

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

МОРОЗОВ КОСТЯНТИН В'ЯЧЕСЛАВОВИЧ

УДК 004.05

ДИСЕРТАЦІЯ
МЕТОДИ І ЗАСОБИ ПОБУДОВИ МОДЕЛЕЙ
ПОВЕДІНКИ НЕБАЗОВИХ ВІДМОВСТІЙКИХ
БАГАТОПРОЦЕСОРНИХ СИСТЕМ

Спеціальність 05.13.05 – Комп'ютерні системи та компоненти
Галузь знань – Інформаційні технології

Подається на здобуття наукового ступеня кандидата технічних наук

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело

_____ К.В. Морозов

Науковий керівник Романкевич Віталій Олексійович,
доктор технічних наук, доцент

Київ - 2021

АНОТАЦІЯ

Морозов К.В. Методи і засоби побудови моделей поведінки небазових відмовостійких багатопроцесорних систем. – Рукопис.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.05 – Комп'ютерні системи та компоненти. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, 2021 р.

Дисертація присвячена проблемі побудови графо-логічних моделей (GL-моделей) небазових відмовостійких багатопроцесорних систем (ВБС) з метою розрахунку їх параметрів надійності шляхом проведення статистичних експериментів із вищезгаданими моделями.

ВБС, що складається з n процесорів та залишається роботоздатною при виході з ладу не більше, ніж m будь-яких з них називається базовою. Всі інші системи, що не відповідають даному критерію, називаються небазовими. Велика кількість реальних систем, особливо систем керування є небазовими.

Один із ефективних методів розрахунку параметрів надійності ВБС, в тому числі небазових, базується на проведенні статистичних експериментів з так званими графо-логічними моделями, що описують поведінку системи в потоці відмов.

Графо-логічна модель являє собою неорієнтований граф, ребрам якого відповідають булеві реберні функції, що залежать від так званого вектору стану системи, кожний елемент котрого відповідає стану одного з процесорів системи і приймає значення 1, якщо той є роботоздатним та 0 в протилежному випадку. Якщо реберна функція приймає нульове значення, відповідне їй ребро виключається з графу моделі. Зв'язність останнього характеризує стан системи в потоці відмов: втрата зв'язності графом відповідає втраті роботоздатності системою.

Відомий ряд методів, що дозволяють будувати різні GL-моделі, кожна з яких має свої переваги та недоліки. Зокрема, так звані MBP-моделі (мінімум

втрачених ребер) мають таку властивість: вони втрачають мінімальне число ребер на векторах із кратністю відмов, що на одиницю перевищує ступінь відмовостійкості системи. Окрім того, вони базуються на циклічних графах, процес визначення зв'язності яких є тривіальним. Проте, алгоритм побудови цього різновиду моделей (як і деяких інших) є придатним лише для базових ВБС.

Побудова GL-моделі для небазової системи може відбуватися шляхом модифікації деякої базової моделі, яка є досить близькою до шуканої, в результаті чого на деякій множині так званих блокованих векторів поведінка моделей починає відрізнятися. Така модифікація може відбуватися шляхом зміни структури графу моделі (проведення додаткових ребер), зміни виразів реберних функцій, або шляхом поєднання цих підходів. В роботі розглядається другий підхід, що базується на модифікації лише виразів реберних функцій, в результаті чого граф моделі залишається циклічним.

В результаті аналізу алгоритму побудови реберних функцій МВР-моделі помічено, що їх вирази складаються з двох частин. Запропоновано метод перетворення GL-моделей за рахунок модифікації виразу будь-якої реберної функції МВР-моделі. Метод базується на зміні як однієї, так і обох частин виразу реберної функції та дозволяє модифікувати модель так, що на деяких векторах стану системи вона, на відміну від оригінальної, починає показувати роботоздатний і/або нероботоздатний стан. Проаналізовано властивості модифікованих моделей, сформульовано та доведено твердження, що дозволяють визначати кількість та множину блокованих векторів відповідно до параметрів модифікації та виду функції, що модифікується. Також виконано аналіз впливу модифікації на поведінку МВР-моделі для випадку, коли модифікована модель використовується в якості допоміжної при її побудові.

Отримано оцінку максимальної кількості векторів, що можуть бути блоковані запропонованим методом. Визначено функції та параметри їх модифікації, які дозволяють досягти цієї кількості блокованих векторів.

Проаналізовано межі впливу для загального випадку модифікації як однієї, так і кількох реберних функцій на зміну поведінки моделі. Вперше виконано

аналіз необхідності модифікації декількох функцій МВР-моделі. Виявлено, що у випадку одночасної модифікації кількох реберних функцій може проявлятися побічний ефект у вигляді блокування векторів із підвищеною кратністю відмов. Отримано критерії, що дозволяють уникнути його у загальному випадку, а також критерії вибору пар функцій для випадку модифікації двох реберних функцій.

Вперше запропоновано аналітичний апарат, що дозволяє визначати вектор стану t -діагностованої ВБС на основі результатів взаємного тестування процесорів, згідно із моделлю Препарати-Метца-Чена (ПМЧ-модель), для довільної топології зв'язків між ними. Метод базується на формуванні та перетворенні булевого рівняння спеціального виду. Сформульовано рекомендації щодо можливих стратегій вибору пар процесорів для проведення подальших тестів. Отримано оцінку ефективності запропонованого методу для заданої множини тестових перевірок, яка може бути представлена за допомогою діагностичного графу.

Вперше запропоновано метод побудови моделей так званих зважених систем, в яких кожний компонент має деяку вагу, що характеризує його вклад в роботоздатність системи в цілому. Зокрема, зважена базова система (або зважена k -з- n система) залишається роботоздатною доти, доки сумарна вага всіх її роботоздатних компонентів є не меншою за деяке порогове значення. Запропонований метод побудови GL-моделей таких систем на основі МВР-моделей базується на n_i -кратному дублюванні відповідних елементів вектору стану системи, вибираючи значення n_i пропорційними вагам відповідних компонентів системи. Метод може бути застосований для випадків ваг, представлених як цілими, так і раціональними числами.

Вперше запропоновано узагальнення методу побудови GL-моделей, що базується на використанні в якості реберних функцій багатозначної логіки. Це дозволяє використовувати їх для побудови моделей систем, для яких, як і їх компонентів характерні більше двох станів роботоздатності. Тобто, у випадках, коли окрім повністю роботоздатного та несправного мають місце додаткові стани часткової роботоздатності. Причому кількості таких станів як у системи,

так і в кожного з її компонентів можуть різнитися між собою. Запропоновано модифікації алгоритмів Прима та Крускала для визначення стану системи відповідно до графу моделі.

Запропоновано метод побудови GL-моделей ієрархічних систем шляхом композиції декількох МВР-моделей. Для об'єднання моделей, що описують стан системи в потоках відмов відповідно процесорів та підсистем запропоновано використовувати допоміжну модель, що відповідає невідмовостійкій системі.

Запропоновано метод побудови моделей систем, що складаються з кількох підсистем та мають ковзний резерв. Метод базується на побудові допоміжних моделей для кожної з підсистем із подальшим поєднанням значень їх реберних функцій у вхідному векторі результуючої моделі.

Запропоновано модель системи, в якій деякі підсистеми можуть виступати в якості «донорів», тобто, при необхідності віддавати частину своїх ресурсів, а деякі інші – в якості «реципієнтів», тобто, приймати ці ресурси для забезпечення своєї роботоздатності. Для її побудови в якості допоміжної запропонована дуальна МВР-модель, що відрізняється від звичайної заміною в виразах реберних функцій операцій кон'юнкції операціями диз'юнкції та навпаки.

Практичне значення одержаних результатів визначається тим, що в ході дисертаційної роботи запропоновано і досліджено ряд методів модифікації МВР-моделей, що дозволяють моделювати поведінку небазових ВБС в потоці відмов. Однією з переваг запропонованих підходів є те, що їх застосування не призводить до зміни структури графа моделі, в результаті чого процедура визначення зв'язності графа є тривіальною і не потребує значних обчислювальних можливостей. Окрім того, запропоновані підходи не потребують проведення додаткових ребер, а, отже, і розрахунку значень пов'язаних із ними функцій, що призводить до додаткового зменшення обчислювальної складності процесу визначення стану системи в порівнянні із методами, що базуються на зміні структури графу моделі. Запропонований аналітичний апарат для аналізу результатів взаємного тестування пар процесорів ВБС дозволяє на основі проведених тестів отримувати вектор стану системи

довільної структури, а також може бути основою для розробки оптимальних або квазіоптимальних (за кількістю необхідних тестів) стратегій вибору наступних тестів в залежності від результатів попередніх.

Розроблено програмне забезпечення, що дозволяє виконувати експерименти з модифікованими GL-моделями, а також визначати вектор стану систему відповідно до результатів тестових перевірок, згідно із запропонованим в роботі методом. Отримані експериментальні дані, що підтверджують достовірність теоретично отриманих результатів.

Ключові слова: відмовостійкі багатопроцесорні системи, графо-логічні моделі, надійність, модель Препарати-Метце-Чена, зважені k -з- n системи, ієрархічні системи, ковзний резерв.

ABSTRACT

Morozov K.V. Methods and tools for building models of behavior of non basic fault tolerant multiprocessor systems. – Manuscript.

Dissertation for the scientific degree of Candidate of Technical Sciences on specialty 05.13.05 – Computer systems and components. National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, 2021.

The thesis is devoted to the problem of building graph-logical models (GL models) of non-basic fault-tolerant multiprocessor systems (FTMS) for calculating their reliability parameters by means of carrying out statistical experiments with the above-mentioned models.

An FTMS consisting of n processors and which remains operational in case of failure of no more than m of any of them is called basic. All other systems which do not meet this criterion are called non-basic. A large number of real systems, especially control systems, are non-basic.

One of the effective methods of calculating the FTMS reliability parameters, including non-basic ones, is based on statistical experiments with so-called graph-logical models, which describe the behavior of the system in the flow of failures.

A graph-logical model is an undirected graph whose edges correspond to Boolean edge functions depending on the so-called system state vector, each element of which corresponds to the state of one of the system processors and takes on the value 1 if it is operational and 0 otherwise. If an edge function takes on a zero value, the corresponding edge is excluded from the model graph. The connectivity of the graph characterizes the state of the system in the flow of failures: the loss of connectivity by the graph corresponds to the loss of operability by the system.

Numerous methods are known for building various GL-models, each of which has its own advantages and disadvantages. In particular, the so-called MLE-models (minimum of lost edges) have the following feature: they lose the minimal number of edges on vectors with a number of failures which exceeds by one the degree of system fault tolerance. In addition, they are based on cyclic graphs, the process of determining

of connectivity for which is trivial. However, the algorithm of building of this type of models (as well as some others) is suitable only for basic FTMS.

Building of a GL-model for a non-basic system can be carried out by modifying some basic model which is close enough to the desired one, as a result of which the behavior of the models starts to differ on a certain set of so-called blocked vectors. Such modification can occur by changing the structure of the model graph (drawing additional edges), changing the expressions of edge functions, or by combining these approaches. The paper considers the second approach based only on modifying of the expressions of edge functions, as a result of which the model graph remains cyclic.

As a result of the analysis of the algorithm of building the edge functions of the MLE-model, it is noticed that their expressions consist of two fragments. A method for transforming of GL-models by modifying the expression of any edge function of the MLE-model is proposed. The method is based on changing either any one or both two fragments of the edge function's expression and allows modifying the model so that on some system state vectors, it starts to show an operable and/or inoperable state in contrast to the original one. The properties of the modified models have been analyzed, the statements have been formulated and proved that allow determining the number and the set of blocked vectors in accordance with modification parameters and type of the modified function. It was also analyzed the effect of modification on the behavior of the MLE-model for case when the modified model is used as an auxiliary one for its building.

An estimate of the maximal number of vectors which can be blocked with the proposed method is obtained. The functions and parameters of their modification, which allow achieving this number of blocked vectors are determined.

The boundaries of influence for general case of modification of both single and several edge functions on change of the behavior of the model is analyzed. For the first time, an analysis of necessity to modify several functions of the MLE-model was done. It was found that in case of simultaneous modification of several edge functions, a side effect in form of blocking of the vectors with increased multiplicity of failures may

appear. Criteria to avoid it in general case, as well as criteria of selection of pairs of functions for the case of modification of two edge functions are obtained.

For the first time, an analytical apparatus that allows to determine the state vector of t -diagnosable FTMS based on the results of mutual testing of processors, according to the Preparata-Metze-Chen model (PMC-model), for arbitrary topology of connections between them has been proposed. The method is based on formation and further transformation of a Boolean equation of a special form. Recommendations regarding possible strategies for choosing processor pairs for further tests are formulated. An estimate of the effectiveness of the proposed method for a given set of test checks, which can be represented using a diagnostic graph is obtained.

For the first time, a method of building models of so-called weighted systems, in which each component has a certain weight that characterizes its contribution to the performance of the system as a whole is proposed. In particular, a basic weighted system (or a weighted k -out-of- n system) remains operable as long as the total weight of all its operable components remains not less than a certain threshold value. The proposed method of building GL-models of such systems based on MLE-models is based on n_i -fold duplication of the corresponding elements of the system state vector, choosing the value n_i proportional to the weights of the corresponding system components. The method can be applied for cases of weights represented by both integers and rational numbers.

For the first time, a generalization of methods of building GL-models, based on the usage of multivalued logic edge functions is proposed. This allows them to be used as models of systems for which, as well as for their components, more than two states of operability are available. I.e. for cases where, in addition to fully operable and faulty states, the additional states of partial operability are present. Moreover, the numbers of states for the system and for each of its components may differ. Modifications of the Prim and Kruskal algorithms for determining of the state of the system in accordance to the model graph are proposed.

A method of building GL-models of hierarchical systems by means of composition of several MLE-models is proposed. To combine the models which

describe states of the system in flows of failures of processors and subsystems respectively, it is proposed to use an auxiliary model corresponding to a non-fault-tolerant system.

A method of building models of systems consisting of several subsystems and having a sliding reserve is proposed. The method is based on building of auxiliary models for each of the subsystems with the subsequent concatenation of the values of their edge functions into the input vector of the resulting model.

A model of the system in which some subsystems can act as "donors", i.e., give some portion of their resources when it is necessary, and some others as "recipients", i.e. utilize these resources to restore own operation is proposed. For its building an auxiliary dual MLE model, which differs from the common one by replacement of conjunction operations with disjunction operations and vice versa in the expressions of edge functions is proposed.

The practical importance of the results obtained is determined by the fact that during the thesis preparation numerous methods of modification of MLE-models were proposed and investigated, that make possible to simulate the behavior of non-basic FTMS in the flow of failures. One of the advantages of the proposed approaches is that their application does not change the structure of the model graph, as a result of that the procedure for determining the graph connectivity is trivial and does not require significant computational capabilities. In addition, the proposed approaches do not require drawing additional edges, and, consequently, calculating the values of the functions associated with them, which leads to an additional decrease in the computational complexity of the process of determining the state of the system comparing to the methods based on changing the structure of the model graph. The proposed analytical apparatus for analyzing the results of mutual testing of pairs of processors of FTMS allows to obtain a system state vector from the tests carried out, for arbitrary structure of the system, and can also become the basis for the development of optimal or quasi-optimal (by the number of tests required) strategies for choosing the following test checks, depending on the results of previous ones.

Software that allows performing experiments with modified GL-models, as well as determining the system state vector according to the results of test checks, according to the method proposed in the paper has been developed. Experimental data which confirm the reliability of the theoretically obtained results have been obtained.

Key words: fault-tolerant multiprocessor systems, graph-logical models, reliability, Preparata-Metze-Chen model, weighted k -out-of- n systems, hierarchical systems, sliding reserve.

АННОТАЦИЯ

Морозов К.В. Методы и средства построения моделей поведения небазовых отказоустойчивых многопроцессорных систем. – Рукопись.

Диссертация на соискание ученой степени кандидата технических наук по специальности 05.13.05 – Компьютерные системы и компоненты. Национальный технический университет Украины «Киевский политехнический институт имени Игоря Сикорского», г. Киев, 2021 г.

Диссертация посвящена проблеме построения графо-логических моделей (GL моделей) небазовых отказоустойчивых многопроцессорных систем (ОМС) в целях расчета их параметров надежности путем проведения статистических экспериментов с вышеупомянутыми моделями.

ОМС, состоящая из n процессоров и остающаяся работоспособной при выходе из строя не более, чем m любых из них называется базовой. Все остальные системы, не соответствующие данному критерию, называются небазовыми. Большое количество реальных систем, особенно систем управления являются небазовыми.

Один из эффективных методов расчета параметров надежности ОМС, в том числе небазовых, основан на проведении статистических экспериментов с так называемыми графо-логическими моделями, описывающими поведение системы в потоке отказов.

Графо-логическая модель представляет собой неориентированный граф, ребрам которого соответствуют булевы реберные функции, зависящие от так называемого вектора состояния системы, каждый элемент которого соответствует состоянию одного из процессоров системы и принимает значение 1, если тот является работоспособным и 0 в противном случае. Если реберная функция принимает нулевое значение, соответствующее ей ребро, исключается из графа модели. Связность последнего характеризует состояние системы в потоке отказов: потеря связности графом соответствует потере работоспособности системой.

Известен ряд методов, позволяющих строить различные GL-модели, каждая из которых имеет свои преимущества и недостатки. В частности, так называемые МВР-модели (минимум выпадающих ребер) имеют такое свойство: они теряют минимальное число ребер на векторах с кратностью отказов, на единицу превышающей степень отказоустойчивости системы. Кроме того, они базируются на циклических графах, процесс определения связности которых является тривиальным. Однако, алгоритм построения этой разновидности моделей (как и некоторых других) является применимым только для базовых ОМС.

Построение GL-модели для небазовой системы может происходить путем модификации некоторой базовой модели, достаточно близкой к искомой, в результате чего на некотором множестве так называемых блокируемых векторов поведение моделей начинает отличаться. Такая модификация может происходить путем изменения структуры графа модели (проведение дополнительных ребер), изменения выражений реберных функций, или путем сочетания этих подходов. В работе рассматривается второй подход, основанный на модификации только выражений реберных функций, в результате чего граф модели остается циклическим.

