УДК 007:519.816

ИНТЕРПОЛЯЦИОННАЯ ТЕОРЕМА ДЛЯ ЛОГИКИ ПРОЦЕССОВ В.ПРАТТА[*]

М.К.Валиев[1]

Предлагается обобщение интерполяционной теоремы для пропозициональной динамической логики [Kowalsky, 2002] на вариант логики процессов, предложенный в [Pratt, 1979].

Введение

Интерполяционная теорема (лемма Крейга [Craig, 1957]) в ограниченной для пропозиционального случая форме утверждает:

если пропозициональная формула f -> g тождественно истинна, то существует формула h такая, что 1) h содержит только пропозициональные переменные, входящие в обе формулы f и g, и 2) формулы f -> h и h -> g тождественно истинны.

Позднее было установлено, что эта теорема верна для многих других логик, в частности, для различных вариантов модальной логики (см. [Максимова, 1979]), но неверна для многих других, в частности, для временной логики фон Вригта [Максимова, 1989]. Довольно долго стоявший вопрос о том, верна ли интерполяционная теорема для пропозициональной динамической логики (ПДЛ) [Fischer-Ladner, 1979], был относительно недавно положительно решен в [Kowalsky, 2002] (временная логика фон Вригта фактически является сильно ограниченным фрагментом ПДЛ). Естественно возникает вопрос, верна ли интерполяционная теорема для различных обобщений ПДЛ, рассматривавшихся в литературе.

Основное утверждение настоящей работы состоит в том, что интерполяционная теорема имеет место для пропозициональной логики процессов (ПЛП), определенной В.Праттом [Pratt, 1979] как расширение ПДЛ. ПЛП содержит некоторый программно-модальный оператор,

---

[*] Работа выполнена при финансовой подержке РФФИ (проекты № 07-01-00637 и 08-01-00241).

[1] *Москва, Миусская пл., 4, Институт прикладной математики РАН, valiev@keldysh.ru*

позволяющий выражать свойства состояний, появляющихся во время выполнения программ (такого типа как «все (или некоторые) состояния, встречающиеся в процессе работы программы, удовлетворяют некоторому свойству» и т.п.). В отличие от ПЛП ПДЛ позволяет выражать только свойства программ в терминах «вход-выход».

1. Пропозициональная логика процессов

1.1. Синтаксис ПЛП

Язык ПЛП содержит два типа переменных: пропозициональные переменные *P, Q,…* и программные переменные *A, B,…* (допускаются индексы). Понятия программы (точнее, схемы программ) и формулы ПЛП определяются следующим образом:

Определение 1.

1) каждая программная переменная является (элементарной) программой, пропозициональная переменная – (элементарной) формулой;
2) если *a, b* – программы, *p, q* – формулы, то *(a||b), (a;b), (a*)* – программы, *(p&q), ~p, aU[p,q]* – формулы.

(a||b) определяет недетерминированный выбор программ *a* или *b* для выполнения, *(a;b)* – последовательное выполнение программ *a* и *b, (a*)* – итерация программы *a*. Заметим также, что операторы *aU[…]* двойственны операторам, введенным в [Pratt, 1979].

При записи программ и формул будем употреблять обычные правила опускания скобок. Как обычно, дизъюнкция *p||q* и импликация *p->q* вводятся как сокращения формул *~(~p&~q)* и *~(p&~q)*.

Заметим, что здесь мы опустили тесты, т.е. программы вида *p?*, и операцию обращения программ, которые требуют отдельного рассмотрения, хотя, по-видимому, интерполяционная теорема остается справедливой и при введении тестов и операции обращения. Заметим, что каждая программа *a* представляет собой регулярное выражение над некоторым множеством элементарных программ. Обозначим через *П* множество всех программ.

1.2. Семантика ПЛП

Семантика ПЛП определяется на моделях **M** типа Крипке, **M** = *<S, T, V>,* где *S* – множество (абстрактных) состояний, *V* сопоставляет каждой элементарной формуле *P* некоторое множество состояний (на которых истинна *P*), *T* сопоставляет каждой элементарной программе *A* некоторое множество конечных траекторий (т.е. конечных последовательностей состояний).

