УДК 518.74

З.М. АСЕЛЬДЕРОВ, А.А. ЛЯЛЕЦКИЙ

ПРАВИЛО КОНТРАРНОГО ЗАКРЫТИЯ И ПОЛНЫЕ РАСШИРЕНИЯ ЛОГИЧЕСКОГО АППАРАТА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ С ПРАВИЛОМ ВХОДНОЙ РЕЗОЛЮЦИИ

Введение

Применение логики в интеллектуальных системах, как правило, предполагает создание и/или применение уже известных и достаточно эффективных методов поиска вывода в классической логике первого порядка. К числу последних относится входная резолюция, часто выбираемая в качестве базового дедуктивного аппарата разнообразных систем логического программирования. Для множеств хорновых дизъюнктов входная резолюция является корректным и полным методом, которая для достижения полноты в случае более широких классов дизъюнктов требует подключения дополнительных логических механизмов. Был получен ряд результатов [1–3] показывающих, что, в общем случае, за счет использования некоторой вспомогательной информации, можно получить полные расширения входной резолюции при построении линейных выводов.

Параллельно, в целях повышения эффективности поиска вывода, изучался вопрос построения древовидного вывода с правилом входной резолюции. В результате входная резолюция приобрела вид так называемой SLD-резолюции, в ходе применения которой строятся SLD-деревья [4,5] и которая широко используется в интеллектуальных системах типа систем логического программирования. SLD-резолюция является корректной и полной для множеств хорновых дизъюнктов, но не сохраняет полноту для множеств произвольных дизъюнктов. Поскольку в текущее время стала актуальной способность интеллектуальных систем проводить формальные рассуждения, используя формулы произвольного вида, естественно, что возник вопрос о том, какими максимально простыми средствами необходимо дополнить SLD-резолюцию для получения полного метода поиска вывода в общем случае. Одно из возможных его решений предлагается в данной работе[1]: полное расширение SLD-резолюции достигается за счет встраивания в SLD-резолюцию достаточно простого правила вывода, названного здесь правилом контрарного закрытия.

Специфической чертой данной работы является то, что упомянутое выше полное расширение SLD-резолюции появляется как «сужение» на дизъюнкты специальных машинно-ориентированных секвенциальных исчислений, разработанных в целях получения корректных и полных методов установления выводимости в классической логике первого порядка, которые, в некотором смысле, сравнимы по эффективности с широко используемыми резолюционными методами [6]. В этой связи секвенциальные исчисления, рассматриваемые здесь, могут выступать в качестве «естественных» полных расширений SLD-резолюции на секвенциальный случай. Сама же ориентация на развитие специальных секвенциальных исчислений была вызвана тем обстоятельством, что использование генценовского подхода [7] к проведению дедукции является намного более естественным и удобным при поиске доказательства, предусматривающего вмешательство человека в процесс дедукции. Заметим, что оптимизация перебора, возникающего из-за возможности разных порядков применения обычных кванторных правил, достигается в работе за счет предварительной сколемизации, впервые предложенной в [8] для формул первого порядка и позже перенесенной в [9] на секвенции. Современное состояние этих проблем подробно изложено в [10].

[1] Исследования, представленные в данной работе, поддержаны Intas-проектом 2000-447.

Предварительные сведения

Рассматривается *секвенциальная форма классической логики первого порядка без равенства*. Известно [9], что установление выводимости любой секвенции может быть сведено к установлению подходящей замкнутой секвенции с элиминированными положительными кванторами. Следовательно, можно считать, что любая секвенция состоит из бескванторных формул, вместо переменных которых могут быть подставлены любые термы, а сами формулы секвенций попарно не имеют общих переменных.

Считаются известными понятия *литеры, терма, формулы* и *дизъюнкта*. Если L – литера, то через $\sim L$ обозначается её *дополнение*. Выражение $F^{\neg}$ обозначает результат одношагового внесения отрицания в формулу F, а $\#$ – пустую формулу. Также отметим, что мы понимаем положительное $\left(P\left[F^{+}\right]\right)$ и отрицательное $\left(P\left[F^{-}\right]\right)$ вхождения формулы F в формулу P в обычном смысле.

