Полиномиально вычислимые Σ–спецификации иерархизированных моделей реагирующих систем.

Глушкова В.Н.

ДГТУ, lar@sfedu.ru

Аннотация. Рассматривается полиномиально-реализуемый класс $\Delta_0 T$– формул, интерпретируемый на многосортных моделях с иерархическими надстройками. Структура надстройки описывается произвольной КС-грамматикой. Для теорий из $\Delta_0 T$– квазитождеств, обладающих свойством нётеровости и конфлюентности, можно построить константную модель. Предикаты и функции сигнатуры модели интерпретируются на исходном КС-списке, достраиваемом в процессе интерпретации. Этот класс формул можно использовать для моделирования систем реального времени. Приводится пример логической спецификации управляющего устройства поведением робота.

The polynomial realized $\Delta_0 T$ –formulas are considered with quantifiers acting on hierarchy lists described by CF-grammars. These formulas are interpreted on a many-sorted model with the hierarchy list superstructure. The constant model is constructed for the Noether confluent theory based on quasi-identities. The signature predicates and functions are interpreted on input CF-list extended on the interpretation process. The $\Delta_0 T$ –formulas theories might be used to model real-time systems. As example a logic specification of robot behaviour is given.

1.Введение

Метод формальной верификации model cheking, успешно используемый в технологии разработки программных систем, основан на темпоральных логиках [1]. В этом подходе реальная система моделируется разновидностью конечного автомата, состояния которого помечаются набором атомных формул исчисления высказывания. Для спецификации функционирования моделируемой системы во времени ее поведение дискретизируется и конкретные значения времени игнорируются. Поэтому средствами темпоральной логики сложно выразить свойства о количественных характеристиках системы. Использование для моделей непрерывного времени общепринятого формализма временных автоматов [2] затрудняется экспоненциальными оценками алгоритмов верификации.

В предлагаемом подходе для спецификации моделируемых систем используется класс Δ_0 – формул многосортного языка ИП, выделенных в концепции семантического программирования [3]. В этой концепции, основанной на теоретико-модельном подходе, особую роль играют многосортные модели с надстройкой из конечных списков, формируемых из элементов исходной модели. В сигнатуру модели с надстройкой явно вводится сорт “ *list*” (список) и стандартные операции и отношения для элементов этого сорта. Списки используются для ограничения области изменения квантифицируемых переменных. А именно, вводятся ограниченные кванторы вида $\forall\, x\in t$, $\exists\, x\in t$, $\forall\, x\subseteq t$, $\exists\, x\subseteq t$, где x – переменная произвольного сорта, t – терм *list*-сорта (t не содержит x); $\in$– отношение принадлежности элемента списку, $\subseteq$ – отношение включения для списков.

2. **Используемые понятия**

Определим многосортную модель M сигнатуры $\sigma = < \hat{I}, C_M, F, R>$, где $\hat{I} = I \cup \{list\}$ – множество сортов, C_M, F, R – множества констант, функций и предикатов соответственно. Каждому сигнатурному символу приписан тип: n- местная функция $f \in F$, $n \geq 0$ имеет тип $<i_1, \ldots i_n, i>$, где $i_1, \ldots i_n, i \in \hat{I}$, i – тип значения функции, остальные символы – типы аргументов. Предикат $r \in R$ имеет тип $<i_1, \ldots i_n>$. Универс M состоит из индексированного семейства носителей $U_j = C_j$, $j \in I$, где C_j – множество констант сорта j; $f: U_{i_1} \times \ldots \times U_{i_n} \to U_i$,

$r \subseteq U_{i_1} \times ... \times U_{i_n}$.