Функции *V* и *T* распространяются на произвольные формулы и программы следующим образом. В этом определении для обозначения истинности формулы p в состоянии s (т.е. указания на то, что *s* входит в *V(p))* используется обозначение *s*|=*p.*

Определение 2.

1) *T(a||b)* равно объединению *T(a)* и *T(b)*.
2) Для траекторий t_1 и t_2 пусть $t_1.t_2$ обозначает траекторию, получаемую идентификацией последнего состояния из t_1 и первого состояния из t_2, если эти состояния совпадают (в противном случае $t_1.t_2$ не определена), и для множеств траекторий T_1 и T_2 множество $T_1.T_2$ определяется как множество траекторий $t_1.t_2$ таких, что t_1 входит в T_1, t_2 входит в T_2; тогда *T(a;b)* определяется как *T(a).T(b)*.
3) *T(a*)* равно объединению множеств {<*s*>*: s* – произвольное состояние из *S}, T(a), T(a.a),,,, T(a.a…a),…*
4) *s*|=*p&q*, если и только если *s*|= *p и s*|= *q*.
5) *s*|=~*p*, если и только если неверно, что *s*|= *p*.
6) *s*|=*aU[p,q]*, если и только если *T(a)* содержит траекторию <s_1,s_2, …,s_k> такую, что s_k|=*q* и s_j|=*p* для всех *j*, *0*<*j*< *k+1*.

Заметим, что для любой программы *a* множество *T(a),* как определено выше, содержит только конечные траектории. Легко видеть, что из данного определения семантики ПЛП следует, что аналог оператора <*a*> пропозициональной динамической логики можно выразить в ПЛП формулой *aU[true, p]*. Краткое обозначение <*a*>*p* для этой формулы мы используем ниже при определении аксиом.

2. Интерполяционная теорема для ПЛП

Теорема. *Если формула ПЛП, имеющая вид p->q, тождественно истинна, то существует формула r такая, что 1) все пропозициональные и программные переменные формулы r входят в обе формулы p и q, и 2) формулы p->r и r->q тождественно истинны.*

Доказательство этой теоремы получается обобщением соответствующего доказательства из [Kowalsky, 2002], где интерполяционная теорема установлена для пропозициональной динамической логики.

Это доказательство состоит из следующих этапов:

i) Вводится понятие proc-U-алгебры, которое обобщает понятие динамической алгебры, введенное в [Pratt, 1991] в качестве алгебраического компаньона для пропозициональной динамической логики.

Proc-U-алгебра – это булева алгебра B, над которой определено множество двухместных модальных операторов $aU[...]$, где a - программа из Π, удовлетворяющих следующим аксиомам:

$$aU[0,x] = aU[x,0] = 0;$$
$$aU[x, y] \leq x;$$
$$aU[x, y] \leq \langle a\rangle(x\&y);$$
$$(x\text{->}y)\&(z\text{->}u)\&\ aU[x,z] \leq aU[y,u];$$
$$(x\text{->}y)\&aU[x, z||u] \leq\ aU[y,z]||aU[y,u];$$
$$(a||b)U[x,y] = aU[x,y]||\ bU[x,y];$$
$$(a;b)U[x,y] = aU[x,\ bU[x,y]];$$
$$(x\&y)||aU[x,(a^*)U[x,y]] \leq (a^*)U[x,y] \leq\ (x\&y)||\langle a^*\rangle(\sim q\&aU[x,y]).$$

В [Valiev, 1983] построена полная аксиоматизация генценовского типа для ПЛП с тестами и операцией обращения программ; из нее легко извлекается аксиоматизация для ограниченного как в данной работе варианта ПЛП. Используя эту аксиоматизацию, для ПЛП можно показать полноту некоторой аксиоматизации гильбертовского типа, которая переводится в алгебраическую форму в виде приведенных выше аксиом.

ii) Для $(a^*)U[...]$ доказывается обобщение теоремы Пратта о представлении оператора $\langle a^*\rangle$. А именно, пусть $!a(p,q)$ обозначает множество

$\{r\text{: } (p\&q)||aU[p,r] \leq r\}$.