В дальнейшем будет использоваться определение (одновременного) *наиболее общего унификатора* в формулировке из [11]. Под *равенством* понимается упорядоченная пара термов $< s\ ,\ t >$, записываемая в виде $s \approx t$.

Пусть L является литерой вида $R(t_1,...,t_n)(\neg R(t_1,...,t_n))$ и M – литерой вида $R(s_1,...,s_n)(\neg R(s_1,...,s_n))$, где R есть предикатный символ и $t_1,...,t_n$, $s_1,...,s_n$ есть термы. Тогда через $\sum(L,M)$ обозначается множество равенств $\{t_1 \approx s_1,...,t_n \approx s_n\}$. В этом случае будем говорить, что литеры L и M являются *эквивалентными по модулю* $\sum(L,M)(L \approx M \bmod \sum(L,M))$.

Мы будем придерживаться обычного понятия *секвенции*. Формулы из антецедента секвенции будут называться ее *посылками*, а формулы из сукцедента – ее *целями*.

В дальнейшем будут рассматриваться только секвенции с одной целью в сукцеденте секвенции. При этом понятие секвенции распространяется на выражения вида $\Gamma \rightarrow \#$, где $\#$ обозначает пустую формулу. Секвенции $\Gamma \rightarrow \#$ будут называться *аксиомами* рассматриваемых секвенциальных исчислений.

Деревья понимаются в обычном смысле. Деревья растут «сверху вниз». В дальнейшем мы будем рассматривать так называемые секвенциальные деревья. *Секвенциальное дерево* – это дерево, каждая вершина которого помечена некоторой секвенцией. Часто вершина дерева будет отождествляться с секвенцией, приписанной ей в качестве метки.

В процессе поиска вывода начальной секвенции S строится так называемое *дерево вывода* Tr относительно S. В начальный момент процесса поиска дерево Tr состоит только из корня, помеченного S, и называется *инициальным деревом*. Если на некотором этапе Tr уже сгенерировано, то последующие вершины дерева порождаются в соответствии с тем или иным правилом вывода, примененным к некоторому листу Tr в случае, когда такое применение правила возможно.

С каждым деревом вывода связывается специальное множество равенств $Eq(Tr)$. $Eq(Tr)$ полагается равным пустому множеству $\varnothing$ для каждого инициального дерева вывода. Если же дерево вывода Tr отлично от инициального, то $Eq(Tr)$ полагается равным $Eq(Tr')\cup\sum(L,M)$, где Tr' – такое дерево вывода, что Tr «выводится» из Tr' применением некоторого подходящего правила вывода, генерирующим соответствующее $\sum(L,M)$ (см. ниже).

Дерево вывода Tr относительно секвенции S называется *деревом доказательства* относительно S тогда и только тогда, когда каждый лист Tr помечен аксиомой и существует одновременный наиболее общий унификатор всех равенств из $Eq(Tr)$.

Пусть $P_1...,P_n$ и G есть формулы $(n \geq 0)$, и S обозначает секвенцию $P_1...,P_n \to G$. *Тогда формульным образом* $\varphi(S)$ называется формула $P_1 \wedge ... \wedge P_n \supset G$, когда $n > 0$, и $\varphi(S)$ есть G, когда $n = 0$. Если G есть #, то $\varphi(S)$ есть #. Заметим, что # всегда полагается общезначимой формулой.

Целеориентированное исчисление секвенций

Описываемое ниже исчисление GS предназначено для установления того, что формула G является логическим следствием формул $P_1...,P_n$ (в этом случае $P_1...,P_n$ и G называются инициальными формулами). Ясно, что для задания GS нам требуется только определить его правила вывода.

