УДК 519.718

**А.М. РОМАНКЕВИЧ, И.В. МАЙДАНЮК, М.И. ХАИТОВ**

***Национальный технический университет Украины «КПИ», Украина***

# ОБ ОДНОМ АЛГОРИТМЕ ФОРМИРОВАНИЯ РЕБЕРНЫХ ФУНКЦИЙ ГРАФО-ЛОГИЧЕСКИХ МОДЕЛЕЙ ОТКАЗОУСТОЙЧИВЫХ МНОГОПРОЦЕССОРНЫХ СИСТЕМ

*Работа посвящена алгоритму формирования реберных функций циклической графо-логической модели (GL-модели) базовой отказоустойчивой многопроцессорной системы (ОМС), которая теряет минимальное количество ребер при отказе компонентов, количество которых превышает базовою степень отказоустойчивости. Приведено ряд утверждений и формул, которые дают возможность формировать рассматриваемую модель в обход относительно сложной процедуры минимизации, что позволяет уменьшить вычислительную сложность процесса формирования данной модели.*

***Ключевые слова:*** *отказоустойчивые многомодульные системы, графо-логические модели, расчет надежности.*

## Введение

Задача повышения надежности цифровых вычислительных систем становится все более актуальной в последние десятилетия. Это касается, в первую очередь, систем управления критическими и сложными объектами, отказ которых может привести к значительным потерям, в том числе к человеческим жертвам (самолеты, космические ракеты, АЭС и др.).

Одним из перспективных направлений в области создания высоконадежных систем управления является построение отказоустойчивых реконфигурируемых многопроцессорных систем (ОМС) [1], свойства и способы расчета надежности которых достаточно интенсивно исследуются [2, 3].

Исследование реакции ОМС на появление отказов, расчет ее надежностных характеристик, поиск «узких» мест являются весьма актуальными задачами, которые можно решить, анализируя поведение ОМС в потоке отказов. Зачастую для такого анализа используются различные математические модели поведения ОМС в потоке отказов, в том числе и рассматриваемая в данной работе графологическая модель.

Одним из универсальных способов расчета надежности (а точнее вероятности безотказной работы) ОМС является проведение статистических экспериментов с графо-логическими моделями (GL-моделями). Для больших систем, не поддающихся декомпозиции, проведение статистического эксперимента и формирование самой модели требует значительных временных и вычислительных ресурсов. В связи с этим, вопросы сокращения времени формирования модели и выполнения статистического эксперимента с ними приобретают особый интерес.

## 1. Постановка задачи

Как отмечалось, в работе рассматривается графо-логическая модель (GL-модель) поведения ОМС в потоке отказов. GL-модель - неориентированный циклический граф R, ребрам которого приписаны определенные булевы функции, аргументами которых являются двоичные индикаторные переменные $x_i$ (i=1..n), отображающие работоспособность соответствующего компонента системы ("0" – отказ, "1" – исправен). Если функция, приписываемая тому или иному ребру, принимает нулевое значение, то это ребро исключается. Связность графа R соответствует работоспособности системы в целом. Последовательность значений (как реальных, так и смоделированных) переменных назовем вектором состояния системы.

ОМС, состоящую из n процессоров и сохраняющую работоспособность при отказе не более, чем m ее компонентов, будем называть базовой и обозначать как K(m,n).

Реальные ОМС часто являются небазовыми, т.е. такими, которые продолжают функционировать при появлении векторов состояния с большим, чем m количеством нулей. Зачастую при решении задачи повышения надежности ограничиваются увеличением кратности отказоустойчивости системы не более, чем на единицу. Модели небазовых ОМС строятся на основе моделей базовых ОМС, причем сложность такой трансформации [4] зависит от количества ребер, которые исчезают в базовой модели. В связи с этим важным является создание базовой GL-модели, в которой при появлении m+1 отказа исчезает 2 ребра (необходимый минимум для потери циклическим графом связности).

На данный момент известно достаточно много способов формирования реберных функций GL-модели. Одним из первых был разработан рекуррентный алгоритм формирования так называемой канонической GL-модели [5]. Далее в [6] предложен алгоритм минимизации канонической модели. Полученная после минимизации GL-модель теряет минимальное число ребер при появлении вектора с m+1 нулевой компонентой, т.е. 2. Обозначим такую модель как 2p-модель. Указанная модель значительно проще, чем каноническая, но при этом сложность ее построения по [6] значительно выше. В свете этого определенный интерес представляет алгоритм формирования 2p-модели с меньшей сложностью.

