в дереве $Tr*\sigma$ к листу с $L*\sigma$ одного наследника с меткой, выводится по правилу P_{up} из Tr относительно E.

Обозначим $IT_d\,(IT_u)$ исчисление литеральных деревьев, полученное из IT добавлением к нему правил P_{in}, P_{out} и P_{down} (P_{in}, P_{out} и P_{up}). Относительно так построенных исчислений имеют место следующие результаты.

Предложение 2. Пусть IS обозначает конечное множество дизъюнктов и C – такой дизъюнкт из IS, что множество $IS/\{C\}$ E-выполнимо. Множество IS является E-не выполнимым тогда и только тогда, когда в исчислении $IT_d\,(IT_u)$ существует вывод вырожденного дерева Δ относительно C с $IS \bigcup AFR(S)$ в качестве исходного множества.

Доказательство (схема). Доказательство этого предложения может быть получено, следуя идеям установления корректности и полноты секвенциальных исчислений с равенством, рассмотренным в [12]. Q.E.D.

Следствие 2. Пусть IS обозначает конечное множество хорновых дизъюнктов и C – такой положительный дизъюнкт из IS, что множество $IS/\{C\}$ E-выполнимо. Множество IS является E-не выполнимым тогда и только тогда, когда в исчислении $IT_d\,(IT_u)$ без правила CC существует вывод вырожденного дерева Δ относительно C с $S \bigcup AFR(S)$ в качестве исходного множества.

Доказательство "повторяет" рассуждения из доказательства следствия 1. Q.E.D.

Аксиомы функциональной рефлексивности являются необходимыми для полноты исчислений IT_d и IT_u. В этом можно убедиться, если рассмотреть следующий пример E-невыполнимого множества $IS = \{a = b, R_1(h_1(z,z)), R_2(h_2(u,u)), \neg R_1(g_1(x,x) \vee \neg R_2(g_2(y,y)), g_1(f(a), f(b)) = h_1(f(a), f(b)), g_2(f(a), f(b)) = h_2(f(a), f(b))\}$, которого не имеется возможности построить вывод без использования аксиом функциональной рефлексивности ни в исчислении IT_d, ни в исчислении IT_u.

Заметим, из следствия 2 легко получается полнота входной резолюции с парамодуляцией.

5. Заключение

Проведенные исследования показывают, что древовидная форма поиска вывода дает, по сравнению с линейной организацией процесса дедукции, ряд преимуществ, вызванных следующими обстоятельствами.

"Разбиение" обычной резолюции на несколько простых правил позволяет оттенить влияние каждого правила на процесс поиска вывода. Так, в частности, оказалось, что SLD-метод (так же, как и входная резолюция) может быть получен просто удалением правила CC из исчисления литеральных деревьев. И наоборот, "добавление" к SLD-методу правила CC приводит к полному его расширению для логики без равенства.

Заметим, что предложенные в работе расширения исчисления литеральных деревьев парамодуляционной техникой обладают аналогичными ствойствами. К недостаткам предложенных парамодуляционных расширений относится необходимость присоединения аксиом функциональной рефлексивности для получения полного метода для логики с равенством. Однако следует ожидать, что техника парамодуляционного расширения, использованная в [13] для табличных методов, может быть встроена и в исчисления литеральных деревьев.

Приведенные свойства исчислений литеральных деревьев могут быть использованы при попытке построения таких расширений SLD-резолюции, уже реализованной в некоторой интеллектуальной системе, которые должны превратиться в методы поиска опровержения, полные как для классической логики при работе как без равенства, так и вместе с ним.

СПИСОК ЛИТЕРАТУРЫ

1. Robinson J.A. A machine-oriented logic based on resolution principle / J.A. Robinson // Journal of the ACM. – 1965. – Vol. 12. – P. 23 – 41.
2. Kowalski R. Linear Resolution with selection function / R. Kowalski, D. Kuehner // Artificial Intelligence. – 1971. – Vol. 2. – P. 227 – 260.
3. Apt K.R. Contributions into the theory of logic programming / K.R. Apt, M.H. van Emden // Journal of the ACM. – 1982. – Vol. 3, N 29. – P. 841 – 862.
4. Lloyd J.V. Foundations of logic programming / Lloyd J.V. – Springer, Berlin: Springer, 1987. – 476 p.
5. Nilsson U. Logic, programming and PROLOG / U. Nilsson, J. Maluszynski. – New York: John Wiley & Sons, Inc., 1990. – 304 p.
6. Handbook of logic in artificial intelligence and logic programming / D.M. Gabbay, J. Christopher, Ch.J. Hogger et al. // Oxford University Press. – 1994. – Vol. 2: Deduction methodologies. – 508 p.
7. Stickel M.A. Prolog Technology Theorem Prover / M.A. Stickel // New Generation Comp. – 1984. – Vol. 4. – P. 371 – 383.
8. Stickel M.A. Prolog Technology Theorem Prover: Implementation by an Extended Prolog Compiler / M.A. Stickel // Lecture Notes in Computer Science. – 1986. – Vol. 232. – P. 573 – 587.
9. Афонін А.О. Про машинно-орієнтовані числення секвенціального типу для класичної логіки першого порядку / А.О. Афонін // Наукові записки. Національний університет "Києво-Могилянська Академія". Комп'ютерні науки. – К.: НаУКМА, 2009. – № 99. – С. 23 – 28.
10. Минц Г.Е. Теорема Эбрана / Г.Е. Минц // Математическая теория логического вывода. – М.: Наука, 1967. – С. 311 – 350.
11. Chang C. Symbolic logic and mechanical theorem proving / C. Chang, R. Lee. – Orlando, FL, USA: Academic Press, Inc., 1997. – 331 p.
12. Lyaletski A. Computer-oriented calculus of sequent tree / A. Lyaletski // Lecture Notes in Computer Science.– Springer-Verlag, Heidelberg. – 2004. – Vol. 2942. – P. 213 – 230.
13. Paskevich A. Connection tableaux with lazy paramodulation / A. Paskevich // Journal of Automated Reasoning. – 2008. – Vol. 40, N 2-3. – P. 179 – 194.

Стаття надійшла до редакції 15.01.2010