УДК 629.7.05 + 004.41

К ВОПРОСУ О ПОСТРОЕНИИ СПЕЦИФИКАЦИИ ДЛЯ БОРТОВЫХ УПРАВЛЯЮЩИХ АЛГОРИТМОВ РЕАЛЬНОГО ВРЕМЕНИ

© 2011 А. А. Тюгашёв, А. Ю. Богатов

Самарский государственный аэрокосмический университет
имени академика С. П. Королёва
(национальный исследовательский университет)

Предлагается подход к решению проблемы спецификации управляющих алгоритмов реального времени, основанный на специально построенной формальной теории. Рассматривается возможность автоматизации синтаксической редукции спецификации управляющих алгоритмов.

Математическая модель, спецификация, управляющий алгоритм реального времени, эквивалентные преобразования, функциональная задача, система уравнений, бинарное дерево, предикат.

Введение

Основными требованиями, предъявляемыми к спецификации программ, являются точность, однозначность и полнота, а также возможность формальной верификации построенной спецификации.

С точки зрения модели пред- и постусловий Хоара [1] классическая последовательная или параллельная программа π работает корректно: $(A)\pi(B)\equiv 1$, если для любого набора данных, на котором истинен предикат A, после выполнения π получаем выходные данные, на которых истинен предикат B. В случае программ, основанных на алгоритмах реального времени, данное условие неприменимо и должно быть заменено следующим: $(A(D_0,t_0))\pi(B(D_1(t_1),D_2(t_2),...,D_k(t_k)))$,

где

$A(D_0,t_0)$ означает корректное задание начальных условий на момент времени t_0,

$B(D_1(t_1),D_2(t_2),...,D_k(t_k))$ означает, что в результате выполнения программы π были корректно выполнены целевые задачи в моменты времени $t_1,...,t_k$. Данная особенность приводит к существенному усложнению спецификации программ, реализующих управляющие алгоритмы реального времени (УА РВ). В работе рассматриваются методы синтаксической редукции спецификации УА РВ.

Алгебраическая система УА РВ

Управляющие алгоритмы реального времени можно представить в виде четвёрок объектов [2]:

$$УА\ РВ=\{<f_i,t_i,\tau_i,\bar{l}_i>\},\ i=\overline{1,N},$$

где f_i – функциональная задача (действие);

t_i – момент начала выполнения действия (целое неотрицательное число);

τ_i – длительность действия (целое неотрицательное число);

$\bar{l}_i$ – логический вектор, обуславливающий выполнение функциональной задачи.

Далее, пользуясь терминологией А. И. Мальцева [3], введём в рассмотрение двухосновную алгебраическую систему

$$<U,L;F;R>,$$

где

U – множество УА РВ в смысле наборов четвёрок объектов,

L – множество логических условий,

F – множество операций, определённых на декартовом произведении $U\times L$,

R – множество отношений между элементами множеств U и L.

Для описания алгебраической системы УА РА в работах [2, 4] было предложено исчисление управляющих алгоритмов реального времени. Данная формальная система представляет собой исчисление предикатов первого порядка. Ниже приводится определение терма, а также набор функциональных и предикатных символов расширенного исчисления УА РВ.

Определение терма вводится рекурсивно в соответствии со следующими правилами:

1) Символ функциональной задачи есть терм.

2) Если $T1$ и $T2$ – термы, а x – целое неотрицательное число, то

$T1$ СН $T2$,

$T1$ СК $T2$,

$T1 \to T2$,

Н $(T1, T2, x)$,

ЗА $(T1, T2, x)$,

$@(T1, x)$

являются термами.

