ՀԱՅԱՍՏԱՆԻ ԳԻՏՈՒԹՅՈՒՆՆԵՐԻ ԱԶԳԱՅԻՆ ԱԿԱԴԵՄԻԱՅԻ ԶԵԿՈՒՅՑՆԵՐ
ДОКЛАДЫ НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК АРМЕНИИ

Том 98 1998 №2

ИНФОРМАТИКА

УДК 519.71

**К. Б. Амбарцумян, З. А. Галстян, А. Е. Карапетян,
Г. Г. Мадатян, К. В. Шахбазян**

Алгоритм стабилизационного эксперимента для произвольного конвейера и его программная реализация на языке Microsoft Visual C++

(Представлено академиком НАН Армении Ю.Г.Шукуряном 24/ III 1998)

В статье предлагается алгоритм стабилизационного эксперимента для конвейерных систем и его программная реализация на языке Microsoft Visual C++.

Введем необходимые понятия и обозначения. Конвейером будем называть сеть из функциональных элементов, обладающих единичной задержкой ([1]). Рассмотрим класс помеченных графов $G=(V,U)$. Такие графы будем называть конвейерными графами. Каждому конвейерному графу G соответствует конвейер $K(G)$, и обратно, каждому конвейеру K соответствует конвейерный граф $G(K)$. Через G^* будем обозначать фактор-граф графа G. Пусть B – произвольная вершина графа G^*, тогда ей соответствует некоторая бикомпонента $B=(V_B,U_B)$ графа G. Если q – внутреннее состояние конвейера, то q_B – внутреннее состояние бикомпоненты B при состоянии конвейера q.

Стабилизационный эксперимент состоит в выяснении возможности перехода конвейера при фиксированном состоянии входов в некоторое устойчивое состояние.

Алгоритм стабилизационного эксперимента для произвольного конвейера. Предлолагаем, что каждой бикомпоненте $B=(V_B,U_B)$ графа G сопоставлен вес $L(B)$. Алгоритм заключается в выполнении последовательности шагов конвейера и сравнении каждой пары последовательных значений. Вес $L(B)$ используется для выяснения стабильности бикомпоненты B. А именно $L(B)$ – это число шагов, в течение которых достигается стабильное состояние, если оно есть. Пусть B – произвольная вершина графа G^*. Легко видеть, что прежде чем решать задачу стабилизации для B, нужно убедиться в стабильности всех бикомпонент B', предшествующих B. Лишь после стабилизации

всех предшествующих бикомпонент может начаться эксперимент для B. В процессе моделирования шагов конвейера $K(B)$ динамически определяется момент начала эксперимента для B. Будем говорить, что начиная с этого шага и до установления факта стабилизации бикомпонента B находится в состоянии тестирования. Как говорилось в (2), B может находиться в этом состоянии не более чем $k(B)|A|^{\frac{|U_B|}{k(B)}}$ шагов, где A – конечный алфавит, $k(B)$ – индекс расщепления (3), а U_B множество дуг B.

Алгоритм

Шаг 1. $y := q_0$ (начальное состояние конвейера).

Просмотр всех вершин B графа G^*, различающий следующие случаи:

1) если B не имеет предшественников, то $\sigma(B) := 2$ (т.е. все такие вершины приводятся в состояние тестирования); $n(B) := 0$ (счетчик числа шагов тестирования);
2) если B имеет предшественников, то $\sigma(B) := 1$ (тестирование еще не начато).

Шаг 2. Очередной шаг конвейера, т.е. вычисление q – нового состояния конвейера (или, что то же самое, нового выхода).

Шаг 3. Если $y = q$, то стабилизация достигнута. Конец алгоритма.

Шаг 4. Если $y \neq q$, то для каждой B рассмотреть следующие случаи:

1) если $\sigma(B) = 2$ и B – пустая, то $\sigma(B) := 3$ (тестирование окончено);
2) если $\sigma(B) = 2$, B – непустая и $y_B = q_B$, то $\sigma(B) := 3$;
3) если $\sigma(B) = 2$, B – непустая, $y_B \neq q_B$ и $L(B) = n(B)$, то стабильности нет. Конец алгоритма;
4) если $\sigma(B) = 2$, B – непустая, $q_B \neq y_B$, $n(B) < L(B)$, то $n(B) := n(B) + 1$.