В результате анализа алгоритма построения реберных функций МВР-модели замечено, что их выражения состоят из двух частей. Предложен метод преобразования GL-моделей за счет модификации выражения любой реберной функции МВР-модели. Метод основан на изменении как одной, так и обеих частей выражения реберной функции и позволяет модифицировать модель так, что на некоторых векторах состояния системы она, в отличие от оригинальной, начинает показывать работоспособное и/или неработоспособное состояние. Проанализированы свойства модифицированных моделей, сформулированы и доказаны утверждения, позволяющие определять количество и множество блокируемых векторов в соответствии с параметрами модификации и вида модифицируемой функции. Также выполнен анализ влияния

модификации на поведение МВР-модели для случая, когда модифицированная модель используется в качестве вспомогательной при ее построении.

Получена оценка максимального количества векторов, которые могут быть заблокированы предложенным методом. Определены функции и параметры их модификации, которые позволяют достичь этого количества блокируемых векторов.

Проанализированы границы влияния для общего случая модификации как одной, так и нескольких реберных функций на изменение поведения модели. Впервые выполнен анализ необходимости модификации нескольких функций МВР-модели. Обнаружено, что в случае одновременной модификации нескольких реберных функций может проявляться побочный эффект в виде блокировки векторов с повышенной кратностью отказов. Получены критерии, позволяющие избежать его в общем случае, а также критерии выбора пар функций для случая модификации двух реберных функций.

Впервые предложен аналитический аппарат, позволяющий определять вектор состояния t -диагностируемой ОМС на основе результатов взаимного тестирования процессоров, согласно модели Препараты-Метце-Чена (ПМЧ-модель), для произвольной топологии связей между ними. Метод основан на формировании и преобразовании булевого уравнения специального вида. Сформулированы рекомендации относительно возможных стратегий выбора пар процессоров для проведения дальнейших тестов. Получена оценка эффективности предложенного метода для заданного множества тестовых проверок, которое может быть представлено с помощью диагностического графа.

Впервые предложен метод построения моделей так называемых взвешенных систем, в которых каждый компонент имеет некоторый вес, характеризующий его вклад в работоспособность системы в целом. В частности, взвешенная базовая система (или взвешенная k -из- n система) остается работоспособной до тех пор, пока суммарный вес всех ее работоспособных компонентов остается не меньшим некоторого порогового значения.

Предложенный метод построения GL-моделей таких систем на базе МВР моделей основан на n_i -кратном дублировании соответствующих элементов вектора состояния системы, выбирая значение n_i пропорциональными весам соответствующих компонентов системы. Метод может быть применен для случаев весов, представленных как целыми, так и рациональными числами.

Впервые предложено обобщение метода построения GL-моделей, основанное на использовании в качестве реберных функций многозначной логики. Это позволяет использовать их для построения моделей систем, для которых, как и их компонентов характерны более двух состояний работоспособности. То есть, в случаях, когда кроме полностью работоспособного и неисправного имеют место дополнительные состояния частичной работоспособности. Причем количества таких состояний как у системы, так и у каждого из ее компонентов могут различаться между собой. Предложены модификации алгоритмов Прима и Краскала для определения состояния системы в соответствии с графом модели.

Предложен метод построения GL-моделей иерархических систем путем композиции нескольких МВР-моделей. Для объединения моделей, описывающих состояние системы в потоках отказов соответственно процессоров и подсистем предложено использовать вспомогательную модель, отвечающую неотказоустойчивой системе.

Предложен метод построения моделей систем, состоящих из нескольких подсистем, и имеющих скользящий резерв. Метод основан на построении вспомогательных моделей для каждой из подсистем с последующим сочетанием значений их реберных функции во входном векторе результирующей модели.

Предложена модель системы, в которой некоторые подсистемы могут выступать в качестве «доноров», то есть при необходимости отдавать часть своих ресурсов, а некоторые другие – в качестве «реципиентов», то есть принимать эти ресурсы для обеспечения своей работоспособности. Для ее построения в качестве вспомогательной предложена дуальная МВР-модель,

отличающаяся от обычной заменой в выражениях реберных функций операций конъюнкции операциями дизъюнкции и наоборот.

Практическое значение полученных результатов определяется тем, что в ходе диссертационной работы предложен и исследован ряд методов модификации МВР-моделей, позволяющих моделировать поведение небазовых ОМС в потоке отказов. Одним из преимуществ предложенных подходов является то, что их применение не приводит к изменению структуры графа модели, в результате чего процедура определения связности графа является тривиальной и не требует значительных вычислительных возможностей. Кроме того, предложенные подходы не требуют проведения дополнительных ребер, а, следовательно, и расчета значений, связанных с ними функций, что приводит к дополнительному уменьшению вычислительной сложности процесса определения состояния системы по сравнению с методами, основанными на изменении структуры графа модели. Предложенный аналитический аппарат для анализа результатов взаимного тестирования пар процессоров ОМС позволяет на основе проведенных тестов получать вектор состояния системы произвольной структуры, а также может быть основой для разработки оптимальных или квазиоптимальных (по количеству необходимых тестов) стратегий выбора следующих тестов в зависимости от результатов предыдущих.

Разработано программное обеспечение, позволяющее выполнять эксперименты с модифицированными GL-моделями, а также определять вектор состояния системы согласно результатам тестовых проверок, согласно предложенному в работе методу. Получены экспериментальные данные, подтверждающие достоверность теоретически полученных результатов.

Ключевые слова: отказоустойчивые многопроцессорные системы, графологические модели, надежность, модель Препараты-Метце-Чена, взвешенные k -из- n системы, иерархические системы, скользящий резерв.

Список публікацій здобувача

1. Romankevich A.M. Hierarchical Graph-Logical Models of Multiprocessor Systems Based on Grouping of Their Components / A.M. Romankevich, K.V. Morozov, V.A. Romankevich // IJCSNS International Journal of Computer Science and Network Security. – 2019. – Vol. 19, No.10. – P. 138-143. *Автором запропоновано використання проміжних допоміжних моделей для поєднання моделей, що відповідають підсистемам нижнього та поточного рівня ієрархії. (Закордонне видання, реферується наукометричною базою «Web of Science»)*
2. Romankevich A.M. Graph-Logic Models of Hierarchical Fault-Tolerant Multiprocessor Systems / A.M. Romankevich, K.V. Morozov, V.A. Romankevich // IJCSNS International Journal of Computer Science and Network Security. – 2019. – Vol. 19, No.7. – P. 151-156. *Автором запропоновано метод побудови GL моделей розглянутих ієрархічних систем. (Закордонне видання, реферується наукометричною базою «Web of Science»)*
3. Морозов К.В. О характере влияния модификации рёберных функций GL-модели на её поведение в потоке отказов / К.В. Морозов, А.М. Романкевич, В.А. Романкевич // Радиоелектронні і комп'ютерні системи. – №6, 2016. – С.108–112. *Автором виконано аналіз границь зміни поведінки моделі в залежності від кількості модифікованих реберних функцій. (Реферується наукометричними базами «Україніка наукова», «Index Copernicus»)*
4. Морозов К.В. Про модифікацію графо-логічної моделі / К.В. Морозов, В.О. Романкевич, К.Р. Потапова // Інформаційні технології та комп'ютерна інженерія. – Том 3, № 34. – 2015. – С.42-49. *Автором сформульовано критерії вибору пар функцій, що можуть бути сумісно модифіковані без появи побічного ефекту (блокування векторів із підвищеною кратністю відмов).*
5. Романкевич В.А. Об одном методе модификации рёберных функций GL-моделей / В.А. Романкевич, К.В. Морозов, А.П. Фесенюк // Радиоелектронні і комп'ютерні системи. – №6, 2014. – С.95-99. *Автором запропоновано спосіб модифікації реберної функції GL-моделі, сформульовано та доведено твердження щодо множини векторів, блокуваних в результаті такої*

модифікації. (Реферується наукометричними базами «Україніка наукова», «Index Copernicus»)

6. Романкевич В.А. Определение достаточного уровня отказоустойчивости для обеспечения заданной вероятности безотказной работы многопроцессорной системы / В.А. Романкевич, А.П. Фесенюк, К.В. Морозов // Journal of Qafqaz university – Mathematics and computer science. – Volume 1, Number 2. – 2013. – pp.118-126. *Автору належить ідея використання підмножин додаткових векторів для уникнення недоліку в запропонованому алгоритму пошуку множини векторів стану системи. (Закордонне видання)*
7. Романкевич А.М. Об одной GL модели системы со скользящим резервом / А.М. Романкевич, В.А. Романкевич, К.В. Морозов // Радіоелектронні і комп'ютерні системи. – №5, 2013. – С.333-336. *Автором розроблено алгоритм побудови моделі системи. (Реферується наукометричними базами «Україніка наукова», «Index Copernicus»)*
8. Романкевич А.М. Об одной задаче реконфигурирования в многопроцессорных системах / А.М. Романкевич, Мораведж Сейед Милад, В.А. Романкевич, К.В. Морозов // Радіоелектронні і комп'ютерні системи. – №5, 2012. – С.91-94. *Автором запропоновано метод оцінки кількості можливих розподілів функцій по процесорах. (Реферується наукометричними базами «Україніка наукова», «Index Copernicus»)*
9. Майданюк И.В. Об одном свойстве GL-модели с минимальным количеством теряемых рёбер / И.В. Майданюк, К.В. Морозов, Е.Р. Потапова, А.В. Шурига // Науковий вісник Чернівецького університету. Сер.: Комп'ютерні системи та компоненти. – 2010. – Т.1, вип. 2. – С.31-34. *Автором доведено твердження про кількість функцій, що приймають нульове значення для другого та третього з розглянутих випадків.*
10. Морозов К.В. О модификации графо-логической модели для систем и их компонентов с множественными состояниями / К.В. Морозов, В.А. Романкевич, Т.Г. Сапсай // Системный анализ и информационные технологии: материалы 19 й Международной научно-практической

- конференции SAIT 2017. – К.: УНК «ИПСА» КПИ им. Игоря Сикорского, 2017. – С.94. *Автору належить ідея використання функцій багатозначної логіки в якості реберних функцій GL моделей, а також спосіб визначення стану системи відповідно до таких моделей.*
11. Морозов К.В. GL-модель иерархической системы с процессорами на всех уровнях иерархии / К.В. Морозов, Е.Р. Потапова, Н.К. Кичигин // Комп'ютерні системи і проектування технологічних процесів та обладнання: Матеріали всеукраїнської науково-практичної конференції (17-19 лютого 2016р). – Чернівці: ЧФ НТУ «ХП», 2016. – С.35-37. *Автору належить ідея використання проміжного рівня ієрархії для поєднання моделей, що описують поведінку системи в потоці відмов процесорів та підсистем.*
 12. Морозов К.В. О самотестировании многопроцессорных систем / К.В. Морозов, Е.Р. Потапова // Системный анализ и информационные технологии: материалы 18-й Международной научно-практической конференции SAIT 2016. – К.: УНК «ИПСА» НТУУ «КПИ», 2016. – С.396. *Автором запропоновано формальний метод визначення стану процесорів системи за результатами проведених тестових випробувань.*
 13. Морозов К.В. О графо-логических моделях взвешенных систем / К.В. Морозов, В.А. Романкевич, В.В. Олейник // Інформаційні технології в освіті, техніці та промисловості. Матеріали II всеукраїнської науково-практичної конференції молодих учених і студентів. – Івано-Франківськ – 2015. – С. 206-207. *Автором запропоновано метод побудови моделі системи із заданими цілими або дробовими вагами (кожного з її компонентів).*
 14. Потапова К.Р. Формування GL-моделей для систем з ієрархічною структурою / К.Р. Потапова, К.В. Морозов, М.І. Трошина // Прикладна математика та комп'ютинг (ПМК-2015). Збірник тез доповідей. – Київ, НТУУ "КПІ". – 2015. – С. 156 160. *Автором запропоновано метод побудови GL моделей систем з ієрархічною структурою.*
 15. Романкевич А.М. Об одном условии модификации рёберных функций GL-модели / А.М. Романкевич, В.А. Романкевич, К.В. Морозов // Системный

- анализ и информационные технологии: материалы 17-й Международной научно-практической конференции SAIT 2015. - К.: УНК «ИПСА» НТУУ «КПИ», 2015. – С.106. *Автором проаналізовано границі зміни поведінки моделі в залежності від кількості модифікованих реберних функцій.*
- 16.Романкевич А.М. О модификации реберных функций GL-моделей / А.М. Романкевич, Т.Г. Сапсай, К.В. Морозов // Інформаційно-керуючі системи на залізничному транспорті. Тези доповідей. – 2013. – №4 (додаток). – С. 36. *Автором запропоновано спосіб модифікації реберних функцій 2p-моделі*
- 17.Гроль В.В. Упрощенная 3 отказоустойчивая GL-модель для отказоустойчивых многопроцессорных систем / В.В. Гроль, К.В. Морозов, В.А. Романкевич // Інформаційно-керуючі системи на залізничному транспорті. Тези доповідей. – 2012. – №5. – С. 90. *Автором запропоновано метод побудови спрощених базових 3-відмовостійких GL моделей.*
- 18.Романкевич В.О. Базова K(3,n) GL-модель, яка будується методом зсуву змінних / В.О. Романкевич, К.В. Морозов // Прикладна математика та комп'ютинг (ПМК 2012). Збірник тез доповідей. – Київ, НТУУ "КПІ". – 2012. – С.230-235. *Автором запропоновано алгоритм побудови базових GL моделей, що базується на зсуві змінних.*
- 19.Романкевич В.А. Построение GL моделей поведения в потоке отказов для систем со скользящим резервом / В.А. Романкевич, Мораведж Милад, К.В. Морозов // Тези доповідей Третьої міжнародної науково-практичної конференції «Методи та засоби кодування, захисту й ущільнення інформації». – Вінниця, ВНТУ. – 2011. – С.36-37. *Автору належить ідея побудови системи із ковзним резервом на основі комбінації кількох базових моделей.*
- 20.Романкевич В.А. О расчёте надёжности отказоустойчивых многопроцессорных систем, подсистемы которых имеют общие процессоры / В.А. Романкевич, А.П. Фесенюк, К.В. Морозов // Інформаційно-керуючі системи на залізничному транспорті. Тези доповідей. – 2010. – №4 (додаток). – С. 6. *Автором розроблено методику оцінки похибки для запропонованого в*

роботі методу розрахунку ймовірності безвідмовної роботи ієрархічних ВБС, підсистеми яких можуть мати спільні процесори.

21. Романкевич В.А. Об оптимизации вычисления рёберных функций в GL-моделях / В.А. Романкевич, К.В. Морозов // Інформаційні технології та комп'ютерна інженерія. Тези доповідей Міжнародної науково-практичної конференції. – Вінниця, ВНТУ. – 2010. – С.379-380. *Автором запропоновано метод оптимізованого розрахунку виразів реберних функцій для базових GL-моделей.*
22. Потапова К.Р. Комп'ютерна програма «Програма підтримки експериментів з графо-логічними моделями відмово стійких багатопроцесорних систем» / К.Р. Потапова, М.І. Трошина, К.В. Морозов, Б.А. Корнейчик // Свідоцтво про реєстрацію авторського права на твір №59822 від 27.05.2015р. *Автором розроблені алгоритми побудови моделей ВБС відповідно до їх параметрів.*
23. Шурига О.В. Комп'ютерна програма «Програма формування та оцінки складності реберних функцій графо-логічної моделі поведінки відмовостійких багатопроцесорних систем у потоці відмов методом поділу множини змінних на дві рівні частини і методом нерівного поділу» / О.В. Шурига, В.О. Романкевич, К.В. Морозов // Свідоцтво про реєстрацію авторського права на твір №39445 від 02.08.2011р. *Автором розроблені алгоритми формування виразів реберних функцій графо-логічної моделі відповідно до заданих параметрів системи.*

ЗМІСТ

АНОТАЦІЯ	2
ЗМІСТ	22
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	26
ВСТУП.....	27
РОЗДІЛ 1. РОЗРАХУНОК ПОКАЗНИКІВ НАДІЙНОСТІ ВІДМОВОСТІЙКИХ БАГАТОПРОЦЕСОРНИХ СИСТЕМ.....	35
1.1. Методи розрахунку показників надійності.....	35
1.1.1 Поняття надійності та її показники	35
1.1.2 Методи розрахунку показників надійності ВБС.....	38
1.2. Системи типу k - z - n	40
1.2.1 Послідовні k - z - n системи	41
1.2.2 Системи типу (n, f, k)	41
1.2.3 Зважені k - z - n системи.....	42
1.2.4 Системи k - z - n із багатьма станами	42
1.3. Методи розрахунку показників надійності систем типу k - z - n	44
1.3.1 Ймовірність безвідмовної роботи системи з однаковими незалежними компонентами.....	44
1.3.2 Системи k - z - n з незалежними компонентами.....	46
1.3.2.1 Підхід на основі мінімальних шляхів та мінімальних розрізів ...	47
1.3.2.2 Підхід Барлоу-Хідтмана на базі породжуючої функції	48
1.3.2.3 Підхід Белфора на базі перетворення Фур'є	49
1.3.2.4 Підхід Рушді на базі симетричної булевої функції	50
1.3.2.5 Метод вкладених марківських ланцюгів	52
1.4. Методи розрахунку показників надійності зважених систем.....	53
1.4.1 Зважені k - z - n системи.....	53
1.4.2 Зважені послідовно-паралельні системи з багатьма станами.....	54
1.5. GL-моделі	59
1.5.1 GL-моделі та методи їх побудови.....	59