## 2. Базовый алгоритм формирования модели

Вначале скажем, что основная идея алгоритма построения канонической модели заключается в последовательном разделении множества компонентов (индикаторных переменных) системы на два равных или почти равных непересекающихся подмножества и перебор всех вариантов распределения по ним текущего количества отказов, к которым система должна быть устойчива. Этот процесс продолжается до тех пор, пока количество отказов в текущем подмножестве либо не станет равным мощности подмножества, либо не будет равно единице. Далее записывается функция, соответствующая каждому такому распределению, и приписывается одному ребру графа модели.

Рассмотрим процесс разбиения множества переменных более детально.

### 2.1. Дерево разбиения множества индикаторных переменных

Обозначим множество всех переменных модели K(m,n) как $\alpha_1=\{x_1,x_2,...,x_n\}$. Как было сказано, в процессе построения модели проводится последовательное деление множества $\alpha_1$ пополам. Множества, полученные в результате такого разбиения, будем нумеровать определенным образом: номера множеств, на которые непосредственно разделяется множество $\alpha_\lambda$ ($\lambda$ - номер множества) , определяются как $\alpha_{2\cdot\lambda}$, $\alpha_{2\cdot\lambda+1}$, а их мощности соответственно $[|\alpha_\lambda|/2]$ и $]|\alpha_\lambda|/2[$. Например, множество $\alpha_1$ разбивается на множества $\alpha_2=\{x_1,...,x_{[n/2]}\}$ и $\alpha_3=\{x_{[n/2]+1},...,x_n\}$ с мощностями $[n/2]$ и $]n/2[$, соответственно, где $[x]$ – целая часть от x, $]x[$ – округление к ближайшему большему целому. Ясно, что $\alpha_\lambda=\alpha_{2\cdot\lambda}\cup\alpha_{2\cdot\lambda+1}$.

Вообще деление пополам множества можно производить как угодно, но условимся о следующем:

– все элементы $\alpha_\lambda$ всегда упорядочены по возрастанию индексов; вследствие этого будем иметь дело не с множеством, а с кортежем, но по-прежнему будем называть его множеством,

– разделение множества $\alpha_\lambda$ на два будем производить таким образом, что в $\alpha_{2\cdot\lambda}$ входят переменные с меньшими порядковыми номерами.

Процесс разбиения множества переменных можно представить в виде дерева, на котором вершины обозначаются соответствующими множествами и их номерами (рис. 1.). Если множество А является подмножеством В, то между соответствующими вершинами есть путь.

На дереве можно выделить уровни, которые соответствуют итерациям разбиения множества $\alpha_1$ пополам. Понятно, что на уровне i (уровни нумеруются с нуля) находятся вершины с номерами $2^i,\dots,2^{i+1}-1$, и они упорядочены по возрастанию слева направо. Таким образом, по номеру множества $\alpha_\lambda$ можно определить уровень, которому это множество принадлежит: $i(\lambda)=[\log_2\lambda]$. На рис. 1 приведен пример такого дерева для модели K(4,15), на котором кроме самих множеств, представлены их номера.