Списочная надстройка $U_{list} = D_G(C)$ модели M состоит из иерархизированных списков, структура которых задается некоторой КС-грамматикой $G = (V, P)$, где V, P – множества символов и правил соответственно; $V = N \cup T$, где N, T – множества нетерминальных и терминальных символов. Каждому символу $A \in V$ приписан атрибут сорт $\rho(A)$ из множества сортов I. Множество $D_G(C)$ определяется как наименьшее множество всех списков $<t_1, ..., t_n>$, формируемых для каждого правила $A \to X_1 ... X_n \in P$, $n \geq 1$, $X_i \in I$ следующим образом: если $X_i \in T$, то t_i – произвольная константа из $C_{\rho(X_i)}$, в противном случае t_i – произвольный список сорта $\rho(X_i)$. Список $<t_1, ..., t_n>$ имеет сорт $\rho(A)$.

Δ_0–формулы определяются с использованием всех логических связок и ограниченных кванторов. Будем использовать лишь ограниченные кванторы вида $\forall x \in t$, $\forall x \overset{*}{\in} t$, где t – переменная сорта $list$, отношение $\overset{*}{\in}$ является рефлексивным транзитивным замыканием $\in$. Пусть $\bar{x}$ обозначает индексированную последовательность переменных x (многосортных), $\overset{\circ}{\in}$ – отношение $\in$ или $\overset{*}{\in}$, $\prec$ – отношение «левее» для списочных объектов.

Определение 2. Δ_0–формула вида

$$(\forall x_1 \overset{\circ}{\in} t_1) \ldots (\forall x_m \overset{\circ}{\in} t_m)\ (y_1 \prec z_1) \ldots (y_p \prec z_p)\ \Phi(\bar{x}, \bar{t}),\ m \geq 1,\ p \geq 0 \quad (1)$$

называется $\Delta_0 T$ – формулой, если $y_j, z_j \in (\bar{x}, \bar{t})$, $1 \leq j \leq p$, и переменные префикса удовлетворяют условию: для всех $1 \leq i \leq m$, $k \leq i$ выполняется: $t_{i+1} = t_i$ или $t_{i+1} = x_k$. Если $t_{i+1} = x_k$, то для всех $l \leq i$: $x_{i+l} \neq x_l$, $x_{i+l} \neq t_l$.

Если представить все переменные префикса узлами с дугами, ведущими от t_i к x_i, то получится дерево с корнем t_1.

Для спецификации реагирующих систем, функционирующих неограниченно долго, выразительных возможностей Δ_0–формул не достаточно.

Определение 3. Формула, полученная из $\Delta_0 T$– формулы навешиванием неограниченного квантора всеобщности, называется ΠT – формулой.

Модель M строится по теории Th из квазитождеств с отрицаниями вида:

$$(\forall x_1 \overset{\circ}{\in} t_1) \ldots (\forall x_m \overset{\circ}{\in} t_m)\ (y_1 \prec z_1) \ldots (y_p \prec z_p)(\varphi(\bar{x}, \bar{t}) \to \psi(\bar{x}, \bar{t})), \quad (2)$$

Формула $\psi(\varphi)$ – это конъюнкция атомных формул (или их отрицаний) вида: r, $\tau_1 = \tau_2$, $(f = \tau)$, τ_1, τ_2 – термы сигнатуры σ.

Для теорий, обладающих свойством нётеровости и конфлюентности, можно построить индуктивно вычислимую модель из констант C_M на основе правила вывода "modus ponens". При интерпретации аксиом теории арифметические операции считаются встроенными. Отрицание для предикатов интерпретируется по принципу "замкнутого мира". Интерпретация начинается, исходя из Th_0 –"подтеории" фактов, а именно –формул вида $r(\bar{c})$ или $\neg r(\bar{c})$, $f(\bar{c}) = c_n$, $\bar{c} \in C_M^*$, $c_n \in C_M$ и списка s_0 - начального значения переменной t_1 из (2). Списку s_0 в грамматике G соответствует вывод $A \overset{*}{\Rightarrow} \alpha_0$. Некоторые предикаты сигнатуры σ согласованы с нетерминальными символами из N, например, совпадают по имени. Выделенные аксиомы Ax теории Th также согласованы с правилами грамматики G и им приписана последовательность правил $p \in P^*$. Последовательность правил p применяется к выводу в грамматике G, построенному до момента интерпретации аксиомы Ax.