1. *Правила расщепления цели*. Эти правила предназначены для элиминации главной логической связки рассматриваемой цели. Результатом применения каждого такого правила является новая секвенция (секвенции) с новой целью (целями) и, возможно, новыми посылками. Отметим, каждое правило расщепления цели генерирует $\sum(L,M)$, равное пустому множеству $\varnothing$.

$$\frac{\Gamma \to F \supset G}{\Gamma, F \to G}; \qquad \frac{\Gamma \to F \supset G}{\Gamma, G^{\neg} \to F};$$

$$\frac{\Gamma \to F \vee G}{\Gamma, F^{\neg} \to G}; \qquad \frac{\Gamma \to F \vee G}{\Gamma, G^{\neg} \to F};$$

$$\frac{\Gamma \to F \wedge G}{\Gamma \to F \Gamma \to G}; \qquad \frac{\Gamma \to \neg F}{\Gamma \to F^{\neg}}.$$

2. *Правило дублирования посылки*.

$$\frac{\Gamma_1, F[M^+], \Gamma_2 \to L}{\Gamma_1, F', F[M^+], \Gamma_2 \to L},$$

где $L \approx M \operatorname{mod} \sum(L,M)$ и F' есть вариант формулы F.

3. *Правила расщепления цели*. Порядок применения правил этого типа “управляется” литерой L из сукцедента. Отметим, что везде ниже M обозначает литеру, удовлетворяющую условию: $L \approx M \operatorname{mod} \sum(L,M)$.

$$\frac{\Gamma_1, F[M^-] \supset G, \Gamma_2 \to L}{\Gamma_1, (F[M^+])^{\neg}, \Gamma_2 \to \Gamma \Gamma_1, \Gamma_2 \to G^{\neg}}; \qquad \frac{\Gamma_1, F \supset G[M^+], \Gamma_2 \to L}{\Gamma_1, (G[M^+])^{\neg}, \Gamma_2 \to \Gamma \Gamma_1, \Gamma_2 \to F};$$

$$\frac{\Gamma_1, F \vee G[M^+], \Gamma_2 \to L}{\Gamma_1, G[M^+], \Gamma_2 \to L \Gamma_1, \Gamma_2 \to \neg F}; \qquad \frac{\Gamma_1, F[M^+] \vee G, \Gamma_2 \to L}{\Gamma_1, F[M^+], \Gamma_2 \to L \Gamma_1, \Gamma_2 \to \neg G};$$

$$\frac{\Gamma_1, F \wedge G[M^+], \Gamma_2 \to L}{\Gamma_1, G[M^+], F, \Gamma_2 \to L}; \qquad \frac{\Gamma_1, F[M^+] \wedge G, \Gamma_2 \to L}{\Gamma_1, F[M^+], G, \Gamma_2 \to L};$$

$$\frac{\Gamma_1, \neg(F[M^-]), \Gamma_1 \to L}{\Delta_1, F^{\neg}[M^+], \Delta_2 \to L}; \qquad \frac{\Gamma_1, M, \Gamma_2 \to L}{\Gamma_1, M, \Gamma_2 \to \#}.$$

4. Правило контрарного закрытия (CC-правило). Пусть Tr – некоторое дерево вывода и Br – его ветвь с листом Lf, помеченная секвенцией $\Gamma \to L$, где Γ – последовательность формул и L – литера. Пусть ветвь Br содержит секвенцию $\Gamma' \to M$, где Γ' является последовательностью формул и M – такая литера, что $\sim L \approx M \bmod \sum(\sim L, M)$. Если дерево Tr' получено из Tr с помощью присоединения к Lf одного наследника, помеченного секвенцией $\Gamma \to \#$, то говорится, что Tr' выводимо из Tr с помощью *правила контрарного закрытия (CC-правила)*. При этом множество $Eq(Tr')$ определяется равным $Eg(Tr) \cup \sum(\sim L, M)$.

Относительно исчисления GS имеет место следующее утверждение.

ПРЕДЛОЖЕНИЕ 1. Пусть формулы $P_1,...,P_n$ образуют конечное непротиворечивое множество формул. Тогда формула G является логическим следствием формул $P_1,...,P_n$ в том и только том случае, когда существует дерево доказательства относительно секвенции $P_1,...,P_n, \neg G \to G$ в исчислении GS.

