Мальцевские чтения
Новосибирск (2005)
Тезисы докладов

Вычислимые Σ-спецификации иерархизированных моделей

В. Н. Глушкова

В рамках концепции Σ-программирования [**1**] в работах [**2, 3**] введен класс $\Delta_0 T$-формул, отличающихся от Δ_0-формул иерархизированным префиксом. Эти формулы имеют вид

$$(\forall x_1 \dot{\in} t_1) \dots (\forall x_m \dot{\in} t_m)(y_1 \prec z_1) \dots (y_p \prec z_p)\Phi(\bar{\alpha})$$

$m \geqslant 1$, $p \geqslant 0$, $\bar{\alpha}$ — последовательность переменных $x_1, \dots x_m$, $t_1, \dots, t_m$; $y_j, z_j \in \bar{\alpha}$, $1 \leqslant j \leqslant p$. Представляя все переменные префикса узлами с дугами от узла t_i к узлу x_i, получим дерево с корнем t_1. $\Delta_0 T$-формулы интерпретируются на моделях с иерархизированной надстройкой $D_G(C)$. Надстройка задается некоторой КС-грамматикой $G = (I, P)$ (I — множество символов, P — множество правил) над множеством констант $C = \{C_a\}_{a \in T}$, где $I = T \cup N$, T, N — множества терминальных и нетерминальных символов соответственно. Множество $D_G(C)$ определяется как наименьшее множество всех списков $\langle t_1, \dots, t_n \rangle$, формируемых для каждого правила $A \to X_1 \dots X_n \in P$, $n \geq 1$, $X_i \in I$, грамматики следующим образом: если $X_i \in T$, то t_i — произвольная константа из C_{X_i}, иначе ($X_i \in N$) — произвольный список сорта X_i. Список $\langle t_1, \dots, t_n \rangle$ имеет сорт A, $D_G(C) = \bigcup\limits_{A \in N} D_G^A(C)$.

Обозначим сигнатуру модели $\mathbf{M}$ через $\sigma = \langle I \cup \{list\}, C, F, R \rangle$, I — множество сортов модели, $list$ сорт списков из $D_G(C)$; F, R — множества функций и предикатов соответственно. Деревья разбора КС-грамматики G представляются в надстройке модели списком; отношение $\dot{\in}$ обозначает отношение принадлежности элемента списку или его транзитивное замыкание и соответствует отношению подчинения узлов дерева, $\prec$ обозначает отношение "левее".

Будем называть модель $CF(\mathbf{M})$ N-структурированным представлением модели M, если они имеют одни и те же носители, исключая носители нетерминальных сортов. Множество $D_G^A(C)$ является носителем сорта A, $A \in N$ в модели $CF(\mathbf{M})$.

Рассмотрим теорию из $\Delta_0 T$-квазитождеств, для которых формула $\Phi(\bar{\alpha})$ является импликацией вида: $\varphi(\bar{x}, \bar{t}) \to \psi(\bar{x}, \bar{t})$, где φ (ψ) конъюнкция атомных формул вида p, $\tau_1 = \tau_2$ (r, $f = \tau$); p, r, f — предикатные и функциональный символы τ, τ_1, τ_2 — термы. Пусть теория $Th = T_0 \cup Tr \cup H$ включает конечные множества T_0 замкнутых атомарных формул вида $f(\bar{c}) = c_i, p(\bar{c})$ и $Tr : \{c_i \in c_j \mid c_i, c_j \in C_t, C_t \subseteq C\}$. Отношения из Tr определяют дерево tr, в котором константы c_i являются узлами, такими что $sort(c_i) \in I$, $sort(c_j) \in N$; дереву tr соответствует некотороая последовательность правил грамматики G. Теория $H-$ это конечное множество универсальных $\Delta_0 T$-квазитождеств.

Теория Th трактуется как система определений функций и отношений, заданных на дереве tr, узлы которого представлены константами сответствующего сорта из C.

Для существования модели на теорию накладываются такие же ограничения как в [**4**].