Функциональные символы расширенного исчисления УА РВ

Функциональный символ СН описывает операцию алгебраической системы «совпадение по началу». Пусть даны термы

$$T1 = < T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} >$$

и

$$T2 = < T2, t_{T2}, \tau_{T2}, \overline{l_{T2}} >,$$

тогда

$$T3 = T1 \text{СН} T2 = < T3, t_{T3}, \tau_{T3}, \overline{l_{T3}} >.$$

Терм $T3$ содержит описания всех функциональных задач, входящих в $T1$ и $T2$:

$$t_{T3} = t_{T1},\ \tau_{T3} = \max(\tau_{T1}, \tau_{T2});$$

логические векторы, обусловливающие выполнение функциональных задач, не изменяются.

Функциональный символ СК описывает операцию алгебраической системы «совпадение по концу». Пусть даны термы

$$T1 = < T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} >$$

и

$$T2 = < T2, t_{T2}, \tau_{T2}, \overline{l_{T2}} >,$$

тогда

$$T3 = T1 \quad \text{СК} \quad T2 = < T3, t_{T3}, \tau_{T3}, \overline{l_{T3}} >.$$

Терм $T3$ содержит описания всех функциональных задач, входящих в $T1$ и $T2$. При этом для всех компонентов – функциональных задач, вошедших в состав $T3$ из участвующего в операции УА РВ с меньшей суммарной длительностью, необходимо провести операцию нормировки (сдвига) момента старта на величину

$$\Delta t_{T3} = \begin{cases} t_{T1} - t_{T2}, & \text{если } t_{T1} > t_{T2}, \\ t_{T2} - t_{T1}, & \text{если } t_{T2} > t_{T1}, \end{cases}$$

$$\tau_{T3} = \max(\tau_{T1}, \tau_{T2}).$$

Логические векторы, обусловливающие выполнение функциональных задач, не меняются.

Функциональный символ $\to$ описывает операцию непосредственного следования. Пусть даны термы

$$T1 = < T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} >$$

и

$$T2 = < T2, t_{T2}, \tau_{T2}, \overline{l_{T2}} >,$$

тогда

$$T3 = T1 \to T2 = < T3, t_{T3}, \tau_{T3}, \overline{l_{T3}} >,$$

где

$$t_{T3} = t_{T1},\ \tau_{T3} = \tau_{T1} + \tau_{T2}.$$

Логические вектора не меняются.

Функциональный символ Н описывает тернарную операцию «наложение со сдвигом». Пусть даны термы

$$T1 = < T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} >,$$

$$T2 = < T2, t_{T2}, \tau_{T2}, \overline{l_{T2}} >$$

и

величина сдвига t_S $(t_S \in Z_{\geq 0})$,

тогда

$$T3 = H(T1, T2, t_S) = < T3, t_{T3}, \tau_{T3}, \overline{l_{T3}} >.$$

Терм $T3$ содержит описания всех функциональных задач, входящих в $T1$ и $T2$, при этом

$$t_{T3} = t_{T1} + t_S,\ \tau_{T3} = \max(\tau_{T1}, \tau_{T2} + t_S).$$

Логические векторы, обусловливающие выполнение функциональных задач, не изменяются.

Функциональный символ ЗА описывает тернарную операцию «следование со сдвигом». Пусть даны термы

$T1 = \langle T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} \rangle$,

$T2 = \langle T2, t_{T2}, \tau_{T2}, \overline{l_{T2}} \rangle$

и величина сдвига t_S $(t_S \in Z_{\geq 0})$,

тогда

$T3 = ЗА(T1, T2, t_S) = \langle T3, t_{T3}, \tau_{T3}, \overline{l_{T3}} \rangle$.

Терм $T3$ содержит описания всех функциональных задач, входящих в $T1$ и $T2$, при этом

$t_{T3} = t_{T1}$, $\tau_{T3} = \tau_{T1} + \tau_{T2} + t_S$.

Логические векторы не изменяются.

Функциональный символ @ описывает операцию привязки начала выполнения УА к абсолютному значению времени. Пусть дан терм

$T = \langle T, t_T, \tau_T, \overline{l_T} \rangle$

и значение момента времени

t_0 $(t_0 \in Z_{\geq 0})$,

тогда

$T1 = \langle T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} \rangle$.