Схема доказательства. Прежде всего заметим, что теорема Эрбрана в формулировке работы [9] и некоторые другие результаты из [9] позволяют ограничиться рассмотрением только пропозиционального случая выводимости в исчислении GS, т.е. рассмотрением только таких деревьев вывода Tr, у которых $\sum(Tr) = \varnothing$. Для пропозиционального случая полнота исчисления GS может быть доказана индукцией по числу бинарных логических связок в инициальной секвенции, исследуемой на выводимость, а корректность может быть легко проверена с помощью понятия формульного образа $\varphi(S)$ секвенции $S.Q.E.D.$

Модификации исчисления GS

Отдельного обсуждения заслуживает случай, когда мы рассматриваем секвенции вида $M_{1,1} \vee ... \vee M_{1,r1,} ... M_{n,1} \vee ... \vee M_{n,m} \to L_1 \wedge ... \wedge L_k$, где $M_{1,1},...,M_{n,m}, L_1,...,L_k$ – литеры. Поскольку каждая формула первого порядка может быть приведена к конъюнктивной (дизъюнктивной) нормальной форме такими преобразованиями, которые сохраняют логическую эквивалентность, то на основании результатов из [9] легко показать, что установление выводимости любой секвенции произвольного вида эквивалентно установлению выводимости некоторой подходящей секвенции вида $M_{1,1} \vee ... \vee M_{1,r1},...,M_{n,1} \vee ... \vee M_{n,m} \to L_1 \wedge ... \wedge L_k$. Вот почему в данном разделе мы рассматриваем только секвенции такого вида. В этой связи отметим, что если G есть $L_1 \wedge ... \wedge L_k$, то $G^{\neg}$ есть $\sim L_1 \vee ... \vee \sim \mathrm{L}_k$.

Учитывая вышесказанное, исчисление GS может быть трансформировано в исчисление LS “литеральных” секвенций, которое имеет следующие правила:

1. Правило расщепления цели. Приводимое ниже правило обобщает правило “расщепления конъюнкции в цели” исчисления GS, если рассматривать $\wedge$ как многоместную операцию:

$$\frac{\Gamma \to L \ \wedge \ldots \wedge L_m}{\Gamma \to L_1, \ldots, \Gamma \to L_m}.$$

2. Правило расщепления посылки. Это правило может быть применено, когда цель рассматриваемой секвенции является литерой:

$$\frac{\Gamma_1, A_1 \vee \ldots \vee A_n \vee M \vee B_1 \vee \ldots \vee B_r, \Gamma_2 \to L}{\Gamma' \to -A_1', \ldots, \Gamma' \to -A_n', \Gamma' \to -B_1', \ldots, \Gamma' \to -B_r'} \ ,$$

где $A_1, \ldots, A_n, B_1, \ldots, B_r$, L и M – литеры, $L \approx M \bmod \sum(L, M)$ и $A_1', \ldots, A_n', B_1', \ldots, B_r'$ – новые варианты литер $A_1, \ldots, A_n, B_1, \ldots, B_r$ соответственно и Γ' является последовательностью $\Gamma_1, A_1 \vee \ldots \vee A_n \vee M \vee B_1 \vee \ldots \vee B_r, \Gamma_2$.

Исчисление LS имеет такое же $\#$-правило и СС-правило, как и исчисление GS.

ПРЕДЛОЖЕНИЕ 2. Пусть дизъюнкты $P_1, \ldots, P_n$ образуют конечное непротиворечивое множество дизъюнктов. Конъюнкция литер G является логическим следствием дизъюнктов $P_1, \ldots, P_n$ в том и только том случае, когда существует дерево доказательства относительно секвенции $P_1, \ldots, P_n$, $G^{\neg} \to G$ в исчислении LS.

Схема доказательства. В соответствии с предложением 1, существует дерево доказательства Tr относительно секвенции $P_1, \ldots, P_n$, $G^{\neg} \to G$ в исчислении GS. Очевидно, что Tr содержит применение только таких правил вывода исчисления GS, которые имеют соответствующие аналоги в исчислении $LS.Q.E.D.$

Особенностью исчисления LS является то, что антецеденты выводимых секвенций совпадают с антецедентом соответствующей инициальной секвенции. Это позволяет рассматривать антецедент этой инициальной секвенции как множество входных дизъюнктов и преобразовывать каждое дерево вывода Tr исчисления LS в дерево $\gamma(Tr)$, листья которого совпадают с листьями дерева Tr, и помеченное целями соответствующих секвенций (и только ими). Мы будем называть такое дерево $\gamma(Tr)$ деревом целей, соответствующим дереву Tr. Таким образом, имеется простой способ перехода от исчисления LS к SLD-резолюции.

