

Калентьев А.А., Тюгашев А.А.

ПРОБЛЕМА СЕМАНТИКО-СИНТАКСИЧЕСКОЙ ЭКВИВАЛЕНТНОСТИ МОДЕЛЕЙ УПРАВЛЯЮЩИХ АЛГОРИТМОВ РЕАЛЬНОГО ВРЕМЕНИ

Вопрос о синтаксисе и семантике программ, толковании, взаимосвязи и различии данных понятий волнует ученых, разработчиков программного обеспечения с момента самого зарождения понятия о программировании, возможно, даже еще с более раннего времени.

Поскольку в системе ГРАФКОНТ, разрабатываемой в качестве системы автоматизированной разработки программ (CASE-системы), одним из используемых входных языков является символический язык, построенный на основе формального исчисления (теории) управляющих алгоритмов (УА), интерпретируемой на алгебраической системе УА с основными операциями СН (совпадение по началу), СК (совпадение по концу), \supseteq (следование) во временном пространстве и + (логический выбор), \Rightarrow (следствие) в логическом пространстве очевидно, что первый тип семантической эквивалентности синтаксически различных конструкций связан с интерпретацией формул символического входного языка и использованием аксиом и правил вывода исчисления УА для эквивалентных преобразований формул.

Пример:

$$t1 := (f001 \text{ CH } f002) \text{ CH } f003$$

$$t2 := (f001 \text{ CH } f003) \text{ CH } (f003 \text{ CH } f002)$$

$$t3 := (a1=0) \Rightarrow t1 + (a1=1) \Rightarrow t2$$

$$t4 := t1$$

Здесь структура термы $t1$ и $t2$ синтаксически различна, а семантически они описывают один и тот же управляющий алгоритм за счет семантики операции СН. Более того, значимость проверки значения логической переменной $a1$ в терме $t3$ вообще утеряна, и в итоге термы $t1$, $t2$, $t3$ и $t4$ семантически эквивалентны между собой.

Чтобы дать точное определение семантически эквивалентных управляющих алгоритмов реального времени (УА РВ), необходимо дать определение понятию семантики УА РВ.

Для этого обратимся к функциям УА РВ. Главная их задача – координировать работу в реальном времени функциональных программ низшего уровня, управляющих работой от-

дельных приборов, систем, агрегатов на борту космического аппарата (КА), в зависимости от складывающейся на борту ситуации, которая находит свое отражение в значениях вектора логических переменных.

В связи с этим дадим следующее определение семантики УА (включать в нужные моменты времени нужные функциональные программы (ФП) в зависимости от значений логического вектора):

$$УА = \{ \langle f_i, t_i, \tau_i, I_i \rangle \}, i = 1, N,$$

где f_i - идентификатор i -ой запускаемой на выполнение ФП,

t_i - момент запуска i -ой ФП,

τ_i - длительность i -ой ФП,

I_i - логический вектор, обуславливающий запуск ФП.

То есть, управляющий алгоритм рассматривается как набор четверек, каждая из которых описывает один запуск на выполнение функциональной программы в рамках решения целевой задачи.

Еще одной формой представления, уже фактически на уровне реализации УА РВ, является его многоходовая модель, фактически отражающая логико-временную структуру управляющей программы. Графическим образом многоходовой модели, наглядно описывающим ее и одновременно являющимся одним из главных видов технической документации на УА РВ, является временная диаграмма.

Поскольку интуитивно ясно, что для одной и той же семантики УА РВ в приведенном выше смысле возможно несколько реализаций в виде различных логико-временных схем, то и изображать временную диаграмму можно по-разному.

Простейший вариант синтаксической разницы (понимаемой в данном случае как разницы логико-временных структур реализующих УА РВ управляющих программ) получается в рамках одного входа (включения) УА за счет возможности разной последовательности проверки логических переменных, как это показано на рисунке 1.