Терм $T1$ содержит описания всех функциональных задач, входящих в T, при этом $t_{T1} = t_0$, $\tau_{T1} = \tau_T$ и логические векторы не изменяются.

Предикатные символы расширенного исчисления УА РВ

Предикатные символы, одноимённые с функциональными, будем обозначать курсивом. Пусть U – множество управляющих алгоритмов реального времени.

Предикатный символ *СН* описывает бинарное отношение «совпадение по началу» на декартовом произведении $U \times U$. Для термов

$T1 = \langle T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} \rangle$

и

$T2 = \langle T2, t_{T2}, \tau_{T2}, \overline{l_{T2}} \rangle$

предикат $T1\,СН\,T2$ истинен, если $t_{T1} = t_{T2}$, и ложен в противном случае.

Предикатный символ *СК* описывает бинарное отношение «совпадение по концу» на декартовом произведении $U \times U$. Для термов

$T1 = \langle T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} \rangle$

и

$T2 = \langle T2, t_{T2}, \tau_{T2}, \overline{l_{T2}} \rangle$

предикат $T1\,СК\,T2$ истинен, если $t_{T1} + \tau_{T1} = t_{T2} + \tau_{T2}$.

Предикатный символ $\Rightarrow$ описывает бинарное отношение временного следования на декартовом произведении $U \times U$. Для термов

$T1 = \langle T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} \rangle$

и

$T2 = \langle T2, t_{T2}, \tau_{T2}, \overline{l_{T2}} \rangle$

предикат $T1 \Rightarrow T2$ истинен, если $t_{T1} + \tau_{T1} = t_{T2}$, и ложен в противном случае.

Предикатный символ < описывает бинарное отношение предшествования на декартовом произведении $U \times U$. Для термов

$T1 = \langle T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} \rangle$

и

$T2 = \langle T2, t_{T2}, \tau_{T2}, \overline{l_{T2}} \rangle$

предикат $T1 < T2$ истинен, если $t_{T1} < t_{T2}$, и ложен в противном случае.

Предикатный символ << описывает бинарное отношение сильного предшествования на декартовом произведении $U \times U$. Для термов

$T1 = \langle T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} \rangle$

и

$T2 = \langle T2, t_{T2}, \tau_{T2}, \overline{l_{T2}} \rangle$

предикат $T1 << T2$ истинен, если $t_{T1} + \tau_{T1} < t_{T2}$, и ложен в противном случае.

Предикатный символ <> описывает отношение несовместности по времени. Это бинарное отношение на декартовом произведении $U \times U$, где U – множество управляющих алгоритмов реального времени. Для термов

$T1 = \langle T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} \rangle$

и

$T2 = \langle T2, t_{T2}, \tau_{T2}, \overline{l_{T2}} \rangle$

предикат $T1 <> T2$ истинен, если $(t_{T1} + \tau_{T1} < t_{T2}) \vee (t_{T2} + \tau_{T2} < t_{T1})$, и ложен в противном случае.

Предикатный символ </> описывает бинарное отношение несовместности по логике на декартовом произведении $U \times U$. Для термов

$$T1 = < T1, t_{T1}, \tau_{T1}, \overline{l_{T1}} >$$

и

$$T2 = < T2, t_{T2}, \tau_{T2}, \overline{l_{T2}} >$$

предикат </> истинен, если несовместны логические векторы, обусловливающие выполнение $T1$ и $T2$, и ложен в противном случае.

Построение спецификации с помощью формул исчисления УА РВ

Будем называть функциональную задачу вполне определённой, если для неё известны время начала t, длительность τ и обусловливающий её выполнение логический вектор α. В противном случае будем называть функциональную задачу частично определённой.