1.5.2	Методи модифікації GL-моделей	59
	Висновки до розділу 1	61
РОЗДІЛ 2. МОДИФІКАЦІЯ РЕБЕРНИХ ФУНКЦІЙ БАЗОВИХ GL-МОДЕЛЕЙ		62
2.1.	Межі модифікації базової $K(t, n)$ моделі шляхом зміни однієї реберної функції.....	63
2.2.	Блокування векторів шляхом посилення	66
2.3.	Граничний випадок блокування векторів шляхом посилення.....	72
2.4.	Оцінка максимальної можливої кількості векторів, що можуть бути блокованими запропонованим методом	74
2.5.	Блокування векторів шляхом послаблення.....	81
2.6.	Граничний випадок блокування векторів шляхом послаблення	86
2.7.	Комбіноване блокування векторів	88
2.8.	Модифікація обох частин однієї реберної функції	91
2.9.	Граничні випадки модифікації	98
	Висновки до розділу 2	101
РОЗДІЛ 3. МОДИФІКАЦІЯ ДЕКИЛЬКОХ РЕБЕРНИХ ФУНКЦІЙ МВР-МОДЕЛІ.....		103
3.1.	Аналіз загального випадку модифікації.....	104
3.2.	Модифікація двох реберних функцій	106
	Висновки до розділу 3	122
РОЗДІЛ 4. ВСТАНОВЛЕННЯ СТАНІВ ПРОЦЕСОРІВ ВБС В ПОТОЦІ ВІДМОВ ЗА РЕЗУЛЬТАТАМИ ЇХ ВЗАЄМНИХ ТЕСТОВИХ ПЕРЕВІРОК .		124
4.1.	Формальне представлення результатів окремих тестових перевірок ...	126
4.2.	Проведення додаткових тестів	131
4.3.	Оцінка ефективності методу для заданого набору тестів	134
	Висновки до розділу 4	139
РОЗДІЛ 5. МОДЕЛІ ІНШИХ НЕБАЗОВИХ СИСТЕМ		141
5.1.	Системи з більш ніж двома станами роботоздатності.....	141
5.1.1	GL-моделі систем з більш ніж двома станами роботоздатності	142

5.1.2 Побудова моделі простої системи з S станами роботоздатності.....	144
5.2. Зважені системи	150
5.2.1 Моделі систем з цілими вагами	150
5.2.2 Випадок систем, що містять компоненти з від'ємними вагами	153
5.2.3 Зважені системи з раціональними вагами	154
5.3. Ієрархічні системи	154
5.3.1 Побудова моделей ієрархічних систем	155
5.3.2 Системи, що містять процесори лише на нижніх рівнях ієрархії ...	156
5.3.3 Системи, що містять процесори на всіх рівнях ієрархії.....	159
5.4. Системи з ковзним резервом	162
5.4.1 Модель системи з ковзним резервом	163
5.4.2 Модель систем з «асиметричним» ковзним резервом.....	166
5.4.3 Двоїста модель.....	167
5.4.4 Модель пари підсистем «донор»-«реципієнт»	168
5.4.5 Модель «симетричної» пари підсистем «донор»-«реципієнт».....	169
5.4.6 Обмеження кількості «донорських» процесорів.....	171
5.4.7 Загальний випадок систем з ковзним резервом	173
Висновки до розділу 5	175
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ	178
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	181
Додаток А.....	195
Додаток Б.....	199
Додаток В	208
Додаток Г	212
Додаток Д.....	216
Додаток Е	219
Додаток Ж	225
Додаток И.....	228
Додаток К	231
Додаток Л	234

Додаток М 241

Додаток Н..... 243

Додаток П..... 245

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- GL-модель – графо-логічна модель поведінки ВБС в потоці відмов;
- $K(m, n)$ – базова ВБС або її GL-модель;
- ВБС – відмовостійкі багатопроцесорні системи;
- ДДНФ – досконала диз’юнктивна нормальна форма;
- МВР – мінімум втрачених (випадаючих) ребер;
- ПМЧ – Препарата-Метце-Чена;
- ПРЦ – попарні реберні цикли;
- СК – система керування;
- СКЗ – система критичного застосування.

ВСТУП

Актуальність теми. В сучасному світі все більш поширеною стає тенденція до зменшення фізичної участі людини в різних галузях діяльності. З одного боку це дозволяє збільшити якість виконуваних процесів завдяки подоланню фізіологічних обмежень людини (швидкість реакції, обмежена концентрація уваги тощо), мінімізації впливу людського фактору (можливість помилок, вплив емоційного та фізіологічного стану і т.д.), а з іншого – звільнити оператора від виконання монотонних та низькоінтелектуальних завдань. Окрім того, деякі досить складні задачі на даному етапі науково-технічного прогресу принципово не можуть виконуватися за безпосередньої участі людини (наприклад, далекі космічні місії). В таких об'єктах задачі керування частково чи повністю покладаються на так звану систему керування (СК), що виконує обробку сигналів з датчиків, формуючи керуючі сигнали. Зазвичай сучасні СК будуються на основі одного чи декількох мікропроцесорів.

В деяких випадках для системи висуваються підвищені вимоги по надійності. Це зокрема стосується так званих систем критичного застосування (СКЗ), вихід з ладу яких може призвести до значних матеріальних, фінансових або екологічних збитків, загрожувати життю та здоров'ю людей, безпеці країни і т.п. (електростанції, заводи, транспорт, особливо авіаційний та залізничний, банківські системи, космічні апарати, військова техніка тощо). СК таких об'єктів також мають бути високонадійними, для чого можуть бути побудовані на основі так званих відмовостійких багатопроцесорних систем (ВБС), що за рахунок внесення того чи іншого типу надлишковості здатні зберігати свою роботоздатність навіть при відмові деякої множини своїх процесорів.

Розробникам необхідні інструменти для оцінки параметрів надійності створюваної ними ВБС. Зазначимо, що деякі СК складаються з десятків або навіть сотен процесорів, які можуть бути різних типів та мати різні характеристики. В результаті цього, задача розрахунку параметрів надійності складних ВБС керування іноді є досить складною. Існуючі методи розрахунку

таких параметрів не завжди є досить оптимальними для використання на практиці.

Таким чином, дослідження в області розрахунку надійності ВБС є актуальними, зокрема і для України, яка є одним з провідних виробників авіаційної, космічної та військової техніки та активно розвиває свою промисловість.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційні дослідження виконувалися на кафедрі системного програмування та спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» в рамках науково-дослідницьких держбюджетних робіт:

- «Спеціалізована комп'ютерна система діагностування та розрахунку надійності реконфігурованих відмовостійких багатопроцесорних систем», № держреєстрації 0110U000262 (2010–2011 рр.).
- «Методи та засоби підвищення надійності та інформаційної стійкості багатопроцесорних систем для об'єктів критичного призначення», № держреєстрації 0113U000716 (2013-2014рр.);
- «Методи оцінки та забезпечення необхідного рівня технічної безпеки роботи спеціалізованих багатопроцесорних систем управління», № держреєстрації 0115U000323 (з 2015р.).

Матеріали також планується залучити до наукової теми, що наразі подана на конкурс в Міністерство освіти і науки України.

Мета і завдання дослідження. Метою роботи є зменшення часу виконання експерименту з GL-моделлю небазових відмовостійких багатопроцесорних систем шляхом її модифікації, а також оцінка повноти множини взаємоперевірок процесорів у системі з довільною топологією.

Основні задачі дослідження у відповідності до поставленої мети полягають в наступному.

1. Аналіз сучасних методів розрахунку показників надійності ВБС та методів моделювання поведінки ВБС в потоці відмов. Обґрунтування вибору

методу розрахунку показників надійності ВБС шляхом виконання статистичних експериментів з GL-моделями.

2. Аналіз методів побудови GL-моделей небазових систем. Обґрунтування важливості розробки методів модифікації GL-моделей шляхом зміни їх реберних функцій.

3. Аналіз властивостей MBR-моделі (моделі з мінімальним числом втрачених ребер) як різновиду GL-моделей. Розробка методу модифікації MBR-моделі шляхом зміни однієї з її реберних функцій. Аналіз множини векторів стану системи, що блокуються, в залежності від виду модифікованої функції та інших параметрів модифікації.

4. Аналіз результатів модифікації у випадку зміни кількох реберних функцій MBR-моделі. Пошук критеріїв вибору пар реберних функцій MBR-моделі, сумісних для модифікації.

5. Розробка методу та алгоритму формалізації аналізу результатів взаємного тестування пар процесорів та отримання вектору стану системи згідно з моделлю Препарати-Метса-Чена. Оцінка ефективності методу для заданої множини тестових перевірок.

6. Розробка методів побудови GL-моделей систем, компоненти яких мають різну вагу, а також систем, які, як і їх компоненти, мають більш ніж два стани роботоздатності.

7. Аналіз особливостей побудови моделей ієрархічних систем. Розробка алгоритму побудови GL-моделей систем із ковзним резервом.

Об'єктом дослідження є процеси розрахунку показників надійності складних відмовостійких багатопроцесорних систем та їх діагностування.

Предметом дослідження є методи і засоби формування графо-логічних моделей поведінки відмовостійких багатопроцесорних систем в потоці відмов і виконання з ними статистичних експериментів, а також методи отримання вектору стану таких систем шляхом проведення їх самотестування.

Методи дослідження. При виконанні досліджень дисертаційної роботи використовувалися методи теорії графів, булевої алгебри, теорії множин та

комбінаторного аналізу. При розробці моделей систем, які, як і їх компоненти, мають більш ніж два стани роботоздатності, були також використані методи багатозначної логіки.

Наукова новизна одержаних результатів визначається наступними положеннями.

1. Вперше запропоновано метод перетворення базової GL-моделі за рахунок модифікації реберних функцій MBR-моделі, який призводить до блокування певної множини векторів стану системи, що відрізняють модель від базової, з метою забезпечення її адекватності. На базі аналізу цієї множини виявлена можлива необхідність модифікації декількох функцій MBR-моделі, показана можливість появи побічних ефектів, визначено умови їх появи та характеристики. Вперше сформульовано критерії вибору реберних функцій, що можуть бути модифіковані без виникнення побічних ефектів для випадку модифікації двох реберних функцій моделі.

2. Вперше запропоновано аналітичний апарат, що базується на результатах заданої множини взаємних перевірок процесорів ВБС і який дозволяє за рахунок формування та перетворення булевого рівняння спеціального виду визначити якість заданої множини взаємоперевірок: чи вона здатна виявити стани процесорів системи.

3. Вперше запропоновано метод побудови GL-моделей, який за рахунок пі-кратного дублювання відповідних елементів вектору стану системи, вибираючи величини пі пропорційними вагам відповідних компонентів системи, дозволяє будувати моделі так званих зважених систем (компоненти їх мають різну вагу), які в загальному випадку є небазовими.

4. Вперше запропоновано узагальнення методу побудови GL-моделі за рахунок використання функцій багатозначної логіки в якості реберних функцій. Дане узагальнення дозволяє будувати моделі систем, які, як і їх компоненти, мають більш ніж два стани роботоздатності.

5. Вперше запропоновані методи побудови GL-моделей, які за рахунок комбінування MBR-моделей, а для деяких випадків також використання так

званих двоїстих моделей, вперше запропонованих в роботі, дозволяють побудову моделей ієрархічних систем, що складаються з декількох підсистем та в загальному випадку не є базовими.

Практичне значення одержаних результатів визначається тим, що в ході дисертаційної роботи запропоновано і досліджено ряд методів модифікації МВР-моделей, що дозволяють моделювати поведінку небазових ВБС в потоці відмов. Однією з переваг запропонованих підходів є те, що їх застосування не призводить до зміни структури графа моделі, в результаті чого процедура визначення зв'язності графа є тривіальною і не потребує значних обчислювальних можливостей. Запропонований аналітичний апарат для аналізу результатів взаємного тестування пар процесорів ВБС дозволяє на основі проведених тестів отримувати вектор стану системи довільної структури, а також може бути основою для розробки оптимальних або квазіоптимальних (за кількістю необхідних тестів) стратегій вибору наступних тестів в залежності від результатів попередніх.

Отримані результати можуть бути застосовані в організаціях, що займаються проектуванням, експлуатацією та оцінкою показників надійності реконфігурованих відмовостійких багатопроцесорних систем керування складними об'єктами що, зокрема підтверджується довідкою від ІТТФ НАН України.

У навчальному процесі вищої школи результати дисертаційної роботи використовуються на кафедрі системного програмування і спеціалізованих комп'ютерних систем КПП ім. Ігоря Сікорського в курсі лекцій «Тестування, надійність, контроль та діагностика комп'ютерних систем», що підтверджується відповідним актом.

Особистий внесок здобувача. Всі результати, наведені в основній частині дисертаційної роботи, отримані здобувачем самостійно.

В [9] здобувачем доведено твердження про кількість функцій, що приймають нульове значення для другого та третього з розглянутих випадків. В [8] автором запропоновано метод оцінки кількості можливих розподілів функцій

по процесорах. В [7] здобувачем розроблено алгоритм побудови моделі системи. В [6] дисертанту належить ідея використання підмножин додаткових векторів для уникнення недоліку в запропонованому алгоритму пошуку множини векторів стану системи. В [5], [16] автором запропоновано спосіб модифікації реберної функції GL моделі, сформульовано та доведено твердження щодо множини векторів, блокованих в результаті такої модифікації. В [4] дисертантом сформульовано критерії вибору пар функцій, що можуть бути сумісно модифіковані без появи побічного ефекту (блокування векторів із підвищеною кратністю відмов). В [3], [15] автором дисертації виконано аналіз границь зміни поведінки моделі в залежності від кількості модифікованих реберних функцій. В [2], [14] дисертантом запропоновано метод побудови GL моделей розглянутих ієрархічних систем, а в [1], [11] – використання проміжних допоміжних моделей для поєднання моделей, що відповідають підсистемам нижнього та поточного рівня ієрархії. В [21] здобувачем запропоновано метод оптимізованого розрахунку виразів реберних функцій для базових GL моделей. В [20] дисертантом розроблено методику оцінки похибки для запропонованого в роботі методу розрахунку ймовірності безвідмовної роботи ієрархічних ВБС, підсистеми яких можуть мати спільні процесори. В [19] здобувачу належить ідея побудови системи із ковзним резервом на основі комбінації кількох базових моделей. В [17], [18] дисертантом запропоновано алгоритм побудови спрощених базових 3 відмовостійких GL моделей, що базується на зсуві змінних. В [13] автором запропоновано метод побудови моделі системи із заданими цілими або дробовими вагами (кожного з її компонентів). В [12] дисертантом запропоновано формальний метод визначення стану процесорів системи за результатами проведених тестових випробувань. В [10] автору дисертації належить ідея використання функцій багатозначної логіки в якості реберних функцій GL моделей, а також спосіб визначення стану системи відповідно до таких моделей. В [22], [23] здобувачем розроблені алгоритми побудови моделей ВБС відповідно до їх параметрів.

Апробація результатів дисертації. Основні результати дисертації доповідались та обговорювались на 14 науково-технічних конференціях, з них 10 – міжнародних:

- Науково-технічних семінарах кафедри системного програмування і спеціалізованих комп'ютерних систем КПІ ім. Ігоря Сікорського, м. Київ (2010 – 2020 рр.)
- 17-й ÷ 19-й Международной научно-практической конференции SAIT, УНК «ИПСА» Национального технического университета Украины «Киевский политехнический институт имени Игоря Сикорского» (2015 - 2017 рр.);
- 23-й, 25-й та 26-й Міжнародній науково-практичній конференції «Інформаційно-керуючі системи на залізничному транспорті», м. Алушта (2010, 2012, 2013 рр.);
- 3-й Міжнародній науково-практичній конференції «Методи та засоби кодування, захисту та ущільнення інформації», м. Вінниця, травень 2011р.;
- Всеукраїнській науково-практичній конференції «Комп'ютерні системи і проектування технологічних процесів та обладнання», м. Чернівці (17-19 лютого 2016 р.);
- II всеукраїнській науково-практичній конференції молодих учених і студентів «Інформаційні технології в освіті, техніці та промисловості», Івано-Франківськ, 2015 р.;
- Міжнародній науково-практичній конференції «Інформаційні технології та комп'ютерна інженерія», м. Вінниця, ВНТУ (2010 рр.);
- Прикладна математика та комп'ютинг, м. Київ (2012, 2015 рр.);
- 6-й ÷ 7-й міжнародних науково-практичних конференціях «Гарантоздатні (надійні та безпечні) системи, сервіси та технології» (2012 – Севастополь, 2014 – Київ).

Публікації. Основні положення дисертаційної роботи опубліковані в 23 наукових працях, серед яких 9 статей у наукових фахових виданнях (в тому числі

3 статті у закордонних наукових періодичних виданнях, 2 з яких входять до наукометричної бази «Web of Science», та 6 статей у наукових фахових виданнях України, 4 з яких включені до наукометричних баз даних), 12 публікацій в збірниках тез доповідей науково-технічних конференцій (8 з яких – міжнародні); 2 свідоцтва про реєстрацію авторського права на твір.

Структура та обсяг дисертації. Дисертаційна робота складається із вступу, п'яти розділів, висновків і додатків. Загальний обсяг роботи становить 248 сторінок друкованого тексту (серед яких основна частина – 154 сторінки, додатки – 54 сторінки), 16 рисунків, 12 таблиць, 13 додатків та списку використаної літератури на 128 найменувань.

РОЗДІЛ 1.

РОЗРАХУНОК ПОКАЗНИКІВ НАДІЙНОСТІ ВІДМОВСТІЙКИХ БАГАТОПРОЦЕСОРНИХ СИСТЕМ

1.1. Методи розрахунку показників надійності

1.1.1 Поняття надійності та її показники

Надійність – це властивість об'єкта зберігати в часі у встановлених межах значення всіх параметрів, що характеризують його здатність виконувати необхідні функції в заданих режимах і умовах застосування, технічного обслуговування, зберігання і транспортування. Надійність є комплексною властивістю, що в залежності від призначення об'єкта та умов його застосування може включати безвідмовність, довговічність, ремонтпридатність і збереженість або певні поєднання цих властивостей [1].

Показниками надійності є кількісні характеристики надійності, які вводяться відповідно до правил статистичної теорії надійності [1]. Серед показників надійності можна виділити одиничні, тобто такі, що характеризують лише одну з властивостей надійності об'єкта, та комплексні – що характеризують одразу декілька таких властивостей [1].