Пусть правая часть (2) содержит предикаты $Q_1, ... Q_m$ (в порядке их следования) и предикат $Q \in \{Q_1, ... Q_m\}$ определен на области $D_1 \times .... \times D_n$, тогда $Q \overset{*}{\Rightarrow} имя(D_1) ... имя(D_n)$.

Если это не вызывает противоречия, можно считать, что *имя*(D_j)=D_j, $1 \le j \le n$, тогда $Q \overset{*}{\Rightarrow} D_1 \dots D_n$. Если предикат Q имеет тип $<i_1, \dots i_n>$, то $\rho(D_j) = i_j$. Пусть $\overline{d} = <d_1, \dots d_n>$ составляет набор констант, на котором интерпретируется предикат Q в аксиоме Ax. Последовательность правил p составлена так, что $Q \overset{*}{\underset{p}{\Rightarrow}} X_1 \dots X_n$, $X_j \in T$, $1 \le j \le n$, $\rho(X_j) = \rho(d_j)$, поэтому в качестве терминальных символов X_j в последнем выводе берутся константы d_j. Таким образом, список, на котором интерпретируется теория *Th,* расширяется и дерево, соответствующее исходному списку, достраивается. В результате получается дерево, отражающее последовательность шагов интерпретации предикатов и функций, в результате которой область их определения расширяется до получения "неподвижной точки" вычисления.

Ограничивая зависимость переменных, входящих в префикс $\Delta_0 T$- формул, можно выделить класс полиномиально реализуемых формул относительно «размера» списка, определяемого количеством узлов соответствующего дерева вывода в G. Верификация модели M состоит в проверке свойств, которые формализуются теорией из произвольных $\Delta_0 T$ – формул, проверяемых за полиномиальное время. Причем степень полинома зависит от вида грамматики и префикса формул. Справедлива теорема.

Теорема. Произвольная $\Delta_0 T$–формула (*1*) с префиксом длины m интерпретируется на модели M с временной сложностью $O(n^{m+1})$ относительно мощности списков.

3. Пример

Приведем логическую спецификацию из $\Delta_0 T$ и ΠT – формул поведения управляющего устройства на примере робота, обслуживающего технологический автомат [4]. Роботизированный комплекс состоит из робота, позиции загрузки заготовок (*lp*), позиции разгрузки деталей (*ulp*) и технологического автомата (*ap*). Напротив позиций находятся соответствующие датчики. Цикл функционирования робота начинается с позиции загрузки.

ЦИКЛ:

1. На позиции *lp* для загрузки детали робот выдвигает исполнительный механизм (за время 3 сек.); берет заготовку, сжимая захват (1 сек.), вдвигает исполнительный механизм (3 сек) и начинает перемещение вправо к автомату до срабатывания датчика положения *ap*.
2. На позиции *ap* с целью установки заготовки на автомате робот выдвигает исполнительный механизм, разжимает захват (1 сек.), вдвигает исполнительный механизм. После ожидания 3 мин. робот разгружает автомат, повторяя те же процедуры, что и на позиции *lp*. Далее робот двигается влево к позиции разгрузки до срабатывания датчика положения *ulp*.
3. Деталь разгружается на транспортер за время 7 сек. Робот перемещается влево до срабатывания датчика *lp* на позицию загрузки и цикл работы комплекса повторяется.

Спецификация системы состоит из нескольких уровней. Поведение робота зависит от сигналов с датчиков его положения:*lp*, *ulp*, *ap* ($\neg lp, \neg ulp, \neg ap$, отрицание означает, что сигнал не вырабатывается). Эта последовательность сигналов представляется списком $mc=<x, y, z...>$, где $x = lp\ (\neg lp)$, $y=ulp\ (\neg ulp)$, $z= ap\ (\neg ap)$ и легко описывается конечным автоматом с начальным состоянием, помеченным *x*. Этот автомат можно построить по праволинейной грамматике с правилами:

$C \to lp\ C_1 \mid \neg lp\ C_1 \mid \varepsilon$

$C_1 \to ap\ C_2 \mid \neg ap\ C_2$

$C_2 \to ulp\ C \mid \neg ulp\ C$

Обозначим через *Dc* – множество списков, составленных из цепочек символов, порожденных этой грамматикой.