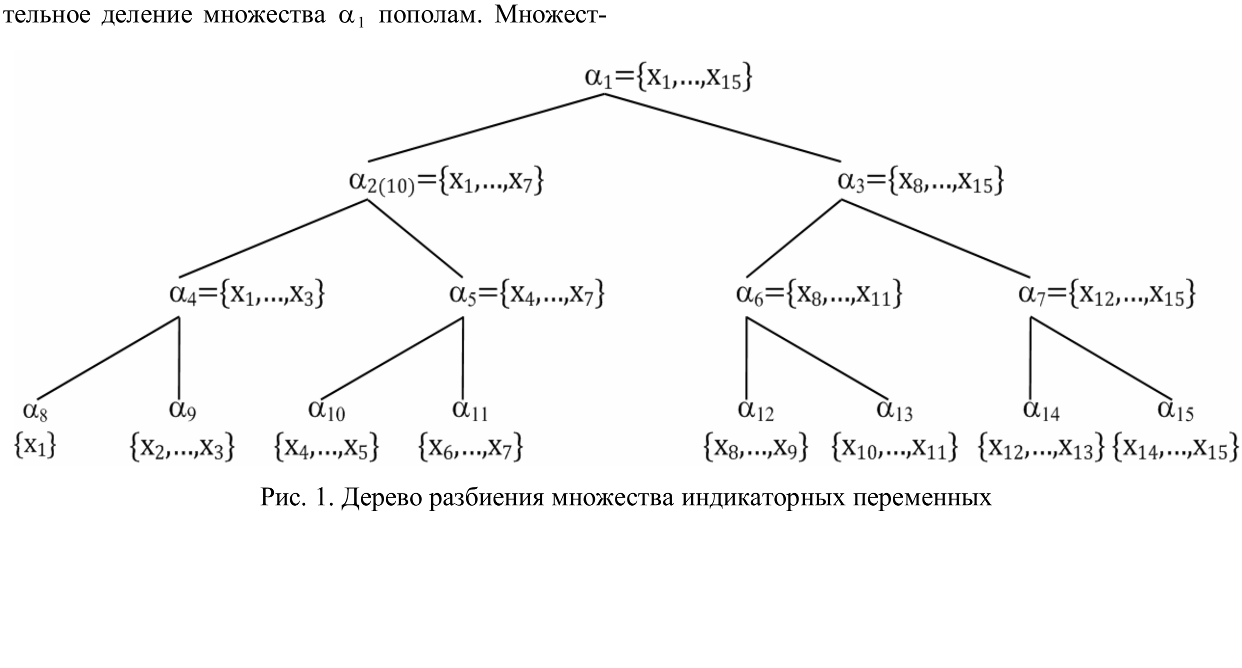


Рис. 1. Дерево разбиения множества индикаторных переменных

Отметим, что разбиение множества $\alpha_\lambda$ выполняется лишь при условии $|\alpha_\lambda| > 2$. Таким образом, нижним является уровень дерева, на котором мощность максимального из множеств равна двум.

Выше изложенное дает возможность однозначно идентифицировать множество по его номеру, т.е. зная номер множества можно определить его содержимое, что позволяет в дальнейших изложениях обходиться только номерами.

### 2.2. Базовый алгоритм формирования реберных функций

Вслед за [минимиз] обозначим через K'(m,n) множество реберных функций GL-модели K(m,n), а $K'(m, \alpha_\lambda)$ – множество реберных функций модели, которые зависят от конкретного множества переменных $\alpha_\lambda$. Каждый элемент из K'(m,n) приписан только одному ребру.

Пусть $w(\varphi)$ – количество равных 0 переменных из множества $\varphi$. Пусть функция $f(m_x,\alpha_\lambda)=0$, если $w(\alpha_\lambda)\geq m_x$, и $f(m_x,n_x)=0$, если $w(X)\geq m_x$, где X произвольное множество мощностью в $n_x$ элементов.

Понятно, что существует достаточно большое множество формульных представлений функции $f(m_x,\alpha_\lambda)$. Далее будет использоваться то, которое получается путем конъюнкции всех реберных функций множества $K'(m_x, \alpha_\lambda)$:

$$f(K'(m_x, \alpha_\lambda)) = \mathop{\&}_{\forall g \in K'(m_x,\alpha_\lambda)} g = f(m_x, \alpha_\lambda), \qquad (1)$$

где g – реберная функция модели $K(m_x, \alpha_\lambda)$.

Справедливость приведенного выражения объясняется тем, что при $w(\alpha_\lambda)\geq m_x$ в множестве $K'(m_x, \alpha_\lambda)$ по крайней мере одна реберная функция примет нулевое значение, и в результате вся конъюнкция будет равна 0.

Большая часть операций в алгоритме формирования множества реберных функций 2р-модели, изложенного в [6], реализует следующее выражение:

$$K'(m_x,\alpha_\lambda) = K'(m_x,\alpha_{2\lambda}) \cup$$
$$\bigcup_{i=1}^{m_x-1} \left[ f(K'(m_x - i, \alpha_{2\lambda})) \vee f(K'(i, \alpha_{2\lambda+1})) \right] \cup$$
$$\cup K'(m_x, \alpha_{2\lambda+1}), \qquad (2)$$

где $|\alpha_\lambda| > m_x$, $m_x - i \leq |\alpha_{2\lambda}|$, $i \leq |\alpha_{2\lambda+1}|$.