Теорема 1. *Всякая ацикличная, определимая во множестве констант сигнатуры ς эквационально корректная теория Th из $\Delta_0 T$ формул имеет модель $\boldsymbol{M}$ из констант сигнатуры с КС-списочной надстройкой $D_G(C)$ изоморфную фактор системе свободной системы, порождаемой σ и задаваемой Th.*

Для теории Th строится стандартная модель, определяемая $T_0 \cup Tr$ и множеством порождающих C в квазимногообразии, определяемом H, по алгоритму [**5**].

Теорема 2. *Произвольная $\Delta_0 T-$формула с префиксом длины m интерпретируется на $N-$ структурированныом представлением модели $\boldsymbol{M}$ ей изоморфном с временной сложностью $O(n^{m+1})$ относительно мощности списков.*

Формулами выделенного класса можно специфицировать контекстно-зависимый синтаксис языков программирования и поведение простых технических систем с безальтернативным поведением [**2, 3**]. Для спецификации свойств сложных вычислительных и технических систем выразительных возможностей этих формул не достаточно, поскольку они не выражают альтернативных вариантов вычислений. Рассмотрим более широкий класс формул, правая часть импликации которых содержит разделительную дизъюнкцию и имеет вид: $\varphi(\bar{x}, \bar{t}) \to \psi_1(\bar{x}, \bar{t}) \oplus \psi_2(\bar{x}, \bar{t})$. Формулами этого вида можно специфицировать асинхронные параллельные реагирующие системы. Поведение таких систем во времени можно описывать посредством переходов, определяемых парой состояний (s, s') до и после выполнения некоторого действия (оператора программы), где состояние — это отображение $s : V \to D$ из множества переменных V в область значений D. Начальные состояния системы описываются формулами $\mathcal{S}$ над начальными значениями переменных. Для отношения переходов R строится формула $\mathcal{R}(V, V')$, представляющая этот переход, V, V' — переменные текущего и следующего состояния.

$\Delta_0 T$-квазитождествами, определяемыми над конкретными (или абстрактными) синтаксическими структурами можно специфицировать логическую семантику программы, которая отражает свойства исходных и конечных состояний, а также свойства промежуточных состояний. Даже если специфиация не является нетеровой (поскольку часто предполагают, что система работает неограниченно долго), можно построить некоторую конечную (графовую) модель, используя метод абстракции. Например, рассмотрим анализируемую в [**6**] параллельную программу со взаимным исключением (m, m' — метки входа и выхода).

$P = m : \textbf{cobegin}\, P_0 \parallel P_1 \,\textbf{coend}\, m'$, где

$P_0 :: l_0 : \textbf{while}\, True\, \textbf{do}$

$NC_0:$ **wait**$(turn = 0)$;
$CR_0:$ $turn := 1$;
end while
l'_0
$P_1 :: l_1:$ **while** $True$ **do**
$NC_1:$ **wait**$(turn = 1)$;
$CR_1:$ $turn := 0$;
end while
l'_1

Приведем фрагмент описания программы, где используются предикаты ACT (активный), ASL (ждущий), определенные на $Prog \rtimes Process \rtimes N \rtimes N \cup Process \rtimes Oper \rtimes N \rtimes N$, которые явно зависят от натуральных значений переменных pc_0, pc_1 — счетчиков команд процессов P_0, P_1. Их значения равны числу выполненных операторов в соответствующих процессах до начала исполнения оператора $oper$, для которого выполнено $ACT(P_i, L : oper, pc_0, pc_1)$.