Количество переходов в автомате задаёт "внешнее" дискретное время (переменная *n* в логической спецификации). На втором уровне используется *КС* – грамматика, описываю-

щая последовательность действий *(Act)* робота: *L* , *La* – загрузка робота заготовкой в позиции *lp* и *ap* соответственно; *Ul* ,*Ula* – разгрузка детали в позиции *ulp* и *ap* ; *Rmove*, *Lmove* – движение робота направо и налево; *Stopl*, *Stopa*, *Stopul* – остановка робота в соответствующей позиции; *Wait* – ожидание; *Br* – поломка устройства управления робота. Действия робота влияют на его состояния (символ *St*). В данном примере состояние характеризуется "непрерывным" временем *Ctime* и дискретным *Dtime*, задаваемым натуральным числом. Значением сорта *Ctime* являются сегменты вида $<t_1,\ t_2>$, $t_1,\ t_2$ – константы, причем < заменяется на (или [в зависимости от того, включена левая граница в сегмент времени или нет, аналогично для >.

При спецификации поведения робота абстрагируются от значения времени движения робота от одной позиции до другой. Положение робота определяется сигналами с датчиков положения, являющимися входными к устройству управления робота. Выходными являются сигналы, подаваемые на исполнительные механизмы робота, для осуществления перемещений и работы захвата.

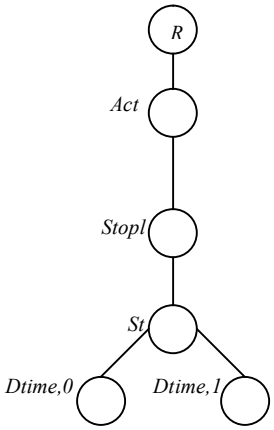
Правила грамматики *G* имеют вид:

1. $R \rightarrow \{Act\}^*$
2. $Act \rightarrow L \mid La \mid Ul \mid Ula \mid Rmove \mid Lmove \mid Stopl \mid Stopa \mid Stopul \mid Wait \mid Br$
3. $L \rightarrow St$
4. $La \rightarrow St$
5. $Ul \rightarrow St$
6. $Ula \rightarrow St$
7. $Rmove \rightarrow St$
8. $Lmove \rightarrow St$
9. $Stopl \rightarrow St$
10. $Stopa \rightarrow St$
11. $Stopul \rightarrow St$
12. $Wait \rightarrow St$
13. $Br \rightarrow St$
14. $St \rightarrow Ctime\ Dtime \mid Dtime\ Dtime$
15. $Ctime \rightarrow Dtime \mid (Dtime, Dtime) \mid [Dtime, Dtime) \mid (Dtime, Dtime] \mid [Dtime, Dtime]$

Dtime – класс лексем, значениями которых являются натуральные числа, вычисляемые во время интерпретации теории *Th.*

В теории *Th* переменные в формулах обозначаются мнемонично в соответствии с их сортом: $\rho(st) = \rho(St)$, $\rho(act) = \rho(Act)$, $\rho(n) = \rho(t) = \rho(Dtime)$, $\rho(ct) = \rho(Ctime)$. Предикаты *Lp, Ap, Ulp* определены на множестве *Dtime*,*Lp* (*n*) истинен, если робот находится на позиции загрузки, аналогично для *Ap* (*n*) – на позиции обрабатывающего автомата, *Ulp* (*n*) – на позиции разгрузки. Перечислим области определения остальных предикатов: *Stopl, Stopa, Stopul, Rmove, Lmove, Br* $\subseteq$ *Dtime*$\times$ *Dtime; L, La ,Ul, Ula, Wait*$\subseteq$ *Ctime*$\times$ *Dtime*. В формулах используются стандартные функции *head* $(<x_1,...x_n>) = x_1$, *tail* $(<x_1,...x_n>) = <x_2,...x_n>$ и функция *Mc*: *Dtime*$\rightarrow$ *Dc, Dc* – множество списков из сигналов датчиков.