При первом обращении к (2) вместо переменной $m_x$ подставляются собственно значение m, а $\alpha_\lambda = \alpha_1$. Далее итерационно формируются соответствующие множества K'. Напомним, что существует два тривиальных случая, при появлении которых не выполняется обращение к правой части выражения (2), а соответствующие множества формируются следующим образом:

$$K'(1, \alpha_\lambda) = f(1, \alpha_\lambda) = \mathop{\&}_{\forall x \in \alpha_\lambda} x, \qquad (3)$$

$$K'(|\alpha_\lambda|, \alpha_\lambda) = f(|\alpha_\lambda|, \alpha_\lambda) = \bigvee_{\forall x \in \alpha_\lambda} x. \qquad (4)$$

Таким образом, посредством введенных в данной работе понятий, описаны ключевые операции алгоритма формирования реберных функций модели, предложенного в [минимиз]. Заметим, что сложность формирования 2р-модели по рассматриваемому алгоритму, можно оценить сверху величиной $n \cdot m^{\log 2(n/m)}$ обращений к выражению (2).

Рассмотрим другую идею решения той же задачи и соответствующий алгоритм.

## 3. Ускоренный алгоритм формирования реберных функций 2р-модели

Для начала условимся, что порядок формирования конъюнкции (1) соответствует последовательности вызовов выражения (2) слева направо. В этом случае в общем виде функция $f(m_x,\alpha_\lambda)$ может быть записана следующей скобочной формой:

$$f(m_x,\alpha_\lambda)=(f(m_x,\alpha_{2\lambda})\vee f(0,\alpha_{2\lambda+1}))\&(f(m_x-1,\alpha_{2\lambda})\vee$$
$$\vee f(1,\alpha_{2\lambda+1}))\& \ldots \& (f(0,\alpha_{2\lambda})\vee f(m_x,\alpha_{2\lambda+1})). \qquad (5)$$

Будем считать, что выражение (5) представляет собой конъюнкцию дизъюнкций двух компонент, причем компонента (которая также является функцией) может быть представлена подобным образом. В случае $m_x - i > |\alpha_{2\lambda}|$ либо $i > |\alpha_{2\lambda+1}|$ соответствующая дизъюнкция компонент (а именно $f(m_x-i, \alpha_{2\lambda}) \vee f(i, \alpha_{2\lambda+1})$) не входит в состав выражения (5). Следовательно, если $|\alpha_\lambda| \geq 2m_x$, то конъюнкция состоит из $m_x+1$ дизъюнкции, в противном случае $|\alpha_\lambda| - m_x + 1$. Далее к выражению (5) будем обращаться как к форме представления функции, подставляя соответствующие значения вместо $m_x$.

Введем в форму (5) индексацию компонент:

$$f(m_x,\alpha_\lambda) = \mathop{\&}_{j=1}^{k} (f_{j,1} \vee f_{j,2}), \qquad (6)$$

где $k=\min(m_x+1, |\alpha_\lambda|-m_x+1)$, т.е. общее количество дизъюнкций в конъюнкции.

Фактически $f_{j,1}$ – функция $f(m_x - \left]\frac{m_x-k+1}{2}\right[ - j+1, \alpha_{2\lambda})$, а $f_{j,2}$ – $f(\left]\frac{m_x-k+1}{2}\right[ + j-1, \alpha_{2\lambda+1})$. Первый индекс соответствует порядковому номеру дизъюнкции, а второй – номеру компоненты в дизъюнкции.

Приведем ряд утверждений, которые позволят путем простых преобразований формировать одни функции, используя компоненты другой.

Как сказано выше, можно выделить два варианта соотношения между величинами $|\alpha_\lambda|$ и m. Рассмотрим случай $|\alpha_\lambda|\geq 2m$.

*Утверждение 1.* Функция $f(m\text{-}1,\alpha_\lambda)$ может быть сформирована на основе компонент функции $f(m,\alpha_\lambda)$ представленной в виде (6), следующим образом:

$$f(m\text{-}1,\alpha_\lambda)=\mathop{\&}_{j=1}^{m}(f_{j+1,1}\vee f_{j,2}), \qquad (7)$$

где $f_{j+1,1}$, $f_{j,2}$ – компоненты функции $f(m,\alpha_\lambda)$.