Тогда имеет место

Утверждение 1. $(a^*)U[p,q]= min\ !a(p,q)$.

iii) В [Kowalsky, 2002] для динамических алгебр доказано свойство суперамальгамируемости (это свойство было определено Максимовой как очень сильное средство для исследования интерполяционного свойства для суперинтуиционистских и модальных логик (см. [Максимова, 1979]) и было обобщено для мультимодальных логик в [Madarasz, 1999]). Результат Ковальского переносится на proc-U-алгебры (при этом используется вышеприведенное утверждение 1.

iv) И наконец, интерполяционная теорема для ПЛП получается в результате применения теоремы из [Madarasz, 1999] (обобщающей результат из [Максимова, 1979]) о том, что для большого класса мультимодальных логик справедливость интерполяционной теоремы эквивалентна выполнению свойства суперамальгамируемости для соответствующих классов алгебр – компаньонов этих логик. ПЛП и proc-U-алгебры удовлетворяют условиям теоремы.

К сожалению, каждый из этапов i)-iii) довольно объемен и, к тому же, использует довольно сложные абстрактные понятия из теории булевых алгебр с модальными операторами; это не позволяет входить в дальнейшие подробности.

Интерполяционную теорему можно распространить и на случаи, когда элементарные программы интерпретируются как бинарные отношения или как частичные или тотальные функции. Кроме того, как уже было замечено выше, интерполяционная теорема, по-видимому, остается справедливой при добавлении тестов и операции обращения программ в определение допустимых программ. Гораздо более сложной представляется изучение интерполяционной теоремы для программных логик с более сложной семантикой, например, для логики процессов из [Harel et al, 1980] или пропозиционального μ-исчисления [Kozen, 1983].

Список литературы

[Максимова, 1979] Максимова Л.Л. Интерполяционные теоремы в модальных логиках и амальгамируемые многообразия топобулевых алгебр. // Алгебра и логика. 1979, т.18, № 5, 556-586.

[Максимова, 1989] Максимова Л.Л. Интерполяция, свойство Бета и временная логика «завтра». Препринт №30, Сиб. Отделение АН СССР, Институт математики, 1989.

[Craig, 1957] Craig W. Linear reasoning. A new form of the Herbrand-Gentzen theorem. // Journal of Symb. Logic. 1957, v. 22, 250-268.

[Fischer-Ladner, 1979] Fischer M.J., Ladner R.E. Propositional dynamic logic of regular programs. // Journal of Comput. Syst. Sci.. 1979, v. 18, 194-211.

[Harel et al, 1980] Harel D., Kozen D., Parikh R. Process logic: expressiveness, decidabikity, completeness. //21st IEEE Symp. on Foundat. Comput. Sci., 1980, 129-142.

[Kowalsky, 2002] Kowalsky T. PDL has interpolation. // Journal of Symb. Logic. 2002, v. 67, № 3, 933-946.

[Kozen, 1983] Kozen D. Results on the propositional mu-calculus. // Theor. Comput. Sci. 1983, v. 27, 333-354.

[Madarasz, 1999] Madarasz J.X. Interpolation and amalgamation: pushing the limits. // Part I. Studia Logica. 1998, v. 61, 311-345, Part II. Studia Logica. 1999, v. 62, 1-19.

[Pratt, 1979] Pratt V. Dynamic logic. Found. Of Computer Science. Math. Centre Tracts, № 109, Amsterdam, 1979.

[Pratt, 1991] Pratt V. Dynamic algebras: examples, constructions, applications. Studia Logica, 1991, v. 50, 571-601.

[Valiev, 1983] Valiev M.K On axiomatization of process logic. Lect. Notes Comput. Sci., № 148, 1983, 304-313.