1.$ACT(m : P, P_0, P_1) \to ASL(P, NC_0 : wait, 1, 0), ASL(P, NC_1 : wait, 0, 1)$
2.$ASL(P, L_0 : op_0, pc_1) \wedge ASL(P, L_1 : op_1, pc_1) \to ACT(P, L_0 : op_0, pc_0, pc_1) \oplus$
$ACT(P, L_1 : op_1, pc_0, pc_1)$
3.$ACT(P, NC_0 : wait, pc_0, pc_1)$,
$ASL(P, L_1 : op_1, pc_0, pc_1), Val(P, turn, pc_0 - 1, pc_1) = 0 \to$
$ASL(P, CR_0 : assign, pc_0 + 1, pc_1), ASL(P, L_1 : op_1, pc_0 + 1, pc_1)$,
$Val(P, turn, pc_0, pc_1) = Val(P, turn, pc_0 - 1, pc_1)$
4.$ACT(P, NC_0 : wait, pc_0, pc_1)$,
$ASL(P, L_1 : op_1, pc_0, pc_1), Val(P, turn, pc_0 - 1, pc_1) \neq 0 \to$
$ASL(P, NC_0 : wait, pc_0 + 1, pc_1), ASL(P, L_1 : op_1, pc_0 + 1, pc_1)$,
$Val(P, turn, pc_0, pc_1) = Val(P, turn, pc_0 - 1, pc_1)$
5.$ACT(P, CR_0 : assign, pc_0, pc_1)$,
$ACL(P, L_1 : op_1, pc_0, pc_1) \to Val(P, turn, pc_0, pc_1) = 1$,
$ASL(P, l_0 : P_0, pc_0 + 1, pc_1), ASL(P, L_1 : op_1, pc_0 + 1, pc_1)$

В спецификации программа P представляется конкретным КС-списком, определяемым абстрактной (конкретной) грамматикой языка. Свободные переменные pc_0, pc_1 связаны универсальным квантором всеобщности; переменные P_0, P_1, op_0, op_1 сорта "процесс", "оператор" и др. связаны ограниченными кванторами всеобщности по списку, конкретизирующему P.

Представим в виде путей графа всевозможные комбинации готовых к исполнению операторов обоих процессов. Для этого абстрагируемся от значений переменных pc_0, pc_1; пометим узлы графа значениями разделяемой переменной $turn$ и предикатами с аргументами ждущих исполнения операторов $ASL(P, L_0 : op_0), ASL(P, L_1 : op_1)$. Здесь $L_0 : op_0$, $L_1 : op_1$ — произвольные операторы процессов P_0, P_1 соответственно. В графе будем отождествлять узлы, помеченные предикатами ASL с одинаковыми значениями pc_0, pc_1. Начальные 2 узла соответствуют формуле $Val(P, turn, 0, 0) = 0 \oplus Val(P, turn, 0, 0) = 1$, $ASL(P, l_0 : P_0, 0, 0)$, $ASL(P, l_1 : P_1, 0, 0)$ и содержат соответственно значения: $ASL(P, l_0 : P_0)$, $ASL(P, l_1 : P_1)$, $turn = 0$ и $ASL(P, l_0 : P_0)$, $ASL(P, l_1 : P_1)$, $turn = 1$. Каждый узел содержит две дуги, помеченные соответственно предикатами $ACT(P, L_0 : op_0)$, $ACT(P, L_1 : op_1)$, От узла n к узлу m идет дуга, если в спецификации есть аксиомы, левая часть которых удовлетворяет конъюнкции условий, содержащихся в узле n и дуге. Узел m помечен предикатами, содержащимися в правой части аксиомы, конкретизированными соответствующими значениями переменных. В результате получим граф достижимых состояний модели Крипке [**6**]. По этому графу легко убедиться в том, что для приведенной программы обеспечивается свойство взаимного исключения: не могут два оператора с метками CR_0, CR_1 одновременно находиться в состоянии ожидания, формализуемое формулой $\neg\exists i, j(ASL(P, CR_0 : assign, i, j) \wedge ASL(P, CR_1 : assign, i, j))$.

Список литературы

[1] S.S. Goncharov and D.I. Sviridenko. Theoretical aspects of Σ- programming. Mathematical Methods of Specification and Synthesis of Software Systems'85. Proceed. of the Internat.Spring School. Springer-Verlag, April, 1985.

[2] Глушкова В.Н. Оценка сложности реализации логических спецификаций. Кибернетика и системный анализ. 1996, № 4.

[3] V.N.Glushkova.The execution complexity of logical formulas with restricted quantifiers based on CF-grammars.http://www.univ-paris12.fr/ lacl /LCCS2001/proceeding

[4] Ильичева О.А. Средства диагностики ошибок для систем логического прототипирования. Автоматика и телемеханика. 9,1997.

[5] Мальцев А.И. Алгебраические системы. М., 1970.

[6] Э.М. Кларк мл., О.Грамберг, Д. Пелед. Верификация моделей программ.М., 2002.

E-mail: `lar@aaanet.ru`