Распишем функцию $f(m, \alpha_\lambda)$ по (5), отметив ее компоненты штрихами:

$$f(m,\alpha_\lambda)=(f'(m,\alpha_{2\lambda})\vee f'(0,\alpha_{2\lambda+1}))\&(f'(m\text{-}1,\alpha_{2\lambda})\vee$$
$$\vee f'(1,\alpha_{2\lambda+1}))\&$$
$$\&\ldots\&(f'(0,\alpha_{2\lambda})\vee f'(m,\alpha_{2\lambda+1})).$$

Запишем теперь функцию $f(m\text{-}1,\alpha_\lambda)$ в таком же виде:

$$f(m\text{-}1,\alpha_\lambda)=(f(m\text{-}1,\alpha_{2\lambda})\vee f(0,\alpha_{2\lambda+1}))\ \&\ (f(m\text{-}2,\alpha_{2\lambda})\vee$$
$$\vee f(1,\alpha_{2\lambda+1}))\ \&\ldots\&\ (f(0,\alpha_{2\lambda})\vee f(m\text{-}1,\alpha_{2\lambda+1})).$$

Сравнив эти выражения, легко увидеть, что $f(m, \alpha_\lambda)$ включает в себя все компоненты $f(m\text{-}1, \alpha_\lambda)$. Кроме того, отметим, что компонента $f'_{j+1,1}$ функции $f(m, \alpha_\lambda)$ равняется компоненте $f_{j,1}$ функции $f(m\text{-}1, \alpha_\lambda)$, как и $f'_{j,2}=f_{j,2}$, где j=1..m, что свидетельствует о справедливости формулы (7) и всего утверждения. Приведем пример:

$$f(3,7)=(f(3,3)\vee f(0,4))\cdot(f(2,3)\vee f(1,4))\cdot(f(1,3)\vee f(2,4))\cdot$$
$$\cdot(f(0,3)\vee f(3,4));$$
$$f(2,7)=(f(2,3)\vee f(0,4))\cdot(f(1,3)\vee f(1,4))\cdot(f(0,3)\vee$$
$$f(2,4)).$$

Последовательное многократное применение соотношения (7) делает справедливым следующее утверждение:

*Утверждение 2*. Функция $f(m\text{-}i,\alpha_\lambda)$ может быть сформирована из компонент функции $f(m,\alpha_\lambda)$, представленной в виде (6), где i=0…m, следующим образом:

$$f(m\text{-}i,\alpha_\lambda)=\mathop{\&}_{j=1}^{m-i+1}(f_{j+i,1}\vee f_{j,2}), \qquad (8)$$

где $f_{j+i,1}$, $f_{j,2}$ – компоненты функции $f(m,\alpha_\lambda)$.

Теперь рассмотрим случай, когда $|\alpha_\lambda|\leq 2m$. Пусть $m_1=[|\alpha_\lambda|/2]=|\alpha_{2\lambda}|$, а $m_2=]|\alpha_\lambda|/2[=|\alpha_{2\lambda+1}|$.

*Утверждение 3.* Функция $f(m_2\text{-}1, \alpha_\lambda)$ может быть сформирована из компонент функции $f(m_2, \alpha_\lambda)$ представленной в виде (6), следующим образом:

$$f(m_2\text{-}1,\alpha_\lambda)=\mathop{\&}_{j=1}^{m_2}(f_{j+1,1}\vee f_{j,2}), \qquad (9)$$

где $f_{j+1,1}$, $f_{j,2}$ – компоненты функции $f(m_2,\alpha_\lambda)$. В случае нечетного значения $|\alpha_\lambda|$ в запись $f(m_2,\alpha_\lambda)$ по форме (6) для сохранения строгости формулы (9) необходимо добавить на первое место дизъюнкцию вида $(1\vee 0)$.

*Доказательство.* Если $|\alpha_\lambda|$ четно, то данное утверждение является производным утверждения 1, поскольку в этом случае $|\alpha_\lambda|=2\cdot m_2$, т.е. $|\alpha_\lambda|\geq 2\cdot m_2$.

Если же значение $|\alpha_\lambda|$ – нечетно, то в записи $f(m_2, \alpha_\lambda)$ по форме (5) отсутствует дизъюнкция компонентов $(f(m_2, \alpha_{2\lambda})\vee f(0, \alpha_{2\lambda+1}))$ , так как $m_2>|\alpha_{2\lambda}|$.

В то же время, согласно (9), она должна быть испльзована, поскольку компонента $f(0,\alpha_{2\lambda+1})$ (равная 0) присутствует в записи $f(m_2\text{-}1, \alpha_\lambda)$ по форме (5).

Из этого следует, что равенство по формуле (9) может быть нарушено.