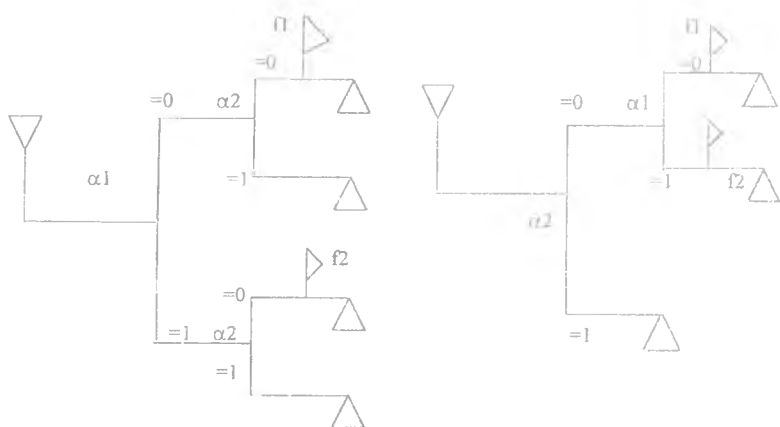


Рис. 1

При этом обе диаграммы семантически эквивалентны. Практический смысл в выявлении семантически эквивалентных, но синтаксически различных структур состоит в возможности проведения квазиоптимизации, приводящей к сокращению, например, объемов потребляемой памяти. В приведенном на рисунке 1 примере, в частности, можно избавиться от лишних логических ветвей, связанных с проверкой логической переменной α !

Вторым видом семантической неоднозначности является, при условии реализации УА в виде многоходовой модели, степень связи между моментом времени проверки значения логической переменной и его актуальностью. Данный вид неоднозначности иллюстрирует рисунок 2. Надо ли проверять значение логической переменной при каждом новом (в иной момент времени) включении управляющего алгоритма? Можно значительно сократить количество проверок условий, если допустить, что нас интересует значение логической переменной только при его первой проверке. В то же время, возможна и иная ситуация, когда актуальным значением логической переменной является ее значение на момент начала выполнения всего алгоритма, а не на момент первой проверки именно ее значения в каком-либо входе

Третий вид семантической неоднозначности – неоднозначность, также связанная с логическими переменными. Он связан с возможным наличием логических переменных двух

сортос – тех, которые меняют свое значение на протяжении выполнения алгоритма, и тех, которые не меняют

Наконец, четвертый вид семантической неоднозначности связан с иным подходом к определению семантики УА РВ в случае, когда важной является именно последовательность действий – то есть, в рамках одного включения (момента времени) необходимо задавать также, какие ФП необходимо выполнять раньше, какие за ними и какие проверки значений логических переменных осуществлять после этого.

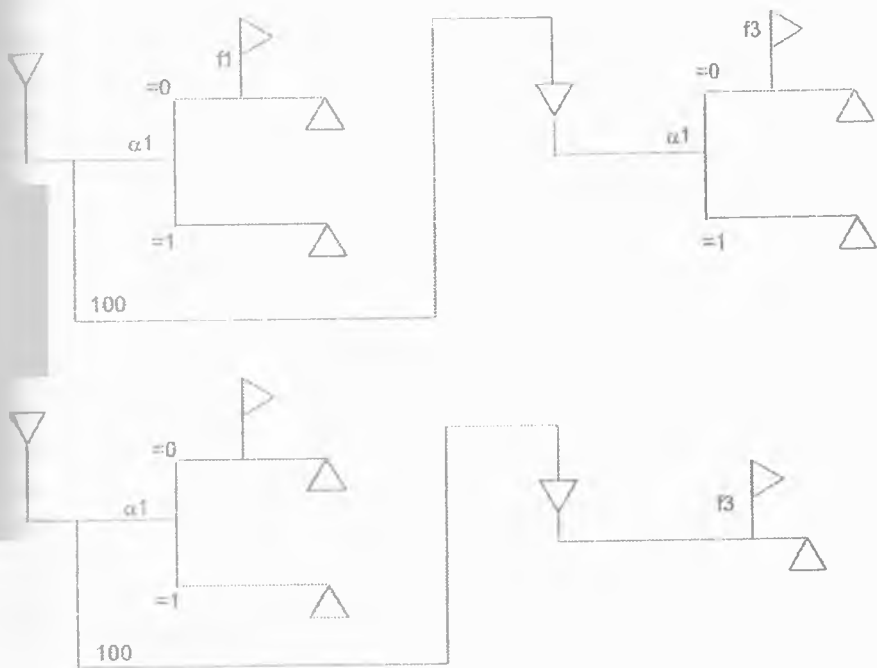


Рис. 2