Відповідно до [1] до одиничних показників надійності відносять наступні:

I. З точки зору безвідмовності (показники безвідмовності):

1. ймовірність безвідмовної роботи – ймовірність того, що в межах заданого наробітку відмови об'єкту не виникне;
2. гама-процентний наробіток до відмови – наробіток, впродовж якого відмови об'єкту не виникне з імовірністю γ , що виражена у відсотках;
3. середній наробіток до відмови – математичне сподівання наробітку об'єкту до першої відмови;

4. середній наробіток на відмову (наробіток на відмову) – відношення сумарного наробітку відновлюваного об'єкта до математичного сподівання числа його відмов впродовж цього наробітку;

5. інтенсивність відмов – умовна щільність ймовірності виникнення відмови об'єкту, що визначається за умови, що до розглянутого моменту відмова не виникла;

6. параметр потоку відмов – відношення математичного сподівання числа відмов відновлюваного об'єкту за достатньо малий його наробіток до значення цього наробітку;

7. осереднений параметр потоку відмов – відношення математичного сподівання числа відмов відновлюваного об'єкту за скінченний його наробіток до значення цього наробітку.

II. З точки зору довговічності (показники довговічності):

1. гама-процентний ресурс – сумарний наробіток, впродовж якого об'єкт не досягне граничного стану з ймовірністю γ , що виражена у відсотках;

2. середній ресурс – математичне сподівання ресурсу;

3. гама-процентний термін служби – календарна тривалість експлуатації, впродовж котрої об'єкт не досягне граничного стану з ймовірністю γ , що виражена у відсотках;

4. середній термін служби – математичне сподівання терміну служби.

III. З точки зору ремонтпридатності (показники ремонтпридатності):

1. ймовірність відновлення – ймовірність того, що час відновлення роботоздатного стану об'єкта не перевищить заданого значення;

2. гама-процентний час відновлення – час, впродовж якого відновлення роботоздатності об'єкта буде здійснено з ймовірністю γ , що виражена у відсотках;

3. середній час відновлення – математичне сподівання часу відновлення роботоздатного стану об'єкта після відмови;

4. інтенсивність відновлення – умовна щільність ймовірності відновлення роботоздатного стану об'єкту, визначена для розглянутого моменту часу, за умови, що до цього моменту відновлення не було завершено;

5. середня трудомісткість відновлення – математичне сподівання трудомісткості відновлення об'єкту після відмови.

IV. З точки зору збереженості (показники збереженості):

1. гамма-процентний строк збереженості – строк збереженості, що досягається об'єктом із заданою ймовірністю γ , що виражена у відсотках;

2. середній термін збереженості – математичне сподівання терміну збереженості.

До комплексних показників надійності відносять [1]:

1. Коефіцієнт готовності – ймовірність того, що об'єкт виявиться в роботоздатному стані в довільний момент часу, окрім запланованих періодів, впродовж котрих застосування об'єкту за призначенням не передбачено.

2. Коефіцієнт оперативної готовності – ймовірність того, що об'єкт виявиться в роботоздатному стані в довільний момент часу, окрім запланованих періодів, впродовж котрих застосування об'єкту за призначенням не передбачено, і, починаючи, з цього моменту, працюватиме безвідмовно впродовж заданого інтервалу часу.

3. Коефіцієнт технічного використання – відношення математичного сподівання сумарного часу перебування об'єкту в роботоздатному стані за деякий період експлуатації до математичного сподівання сумарного часу перебування об'єкту в роботоздатному стані та простоїв, обумовлених технічним обслуговуванням та ремонтом за той же період.

4. Коефіцієнт збереження ефективності – відношення значення показника ефективності використання об'єкта за призначенням за певну

тривалість експлуатації до номінального значення цього показника, розрахованого за умови, що відмови об'єкта впродовж того ж періоду не виникає.

Однією з найбільш важливих характеристик серед вищенаведених є ймовірність безвідмовної роботи, що в рамках даної роботи зазвичай і мається на увазі при розрахунку показників надійності.

1.1.2 Методи розрахунку показників надійності ВБС

Однією з ключових особливостей відмовостійких багатопроцесорних систем (ВБС) є, очевидно, їх відмовостійкість [2]-[7], тобто, здатність зберігати роботоздатний стан навіть при відмові деякої множини своїх процесорів. В результаті, процес розрахунку показників їх надійності, зокрема, ймовірності безвідмовної роботи стає нетривіальним.

Серед методів розрахунку показників надійності ВБС можна виділити дві основні групи:

1. Методи, що базуються на побудові аналітичних виразів для безпосереднього розрахунку показників надійності ВБС.
2. Методи, що базуються на проведенні статистичних експериментів з моделями поведінки ВБС в потоці відмов.

Методи з першої групи дозволяють отримувати аналітичні вирази для безпосереднього розрахунку показників надійності системи на базі значень показників надійності її компонентів [8], в результаті чого власне процес розрахунку має помірну розрахункову складність, а результати розрахунків можуть бути отримані із довільним значенням точності. Зокрема, до таких методів можна віднести наступні: методи послідовно-паралельної редукції [9] - [12], методи центрального розкладання (розкладання структурної функції) [11], [13] - [16], методи мінімальних шляхів та перерізів [11], [15], [17], [18], методи включення-виключення [11], методи суми неперетинних добутоків [11], [19] - [25], методи Марківських ланцюгів [11], [26] - [31], методи перетворення зірка-трикутник [11], [32] - [34], методи, що базуються на

двополюсних графах [35] - [37] тощо [35], [38] - [52]. Для більш складних систем можуть використовуватися наближені методи, серед яких методи включення-виключення, суми неперетинних добутків [11], мінімальних шляхів та перерізів [11], [38], [53].

Зокрема, аналітичні методи розрахунку показників надійності можуть застосовуватися для так званих базових або k -з- n систем, які будуть розглянуті в наступних параграфах. На жаль, для більш складних систем процес побудови аналітичних виразів іноді виявляється вкрай складним, що робить практично неможливим застосування вищезгаданих методів для розрахунку їх показників надійності.

Методи, що базуються на проведенні статистичних експериментів дозволяють отримати параметри надійності ВБС із деякою точністю, яка в загальному випадку залежить від кількості проведених експериментів. Згідно з даними підходом, при кожному експерименті випадково згенерований вектор стану системи (вектор, кожний елемент якого відповідає стану окремого компоненту системи в потоці відмов) подається на відповідну модель, котра дозволяє визначити стан ВБС в потоці відмов залежно від його значення. В якості таких моделей поведінки ВБС в потоці відмов можуть використовуватися, зокрема, блок-діаграми надійності [54] – [57], дерева відмов [58], Марківські моделі [59], мережі Петрі [60], а також GL-моделі (графо-логічні моделі), які більш детально розглядаються в наступних параграфах.

Оцінка параметрів надійності ВБС формується на базі результатів проведених експериментів із врахуванням ймовірностей виникнення відповідних векторів стану. Відомими є методи, що дозволяють дещо підвищити точність оцінки при збереженні кількості експериментів в порівнянні з тривіальним підходом. Проте, зменшення розрахункової складності моделей з метою з одного боку підвищення точності оцінки за рахунок збільшення кількості проведених експериментів, а з іншого – зменшення часу та вартості отримання такої оцінки є актуальною задачею.

1.2. Системи типу k -з- n

Системи типу k -з- n (k -out-of- n) є досить поширеними серед відмовостійких систем. За класифікацією, запропонованою в [11] можна виділити два основних типи таких систем: k -з- n :G та k -з- n :F. Система k -з- n :G складається з n компонентів та залишається роботоздатною, якщо роботоздатними є щонайменше k з них. Система k -з- n :F виходить з ладу, якщо хоча б k з n її компонентів вийшли з ладу. Очевидно, що система k -з- n :G є в той же час і системою $(n - k + 1)$ -з- n :F, і навпаки, система k -з- n :F є одночасно і системою $(n - k + 1)$ -з- n :G [11].

За іншою термінологією n -процесорна ВБС, що залишається роботоздатною рівно доти, доки відмовило не більше ніж m будь-яких її процесорів називається базовою та позначається $K(m, n)$. Відзначимо, що це в свою чергу відповідає визначенню системи $(m + 1)$ -з- n :F. Всі інші системи вважаються небазовими.

На жаль, велика кількість реальних систем не є базовими. Зокрема це стосується систем керування складними об'єктами, які можуть включати процесори різних типів із різними характеристиками та наборами операцій, в результаті чого роботоздатність системи визначається не тільки кратністю несправностей, а й тим, які саме процесори вийшли з ладу.

Відзначимо, що для таких систем аналітичні методи розрахунку параметрів надійності не завжди є ефективними, зокрема через занадто високу складність як безпосередньо відповідних виразів, так і процесу їх побудови. Для оцінки параметрів надійності таких ВБС можуть бути застосовані методи, що базуються на проведенні статистичних експериментів.

Компоненти в системі можуть мати однакові (системи з однаковими компонентами) або різні значення показників надійності. Вихід з ладу кожного з компонентів системи може бути незалежним від станів інших компонентів (системи з незалежними компонентами), або залежати від них за деяким законом.

Найбільш простим випадком є системи з однаковими і незалежними компонентами.

1.2.1 Послідовні k -з- n системи

Послідовна k -з- n :F система складається із n лінійно розміщених компонентів і виходить з ладу тоді й тільки тоді, коли щонайменше k послідовних її компонентів виходить з ладу. Узагальненням даної системи є послідовна k -в- m -з- n :F (consecutive k -within- m -out-of- n :F) система [11], [61], [62], що складається з n лінійно розміщених компонентів і виходить з ладу тоді й тільки тоді, коли m її послідовних компонентів включають щонайменше k несправних компонентів. В літературі такі системи мають декілька альтернативних імен, зокрема k -послідовна- m -з- n (k -consecutive- m -out-of- n) системи [62] та послідовні k -з- m -із- n (consecutive k -out-of- m -from- n :F) системи [63], [64]. Послідовна k -з- n :F система та система типу k -з- n :F є окремими випадками k -в- m -з- n :F системи для, відповідно, $m = k$ та $m = n$ [11].

В [65] запропоновано метод оцінки показників надійності таких систем.

1.2.2 Системи типу (n, f, k)

Система (n, f, k) складається з n лінійно або циклічно розміщених компонентів і виходить з ладу тоді і тільки тоді, коли виходить з ладу щонайменше f її компонентів, або ж щонайменше k її послідовних компонентів. Ідея (n, f, k) -систем вперше була запропонована в [66].

Вираз надійності (n, f, k) -системи для випадку однакових незалежних компонентів був отриманий в [67]:

$$R_C(n, f, k) = \sum_{j=0}^{f-1} N_C(j, n, k) p^{n-j} q^j,$$

де

$$N_C(j, n, k) = \frac{n}{n-j} \sum_{i \geq 0} (-1)^i \binom{n-j+1}{i} \binom{n-kj}{n-j}$$

є кількістю способів впорядкування j несправних та $n - j$ роботоздатних компонентів в циклі, що не містить k послідовних несправних компонентів.

Для випадку лінійних і кільцевих систем із неоднаковими компонентами, надійність може бути розрахована за допомогою Марківського ланцюга. Цей метод запропоновано в [68]. Для випадку лінійних (n, f, k) -систем запропонований алгоритм має складність $O(nN^2)$. Для випадку ж кільцевих систем складність алгоритму оцінюється як $O(knN^2)$ [11].

1.2.3 Зважені k -з- n системи

В [69] була запропонована модифікація систем k -з- n :G, що отримала назву зважена (weighted) k -з- n :G система. В такій системі кожний компонент має деяку вагу w_i , $w_i > 0$, де i – номер компонента. Загальна вага всіх компонентів – w :

$$w = \sum_{i=1}^n w_i$$

Система є роботоздатною, коли сумарна вага всіх її роботоздатних компонентів є не меншою, ніж деяке значення k , при цьому, в даному випадку воно може бути більшим, ніж n , оскільки одиниці розмірності є різними, проте, $k \leq w$.

Зважена k -з- n :G система (якщо всі w_i є цілими) еквівалентна зваженій $(w - k + 1)$ -з- n :F системі, що виходить з ладу, якщо загальна вага несправних компонентів стає не меншою, ніж $(w - k + 1)$. Окрім того, звичайна k -з- n :G система є частковим випадком зваженої k -з- n :G системи, в якій вага кожного з компонентів має значення 1 [11].

1.2.4 Системи k -з- n із багатьма станами

В попередніх параграфах передбачалося, що система, як і її компоненти можуть перебувати лише в двох станах: роботоздатному та несправному (бінарні системи). Такі системи дійсно досить часто використовуються на практиці. Проте, для деяких систем подібне припущення не дозволяє з достатньою

точністю описати їхню поведінку в потоці відмов. Так, окрім повністю роботоздатного та повністю несправного стану система та/чи її компоненти можуть мати також стани часткової роботоздатності. Відзначимо, що в загальному випадку кількість станів як в системі, так і в кожного з її компонентів може відрізнятися.

Іншими словами, система та/або її компоненти можуть знаходитися в одному з $M + 1$ можливих станів: $0, 1, 2, \dots, M$, де 0 відповідає абсолютно несправному стану, а M – в повній мірі роботоздатному [11]. Така система більш точно може бути названа як дискретна система з багатьма станами, оскільки її стан/стан її компонентів можуть бути предтавлені цілими числами [11]. Існують також так звані неперервні системи з багатьма станами, в яких стани можуть представлятися дійсними числами з певного інтервалу [70].

Для бінарних систем та компонентів одним з основних показників надійності є ймовірність безвідмовної роботи, або ж, іншими словами ймовірність перебування в стані 1. З її значення легко може бути отримана і ймовірність відмови системи/компоненту, тобто, ймовірність потрапляння її/його до стану 0. Таким чином, значення надійності системи (ймовірність її безвідмовної роботи) однозначно визначає розподіл станів бінарної системи [11].

Для системи з багатьма станами передбачається, що розподіл станів її компонентів є відомим. Важливим показником, що потребує визначення, є розподіл станів такої системи, тобто, ймовірності її перебування у кожному з можливих станів [11].

Згідно з [71] стан послідовної системи з багатьма станами дорівнюватиме стану «найгіршого» з її компонентів (тобто, такого, значення стану котрого є найменшим), а стан паралельної системи з багатьма станами дорівнюватиме стану «найкращого» з її компонентів (тобто, такого, значення стану котрого є найбільшим). Аналогічно може бути визначена і система типу k -з- n з багатьма станами [72].

Так, система k -з- n :G з багатьма станами, що містить n компонентів перебуватиме в стані, що дорівнюватиме стану «найгіршого» з k її найкращих

компонентів [73]. Така система більш точно називається простою системою k -з- n з багатьма станами [11] адже значення k є однаковим для кожного зі станів. На парактиці ж це не завжди є так.

В [74] запропоновано узагальнення k -з- n систем з багатьма станами. Так, узагальнена система k -з- n :G із багатьма станами, містить n компонентів та знаходитиметься в стані не нижчому за j ($1 \leq j \leq M$), якщо існує таке ціле l ($j \leq l \leq M$) таке, що принаймні k_l її компонентів знаходяться в стані l або вищому.

Згідно з даним визначенням значення k_j можуть відрізнятися для різних значень j . В загальному випадку значення k_j не мають бути впорядкованими, проте, іноді виділяють два спеціальних випадки [11]:

- Так звані зростаючі k -з- n системи з багатьма станами, у яких $k_1 \leq k_2 \leq \dots \leq k_M$;
- Так звані спадні k -з- n системи з багатьма станами, у яких $k_1 \geq k_2 \geq \dots \geq k_M$.

1.3. Методи розрахунку показників надійності систем типу k -з- n

1.3.1 Ймовірність безвідмовної роботи системи з однаковими незалежними компонентами

Система з однаковими незалежними компонентами складається з n компонентів, що, мають однакові ймовірності відмови, і значення цих ймовірностей не залежить від виходу з ладу тих чи інших компонентів.

Ймовірність того, що в такій системі роботоздатними є рівно k компонентів може бути розрахована за формулою [11], [75]:

$$P_e(t, k, n) = C_n^k p(t)^k q(t)^{n-k}, k = 0, 1, 2, \dots, n,$$

де:

$p(t)$ – ймовірність безвідмовної роботи компонента системи;

$q(t) = 1 - p(t)$ – ймовірність відмови компонента системи;

$C_n^k = \frac{n!}{k!(n-k)!}$ – біноміальний коефіцієнт (число сполук з n по k) [76].

Ймовірність того, що в системі із n однаковими і незалежними компонентами вийшли з ладу рівно k із них може бути розрахована за формулою [11]:

$$P_{\geq}(t, k, n) = \sum_{i=k}^n P_e(t, i, n) = \sum_{i=k}^n C_n^i p(t)^i q(t)^{n-i}.$$

Це еквівалентно тому, що в системі вийшли з ладу менше ніж k компонентів, тому [11]:

$$P_{\geq}(t, k, n) = Q_{<}(t, k, n) = \sum_{i=0}^{k-1} Q_e(t, i, n) = \sum_{i=0}^{k-1} C_n^i q(t)^i p(t)^{n-i}$$

Ймовірність того, що в системі із n однаковими і незалежними компонентами роботоздатними є не більше ніж k із них може бути розрахована за формулою [11]:

$$P_{\leq}(t, k, n) = \sum_{i=0}^k P_e(t, i, n) = \sum_{i=0}^k C_n^i p(t)^i q(t)^{n-i}.$$

Надійність системи дорівнює ймовірності того, що кількість роботоздатних компонентів є більшою або рівною k [11]:

$$R(k, n) = P_{\geq}(k, n) = \sum_{i=k}^n P_e(i, n) = \sum_{i=k}^n C_n^i p^i q^{n-i}$$

Якщо до даного виразу застосувати основне розкладання до компонента n , отримане в [77], то надійність k -з- n :G системи з однаковими незалежними компонентами може бути виражена в наступний спосіб [11]:

$$\begin{aligned} R(k, n) &= \Pr(k-1, n-1) + (1-p)R(k, n-1) = \\ &= p(R(k-1, n-1) - R(k, n-1)) + R(k, n-1) = \\ &= p \cdot \Pr(\text{рівно } k-1\text{-з-}n-1 \text{ компонентів є роботоздатними}) + \\ &\quad + R(k, n-1) = C_{n-1}^{k-1} p^k q^{n-k} + R(k, n-1), \end{aligned}$$

де \Pr позначає ймовірність.

Після перегрупування членів рівняння, отримаємо наступний вираз [11]:

$$R(k, n) - R(k-1, n-1) = C_{n-1}^{k-1} p^k q^{n-k} \text{ для } n \geq k.$$

Це рівняння показує підвищення надійності системи при збільшенні кількості її компонентів від $n - 1$ до n . Відзначимо, що в міру збільшення n ступінь підвищення надійності зменшуватиметься [11].