Путем дописывания в начало записи $f(m_2, \alpha_\lambda)$ по (6) выражения $(1\vee 0)$, можно избежать указанной ситуации. Покажем это на следующем примере:

$$f(2,3)=(f(1,1)\vee f(1,2))\ (f(0,1)\vee f(2,2))=$$
$$=(\ 1\ \vee\ 0\ )\ (f(1,1)\vee f(1,2))\ (f(0,1)\vee f(2,2))$$
$$f(1,3)=(f(1,1)\vee\ 0\ )\ (f(0,1)\vee f(1,2))$$

*Утверждение 4.* Функция $f(m_2+1, \alpha_\lambda)$ может быть сформирована из компонент функции $f(m_2, \alpha_\lambda)$ представленной в виде (6) следующим образом:

$$f(m_2+1,\alpha_\lambda)=\mathop{\&}_{j=1}^{m_1}(f_{j,1}\vee f_{j+1,2}), \qquad (10)$$

где $f_{j,1}$, $f_{j+1,2}$ – компоненты функции $f(m_2,\alpha_\lambda)$.

*Доказательство.* Распишем обе функции по форме (5):

$$f(m_2,\alpha_\lambda)=(f'(m_2, \alpha_{2\lambda})\vee f'(0, \alpha_{2\lambda+1}))\&$$
$$\&(f'(m_2\text{-}1,\alpha_{2\lambda})\vee$$
$$\vee f'(1,\alpha_{2\lambda+1}))\&\ldots\&(f'(0,\alpha_{2\lambda})\vee f'(m_2,\alpha_{2\lambda+1})),$$

$$f(m_2+1, \alpha_\lambda)=\ \cancel{(f(m_2+1,\alpha_{2\lambda})\vee f(0,\alpha_{2\lambda+1}))}\ \&$$
$$\&(f(m_2,\alpha_{2\lambda})\vee f(1,\alpha_{2\lambda+1}))\&(f(m_2\text{-}1,\alpha_{2\lambda})\vee$$
$$\vee f(2,\alpha_{2\lambda+1}))\&\ldots\&$$
$$\&(f(0,\alpha_{2\lambda})\vee f(m_2+1,\alpha_{2\lambda+1}))\ \&$$
$$\cancel{\&\ (f(0,\alpha_{2\lambda})\vee f(m_2+1,\alpha_{2\lambda+1}))}$$

Зачеркнутые дизъюнкции отсутствуют в представлении $f(m_2+1, \alpha_\lambda)$, потому что $m_2+1>|\alpha_{2\lambda}|$ и $m_2+1>|\alpha_{2\lambda+1}|$.

Легко видеть, что в $f(m_2, \alpha_\lambda)$ присутствуют все компоненты функции $f(m_2+1, \alpha_\lambda)$.

Поскольку компонента $f'_{j,1}$ из функции $f(m_2, \alpha_\lambda)$ равняется компоненте $f_{j,1}$ функции $f(m_2+1, \alpha_\lambda)$, как и $f'_{j+1,2}=f_{j,2}$, где $j=1..m_1$, то становится очевидным справедливость выражения (10).

Приведем пример:

$$f(4,7)=(f(3,3)\vee f(1,4))\,(f(2,3)\vee f(2,4))\,(f(1,3)\vee f(3,4))\,(f(0,3)\vee f(4,4));$$

$$f(5,7)=(f(3,3)\vee f(2,4))\,(f(2,3)\vee f(3,4))\,(f(1,3)\vee f(4,4)).$$

Многократное применение формул (9) и (10) подтверждает справедливость следующих выражений:

$$f(m_2-i,\alpha_\lambda)=\mathop{\&}_{j=1}^{m_2-i+1}(f_{j+i,1}\vee f_{j,2}), \qquad (11)$$

$$f(m_2+i,\alpha_\lambda)=\mathop{\&}_{j=1}^{m_1-i+1}(f_{j,1}\vee f_{j+i,2}), \qquad (12)$$

где $f_{j+i,1}$, $f_{j,2}$, $f_{j,1}$, $f_{j+i,2}$ – компоненты функции $f(m_2,\alpha_\lambda)$.

Строгость формулировки требует представить выражение (2) для ускоренного алгоритма в следующей форме:

$$K'(m_x,\alpha_\lambda)=K'(m_x,\alpha_{2\lambda})\cup \bigcup_{i=1}^{m_x-1}\left[f(m_x-i,\alpha_{2\lambda})\vee f(i,\alpha_{2\lambda+1})\right]\cup \cup K'(m_x,\alpha_{2\lambda+1}), \qquad (13)$$

где $|\alpha_\lambda|>m_x$, $m_x-i\le|\alpha_{2\lambda}|$, $i\le|\alpha_{2\lambda+1}|$.