В начальный момент времени *t =0, n=1* и на *1*-ом шаге вычисления выполняется предикат *Stopl*(*0,1*); *Mc* (*1*) =*mc*, где *mc*$\in$ *Dc*. В формулах 1-11 переменные *t,n* и переменная *ct* связаны ограниченным квантором $\forall\ st \in^* act,\ \forall\ t,n,\ ct \in st$. В формулах 12-17 переменная *n* связана неограниченным квантором $\forall$; $s_0 = <<<<\ 0,1>>>>$ – начальное значение списка, на котором интерпретируется теория; составляющие его списки в порядке глубины вложенности имеют сорта $\rho(R)$, $\rho(Act)$, $\rho(Stopl)$, $\rho(0) = \rho(1) = \rho(Dtime)$. Списку s_0 соответствует дерево T_0 :



В формулах теории справа в квадратных скобках приводится последовательность правил грамматики G, достраивающих дерево T_0.

Аксиомы теории:

1. $Stopl(t,n)$, $Lp\ (n) \rightarrow L([t\ ,t+7),\ n)$, $Mc\ (n+1) = tail(Mc\ (n))$ [1; 2.1; 3; 14.1; 15.3]
2. $L(ct,n)$, $Ap(n+1) \rightarrow Rmove\ (ct\ [2],\ n+1)$, $Stopa\ (ct\ [2],\ n+1)$ [1; 2.5; 7; 14.2;1; 2.8; 10; 14.2]
3. $Stopa\ (t,n) \rightarrow Ula\ ([t\ ,t+7],\ n)$ [1; 2.4; 6; 14.1; 15.2]
4. $Ula\ (ct,n) \rightarrow Wait\ ((ct\ [2],\ ct\ [2]+180),\ n)$ [1; 2.10; 11; 14.1; 15.2]
5. $Wait\ (ct,n) \rightarrow La\ ([ct\ [2],\ ct\ [2]+3),\ n)$, $Mc\ (n+1) = tail(Mc\ (n))$ [1; 2.2; 4; 14.1; 15.2]
6. $La\ (ct,n)$, $Ulp\ (n+1) \rightarrow Lmove\ (ct\ [2],\ n+1)$, $Stopul\ (ct\ [2],\ n+1)$ [1;2.6; 8; 14.2; 1; 2.9; 11; 14.2]
7. $Stopul\ (t,n\ ) \rightarrow Ul\ ([t,t+7),\ n)$, $Mc\ (n+1) = tail(Mc\ (n))$ [1; 2.3; 5; 14.1; 15.2]
8. $Ul\ (\ ct,n)$, $Lp(n+1) \rightarrow Lmove\ (ct\ [2],\ n+1)$, $Stopl\ (ct\ [2],\ n+1)$ [1; 2.6; 8; 14.2; 1; 2.7; 9; 14.2]
9. $Ul\ \ (ct,n)$, $\neg\ Lp\ (n+1) \rightarrow Br\ (ct\ [2],\ n+1)$ [1; 2.11; 13; 14.2]
10. $La\ (\ ct,n)\ ,\ \neg\ Ulp\ (n+1) \rightarrow Br\ (ct\ [2],\ n+1)$ [1; 2.11; 13; 14.2]
11. $L\ (\ ct,n)$, $\neg\ Ap(n+1) \rightarrow Br\ (ct\ [2],\ n+1)$ [1; 2.11; 13; 14.2]
12. $head\ (Mc\ (n)) =$ “lp” $\rightarrow Lp\ (n)$
13. $head\ (Mc\ (n)) =$ “$\neg\ lp$” $\rightarrow \neg\ Lp\ (n)$.
14. $head\ (Mc\ (n)) =$ “ap” $\rightarrow Ap(n)$
15. $head\ (Mc\ (n)) =$ “$\neg ap$” $\rightarrow \neg\ Ap(n)$
16. $head\ (Mc\ (n)) =$ “ulp” $\rightarrow Ulp\ (n)$
17. $head\ (Mc\ (n)) =$ “$\neg ulp$” $\rightarrow \neg\ Ulp\ (n)$