Середній час до відмови для загальної системи без відновлення (припускаючи, що якщо t прямує до нескінченості, а поведінка системи в потоці відмов описується функцією $R(t)$) має вигляд [11]:

$$T = MTTF = \int_0^{\infty} R(t) dt.$$

Для системи типу k -з- n :G, що містить компоненти з експоненціальним розподілом відмов і постійною інтенсивністю відмов λ , надійість компонента [11]:

$$p(t) = e^{-\lambda t}, t \geq 0.$$

Тому середній час до відмови одного компонента дорівнює $(1 / \lambda)$. Середній час до відмови всієї системи може бути розрахований за формулою [11]:

$$T = \int_0^{\infty} \sum_{m=k}^n (-1)^{m-k} C_{m-1}^{k-1} C_n^m (e^{-\lambda t})^m dt$$

$$T = \sum_{m=k}^n \frac{(-1)^{m-k}}{\lambda m} C_{m-1}^{k-1} C_n^m$$

В [78] запропоновано більш простий вираз для значення T :

$$T = \frac{1}{\lambda} \sum_{m=k}^n \frac{1}{m}.$$

1.3.2 Системи k -з- n з незалежними компонентами

Для систем із компонентами, що мають різні значення показників надійності, розрахунок ймовірності безвідмовної роботи стає дещо складнішою задачею. Складність даної задачі полягає в тому, що ймовірності появи тієї чи іншої комбінації станів процесорів (із однаковою кількістю роботоздатних та однаковою кількістю несправних процесорів) в загальному випадку є різною, а

тому підхід, описаний вище не може бути використаний. Прямий перебір всіх можливих комбінацій в свою чергу вимагає виконання дуже великої кількості розрахунків.

1.3.2.1 Підхід на основі мінімальних шляхів та мінімальних розрізів

Представимо систему у вигляді графа, ребра якого відповідають компонентам системи. Надійність системи дорівнюватиме ймовірності того, що роботоздатним є хоча б один з мінімальних шляхів. Ненадійність же системи дорівнюватиме ймовірності того що, принаймні, один мінімальний розріз відмовив. Для того, що б мінімальний шлях працював, усі його компоненти повинні працювати. Для того щоб відмовив мінімальний розріз, необхідно щоб усі його компоненти відмовили. В k - z - n : G системах є C_n^k мінімальних шляхів і C_n^{n-k+1} мінімальних розрізів. Кожний мінімальний шлях містить рівно k різних компонентів, а кожний мінімальний розріз – рівно $(n - k + 1)$ різних компонентів. Таким чином, усі мінімальні шляхи і розрізи є відомими. Необхідно визначити ймовірність того, що, принаймні, в одному з мінімальних шляхів всі компоненти є роботоздатними, або ж ймовірність того, що, принаймні, в одному з мінімальних розрізів всі компоненти відмовили [11].

Для оцінки надійності k - z - n : G систем, у яких відомі всі мінімальні шляхи і розрізи, можна застосувати метод включення-виключення [11]. Але цей метод має свої недоліки. В [79] та [80] запропоновані покращені варіанти методу включення-виключення для оцінки надійності k - z - n : G систем. У цих покращених алгоритмах виключається скорочення змінних. Проте, обидва ці алгоритми за своєю суттю є алгоритмами перебору [11].

Наприклад, формула, запропонована в [79], має вигляд:

$$R(k, n) = \sum_{i=k}^n (-1)^{i-k} C_{i-1}^{k-1} \sum_{j_1 < j_2 < \dots < j_i} \prod_{l=1}^i p_{j_l}$$

В цьому виразі для кожного значення i , внутрішній член суми дає ймовірність того, що i -й компонент є роботоздатним незалежно від того, чи

роботоздатні інші $(n - i)$ компонентів. Загальне число членів у внутрішній сумі ряду дорівнює C_n^i . Якщо всі компоненти системи є однаковими і незалежними, вираз можна переписати в наступному вигляді [11]:

$$R(k, n) = \sum_{i=k}^n C_n^i C_{i-1}^{k-1} (-1)^{i-k} p^i.$$

Існують методи оцінки надійності k -з- n :G систем, набагато більш ефективні, ніж методи включення-виключення (та його покращені варіанти).

1.3.2.2 Підхід Барлоу-Хідтмана на базі породжуючої функції

В [81] запропоновано метод розрахунку безвідмовної роботи систем k -з- n :G. Метод базується на побудові виразу породжуючої функції виду:

$$g_n(z) = \prod_{i=1}^n (q_i + p_i z) = \sum_{j=0}^n P_e(j, n) z^j,$$

де p_i та q_i – ймовірність відповідно безвідмовної роботи та відмови i -го компонента n -компонентної системи, а z – деяка змінна. Можна помітити, що після розкриття дужок у виразі у якості коефіцієнтів при кожному із z^j стоятимуть $P_e(j, n)$ – ймовірності того, що в роботоздатному стані знаходиться рівно j з n компонентів. Ймовірність того, що в роботоздатному стані знаходиться щонайменше k з n компонентів можна знайти за допомогою виразу [11]:

$$P(k, n) = \sum_{j=k}^n P_e(j, n).$$

Виходячи з цього виразу, можна показати, що має місце рекурентне співвідношення:

$$P_e(j, i) = q_i P_e(j, i - 1) + p_i P_e(j - 1, i - 1),$$

де $0 \leq i \leq n, 0 \leq j \leq n$. Причому

$$P_e(-1, i) = P_e(i + 1, i) = 0,$$

$$P_e(0, 0) = 1.$$

Загальна кількість операцій, які необхідно виконати для розрахунку даним методом: $(n - k)(1,5n + 1,5k + 4) + 3k$ [11].

1.3.2.3 Підхід Белфора на базі перетворення Фур'є

Ще один підхід, описано в [76]. Вираз породжуючої функції може бути переписаний наступним чином:

$$g_n(z) = P_\pi \prod_{i=1}^n (1 + a_i z) = P_\pi (1 + A(z)),$$

де $P_\pi = \prod_{i=1}^n p_i$, $a_i = \frac{q_i}{p_i}$, $A(z) = \prod_{i=1}^n b_i z^i$, $b_i = \sum_{1 \leq j_1 \leq j_2 \leq \dots \leq j_i \leq n} a_{j_1} a_{j_2} \cdot \dots \cdot a_{j_i}$.

Ймовірність безвідмовної роботи системи, що є роботоздатною, якщо хоча б k із n її компонентів знаходяться в роботоздатному стані може бути визначена наступним чином:

$$P(k, n) = P_\pi \left(1 + \sum_{i=1}^{n-k} b_i \right).$$

Припустимо, що $n = 2^m$, де m – ціле число. Зазначимо, що $A(z)$ є поліноміальною функцією n -го порядку. Нехай $A_1(z)$ і $A_2(z)$ – деякі поліноміальні функції порядку $\left(\frac{n}{2}\right)$, причому

$$1 + A(z) = (1 + A_1(z))(1 + A_2(z)).$$

Розкривши дужки отримуємо:

$$1 + A(z) = (1 + A_1(z))(1 + A_2(z)) = 1 + A_1(z) + A_2(z) + A_1(z)A_2(z).$$

Тоді:

$$A(z) = A_1(z) + A_2(z) + A_1(z)A_2(z).$$

Аналогічне розкладання є справедливим і для функцій $A_1(z)$ та $A_2(z)$. Вирази цих функцій мають вигляд:

$$A_1(z) = c_1 z + c_2 z^2 + \dots + c_{\frac{n}{2}} z^{\frac{n}{2}},$$

$$A_2(z) = d_1 z + d_2 z^2 + \dots + d_{\frac{n}{2}} z^{\frac{n}{2}}.$$

Визначимо дві дискретні функції, що відповідають $A_1(z)$ та $A_2(z)$:

$$f_1(x) = \begin{cases} c_{x+1}, & \text{якщо } x = 0, 1, 2 \dots \frac{n}{2} - 1, \\ 0, & \text{в іншому випадку} \end{cases}$$

$$f_2(x) = \begin{cases} d_{x+1}, & \text{якщо } x = 0, 1, 2 \dots \frac{n}{2} - 1. \\ 0, & \text{в іншому випадку} \end{cases}$$

Область визначення даних функцій – $\{0, 1, 2, \dots, \frac{n}{2} - 1\}$. Згортка цих двох функцій має вигляд:

$$f(x) = f_1(x) * f_2(x) \equiv \sum_{y=0}^{\frac{n}{2}-1} f_1(x-y) f_2(y).$$

Область визначення отриманої функції: $\{0, 1, 2, \dots, n-1\}$. Це відповідає тому факту, що добуток двох поліноміальних функцій порядку $\frac{n}{2}$ є поліноміальною функцією порядку n . Значення функції $f(x)$ при $x \in \{0, 1, 2, \dots, n-1\}$ є коефіцієнтами поліноміальної функції, що є добутком $A_1(z)A_2(z)$, а за її виразом можна отримати коефіцієнти функції $A(z)$ [11].

За теоремою про згортку для перетворення Фур'є, перетворення Фур'є згортки двох функцій дорівнює добутку перетворень Фур'є цих двох функцій. Для знаходження коефіцієнтів добутку двох поліноміальних функцій необхідно спочатку виконати перетворення Фур'є для двох дискретних функцій, що їм відповідають, знайти добуток результатів цих перетворень і виконати зворотне перетворення Фур'є. Для вищеприведеного випадку розглянутий метод для знаходження добутку виразів $1 + A_1(z)$ та $1 + A_2(z)$ вимагає $T(n) = 15n \log_2 n + 11n - 2$ операцій (за умови використання швидкого перетворення Фур'є) [11].

1.3.2.4 Підхід Рушді на базі симетричної булевої функції

Інший метод для систем із незалежними компонентами запропоновано в [77]. Нехай стан кожного елементу системи визначається змінною x_i (i – номер компонента), що має значення 1, якщо компонент є роботоздатним та 0 в

протилежному випадку. Вираз, що приймає значення 1, якщо вийшли з ладу не більш ніж k з n компонентів і 0 в протилежному випадку позначатимемо $S(k, n)$.

Надійність системи, що є роботоздатною, якщо хоча б k із n її компонентів є в роботоздатному стані, позначимо $P(k, n)$. Вона може бути визначена як ймовірність того, що $S(k, n) = 1$, тобто:

$$P(k, n) = p(S(k, n) = 1).$$

Ненадійність цієї системи, яку позначимо $Q(k, n)$, може бути визначена як:

$$Q(k, n) = p(S(k, n) = 0),$$

або:

$$Q(k, n) = p(\bar{S}(k, n) = 1).$$

Легко помітити наступні рекурентні залежності:

$$S(j, i) = x_i S(j - 1, i - 1) \vee \bar{x}_i S(j, i - 1),$$

$$\bar{S}(j, i) = x_i \bar{S}(j - 1, i - 1) \vee \bar{x}_i \bar{S}(j, i - 1).$$

Причому вважатимемо, що:

$$S(0, i) = \bar{S}(i + 1, i) = 1,$$

$$\bar{S}(0, i) = S(i + 1, i) = 0.$$

Мають місце рекурентні співвідношення для надійності та ненадійності системи:

$$P(j, i) = p_i P(j - 1, i - 1) + q_i P(j, i - 1),$$

$$Q(j, i) = p_i Q(j - 1, i - 1) + q_i Q(j, i - 1),$$

де p_i та q_i – відповідно надійність та ненадійність i -го компонента.

Вважатимемо, що:

$$P(0, i) = Q(i + 1, i) = 1,$$

$$Q(0, i) = P(i + 1, i) = 0.$$

Виходячи з того, що $p_i = 1 - q_i$, а $q_i = 1 - p_i$, вирази надійності та ненадійності системи можна записати наступним чином:

$$P(j, i) = p_i P(j - 1, i - 1) + (1 - p_i) P(j, i - 1),$$

$$Q(j, i) = (1 - q_i) p_i Q(j - 1, i - 1) + q_i Q(j, i - 1),$$

або:

$$P(j, i) = p_i(P(j-1, i-1) - P(j, i-1)) + P(j, i-1),$$

$$Q(j, i) = Q(j-1, i-1) + q_i(Q(j, i-1) - Q(j-1, i-1)).$$

Для розрахунку значення $P(k, n)$ або $Q(k, n)$ необхідно розрахувати $k(n - k + 1)$ різних значень $P(j, i)$ або $Q(j, i)$, на розрахунок кожного з яких, як можна побачити, необхідно по 3 арифметичні операції, тобто загалом необхідно виконати $3k(n - k + 1)$ арифметичних операцій [11].

1.3.2.5 Метод вкладених марківських ланцюгів

Для представлення системи $k\text{-}z\text{-}n\text{:}G$ за допомогою марківського ланцюга [11], визначимо простір станів $S = \{s_0, s_1, \dots, s_k\} = \{0, 1, \dots, k\}$, розбиття $S_i = \{i\}$ при $i = 0, 1, \dots, k$ та марківський ланцюг $\{Y_l, l \geq 0\}$:

1. $Y_l = i$, якщо рівно i з компонентів $1, 2, \dots, l$ є роботоздатними ($0 \leq i \leq k$);
2. $Y_l = k$, якщо принаймні k з компонентів $1, 2, \dots, l$ є роботоздатними.

Ймовірність переходу зі стану i в стан j для марківського ланцюга позначимо p_{ij} . Матриця переходів марківського ланцюга матиме вигляд [11]:

$$A_l = (p_{ij})_{(k+1) \times (k+1)} = \begin{bmatrix} q_l & p_l & & & & \\ & q_l & p_l & & & \\ & & \ddots & \ddots & & \\ & & & q_l & p_l & \\ & & & & q_l & p_l \\ & & & & & 1 \end{bmatrix},$$

де $i, j = 0, 1, \dots, k$. Невказані елементи матриці є рівними нулю.

Дана матриця представляє собою ймовірності перебування системи із доданим компонентом l , в стані j ($0 \leq j \leq k$), за умови, що система, яка складається з $(l-1)$ компонента перебувала в стані i ($0 \leq i \leq k$). Для систем $k\text{-}z\text{-}n\text{:}G$ нас цікавить, чи досягло число роботоздатних компонентів значення k або більшого. В зв'язку з цим простір станів системи включає стани $0, 1, \dots, k$, що представляють поступове збільшення кількості роботоздатних компонентів при збільшенні розміру системи. Коли розмір системи досягає n , то ймовірність того, що система перебуває в стані k , і є надійністю системи [11].

Використовуючи матрицю ймовірностей переходів, можна отримати наступні рекурентні вирази:

$$a_0(l) = q_1 a_0(l-1), l \geq 1$$

$$a_j(l) = p_l a_{j-1}(l-1) + q_l a_j(l-1), 1 \leq j \leq k, j \leq l \leq n$$

$$a_k(l) = p_l a_{k-1}(l-1) + a_k(l-1), k \leq l \leq n$$

де $a_j(l)$ – ймовірність того, що в системі, яка складається з l компонентів, роботоздатними є рівно j компонентів при $0 \leq j \leq k$, а $a_k(l)$ – ймовірність того, що в системі, яка складається з l компонентів роботоздатними є принаймні, k з них.

Для вищенаведених виразів мають місце наступні граничні умови:

$$a_0(0) = 1,$$

$$a_j(0) = 0, j > 0,$$

$$a_j(l) = 0, l < j.$$

Надійність системи дорівнюватиме:

$$R(k, n) = a_k(n).$$

Обчислювальна складність даного алгоритму – порядку $O(k(n-k+1))$ [11].

1.4. Методи розрахунку показників надійності зважених систем

1.4.1 Зважені k -з- n системи

Надійність $P(k, n)$ зваженої k -з- n :G системи може бути розрахована за допомогою рекурентного співвідношення, представленого в [69]. При цьому значення $P(j, i)$ відповідає ймовірності того, що в системі з i компонентами загальна вага усіх роботоздатних компонентів є не меншою, ніж j . Тоді рекурентне співвідношення має вигляд:

$$P(j, i) = p_i P(j - w_i, i - 1) + q_i P(j, i - 1).$$

Причому

$$P(j, i) = 1, \quad \text{для } j \leq 0, i \geq 0,$$

$$P(j, 0) = 0, \quad \text{для } j > 0.$$

Варто зазначити, що значення w_i в загальному випадку може бути й не цілим числом.

Більш розширене визначення зважених k -з- n систем запропоноване в [82]: в зваженій k -з- n :F системі з багатьма станами компонент i в стані j має вагу w_{ij} , $1 \leq i \leq n$, $0 \leq j \leq M$, що представляє його внесок у продуктивність системи.

Визначення зваженої k -з- n :F системи з багатьма станами: Кожний компонент системи може бути в $M + 1$ можливих станах: $0, 1, 2, \dots, M$, де M – ідеальний стан системи. Система знаходиться в стані j^F або нижчому ($1 \leq j^F \leq M$), якщо загальна вага всіх компонентів є нижчою або рівною попередньо визначеному значенню k_j^F ($k_1^F < k_2^F < \dots < k_M^F$). Позначивши стан системи через Φ^F , маємо $\Pr(\Phi^F \leq j^F) = \Pr(W \leq k_j^F)$, де $w = \sum_{j=1}^n w_i$. Оскільки стан M – ідеальний стан системи, маємо $\Pr(\Phi^F \leq M) = 1$. Визначена зважена k -з- n :F система з багатьма станами може також бути представлена у вигляді зваженої k -з- n :G системи з багатьма станами.

Нехай $G = \sum_{i=1}^n w_{iM} - W$. Якщо $W \leq k_j^F$, то $G \geq \sum_{i=1}^n w_{iM} - k_j^F$. Нехай $k_j^G \geq \sum_{i=1}^n w_{iM} - k_j^F$ відображає попередньо визначене значення для стану j^G ($j^G = M + 1 - j^F$, $1 \leq j^G \leq M$), тоді маємо ряд $k_1^G < k_2^G < \dots < k_M^G$. Вважатимемо, що система перебуває в стані j^G або вищому, якщо G – більше або дорівнює k_j^G . Маємо $\Pr(\Phi^G \geq j^G) = \Pr(G \geq k_j^G)$. Таким чином, система може бути представлена як зважена k -з- n :G система з багатьма станами. Співвідношення $\Pr(\Phi^G \geq j^G) = \Pr(\Phi^F \leq j^F)$ показує, що розподіл станів зваженої k -з- n :F системи з багатьма станами є тотожним розподілу станів зваженої k -з- n :G системи з багатьма станами [11].