Выше приведенные выражения позволяют решать задачу построения реберных функций базовой 2р-модели, не прибегая к процедуре минимизации, и, что более важно, без формирования канонической модели.

### 3.1. Формализованное представление ускоренного алгоритма

Ниже представлен алгоритм формирования реберных функций базовой 2р-модели в виде описания рекурсивной процедуры, результатом выполнения которой является множество реберных функций. Приведенное описание дано лишь для лучшего понимания процесса получения реберных функций 2р-модели.

Прежде чем перейти непосредственно к описанию процедуры, сделаем некие комментарии.

Пусть F – двумерный массив, хранящий запись функции $f(m, \alpha_\lambda)$ таким образом, что в ячейке F(j,1) находится запись компоненты (т.е. функции) $f_{j,1}$ согласно обозначениям в выражении (6), а в F(j,2) соответственно $f_{j,2}$. K' – множество реберных функций.

Полужирным $\mathbf{K'(m,\alpha_\lambda)}$ обозначим собственно процедуру, о которой идет речь.

При каждом проходе по телу процедуры создается новое множество K' и массив F.

**Процедура $\mathbf{K'(m,\alpha_\lambda)}$**

1. Пусть $K':=\varnothing$, $F[i,j]=0$.
2. Если m=0, то $K'=\{0\}$, перейти к п. 8.
3. Если m=1, то сформировать $f(m,\alpha_\lambda)$ по (3) и записать в K', перейти к п. 8.
4. Если $m=|\alpha_\lambda|$, то сформировать $f(m,\alpha_\lambda)$ по (4) и записать в $K'=f(m,\alpha_\lambda)$, перейти к п. 8.
5. Если $|\alpha_{2\lambda}|\ge 2m$, то сформировать $f(m,\alpha_{2\lambda})$ из множества $\mathbf{K'(m,\alpha_{2\lambda})}$ по (1), записать в F(1,1), по (8) сформировать все остальные допустимые F(j,1);

   иначе сформировать $f(]|\alpha_{2\lambda}|/2[,\alpha_{2\lambda})$ из множества $\mathbf{K'(]|\alpha_{2\lambda}|/2[,\alpha_{2\lambda})}$ по (1), записать в $F(m-](m-\min(m+1,|\alpha_\lambda|-m+1)+1)/2[-]|\alpha_{2\lambda}|/2[+1,1)$, по (11) и (12) сформировать все остальные допустимые F(j,1);
6. Если $|\alpha_{2\lambda+1}|\ge 2m$ сформировать $f(m,\alpha_{2\lambda+1})$ из множества $\mathbf{K'(m,\alpha_{2\lambda+1})}$ по (1), записать в F(m+1,2), по (8) сформировать все остальные допустимые F(j,2).

   иначе сформировать $f(]|\alpha_{2\lambda+1}|/2[,\alpha_{2\lambda+1})$ из множества $\mathbf{K'(]|\alpha_{2\lambda+1}|/2[,\alpha_{2\lambda+1})}$ по (1), записать в $F(]|\alpha_{2\lambda+1}|/2[-](m-\min(m+1,|\alpha_\lambda|-m+1)+1)/2[+1,2)$, по (11) и (12) сформировать все остальные допустимые F(j,2).
7. Преобразовать F в множество функций и записать в K'.
8. Возврат в $\mathbf{K'(m,\alpha_\lambda)}$ множества K'.

**Конец процедуры.**

Нетрудно увидеть, что процедура $\mathbf{K'(m,\alpha_\lambda)}$ отражает расширенное представление выражения (11). Отметим, без дополнительных пояснений, что общее количество вызовов выражения (11) по ускоренному алгоритму можно оценить величиной

$$2^{]\log_2 n[+1}-1\le 4n\,.$$

Таким образом, приведенный ускоренный алгоритм имеет значительно меньшую сложность, чем базовый.

## Заключение

Основным результатом работы является ускоренный алгоритм формирования базовой GL-модели, теряющей минимальное количество ребер при появлении вектора состояния системы с m+1 нулем (так названой 2p-модели). Одним из достоинств предложенного алгоритма является отсутствие достаточно сложной многоэтапной процедуры минимизации. Кроме того, отмеченная в работе особенность - наличие одинаковых компонент в различных реберных функциях - дает возможность ускорить операцию их вычисления.