Теория Th обладает свойством нётеровости, т.к. изменение переменной под квантором $\forall$ ограничено k – количеством элементов в исходном списке mc и $head\ (Mc\ (k+1))$ не определено, т.к. $Mc\ (k+1) = <\ >$. Отметим, что цепочки “$\neg\ lp$” и др. в правой части аксиом 12-17 имеют сорт $string$ и ”$\neg$ ”рассматривается не как логическая операция, а как символ.

Для начального значения функции $Mc\ (1) = <lp,\ ap,\ ulp,\ lp,\ \neg\ ap,\ ulp>$ получаем множество следствий: $Stopl(0,1)$, $Lp\ (1)$, $L\ ([0,7),\ 1)$, $Mc\ (2) = <\ ap,\ ulp,\ lp,\ \neg\ ap,\ ulp>$, $Rmove\ (7,2)$, $Stopa(7,2)$, $Ula([7,14],\ 2)$, $Wait([14,\ 194],\ 2)$, $La([194,\ 197),\ 2)$, $Mc\ (3) = <ulp,\ lp,\ \neg\ ap,\ ulp>$, $Lmove(197,\ 3)$, $Stopul\ (197,\ 3)$, $Ul([197,\ 204],\ 3)$, $Mc\ (4) = <\ lp,\ \neg\ ap,\ ulp>$, $Lmove(204,\ 4)$, $Stopl(204,4)$, $L\ ([204,\ 211),\ 4)$, $Mc\ (5) = <\ \neg\ ap,\ ulp>$, $Br(211,\ 5)$. Полученное множество следствий иерархизируется в соответствии с выводом в грамматике G, полученным в результате правил, приписанных к интерпретируемым аксиомам. В соответствии с ними к дереву T_0 добавляются справа от узла, помеченного символом Act, еще 12 вершин, помеченных этим же символом, связанных ребрами с корнем. К новым вершинам подвешиваются поддеревья с корнями, помеченными символами Lp, $Rmove$, $Stopa$ и т.д. с их состояниями и константами сорта $\rho\ (Dtime)$, полученными в результате интерпретации.

На построенной модели можно проверять истинность произвольных $\Delta_0 T$– формул. Например, формализуем утверждение: если робот стоял на позиции загрузки в момент вре-

мени t на шаге n цикла его функционирования, то на шаге $n+2$ через 197 сек. он начинает разгрузку в течение 7 сек. Ниже приведенная формула верифицируется на заданном списке act сорта $\rho^{*}(Act)$:

$(\forall st \in act)\ (\forall t \in st)(\ \forall n \in st\ )\ (Stopl(t,n) \rightarrow\ Ul([t+197,\ t+204], n+2\ ).$

ЛИТЕРАТУРА

[1]Э.М. Кларк, мл. О. Грамберг, Д. Пелед. Верификация моделей программ. Изд-во Московского центра непрерывного математического образования. Москва, 2002.

[2] R. Alur, C. Courcoubetis, and D.L. Dill. Model-cheking for real-time system. In Proceedings of the 5th Annual Symposium on Logic in Computer Science. IEEE Computer Society Press, 1990, p.414-425.

[3]Goncharov S.S. and Sviridenko D.I. 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. pp. 169-179.

[4]. В.А. Горбатов, М.И. Смирнов, И.С. Хлытчиев. Логическое управление распределенными системами. М., Энергоатомиздат. 1991. 288 с.