Пусть требуется описать спецификацию взаимодействия управляющих алгоритмов, состоящих из N функциональных задач: $f_1,..., f_N$, при этом $K \le N$ задач вполне или частично определены. Тогда спецификацию можно записать в виде

$$\begin{cases} P_1(f_1,..., f_N), \\ P_2(f_1,..., f_N), \\ \mathbf{K} \\ P_M(f_1,..., f_N), \\ f_1: \quad t_1 = t'_1, t_1 = t'_1, a_1 = a'_1, \\ f_2: \quad t_2 = t'_2, t_2 = t'_2, a_2 = a'_2, \\ \mathbf{L} \\ f_K: \quad t_K = t'_K, t_K = t'_K, a_K = a'_K. \end{cases}$$

Здесь $P_i(f_1,..., f_N)$ представляет собой формулу, выражающую композицию операций или отношений алгебраической системы УА РВ для функциональных задач $f_1,..., f_N$.

Моделирование операций и отношений алгебраической системы системой уравнений

Из приведённых выше определений отношений и операций алгебраической системы УА РВ следует, что любое отношение на множестве функциональных задач однозначно определяется отношениями <, >, = на декартовом произведении $N \times N \times \{0,1\}^J$, где J — размерность логического вектора, обусловливающего выполнение функциональных задач данного множества. Аналогично, любая операция на множестве функциональных задач однозначно определяется стандартными операциями умножения и сложения на множестве целых чисел. Это позволяет проводить описание УА РВ с помощью системы алгебраических уравнений относительно времени начала и длительности частично определённых функциональных задач.

Содержательно алгоритм построения алгебраической модели по спецификации УА РВ состоит из следующих шагов:

1) спецификация переводится в ПОЛИЗ;

2) для каждого оператора в записи выполняются следующие правила:

– если оператор выражает отношение на множестве функциональных задач, то определяются время начала, длительность, координаты условного вектора операндов и формулируется уравнение или неравенство, соответствующее данному отношению. Каждая часть полученного соотношения умножается на характеристическую функцию соответствующего логического вектора;

– если оператор определяет операцию на множестве функциональных задач, то определяются время начала, длительность, координаты условного вектора операндов и над ними выполняются преобразования, задаваемые данной операцией.

Оптимизация

Оптимизирующие преобразования алгебраической модели заключаются в нахождении решения полученной системы уравнений относительно переменных, соответствующих времени начала и длительности частично определённых функциональных задач. При этом возможны три варианта:

1) Система является определённой. В этом случае оптимизация может считаться завершённой. Все функциональные задачи являются вполне определёнными. В этом случае спецификация не будет содержать ни одной формулы.

2) Система несовместна. Спецификация некорректна, то есть содержит условия, противоречивые с точки зрения логики или по времени.

3) Система является неопределённой. Решение представляет собой выражение значений логических и временных характеристик одних функциональных задач (свободные переменные системы) через логические и временные характеристики других функциональных задач (базисные переменные). При этом возможна оптимизация, заключающаяся в минимизации выражений свободных переменных через базисные.

Формальные преобразования спецификации УА РВ

Из приведённых выше определений отношений алгебраической системы УА РВ следуют тождества:

1) $T_1\ СН\ T_2 = T_2\ СН\ T_1$,

2) $T_1\ СК\ T_2 = T_2\ СК\ T_1$,

3) $T_1 + T_2 = T_2 + T_1$,

4) $(T_1 \to T_2) \to T_3 = T_1 \to (T_2 \to T_3)$,

5) $(T_1\ СН\ T_2)\ СН\ T_3 = T_1\ СН\ (T_2\ СН\ T_3)$,

6) $(T_1\ СК\ T_2)\ СК\ T_3 = T_1\ СК\ (T_2\ СК\ T_3)$,

7) $(T_1 \to T_2) СН (T_1 \to T_3) = T_1 \to (T_2\ СН\ T_3)$,

8) $(T_1 \to T_2) СК (T_3 \to T_2) = (T_1\ СК\ T_3) \to T_2$,

9) $(T_1 \to T_2) + (T_1 \to T_3) = T_1 \to (T_2 + T_3)$,

10) $(T_1 \to T_2) + (T_3 \to T_2) = (T_1 + T_3) \to T_2$,

11) $(T_1\ СН\ T_2) + (T_1\ СН\ T_3) = T_1\ СН\ (T_2 + T_3)$,

12) $(T_1\ СН\ T_2) + (T_3\ СН\ T_2) = (T_1 + T_3)\ СН\ T_2$,

14) $(T_1\ СК\ T_2) + (T_1\ СК\ T_3) = T_1\ СК\ (T_2 + T_3)$,

15) $(T_1\ СК\ T_2) + (T_3\ СК\ T_2) = (T_1 + T_3)\ СК\ T_2$,

16) $(a_1 = 1) \Rightarrow (T_1 \to T_2) = ((a_1 = 1) \Rightarrow$
$\Rightarrow T_1) \to ((a_1 = 1) \Rightarrow T_2)$,

17) $(a_1 = 1) \Rightarrow (T_1\ СН\ T_2) = ((a_1 = 1) \Rightarrow$
$\Rightarrow T_1) СН ((a_1 = 1) \Rightarrow T_2)$,

18) $(a_1 = 1) \Rightarrow (T_1\ СК\ T_2) = ((a_1 = 1) \Rightarrow$
$\Rightarrow T_1) СК ((a_1 = 1) \Rightarrow T_2)$,

19) $T_1\ СН\ T_1 = T_1$,

20) $T_1\ СК\ T_1 = T_1$,

21) $(\alpha \Rightarrow T) + (\neg\alpha \Rightarrow T) = T$,

22) $(\alpha \Rightarrow (\beta \Rightarrow T)) = (\alpha \wedge \beta) \Rightarrow T$.

С помощью этих тождеств можно осуществить синтаксическую редукцию (эквивалентные преобразования, сокращающие длину формулы) спецификации УА РВ.

Рассмотрим пример. Пусть в спецификации присутствует формула

$$((f_1 \to f_2) СН (f_1 \to f_3))\ СК (f_4 \to (f_2\ СН\ f_3)).$$

Применяя тождества 7, 8, эту формулу можно преобразовать к виду
$(f_1\ СК\ f_4) \to (f_2\ СН\ f_3)$.
При этом происходит уменьшение количества операций в два раза.

В качестве внутреннего представления оптимизируемых формул управляющего алгоритма будем рассматривать их двоичные деревья синтаксического разбора. Оптимизация двоичных деревьев проводится в два этапа:

1) идентификация двоичных поддеревьев;

2) применение к поддеревьям аксиомы сжатия, если это возможно.

Существует несколько способов идентификации одинаковых поддеревьев. Первый подход – метод прямого сравнения, то есть для каждого узла сравниваем все поддеревья. Второй подход – метод простых чисел. Все узлы двоичного дерева нумеруются простыми числами, операции $СН, СК, \to, +$ получают коды 2, 3, 5, 7. Узлы, соответствующие одной и той же операции или одной и той же функциональной задаче, нумеруются одинаковыми числами. Для идентификации поддерева производится перемножение кодов всех узлов, входящих в него. Некоммутативные операции $\to$ и + для различения левого и правого поддеревьев в случае равенства произведений используют приписывание знака минус («—»). Недостатком метода простых чисел является быстрое переполнение. Поэтому вместо него возможно применение метода «целых» чисел, в котором узлы-поддеревья кодируются целыми числами. Однако при этом необходимо иметь в памяти таблицу, описывающую каждый подузел с информацией о его левом, правом поддереве и месте в дереве.