## Литература


*1. Реконфигурируемые мультиконвейерные вычислительные структуры / И.А. Каляев, И.И. Левин, Е.А. Семерников, В.И. Шмойлов, Под общ. ред. И.А.Каляева. – Ростов-на-Дону: ЮНЦ РАН, 2008. – 320 с.*

*2. Многоверсионные системы, технологии, проекты / В.С. Харченко, В.Я. Жихарев, В.М. Илюшко, Н.В. Нечипорук. – Х.: Нац. Аэрокосм. ун-т "Харьк.авиац. ин-т", 2003. – 486 с.*

*3. Kuo Way. Optimal Reliability Modeling / Way Kuo, Ming J. Zuo. – John Willey & Sons, 2002. – 560 p.*

*4. Романкевич А.М. Анализ отказоустойчивых многомодульных систем со сложным распределением отказов на основе циклических GL-моделей / А.М. Романкевич, В.В. Иванов, В.А Романкевич // Электронное моделирование. - 2004. - Т. 26, № 5. - С. 67-81.*

*5. Романкевич А.М. Графо-логические модели для анализа сложных отказоустойчивых вычислительных систем / А.М. Романкевич, Л.Ф. Карачун, В.А. Романкевич // Электронное моделирование. - 2001. – Т. 23, №1. – С. 102-111.*

*6. GL-модель поведения отказоустойчивых многопроцессорных систем с минимальным числом теряемых рёбер / В.А. Романкевич, Е.Р. Потапова, Бахтари Хедаятоллах, В.В. Назаренко // Вісник НТУУ "КПІ" Інформатика, управління та обчислювальна техніка. - 2006. - № 45. - С. 93-100.*


*Поступила в редакцию 30.07.2010*

**Рецензент:** д-р техн. наук, проф. В.В. Гроль, Национальный технический университет Украины «Киевский политехнический институт», Киев.

### ПРО ОДИН АЛГОРИТМ ФОРМУВАННЯ РЕБЕРНИХ ФУНКЦІЙ GL-МОДЕЛЕЙ ВІДМОВОСТІЙКИХ БАГАТОПРОЦЕССОРНИХ СИСТЕМ

***О.М. Романкевич, І.В. Майданюк, М.І. Хаітов***

Робота присвячена алгоритму формування реберних функцій циклічної GL-моделі базової відмовостійкої багатопроцесорної системи (ВБС), яка втрачає мінімальна кількість ребер при відмові компонентів, кількість яких перевищує базовий ступінь відмовостійкості. Приведено ряд тверджень і формул, які дають можливість формувати розглядувану модель в обхід відносно складної процедури мінімізації, що дозволяє зменшити обчислювальну складність процесу формування даної моделі.

**Ключові слова:** Відмовостійкі багатомодульні системи, графо-логічні моделі, розрахунок надійності.

### AN ALGORITHM FOR THE FORMATION OF EDGE FUNCTIONS GL-MODELS OF FAULT-TOLERANT MULTIPROCESSOR SYSTEMS

***A.M. Romankevich, I.V. Maidanyuk, M.I. Khaitov***

The paper is dedicated to algorithm for base fault-tolerant multiprocessor system (FTMS) edge functions forming. As it is proved the system can loose minimal numbers of edges under components' failure the number of which exceeds basic degree of fault tolerance. Few statements and formulas leading to possibility to design reviewed model with eliminating of relatively complex minimization procedure are proved, that help to decrease the computing complexity for process of reviewed model's forming.

**Key words:** Fault-tolerant multimodule systems, graph-logical models, calculation of reliability.

**Романкевич Алексей Михайлович** – д-р техн. наук, проф., проф. кафедры специализированных компьютерных систем Национального технического университета Украины «Киевский политехнический институт», Киев, Украина, e-mail: romankev@scs.ntu-kpi.kiev.ua.

**Майданюк Иван Викторович** – аспирант кафедры специализированных компьютерных систем Национального технического университета Украины «Киевский политехнический институт», Киев, Украина, e-mail: maidanyuk_vanya@ukr.net.

**Хаитов Марат Иномжонович** – студент кафедры специализированных компьютерных систем Национального технического университета Украины «Киевский политехнический институт», Киев, Украина.