

ОПТИМИЗИРУЮЩИЕ ПРЕОБРАЗОВАНИЯ НАД МНОЖЕСТВОМ БОРТОВЫХ УПРАВЛЯЮЩИХ АЛГОРИТМОВ РЕАЛЬНОГО ВРЕМЕНИ

Высокая сложность управляющих алгоритмов реального времени, огромное количество вычислений, а также жёсткие требования, предъявляемые к быстрдействию программ и объёму используемой оперативной памяти, делают весьма актуальной задачу оптимизации управляющих алгоритмов.

Задача оптимизации управляющего алгоритма реального времени π заключается в поиске таких преобразований $\Phi(\pi)$, что выполняются следующие два условия:

1. Эквивалентность. \forall предикатов P и Q выполняется $(P(D_0, t_0))\pi(Q(D_1(t_1), D_2(t_2), \dots, D_k(t_k))) \rightarrow (P(D_0, t_0))\Phi(\pi)(Q(D_1(t_1), \dots, D_k(t_k)))$. $P(D_0, t_0)$ означает корректное задание начальных условий на момент времени t_0 , а $Q(D_1(t_1), D_2(t_2), \dots, D_k(t_k))$ означает, что в результате выполнения программы были корректно выполнены целевые задачи в моменты времени t_1, \dots, t_k .

2. Оптимальность. Для выбранного критерия оптимальности, задаваемого предикатом K , должно выполняться тождество: $(K)\Phi(\pi) \equiv 1$.

В качестве оптимизируемого параметра выбрано суммарное количество операций в управляющем алгоритме.

Для решения поставленной задачи предлагаются следующие подходы:

1. Алгебраический подход. Управляющий алгоритм моделируется системой линейных алгебраических уравнений. Каждое уравнение описывает временную и логическую связь между функциональными задачами алгоритма. Если решение системы существует, то оно соответствует оптимальному алгоритму. Если решения не существует, то в исходном алгоритме присутствуют противоречивые логико-временные условия.

2. Графовый подход. Управляющий алгоритм моделируется взвешенным орграфом. В терминах графовых грамматик – это начальный граф. Затем на основании априорно известных соотношений между группами функциональных задач строятся правила переписывания графа (графовая продукция). Если некоторое правило переписывания позволяет сократить число операций (правая часть правила содержит меньше операций, чем левая), то оно применяется к начальному графу. Данный подход определяет квазиоптимальный управляющий алгоритм.

Разработан метод оптимизации визуального представления временных сечений управляющего алгоритма:

1. Выполняется обход графа исходной логико-временной схемы и для каждой вершины, соответствующей функциональной задаче, выписывается полный логический вектор.

2. Для каждой функциональной задачи сравниваются логические вектора и формируется результирующий вектор по следующему принципу: если во всех логических векторах значение i -ой переменной не меняется, в результирующий вектор записывается это значение, иначе – значение H .

3. Рассчитывается вес для каждой логической переменной.

4. Сортируется массив логических переменных в порядке убывания весов.

5. Синтезируется оптимальная логико-временная схема по следующему принципу: на j -ом шаге изображаются вершины для функциональных задач, зависящих ровно от j первых логических переменных; изображаются соответствующее количество вершин, соответствующих j -ой логической переменной.