Рассмотрим алгоритм проведения эквивалентных преобразований на бинарных деревьях. Этот алгоритм начинает обработку с самых нижних поддеревьев, а затем, поднимаясь выше, охватывает все большее количество узлов. Поддеревья соединяются с помощью узла-операции, причём в конструкции $D_1\ f\ D_2$ (поддерево, над которым в текущий момент работает алгоритм) поддеревья D_1 и D_2 уже отработаны, поэтому по ним надо спуститься максимум на один – два уровня. Таким образом, процесс эквивалентных преобразований носит индуктивный характер. Базисом индукции является исходное двоичное дерево представления алгоритма D_0. Шаги индукции проводятся применением аксиом исчисления управляющих алгоритмов. На каждом шаге индукции получаем эквивалентное предыдущему двоичное дерево D_j. Последовательность $D_0, D_1, ..., D_k$ длины $k+1$ назовём выводом $D_0 \to D_k$.

Заключение

Необходимо заметить, что приведённые методы синтаксической редукции спецификации УА РВ уменьшают количество информационных связей в результирующей управляющей программе, понижая тем самым её структурную сложность (сложность управляющего графа, связанного с потоком управления). Кроме того, алгебраический метод синтаксической редукции формул позволяет проверить корректность задания временных и логических условий управляющего алгоритма. Данный подход приводит к квазиоптимальному решению (как таковая, строго говоря, задача оптимизации в классической постановке не ставится).

Таким образом, метод формальных преобразований обеспечивает квазиоптимальность построенного решения.

Библиографический список

1. Мальцев, А. И. Алгебраические системы [Текст] / А. И. Мальцев. – Москва: Наука, 1970. – 400 с.

2. Касьянов, В. Н. Графы в программировании: обработка, визуализация и применение [Текст] / В. Н Касьянов, В. А. Евстигнеев. – СПб.: БХВ – Петербург, 2003. – 1104 с.

3. Калентьев, А. А. Автоматизированный синтез алгоритмов асинхронного управления техническими системами с множеством дискретных состояний [Текст] / А. А Калентьев. – Самара: СГАУ, 1998. – 204 с.

4. Тюгашев, А. А. Синтез и верификация управляющих алгоритмов реального времени для бортовых вычислительных систем космических аппаратов [Текст]: дис. … д-ра техн. наук / А. А. Тюгашев. – Самара: Изд-во СГАУ, 2007. – 312 с.

CONSTRUCTING THE SPECIFICATION FOR THE ON-BOARD REALTIME CONTROL ALGORITHM

© 2011 A. A. Tyugashev, A. Yu. Bogatov

Samara State Aerospace University named after academician S. P. Korolyov
(National Research University)

An approach to solving the problem of the on-board algorithm specification is proposed. This approach is founded on the calculus of a realtime control algorithm. The possibility of the automation of syntax reduction for the specification is also considered.

Mathematical model, specification, control algorithm, equivalent optimizing transformations, functional task, system of linear equations, binary tree, predicate.

Информация об авторах

Тюгашёв Андрей Александрович, доктор технических наук, профессор кафедры компьютерных систем. Самарский государственный аэрокосмический университет имени академика С. П. Королёва (национальный исследовательский университет). Область научных интересов: разработка ИПИ-технологий на проблемно- ориентированном уровне. E-mail: tau797@mail.ru.

Богатов Артём Юрьевич, аспирант, ассистент кафедры компьютерных систем. Самарский государственный аэрокосмический университет имени академика С. П. Королёва (национальный исследовательский университет). Область научных интересов: теория алгоритмов, CALS-технологии. E-mail: artmbogatov@yandex.ru.

Tyugashev Andrey Alexandrovitch, doctor of technical sciences, professor of the department of computer systems, Samara State Aerospace University named after academician S. P. Korolyov (National Research University), tau797@mail.ru. Area of research: informatics and CALS-technology.

Bogatov Artyom Yurievitch, assistant of the department of computer systems, Samara State Aerospace University named after academician S. P. Korolyov (National Research University), artmbogatov@yandex.ru. Area of research: theory of algorithms and CALS-technology.