1.4.2 Зважені послідовно-паралельні системи з багатьма станами

В традиційній моделі послідовно-паралельних систем з багатьма станами відсутній параметр для опису корисності кожного з компонентів/підсистеми. Розширення моделі послідовно-паралельної системи до моделі зваженої послідовно-паралельної системи з багатьма станами робить модель більш

ефективною та гнучкою. Модель послідовно-паралельних систем з багатьма станами [83], розглядає ймовірність і корисність (як кожного компонента, так і системи в цілому) в кожному стані і тому надає кращі можливості для моделювання (є більш гнучкою та більш узагальненою) в порівнянні з традиційними моделями послідовно-паралельних систем з багатьма станами.

Зважена послідовно-паралельна система з багатьма станами включає N послідовно з'єднаних підсистем, i -а підсистема включає в себе n_i паралельно з'єднаних компонентів, $1 \leq i \leq N$. Система знаходиться в стані j або вищому, якщо в кожній паралельній підсистемі корисність хоч одного з компонентів є рівною або більшою ніж деяке задане значення k_j , що в свою чергу означає, що кожна з підсистем знаходиться в стані j або вищому. Корисність зваженої паралельної системи з багатьма станами визначається як корисність її «найкращого» компонента, а корисність зваженої послідовної системи з багатьма станами визначається як корисність «найгіршого» її компонента. В кожний момент часу, корисність зваженої послідовно-паралельної системи з багатьма станами може бути визначена як:

$$\gamma_s = \min_{1 \leq i \leq N} \left(\max_{n_{i-1}+1 \leq j \leq n_i} (\gamma_j) \right),$$

де γ_s – корисність системи, а γ_j – корисність j -го компонента.

Для зваженої послідовно-паралельної системи з багатьма станами, основними показниками роботоздатності є розподіл станів системи, відповідна кожному стану корисність, надійність, а також очікуване значення корисності системи.

Кожний компонент може бути у $M + 1$ стані: $\{0, 1, 2, \dots, M\}$. Компонент i має ймовірність p_{ij} перебування в стані j , і кожному стану цього компонента відповідає деяке значення корисності u_{ij} . Значення корисності u_{ij} описує можливості компонента (в плані продуктивності) в даному стані. Система може бути описана за допомогою пари матриць: P та U . Матриця розподілу ймовірності компонентів має вигляд:

$$P = \begin{bmatrix} p_{10} & p_{11} & p_{12} & \cdots & p_{1M} \\ p_{20} & p_{21} & p_{22} & \cdots & p_{2M} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{N0} & p_{N1} & p_{N2} & \cdots & p_{NM} \end{bmatrix},$$

а матриця розподілу корисності компонентів:

$$U = \begin{bmatrix} u_{10} & u_{11} & u_{12} & \cdots & u_{1M} \\ u_{20} & u_{21} & u_{22} & \cdots & u_{2M} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{N0} & u_{N1} & u_{N2} & \cdots & u_{NM} \end{bmatrix},$$

де p_{ij} та u_{ij} – відповідно ймовірність та корисність компонента i в стані j , $i \in \{1, 2, \dots, N\}$, $j \in \{0, 1, 2, \dots, M\}$.

Кожний i -й компонент системи в свою чергу може бути описаний за допомогою пари векторів: P_i та U_i . Вектор ймовірності компонента має вигляд: $P_i = [p_{i0} \ p_{i1} \ \cdots \ p_{iM}]$, а вектор корисності компонента: $U_i = [u_{i0} \ u_{i1} \ \cdots \ u_{iM}]$.

Відповідно до визначення, ймовірність того, що зважена послідовно-паралельна система з багатьма станами знаходиться в стані j або вищому ($j = 0, 1, \dots, M$) дорівнює:

$$\Pr(\phi_s \geq j) = \Pr(\gamma_s \geq k_j) = \prod_{i=1}^N \Pr(\phi_{s_i} \geq j) = \prod_{i=1}^N \Pr(\gamma_{s_i} \geq k_j),$$

де ϕ_s – стан системи, ϕ_{s_i} – стан i -ї підсистеми, γ_s – корисність системи, γ_{s_i} – корисність i -ї підсистеми, а k_j – деяке встановлене значення корисності. Рівність показує, що корисність системи, яка перебуває в стані j або вищому має бути не нижчою, ніж значення k_j . Таким чином, і корисність кожної з підсистем має бути не нижчою ніж k_j . Ймовірність того, що кожна з підсистем знаходиться в стані j або вищому:

$$\Pr(\phi_{s_i} \geq j) = \Pr(\gamma_{s_i} \geq k_j) = \Pr\{(\gamma_{n_{i-1}+1} \geq k_j) \cup \dots \cup (\gamma_z \geq k_j) \cup \dots \cup (\gamma_{n_i} \geq k_j)\}$$

де n_i – кількість компонентів в i -й підсистемі, а γ_z – корисність z -го компонента.

Припускаємо, що всі компоненти є незалежними.

Для зважених послідовно-паралельних систем з багатьма станами, надійність може бути визначена як ймовірність того, що система знаходиться в

деякому стані r або вищому, або, іншими словами, ймовірність того, що корисність системи є вищою або рівною деякому визначеному рівню k_r , де $0 \leq r \leq M$. Надійність є функцією від матриці розподілу корисності компонентів U та матриці розподілу ймовірності компонентів P і виглядає як $R_s^r(U, P) = \Pr(\varphi_s \geq r)$. Для розрахунку надійності всієї системи спочатку необхідно розрахувати надійність кожної паралельної підсистеми. Надійність підсистеми можна розрахувати наступним чином:

$$\begin{aligned} R_s^r(U, P) &= \prod_{i=1}^N \Pr(\gamma_{s_i} \geq k_r) = \\ &= \prod_{i=1}^N \Pr\{(\gamma_{n_{i-1}+1} \geq k_r) \cup (\gamma_{n_{i-1}+2} \geq k_r) \cup \dots \cup (\gamma_{n_i} \geq k_r)\}. \end{aligned}$$

Таким чином, зважена послідовно-паралельна система з багатьма станами може бути перетворена в бінарну послідовно-паралельну систему (з двома станами). Для перетворення зваженої послідовно-паралельної системи з багатьма станами в бінарну паралельну систему можна використати наступний метод: для кожного компонента додати між собою ймовірності всіх його станів зі значенням корисності не нижчим ніж k_j , отримане значення і буде надійністю цього компонента у відповідній бінарній послідовно-паралельній системі. Враховуючи значення k_j ($j = 0, 1, \dots, M$), можна розрахувати ймовірність перебування системи в стані j ($j = 0, 1, \dots, M$) або вищому. Окрім значення надійності системи, цікавими також є розподіл станів системи, розподіл корисності системи, очікувана корисність системи.

Простий, але досить неефективний метод розрахунку показників роботоздатності системи базується на перерахунку. Він включає в себе пошук всіх комбінацій станів компонентів та розрахунок відповідних показників системи. Далі ж представлено більш ефективний поетапний алгоритм пошуку показників роботоздатності зваженої послідовно-паралельної системи з багатьма станами [83]:

Алгоритм:

Крок 1. Визначити матриці P та U для структури послідовно-паралельної системи.

Крок 2. Знайти початковий вектор ймовірності станів системи, базуючись на всіх комбінаціях станів компонентів, використавши тензорний добуток $P_v = P_1 \otimes P_2 \otimes \dots \otimes P_{n_s}$. Вектор P_v буде вектором-рядком з $(M + 1)^{n_s} \cdot n_i$ елементами, де кожний елемент представляє ймовірність однієї з комбінацій станів компонентів.

Крок 3. Для $i = 1:N$ знайти $\{U_{v_i}\}$ та U_{v_i} :

$$\{U_{v_i}\} = \{U_{n_{i-1}+1}\} \times \{U_{n_{i-1}+2}\} \times \dots \times \{U_z\} \times \dots \times \{U_{n_i}\},$$

$$U_{v_i} = \max\{U_{v_i}\},$$

де $\{U_z\}$ – множина, елементами якої є елементи U_z , \times – оператор декартового добутку. Операція $\max\{U_{v_i}\}$ замінює множину $\{U_{v_i}\}$ на вектор U_{v_i} , де кожний елемент U_{v_i} є максимальним значенням відповідного значення з множини $\{U_{v_i}\}$.

Крок 4. Для $i = 1:N$, знайти $\{U_v\}$ та U_v :

$$\{U_v\} = \{U_{v_1}\} \times \{U_{v_2}\} \times \dots \times \{U_{v_N}\},$$

$$U_v = \min\{U_v\},$$

де U_v – вектор-рядок із $(M + 1)^{n_s} \cdot n_i$ елементами, в якому кожний елемент відповідає значенню корисності системи для відповідної комбінації станів компонентів. Кожному значенню корисності системи із U_v , відповідне значення з P_v показує ймовірність цього значення корисності системи.

Крок 5. Знайти очікуване значення корисності системи:

$$\bar{U}_s = P_v \cdot U_v,$$

де \cdot – оператор скалярного добутку.

Крок 6. Поєднати всі однакові елементи в U_v та відповідні значення в P_v . В результаті буде отримаємо 2 вектори U'_v та P'_v , які містять тільки різні значення. Використовуючи напередзадані значення k_j , $0 \leq j \leq M$, можна отримати вектор станів системи φ'_v та розподіл станів системи.

Крок 7. Видалити всі значення менші ніж r в U'_v та відповідні значення ймовірностей з P'_v та побудувати вектори U''_v та P''_v . Знайти надійність системи наступним чином:

$$R_s^r(U, P) = P''_v \cdot e,$$

де e – вектор-стовпець, у якого всі елементи дорівнюють 1.

1.5. GL-моделі

1.5.1 GL-моделі та методи їх побудови

GL-моделі або графо-логічні моделі поєднують в собі властивості графів та булевих функцій [84], [85]. GL-модель являє собою неорієнтований граф, кожному ребру котрого відповідає деяка булева реберна функція, що залежить від всіх або деяких елементів вектору стану системи. Якщо реберна функція приймає нульове значення, відповідне ребро виключається з графу. Зв'язність графу моделі відповідає стану системи: втрата зв'язності графом відповідає втраті роботоздатності системою.

Відомими є декілька алгоритмів побудови GL-моделей, які дозволяють будувати моделі, що мають свої переваги та недоліки [84] – [90]. Зокрема, так звані MBR-моделі (мінімум втрачених ребер) [86] базуються на циклічних графах. Внаслідок цього процес визначення їх зв'язності є тривіальним. Окрім того, вони мають ряд інших важливих властивостей, зокрема, втрачають мінімальну кількість ребер на векторах стану системи з $m + 1$ нулем. Цей тип GL-моделей було обрано за основу для подальших досліджень в даній роботі. Приклад MBR-моделі базової системи $K(2, 9)$ представлено на рис. 1.1.

1.5.2 Методи модифікації GL-моделей

Відомі методи побудови GL-моделей дозволяють будувати моделі так званих базових систем, які втрачають роботоздатність тоді і тільки тоді, коли з ладу виходить не менш ніж m з їх процесорів. Такі моделі також називаються базовими. Проте, на практиці деякі реальні системи, зокрема, системи керування,

не є базовими. Наприклад, вихід з ладу деякої групи з m процесорів може призводити до втрати роботоздатності системи, проте, при відмові інших m процесорів система продовжуватиме працювати в звичайному режимі. Для побудови GL-моделей таких систем доводиться тим чи іншим чином модифікувати базові моделі, побудовані вищеописаними методами, з метою забезпечення адекватності моделі поведінці системи в потоці відмов.

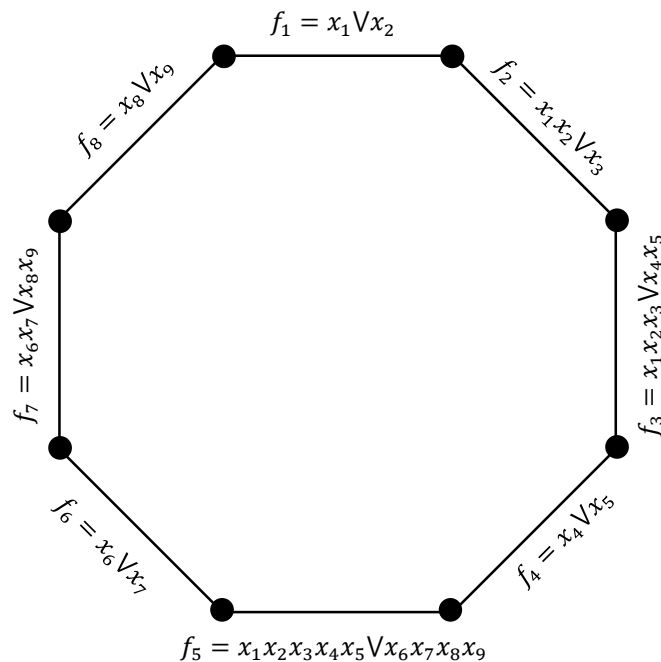


Рис. 1.1. MBP-модель базової системи $K(2, 9)$

Загалом, методи модифікації графо-логічних моделей можна розділити на три групи:

- Методи, що базуються на модифікації графа моделі.
- Методи, що базуються на модифікації реберних функцій.
- Методи, що об'єднують попередні два підходи.

Перша група методів базується на проведенні додаткових ребер (зі своїми функціями) в графі моделі та є вже досить добре вивченою [91], зокрема, вирішено задачу довільної модифікації моделі. Варто зазначити, що при вирішенні даної задачі з'являється проблема виникнення так званих попарних реберних циклів (ПРЦ) [91], для боротьби із якою свого часу було запропоновано метод побудови моделей, в яких ПРЦ не виникають. Проте, дані методи мають і

недоліки. Зокрема, в результаті модифікації моделі змінюється структура її графу (в ньому виникають нові ребра), в результаті чого, він перестає бути циклічним, що ускладнює вирішення задачі визначення його зв'язності.

Методи другої групи [92], [93] базуються на модифікації лише виразів реберних функцій моделі та, таким чином, дозволяють зберегти структуру графа, проте, поки що не є досить вивченими, зокрема, задача довільної модифікації моделі за допомогою них поки що не вирішена. Третя група методів є найменш вивченою, але виглядає потенційно досить перспективною, об'єднуючи переваги описаних вище підходів.

Висновки до розділу 1

В розділі виконано огляд основних показників надійності. Розглянуто методи розрахунку одного з найбільш важливих з них – ймовірності безвідмовної роботи. Серед них можна виділити дві групи: аналітичні методи та методи, що базуються на проведенні статистичних експериментів з моделями поведінки систем в потоці відмов.

Також розглянуті так звані базові, або k -з- n системи. Показаний взаємозв'язок між двома типами систем: k -з- n :F та k -з- n :G. Окрім того, проаналізовано розширення k -з- n систем: (n, f, k) -системи та k -в- m -з- n системи. Виконано огляд основних методів розрахунку показників надійності систем типу k -з- n як для випадку однакових незалежних компонентів, так і для випадку, коли компоненти мають різні показники надійності і є лише незалежними.

Також виконано аналіз окремих розширень таких систем: зважених систем та систем з багатьма станами. Наводяться способи їх опису та розрахунку їх показників надійності.

Розглянуто так звані GL-моделі, що можуть бути використані в якості моделей поведінки відмовостійких багатопроцесорних систем в потоці відмов, зокрема, у методах розрахунку показників надійності, які базуються на проведенні статистичних експериментів. Проаналізовано методи побудови та модифікації таких моделей.

РОЗДІЛ 2.

МОДИФІКАЦІЯ РЕБЕРНИХ ФУНКЦІЙ БАЗОВИХ GL-МОДЕЛЕЙ

Побудова графо-логічної моделі базової системи є вже вирішеною задачею. Більш того, існує декілька способів побудови базових GL-моделей [84] – [90]. Проте, на жаль, реальні системи часто не є базовими. Такі системи і відповідні їм моделі називаються небазовими.

Одне з рішень задачі побудови небазової GL-моделі полягає в модифікації деякої базової моделі. Ця модифікація може бути проведена кількома різними шляхами: модифікація структури графа (наприклад, проведення додаткових ребер [91]), зміна реберних функцій [92], [93] і поєднання цих двох підходів. У даній роботі розглядається другий підхід, що полягає в модифікації тільки реберних функцій. Однак, варто зазначити, що отримані в ній результати можуть бути корисні також і в разі поєднання даного підходу зі зміною структури графа.

Нехай модель $K'(m, n)$ отримана з моделі $K(m, n)$ таким чином, що її поведінка в потоці відмов системи відрізняється від поведінки базової моделі на деякій множині векторів стану. В загальному випадку, множину всіх векторів можна розбити на три взаємно непересічних підмножини (деякі з них можуть бути порожніми): множина векторів, на яких поведінка обох моделей однакова; множина векторів (яку позначимо як A), на яких модифікована модель, на відміну від базової, показує роботоздатний стан системи (далі будемо говорити, що модель *посилена* на цих векторах); і множина векторів (яку позначимо як B), на яких модифікована модель, на відміну від базової, показує нероботоздатний стан системи (далі будемо говорити, що модель *послаблена* на цих векторах).

Таким чином, можна сказати, що в загальному випадку початково задача модифікації базової моделі полягає в отриманні такої моделі $K'(m, n)$, для якої множини A і B будуть співпадати із заданими. Вектори, на яких поведінка моделей відрізняється, тобто вектори, що належать множинам A або B будемо називати блокованими, причому, вектори, що належать множині A називатимемо

блокованими шляхом посилення, а вектори, що належать множині B – блокованими шляхом послаблення.

2.1. Межі модифікації базової $K(m, n)$ моделі шляхом зміни однієї реберної функції

В даному розділі розглядатимуться так звані МВР-моделі, описані в [86], особливістю яких є мінімум ребер, що втрачаються при появі $m + 1$ відмови.