Полнота SLD-резолюции для множеств хорновых дизъюнктов является известным результатом логического программирования. Следующее предложение содержит его.

СЛЕДСТВИЕ (корректность и полнота SLD-резолюции). Пусть $P_1, \ldots, P_n$ являются положительными хорновыми дизъюнктами, и G является конъюнкцией атомарных формул. Цель G является логическим следствием дизъюнктов $P_1, \ldots, P_n$ в том и только в том случае, когда существует дерево доказательства относительно секвенции $P_1, \ldots, P_n \to G$ в исчислении LS (без применений СС-правила).

***Доказательство*.** В соответствии с предложением 2 существует дерево доказательства Tr относительно секвенции $P_1, \ldots, P_n$, $G^{\neg} \to G$ в исчислении LS. Очевидно, что Tr не содержит секвенций, выводимых по СС-правилу. Следовательно, $\gamma(Tr)\sigma$ является SLD-деревом, где σ есть одновременный наиболее общий унификатор всех равенств из $Eq(Tr)$. $Q.E.D.$

Выводы

Полученные результаты показывают, что исчисление LS даёт способы построения полных расширений SLD-резолюции для множеств дизъюнктов произвольного вида посредством встраивания в SLD-резолюцию правила контрарного закрытия или его аналогов. Если же мы интересуемся возможностями полного расширения SLD-резолюции на случай установления выводимости секвенций с произвольными формулами, то можно воспользоваться различными модификациями исчисления GS. Эти свойства исчислений GS и LS определяют методы модификации программных средств в целях получения полных в общем случае расширений логического аппарата разнообразных интеллектуальных систем (например, экспертных систем, систем логического программирования, дедуктивных баз данных и т.д.), использующих SLD-резолюцию в качестве базовой техники поиска логического вывода.

СПИСОК ЛИТЕРАТУРЫ

1. Pool D.L., Goeble R. Gracefully Adding Negation and Disjunction to Prolog // Lecture Notes in Computer Science. – 1986. – Vol. 220. – P. 635 – 641.
2. Stickel M. A. Prolog Technology Theorem Prover // New Generation Comp. – 1984. – Vol. 4. – P. 371 – 383.
3. Stickel M. A. Prolog Technology Theorem Prover: Implementation by an Extended Prolog Compiler // Lecture Notes in Computer Science. – 1986. – Vol. 232. – P. 573 – 587.
4. Apt K.R., van Emden M. H. Contributions into the Theory of Logic Programming // JASM. – 1982. – Vol. 3, N 29. – P. 841 – 862.
5. Lloyd J.V. Foundations of Logic Programming. – Berlin, 1987. – 476 p.
6. Robinson J. A machine-oriented logic based on resolution principle // Journal of the ACM. – 1965. – Vol. 12, N 1. – P. 23 – 41.
7. Gentzen G. Untersuchungen uber das Logische Schliessen // Math. Z. – 1934. – Vol. 39. – P. 176 – 210.
8. Skolem T. Logisch-kombinatorische Untersuchungen uber die Erfullbarkeit oder Beweisbarkeit mathematischer Satze // Skriftner utgit ar Videnskapsselskaper i Kristiania. – 1920. – Vol. 4. – P. 4 – 36.
9. Минц Г.Е. Теорема Эбрана // Математическая теория логического вывода. – М.: Наука, 1967. – С. 311 – 350.
10. Handbook of Automated Reasoning. – Elsevier Science Publishers, 2001. – Vol. 1: Edited by A.Robinson and A.Voronkov. – 1020 p.
11. Чень Ч., Ли Р. Математическая логика и автоматическое доказательство теорем. – М.: Наука, 1987. – 358 с.