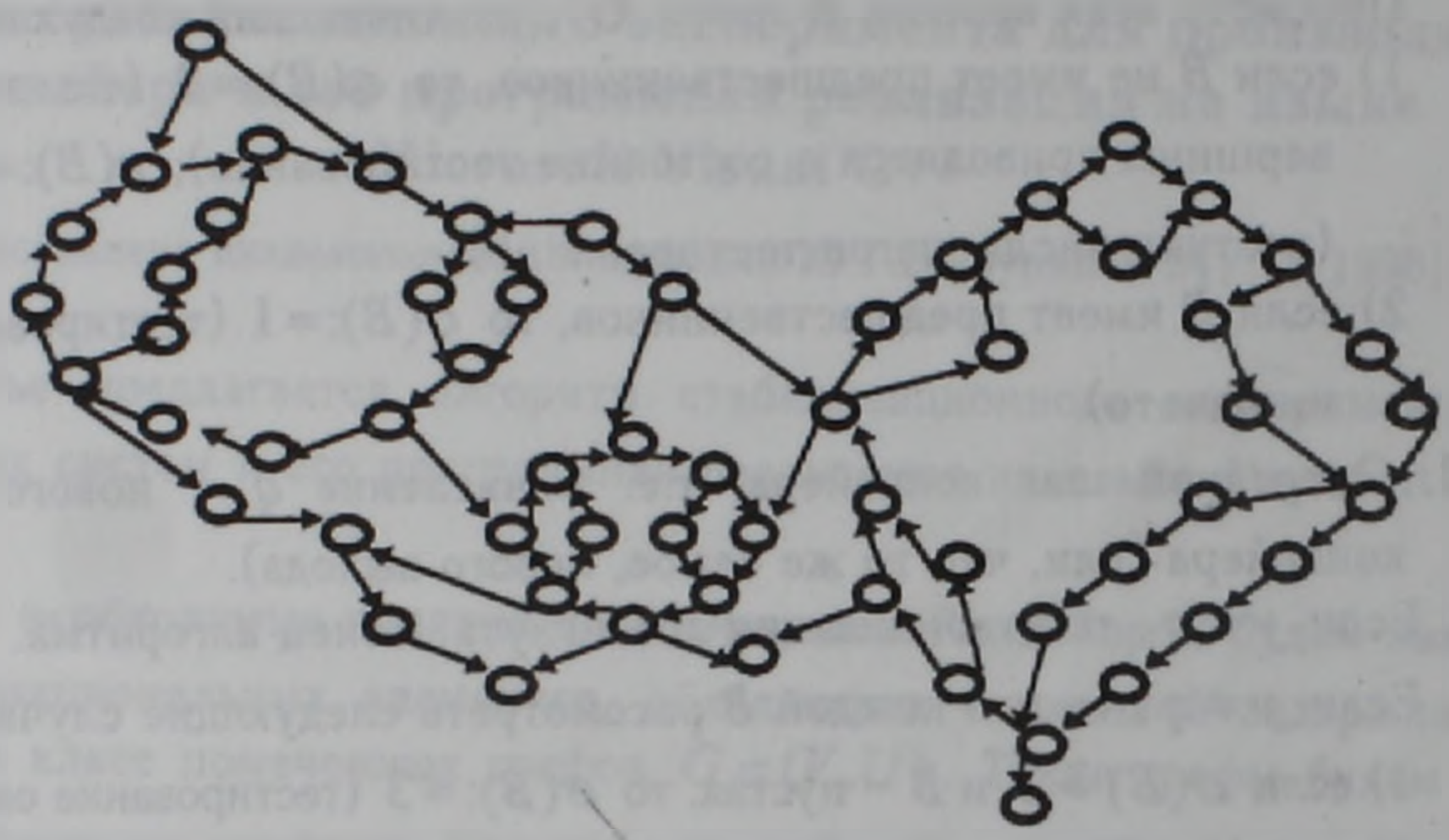
Шаг 5. Просмотр всех бикомпонент B таких, что $\sigma(B) = 1$. Если для B имеет место $\sigma(B) = 1$ и все ее предшественники B' в графе G^* имеют $\sigma(B') = 3$, то для такой бикомпоненты положим $\sigma(B) = 2$ и $n(B) := 0$ (начать тестирование B).