Розглянемо модель $K(m, n)$. Відповідно до алгоритму, запропонованого в [86] при її побудові множина вхідних змінних розбивається на 2 частини. Далі будуються аналогічні моделі для цих частин, для чого, при необхідності, підмножини також розбиваються. Слід зауважити, що деякі реберні функції можуть залежати не від усіх вхідних змінних.

Модель $K(m, n)$, побудована за методом, описаним в [86] містить реберні функції виду $f_p = \kappa_1(i, n_1) \vee \kappa_2(m - i, n_2)$. Для такої функції можна виділити 3 неперетинних підмножини вхідних змінних: V_1 , V_2 та V_3 , де V_1 і V_2 , відповідають розбиттю вхідної множини, а V_3 складається зі змінних, від яких функція не залежить (і може бути порожнім). Потужності підмножин позначимо відповідно n_1 , n_2 та n_3 . При цьому вираз $\kappa_1(i, n_1)$ відповідає кон'юнкції виразів реберних функцій моделі $K_1(i, n_1)$, побудованої для підмножини V_1 . Аналогічно $\kappa_2(m - i, n_2)$ відповідає кон'юнкції виразів реберних функцій моделі $K_2(m - i, n_2)$, побудованої для підмножини V_2 .

В [94] було показано, що кількість реберних функцій МВР-моделі $K(m, n)$ може бути розрахована за формулою:

$$\rho(m, n) = n - m + 1 \quad (2.1)$$

В [95] було доведено, що дана модель $K(m, n)$ на векторах з l нулями втрачає рівно $\varphi(m, l)$ ребер, де:

$$\varphi(m, l) = \begin{cases} 0, & \text{при } l < m \\ l - m + 1, & \text{при } l \geq m \end{cases} \quad (2.2)$$

Лема 2.1. Вираз $\kappa(t, n)$, що є кон'юнкцією виразів реберних функцій моделі $K(t, n)$ на векторі з l нулями приймає нульове значення тоді і тільки тоді, коли $l \geq t$.

Доведення. Для того, щоб кон'юнкція виразів прийняла нульове значення, необхідно і достатньо, щоб значення, рівне нулю мав хоча б один з цих виразів. Ситуація, коли хоча б одна з реберних функцій моделі $K(t, n)$ прийняла нульове значення рівнозначна втраті моделлю відповідного числа ребер, що, як можна побачити з (2.2), відбувається тоді і тільки тоді, коли $l \geq t$. ■

Таким чином, відповідно до вищедоведеної лема, кон'юнкція $\kappa_1(i, n_1)$ приймає нульове значення, тоді і тільки тоді, коли для кількості нулів l_1 в підмножині V_1 справедливо $l_1 \geq i$. Аналогічно кон'юнкція $\kappa_2(m - i, n_2)$ приймає нульове значення, тоді і тільки тоді, коли для кількості нулів l_2 в підмножині V_2 справедливо $l_2 \geq m - i$.

Стан модельованої системи (роботоздатна / відмовила) відповідає зв'язності графа моделі (зв'язний / незв'язний). Циклічний граф, взятий за основу побудови моделі, втрачає зв'язність в разі втрати будь-яких двох або більше ребер.

Розглянемо модифікацію однієї реберної функції такої моделі в загальному випадку, тобто фактично її заміну довільною функцією [93], [96]. Оцінимо межі впливу такої модифікації на поведінку моделі.

Теорема 2.1. Поведінка моделі, отриманої внаслідок довільної модифікації будь-якої однієї реберної функції базової МВР моделі $K(t, n)$ буде відрізнятися від поведінки базової моделі тільки на деяких векторах з t нулями (такі вектори будуть блоковані шляхом послаблення) і/або на деяких векторах з $t + 1$ нулем (такі вектори будуть блоковані шляхом посилення).

Доведення. Нехай модель $K'(t, n)$ отримана з моделі $K(t, n)$ шляхом заміни деякої її реберної функції f_p на деяку іншу функцію f'_p . Побудуємо таблицю залежності кількості втрачених ребер: $(\varphi$ та $\varphi')$ і поведінки моделей $K(t, n)$ та $K'(t, n)$ від кількості нулів l в векторі стану та значень на ньому функцій f_p і f'_p (таблиця 2.1). Стан моделі в таблиці будемо позначати символами "+"

(роботоздатний) і "-" (нероботоздатний). Як вже було сказано, циклічний граф моделі втрачає зв'язність (що відповідає нероботоздатному стану системи), якщо дві або більше її реберних функцій приймають значення, рівне нулю.

Відзначимо, що відповідно до (2.2) на векторах з $l < m$ нулями базова модель не втрачає жодного ребра, тому на таких векторах f_p приймає тільки ненульові значення. На векторах з $m \leq l < n$ нулями f_p може приймати значення як 0, так і 1, а на векторі, що складається лише з нулів ($l = n$), як можна побачити з (2.1 і 2.2), всі реберні функції, в тому числі і f_p , приймають значення, рівне нулю. Функція f'_p , в свою чергу, в залежності від виконаної модифікації, може приймати значення як 0, так і 1 на будь-яких векторах.

Таблиця 2.1.

Залежність кількості втрачених ребер і поведінки моделей від кількості нулів у векторі стану і значень на ньому функцій f_p і f'_p

l	f_p	f'_p	φ	φ'	$K(m,n)$	$K'(m,n)$
$<m$	0	0	0	0	+	+
$<m$	0	1	0	1	+	+
m	0	0	1	1	+	+
m	0	1	1	0	+	+
m	1	0	1	2	+	–
m	1	1	1	1	+	+
$m+1$	0	0	2	2	–	–
$m+1$	0	1	2	1	–	+
$m+1$	1	0	2	3	–	–
$m+1$	1	1	2	2	–	–
$\geq m+2$	0	0	≥ 3	≥ 3	–	–
$\geq m+2$	0	1	≥ 3	≥ 2	–	–
$\geq m+2$	1	0	≥ 3	≥ 4	–	–
$\geq m+2$	1	1	≥ 3	≥ 3	–	–

Як видно з таблиці поведінки вихідної і модифікованої моделей відрізняється тільки в двох випадках:

1) На векторах з m нулями, коли вихідна функція приймає значення 1, а модифікована – 0. В такому випадку базова модель показує роботоздатний стан системи, а модифікована – нероботоздатний. Таким чином, ці вектори будуть блоковані шляхом послаблення.

2) На векторах з $m + 1$ нулем, коли вихідна функція приймає значення 0, а модифікована – 1. В такому випадку базова модель показує нероботоздатний стан системи, а модифікована – роботоздатний. Таким чином, ці вектори будуть блоковані шляхом посилення. ■

Відзначимо також, що відповідно до Таблиці 2.1 вектори з m нулями, на яких f_p приймала значення, рівне нулю, а також вектори з $m + 1$ нулем, на яких вона приймала значення, рівне одиниці, не можуть бути блоковані шляхом модифікації тільки цієї функції.

2.2. Блокування векторів шляхом посилення

Розглянемо підхід, що дозволяє модифікувати довільну реберну функцію базової МВР-моделі, так, щоб блокувати шляхом посилення деякі з векторів стану системи.

Для модифікації моделі $K(m, n)$ пропонується в реберній функції $f_p = \kappa_1(i, n_1) \vee \kappa_2(m - i, n_2)$ певним чином модифікувати вираз $\kappa_1(i, n_1)$, отримавши замість нього вираз $\kappa'_1(i, n_1)$, який відрізняється від оригінального тим, що не приймає нульового значення на деякій множині B підвекторів з $l_1 = i$ нулями. Таким чином, в результаті заміни у вихідній моделі функції f_p на іншу функцію $f_p = \kappa'_1(i, n_1) \vee \kappa_2(m - i, n_2)$, отримаємо модифіковану модель $K'(m, n)$.

Відповідно до описаного вище розбиття множини вхідних змінних на підмножини V_1 , V_2 і V_3 деякий вхідний вектор v , що містить l нулів і має довжину n може бути розбитий на підвектори v_1 , v_2 і v_3 . При цьому довжини цих векторів

відповідно рівні n_1, n_2 і n_3 . Ці вектори містять відповідно l_1, l_2 і l_3 нулів. Очевидно, що $n = n_1 + n_2 + n_3$, а $l = l_1 + l_2 + l_3$.

Перейдемо до розгляду деяких властивостей модифікованої моделі.

Властивість 1

Для запропонованої модифікації множиною блокованих векторів буде об'єднання двох підмножин:

1) Всі вектори, у яких v_1 належить множині B , v_2 – будь-який вектор з $m - i$ нулями і v_3 – будь-який вектор з одним нулем.

2) Вектори, у яких v_1 належить множині B , v_2 – будь-який вектор з $m - i + 1$ нулями і v_3 – який не містить нулів.

Потужність множини блокованих векторів:

$$N = |B| \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}) \quad (2.3)$$

Далі відзначимо, що дана модель $K(m, n)$ може використовуватися в якості службової (проміжної) для побудови однієї з реберних функцій базової моделі, побудованої за [86], яка описує поведінку системи $K_H(M, N)$ в потоці відмов. Заміна моделі $K(m, n)$ на модифіковану модель $K'(m, n)$ дозволяє отримати модифіковану модель $K_H'(M, N)$.

Нехай $\kappa(m, n)$ – кон'юнкція всіх реберних функцій моделі $K(m, n)$, $K(m, n)$, а $\kappa'(m, n)$ – кон'юнкція всіх реберних функцій моделі $K'(m, n)$. Для визначення властивостей моделей, при побудові яких проводиться заміна оригінальної підмоделі $K(m, n)$ на підмодель $K'(m, n)$, необхідно знати множину векторів, на яких значення кон'юнкцій $\kappa(m, n)$ та $\kappa'(m, n)$ відрізняються. Ця множина є аналогом множини B для модифікації моделі $K_H(M, N)$.

Властивість 2

Вектори, на яких значення $\kappa(m, n)$ та $\kappa'(m, n)$ відрізнятимуться це вектори, у яких v_1 належить множині B , v_2 – будь-який вектор з $m - i$ нулями і v_3 – який не містить нулів.

Кількість таких векторів:

$$N = |B| \cdot C_{n_2}^{m-i} \quad (2.4)$$

Перейдемо до доведення справедливості сформульованих положень. Відзначимо, що значення функцій f_p' та f_p відрізнятимуться на деяких векторах. Розглянемо всі можливі значення виразів $\kappa_1(i, n_1)$, $\kappa_1'(i, n_1)$ і $\kappa_2(m-i, n_2)$, а також значення функцій f_p' та f_p в залежності від них (таблиця 2.2). Відзначимо також, що ситуація, коли $\kappa_1(i, n_1) = 1$, а $\kappa_1'(i, n_1) = 0$ в принципі не можлива.

Таблиця 2.2

Значення виразів $\kappa_1(i, n_1)$, $\kappa_1'(i, n_1)$ та $\kappa_2(m-i, n_2)$ і функцій f_p' та f_p

$\kappa_1(i, n_1)$	$\kappa_1'(i, n_1)$	$\kappa_2(m-i, n_2)$	f_p	f_p'
0	0	0	0	0
0	1	0	0	1
1	1	0	1	1
0	0	1	1	1
0	1	1	1	1
1	1	1	1	1

З таблиці видно, що значення функцій f_p та f_p' відрізняються тільки в разі одночасного виконання трьох умов: $\kappa_1(i, n_1) = 0$, $\kappa_1'(i, n_1) = 1$ і $\kappa_2(m-i, n_2) = 0$.

Перші дві умови виконуються тоді і тільки тоді, коли $v_1 \in B$, а третя – коли $l_2 \geq m-i$. Зауважимо, що всі вектори в B містять i нулів, тобто $l_1 = i$.

Проаналізуємо поведінку моделі $K'(m, n)$, отриманої шляхом заміни в моделі $K(m, n)$ реберної функції f_p на f_p' .

Розглянемо, для яких векторів зв'язність графів вихідної і модифікованої моделей буде відрізнятися.

Оскільки модифікованою є лише одна з реберних функцій, то відмінності будуть мати місце тільки в разі, коли вихідна модель втрачає 2 ребра ($f_p = 0$) і граф втрачає зв'язність, а модифікована – 1 ребро ($f_p' = 1$) і граф залишається зв'язним. Протилежна ситуація, коли вихідна і модифікована моделі втрачають відповідно 1 і 2 ребра (тобто $f_p = 1$ і $f_p' = 0$), відповідно до таблиці 2.2 – неможлива.

Відповідно до (2.2), модель $K(m, n)$ втрачає 2 ребра тільки тоді, коли вхідний вектор містить рівно $l = m + 1$ нулів. При цьому виходячи з необхідності відмінності значень функцій f_p та f_p' маємо додаткові умови: $v_1 \in B$ ($l_1 = i$) і $l_2 \geq m - i$.

Згадаймо, що $l_1 + l_2 + l_3 = l$. Виходячи з умов: $l = m + 1$, $l_1 = i$, $l_2 \geq m - i$, і враховуючи, що l, l_1, l_2, l_3 – цілі невід'ємні числа, допустимі лише два варіанти:

- 1) $v_1 \in B$ ($l_1 = i$), $l_2 = m - i$, $l_3 = 1$;
- 2) $v_1 \in B$ ($l_1 = i$), $l_2 = m - i + 1$, $l_3 = 0$.

Сказане стосується векторів, що відповідають *Властивості 1* моделі.

Порахуємо кількість таких векторів:

- 1) Кількість векторів $v_1 \in B$ дорівнює потужності множини B .
- 2) Кількість векторів v_2 з $m - i$ нулями дорівнює $C_{n_2}^{m-i}$.
- 3) Кількість векторів v_2 з $m - i + 1$ нулями дорівнює $C_{n_2}^{m-i+1}$.
- 4) Кількість векторів v_2 з одним нулем дорівнює $C_{n_3}^1 = n_3$.
- 5) Кількість векторів v_3 без нулів дорівнює $C_{n_3}^0 = 1$.

Виконавши нескладні перетворення отримуємо:

$$N = |B| \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}), \text{ що відповідає виразу (2.3).}$$

Далі визначимо, на яких векторах значення $\kappa(m, n)$ та $\kappa'(m, n)$ будуть різними. Вирази $\kappa(m, n)$ і $\kappa'(m, n)$ приймають значення 0 тоді і тільки тоді, коли відповідно моделі $K(m, n)$ і $K'(m, n)$ втрачають хоча б одне ребро. Відмінності між значеннями $\kappa(m, n)$ та $\kappa'(m, n)$ будуть тільки в разі, якщо $K(m, n)$ втрачає одне ребро ($f_p = 0$), а $K'(m, n)$ не втрачає ребер ($f_p' = 1$).

Відповідно до (2.2) модель $K(m, n)$ втрачає одне ребро тільки тоді, коли вхідний вектор v містить рівно $l = m$ нулів. При цьому, виходячи з необхідності відмінності значень функцій f_p та f_p' маємо додаткові умови: $v_1 \in B$ ($l_1 = i$) и $l_2 \geq m - i$.

Виходячи з умов: $l = m$, $l_1 = i$, $l_2 \geq m - i$, і враховуючи, що l, l_1, l_2, l_3 – цілі невід'ємні числа, допустимий лише один варіант: $v_1 \in B$ ($l_1 = i$), $l_2 = m - i$, $l_3 = 0$,

Сказане стосується векторів, що відповідають *Властивості 2* моделі.

В свою чергу, кількість таких векторів:
 $N = |B| \cdot C_{n_2}^{m-i}$, що відповідає виразу (2.4).

Одним з можливих варіантів модифікації моделі $K_I(i, n_I)$ є її заміна на модель, з більшою на 1 степінню відмовостійкості, тобто $K_I(i + 1, n_I)$. В такому випадку вектор B міститиме всі можливі підвектори v_1 з рівно i нулями.

Відповідно до доведених вище властивостей, вектори, на яких поведінка моделей буде відрізнятися це:

1) Вектори, у яких v_1 – будь-який вектор з i нулями, v_2 – будь-який вектор з $m - i$ нулями і v_3 – будь-який вектор з одним нулем.

2) Вектори, у яких v_1 – будь-який вектор з i нулями, v_2 – будь-який вектор з $m - i + 1$ нулями і v_3 – вектор, що не містить нулів.

Кількість векторів, на яких поведінка моделей буде відрізнятися:

$$N = C_{n_1}^i \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1})$$

Відмінності між $\kappa(m, n)$ та $\kappa'(m, n)$ в такому випадку матимуть місце на векторах, у яких v_1 – будь-який вектор з i нулями, v_2 – будь-який вектор з $m - i$ нулями і v_3 – вектор, який не містить нулів.

Кількість таких векторів: $N = C_{n_1}^i \cdot C_{n_2}^{m-i}$.

Приклад

Побудуємо модель $K(3, 11)$ згідно із [86], вона міститиме наступні реберні функції:

$$f_1 = x_1 \vee x_2 \vee x_3$$

$$f_2 = (x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4 x_5 x_6$$

$$f_3 = x_1 x_2 x_3 \vee (x_4 \vee x_5)(x_4 x_5 \vee x_6)$$

$$f_4 = x_4 \vee x_5 \vee x_6$$

$$f_5 = (x_1 \vee x_2)(x_1 x_2 \vee x_3)(x_1 x_2 x_3 \vee x_4 x_5 x_6)(x_4 \vee x_5)(x_4 x_5 \vee x_6) \vee x_7 x_8 x_9 x_{10} x_{11}$$

$$f_6 = x_1 x_2 x_3 x_4 x_5 x_6 \vee (x_7 \vee x_8)(x_7 x_8 \vee x_9)(x_7 x_8 x_9 \vee x_{10} x_{11})(x_{10} \vee x_{11})$$

$$f_7 = x_7 \vee x_8 \vee x_9$$

$$f_8 = (x_7 \vee x_8)(x_7 x_8 \vee x_9) \vee x_{10} x_{11}$$

$$f_9 = x_7 x_8 x_9 \vee x_{10} \vee x_{11}$$

Модифікуватимемо функцію f_8 . При побудові вона мала вигляд $f_8 = \kappa_1(2,3) \vee \kappa_2(1,2)$, і при цьому $V_1 = \{x_7, x_8, x_9\}$, $V_2 = \{x_{10}, x_{11}\}$, $V_3 = \{x_1, x_2, x_3, x_4, x_5, x_6\}$. Модифікуємо цю функцію наступним чином: $f'_8 = \kappa_1(3,3) \vee \kappa_2(1,2)$, в результаті чого отримаємо: $f'_8 = x_7 \vee x_8 \vee x_9 \vee x_{10}x_{11}$.