Шаг 6. Если существуют бикомпоненты, для которых $\sigma(B) = 2$, то $y := q$ и перейти к шагу 2. В противном случае стабилизация достигнута. Конец алгоритм.

Программная реализация алгоритма стабилизационного эксперимента для схемы. Для предложенного алгоритма на языке Microsoft Visual C++ была разработана программа Stabilization Experiment for Conveyer (Ver 1.0), работающая как под управлением операционной системы MS-DOS, так и Microsoft Windows 95. Программа для заданного начального состояния и для заданного постоянного входа (значения которого в Ver 1.0 могут быть только булевски-

ми) выясняет возможность стабилизации схемы. Предполагается, что функциональные элементы схемы вычисляют элементарные булевые функции. Программа на вход принимает схему в виде графа-конвейера. Он определяется числом вершин графа-конвейера (по умолчанию – одна вершина), а также для каждой вершины заданием булевой функции, выходной дуги, определяемой парой вершин, номером аргумента функции и значением на данной дуге.

Программа дает возможность зафиксировать выбранные дуги, откорректировать уже имеющиеся. Пользователю дается возможность запомнить введенный в файл граф (используется расширение .gra) для неоднократного использования.



Для выбранной схемы активизируется алгоритм стабилизации конвейера, после работы которого выдается информация о состоянии стабильности конвейера и количестве сделанных шагов. При желании пользователь может просмотреть пошаговую реализацию алгоритма и в каждый момент времени увидеть состояние любой вершины.

Рассмотрим пример, приведенный на рисунке. Для нахождения стабильности данного конвейера методом "наивного" моделирования необходимо выполнить 2^{60} шагов конвейера, программа же выдает ответ через 15 шагов.

Институт проблем информатики и автоматизации НАН Армении

Կ. Բ. ՀԱՄԲԱՐՁՈՒՄՅԱՆ, Զ. Ա. ԳԱԼՍՏՅԱՆ, Հ. Ե. ԿԱՐԱՊԵՏՅԱՆ,
Գ. Հ. ՄԱԴԱԹՅԱՆ, Կ. Վ. ՇԱՀԲԱԶՅԱՆ

Հարահոսի կայունության փորձարկման ալգորիթմը և նրա ծրագրային իրականացումը Microsoft Visual C++-ով

Հարահոսային կազմակերպումների վերլուծության ժամանակ, կապված վերիֆիկացիայի պրոբլեմի հետ հարց է ծագում հարահոսների կայունության մասին՝ տրված հաստատուն մուտքի համար: Հոդվածում նկարագրված է հարահոսային համակարգերի կայունության փորձարկման ալգորիթմը և նրա ծրագրային իրականացումը Microsoft

Visual C++ *լեզվով: Կայունության փորձը հիմնված է հարահոսներում ընթացող պրոցեսների դեցոմպոզիցիայի հնարավորության վրա:*

ЛИТЕРАТУРА – ԳՐԱԿԱՆՈՒԹՅՈՒՆ

[1] *В.Б.Кудрявцев, С.В.Алешин, А.С.Подколзин*, Введение в теорию автоматов. М., 1985. [2] *К.В.Шахбазян*, Зап. науч. семинаров ПОМИ., т.243, с.15-27 (1998). [3] *К.В.Шахбазян, Т.А.Тушкина, М.М.Лебединский*, Зап. науч. семинаров ЛОМИ., т.159, с.40-46 (1987).