Експерименти показують, що поведінка моделей відрізняється на наступних 39 векторах: {11111010010, 11110110010, 11101110010, 11011110010, 10111110010, 01111110010, 11111001010, 11110101010, 11101101010, 11011101010, 10111101010, 01111101010, 11111000110, 11110100110, 11101100110, 11011100110, 10111100110, 01111100110, 11111010001, 11110110001, 11101110001, 10111110001, 01111110001, 11111001001, 11110101001, 11101101001, 11011101001, 10111101001, 01111101001, 11111000101, 11110100101, 11101100101, 11011100101, 10111100101, 01111100101, 11111110000, 11111101000, 11111100100}.

Якщо підрахувати кількість векторів, керуючись запропонованими вище співвідношеннями (враховуючи, що $i = 2$, $m = 3$, $n_1 = 3$, $n_2 = 2$, $n_3 = 6$), отримаємо:
$$N = C_{n_1}^i \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}) = C_3^2 \cdot (C_2^1 \cdot 6 + C_2^2) = 3 \cdot (2 \cdot 6 + 1) = 39$$

Як бачимо приклад підтверджує справедливість доведених вище положень. Порівняємо множину блокованих векторів із передбаченою теоретично:

1) Вектори, у яких підвектор v_1 – будь-який вектор з 2 нулями (тобто {100, 010, 001}), підвектор v_2 – будь-який вектор з 1 нулем (тобто {10, 01}) і підвектор v_3 – будь-який вектор з 1 нулем (тобто {111110, 111101, 111011, 110111, 101111, 011111}).

2) Вектори, у яких підвектор v_1 – будь-який вектор з 2 нулями (тобто {100, 010, 001}), підвектор v_2 – будь-який вектор з 2 нулями (тобто {00}) і підвектор v_3 – не містить нулів (тобто {111111}).

Як бачимо, вектори з п. 1 відповідають першим 36 з множини отриманих векторів, а вектори з п. 2 – останнім 3.

2.3. Граничний випадок блокування векторів шляхом посилення

Окремо розглянемо ситуацію, коли $i = n_1$. В цьому випадку функція, що модифікується має вигляд: $f_p = \kappa_1(i, i) \vee \kappa_2(m - i, n_2)$. Для побудови модифікованої функції, $f'_p = \kappa'_1(i, i) \vee \kappa_2(m - i, n_2)$, відповідно до запропонованого вище способу необхідно було б побудувати модель $K_1(i + 1, i)$, однак, це неможливо.

Відповідно до Лема 2.1, вираз $\kappa_1(i, i)$ приймає значення 0 тільки в одному випадку: якщо всі i змінних, що входять в множину V_1 приймають значення 0. Таким чином, існує тільки одне значення вектору v_1 , на якому $\kappa_1(i, i) = 0$: це вектор, що складається з i нулів. Отже, існує єдиний можливий варіант модифікації виразу $\kappa_1(i, i)$ запропонованим в попередньому параграфі способом, при якому множина B буде складатися з одного вектору, всі i елементів якого дорівнюють нулю. Вираз $\kappa'_1(i, i)$ відрізнятиметься від $\kappa_1(i, i)$ тільки тим, що прийматиме значення, рівне 1 на цьому векторі, і, отже, ніколи не прийматиме значення 0, тобто, $\kappa'_1(i, i) \equiv 1$.

Вираз модифікованої функції матиме вигляд: $f'_p = 1 \vee \kappa_2(m - i, n_2)$, після спрощення якого отримаємо $f'_p \equiv 1$. Таким чином, відповідне ребро в графі моделі буде присутнім незалежно від значень вектора стану системи. З метою спрощення дане ребро може бути виключене з графа, а інцидентні йому вершини об'єднані в одну. Завдяки тому, що граф є циклічним, обидві моделі (до і після виключення ребра) будуть еквівалентними, так як зв'язність їх графів співпадатиме на будь-якому векторі стану. В результаті, граф моделі залишиться циклічним, але міститиме вже на одне ребро менше, ніж граф вихідної моделі.

Враховуючи те, що множина B складається з одного вектора, всі i елементів якого дорівнюють нулю, вектори, на яких поведінка моделей відрізнятиметься це:

1) Вектори, у яких v_1 – вектор, що складається лише з нулів, v_2 – будь-який вектор з $m - i$ нулями і v_3 – будь-який вектор з одним нулем.

2) Вектори, у яких v_1 – вектор, що складається тільки з нулів, v_2 – будь-який вектор з $m - i + 1$ нулями і v_3 – вектор, який не містить нулів.

Враховуючи, що $|B| = 1$, кількість векторів, на яких поведінка моделей відрізнятиметься: $N = C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}$.

Приклад

Побудуємо модель $K(3, 7)$ згідно з [86], вона міститиме наступні реберні функції:

$$f_1 = x_1 \vee x_2 \vee x_3 x_4$$

$$f_2 = x_1 x_2 \vee x_3 \vee x_4$$

$$f_3 = (x_1 \vee x_2)(x_1 x_2 \vee x_3 x_4)(x_3 \vee x_4) \vee x_5 x_6 x_7$$

$$f_4 = x_1 x_2 x_3 x_4 \vee (x_5 \vee x_6)(x_5 x_6 \vee x_7)$$

$$f_5 = x_5 \vee x_6 \vee x_7$$

Модифікуватимемо функцію f_1 . При побудові вона мала вигляд $f_1 = \kappa_1(2,2) \vee \kappa_2(1,2)$, і при цьому $V_1 = \{x_1, x_2\}$, $V_2 = \{x_3, x_4\}$, $V_3 = \{x_5, x_6, x_7\}$. В результаті модифікації моделі ми отримаємо нову реберну функцію $f'_1 \equiv 1$. Виключивши відповідне їй ребро ми отримаємо нову модель вже з чотирма функціями:

$$f_{H1} = x_1 x_2 \vee x_3 \vee x_4$$

$$f_{H2} = (x_1 \vee x_2)(x_1 x_2 \vee x_3 x_4)(x_3 \vee x_4) \vee x_5 x_6 x_7$$

$$f_{H3} = x_1 x_2 x_3 x_4 \vee (x_5 \vee x_6)(x_5 x_6 \vee x_7)$$

$$f_{H4} = x_5 \vee x_6 \vee x_7$$

Експерименти показують, що поведінка моделей відрізняється на наступних 7 векторах: $\{0010110, 0001110, 0010101, 0001101, 0010011, 0001011, 0000111\}$.

Якщо підрахувати кількість векторів теоретично (враховуючи, що $i = 2$, $m = 3$, $n_1 = 2$, $n_2 = 2$, $n_3 = 3$), отримаємо:

$$N = C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1} = C_2^1 \cdot 3 + C_2^2 = 2 \cdot 3 + 1 = 7.$$

Порівняємо множину блокованих векторів з передбаченою теоретично:

1) Вектори, у яких підвектор v_1 – вектор, що складається лише з нулів (тобто $\{00\}$), підвектор v_2 – будь-який вектор з 1 нулем (тобто $\{10, 01\}$) і підвектор v_3 – будь-який вектор з 1 нулем (тобто $\{110, 101, 011\}$).

2) Вектори, у яких підвектор v_1 – вектор, що складається тільки з нулів (тобто $\{00\}$), підвектор v_2 – будь-який вектор з 2 нулями (тобто $\{00\}$) і підвектор v_3 – вектор, який не містить нулів (тобто $\{111\}$).

Як бачимо, вектори з п. 1 відповідають першим 6 з множини отриманих векторів, а єдиний вектор з п. 2 – останньому з них.

Таким чином, поведінка моделі, отриманої запропонованим вище способом відповідає очікуваній поведінці модифікованої моделі.

2.4. Оцінка максимальної можливої кількості векторів, що можуть бути блокованими запропонованим методом

Оцінимо максимальну кількість векторів, які можуть бути блоковані запропонованим методом. Кількість блокованих векторів розраховується за формулою $N = |B| \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1})$. Максимально можливе значення $|B|$ буде отримано в разі, якщо B включає в себе всі можливі вектори з i нулями, в такому випадку $|B| = C_{n_1}^i$, і кількість блокованих векторів розраховується за формулою $N = C_{n_1}^i \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1})$. Параметри n та m є параметрами базової моделі і не можуть бути змінені. Параметри n_1 , n_2 та n_3 визначаються розбиттям множини вхідних змінних при побудові базової моделі. Це розбиття може бути довільним, і в будь-якому випадку буде отримана базова модель. Параметри n_1 , n_2 та n_3 є цілими числами і на них накладені наступні обмеження: $n_1 + n_2 + n_3 = n$, $n_1 > 0$, $n_2 > 0$, $n_3 \geq 0$. Параметр i визначається видом функції, що модифікується. Ми можемо модифікувати будь-яку функцію моделі, тому параметр i може бути будь-яким цілим числом, для якого справедливо $1 \leq i \leq m - 1$. Окрім того, з логіки побудови моделі маємо додаткові обмеження: $i \leq n_1$ та $m - i \leq n_2$.

Для виконання подальшого аналізу доведемо наступні твердження.

Лема 2.2. $C_n^{m-1} + C_n^m = C_{n+1}^m$ (для $1 \leq m \leq n$).

Доведення. Згадаймо, що $C_n^m = \frac{n!}{m! \cdot (n-m)!}$. Отже, $C_n^{m-1} + C_n^m =$

$$= \frac{n!}{(m-1)! \cdot (n-(m-1))!} + \frac{n!}{m! \cdot (n-m)!} = \frac{n!}{m! \cdot (n-m+1)!} \times$$

$$\times (n-m+1+m) = \frac{n!}{m! \cdot (n-m+1)!} \cdot (n+1) = \frac{(n+1)!}{m! \cdot ((n+1)-m)!} =$$

$$= C_{n+1}^m. \blacksquare$$

Лема 2.3. $k \cdot C_n^{m-1} + C_n^m \leq (k-1) \cdot C_{n+1}^{m-1} + C_{n+1}^m$ (для $1 \leq m \leq n, k \geq 1$).

Доведення. Розглянемо два випадки:

1. $m = 1$. Отже, $k \cdot C_n^{m-1} + C_n^m = k \cdot C_n^0 + C_n^1 = k + n$, а
 $(k-1) \cdot C_{n+1}^{m-1} + C_{n+1}^m = (k-1) \cdot C_{n+1}^0 + C_{n+1}^1 = (k-1) + (n+1) = k + n$.
 Таким чином, для $m = 1$ справедливо: $k \cdot C_n^{m-1} + C_n^m = (k-1) \cdot C_{n+1}^{m-1} + C_{n+1}^m$,
 що відповідає твердженню леми.

2. $m \geq 2$. Відповідно до Лема 2.2 $C_{n+1}^m = C_n^{m-1} + C_n^m$ та
 $C_{n+1}^{m-1} = C_n^{m-2} + C_n^{m-1}$. Отже, $(k-1) \cdot C_{n+1}^{m-1} + C_{n+1}^m = (k-1) \times$
 $\times (C_n^{m-2} + C_n^{m-1}) + (C_n^{m-1} + C_n^m) = k \cdot C_n^{m-1} + C_n^m + (k-1) \cdot C_n^{m-2}$. Для $k \geq 1$ і
 $2 \leq m \leq n$ справедливо: $(k-1) \cdot C_n^{m-2} \geq 0$. Таким чином,
 $(k-1) \cdot C_{n+1}^{m-1} + C_{n+1}^m = k \cdot C_n^{m-1} + C_n^m + (k-1) \cdot C_n^{m-2} \geq k \cdot C_n^{m-1} + C_n^m$, що
 відповідає твердженню леми. \blacksquare

Лема 2.4. $k \cdot C_n^{m-1} + C_n^m \leq C_{n+k}^m$ (для $1 \leq m \leq n, k \geq 1$).

Доведення. Відповідно до Лема 2.3 справедливо: $k \cdot C_n^{m-1} + C_n^m \leq$
 $\leq (k-1) \cdot C_{n+1}^{m-1} + C_{n+1}^m \leq (k-2) \cdot C_{n+2}^{m-1} + C_{n+2}^m \leq \dots \leq (k-k) \cdot C_{n+k}^{m-1} +$
 $+ C_{n+k}^m$. Відмітимо, що $(k-k) \cdot C_{n+k}^{m-1} + C_{n+k}^m = C_{n+k}^m$. Отже, внаслідок
 транзитивності відношення лінійного порядку справедливим буде і
 $k \cdot C_n^{m-1} + C_n^m \leq C_{n+k}^m$, що відповідає твердженню леми. \blacksquare

Лема 2.5. Для фіксованого значення параметрів i та n_1 , максимальну для
 даних параметрів (m, n, i та n_1) кількість векторів стану системи може бути
 блоковано в результаті модифікації функції, що залежить від всіх змінних (тобто
 якщо $n_3 = 0$).

Доведення. Як вже було сказано, для моделі $K(m, n)$ максимальна кількість блокованих векторів (при модифікації однієї реберної функції запропонованим методом) може бути розрахована за формулою: $N = C_{n_1}^i \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1})$. Припустимо, що максимальне число блокованих векторів стану системи для даних параметрів досягається в результаті модифікації функції, для якої $n_3 \geq 1$. Тоді відповідно до Лема 2.4 та зважаючи на те, що $1 \leq i < m$, і, отже, $1 \leq m - i < m$, справедливо: $C_{n_1}^i \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}) \leq C_{n_1}^i \cdot C_{n_2+n_3}^{m-i+1}$. Вираз $C_{n_1}^i \cdot C_{n_2+n_3}^{m-i+1}$, в свою чергу, відповідає ситуації, коли реберна функція залежить від всіх змінних: $n'_1 = n_1$, $n'_2 = n_2 + n_3$, $n'_3 = 0$, $N' = C_{n'_1}^i \cdot (C_{n'_2}^{m-i} \cdot n'_3 + C_{n'_2}^{m-i+1}) = C_{n_1}^i \times \times (C_{n_2+n_3}^{m-i} \cdot 0 + C_{n_2+n_3}^{m-i+1}) = C_{n_1}^i \cdot C_{n_2+n_3}^{m-i+1}$. Отже, модифікація такої функції дозволить блокувати не меншу кількість векторів, ніж якщо $n_3 \neq 0$. ■

Теорема 2.2. Якщо доступна довільна побудова базової $K(m, n)$ моделі, то реберна функція, модифікація якої призводить до блокування максимального для даного сімейства (тобто з тими ж параметрами m і n) моделей числа векторів стану системи, буде знаходитися серед функцій, залежать від всіх змінних.

Під довільною побудовою моделі тут мається на увазі те, що при її побудові можуть бути обрані довільні розбиття множини вхідних змінних. Отже, може бути побудована реберна функція з будь-якими цілими значеннями i , n_1 , n_2 та n_3 , проте, мають виконуватися умови: $1 \leq i < m$, $n_1 + n_2 + n_3 = n$, $n_1 \geq i$, $n_2 \geq m - i$ і $n_3 \geq 0$.

Припустимо, що максимальна кількість блокованих векторів досягається при модифікації деякої функції, для якої $n_3 \geq 1$. Однак, згідно з Лемою 2.5, модифікація функції з такими ж значеннями параметрів i та n_1 , а також $n_3 = 0$, призведе до блокування не меншого числа векторів стану системи.

Відтак, реберна функція, модифікація якої призводить до блокування максимального для даного сімейства моделей числа векторів стану системи буде знаходитися серед функцій, що залежать від всіх змінних. У той же час, можуть існувати реберні функції, які залежать не від усіх змінних і модифікація яких

призводить до блокування такої ж (але не більшої) кількості векторів стану системи. ■

Теорема 2.3. Для фіксованого значення параметра i , при значеннях параметрів $n_1 = \left\lfloor \frac{i \cdot (n+1)}{m+1} \right\rfloor$, $n_2 = n - n_1$, $n_3 = 0$, буде блоковано максимальну кількість векторів стану системи.

Доведення. Згідно з Лемою 2.5, при модифікації функції, що залежить від всіх змінних (тобто для якої $n_3 = 0$), може бути блоковано максимальну для заданих параметрів i та n_1 кількість векторів стану системи.

Розглянемо дві реберні функції, які залежать від всіх вхідних змінних (ці функції в загальному випадку можуть належати двом різним побудовам моделі $K(m, n)$):

$$f_I = \kappa_1(i, n_1) \vee \kappa_2(m - i, n - n_1)$$

$$f_{II} = \kappa_1(i, n_1 - 1) \vee \kappa_2(m - i, n - n_1 + 1)$$

Оцінимо максимальну кількість векторів, блокованих в результаті модифікації кожної з них (за умови модифікації лівої частини виразу функції):

$$N_I = C_{n_1}^i \cdot C_{n-n_1}^{m-i+1} = \frac{n_1! \cdot (n - n_1)!}{i! \cdot (n_1 - i)! \cdot (m - i + 1)! \cdot (n - n_1 - m + i - 1)!}$$

$$N_{II} = C_{n_1-1}^i \cdot C_{n-n_1+1}^{m-i+1} = \frac{(n_1 - 1)! \cdot (n - n_1 + 1)!}{i! \cdot (n_1 - i - 1)! \cdot (m - i + 1)! \cdot (n - n_1 - m + i)!}$$

Виконаємо порівняння цих оцінок:

$$\begin{aligned} \Delta = N_I - N_{II} &= \frac{n_1! \cdot (n - n_1)!}{i! \cdot (n_1 - i)! \cdot (m - i + 1)! \cdot (n - n_1 - m + i - 1)!} - \\ &- \frac{(n_1 - 1)! \cdot (n - n_1 + 1)!}{i! \cdot (n_1 - i - 1)! \cdot (m - i + 1)! \cdot (n - n_1 - m + i)!} = \\ &= \frac{(n_1 - 1)! \cdot n_1 \cdot (n - n_1)! \cdot (n - n_1 - m + i)}{i! \cdot (n_1 - i)! \cdot (m - i + 1)! \cdot (n - n_1 - m + i)!} - \\ &- \frac{(n_1 - 1)! \cdot (n - n_1)! \cdot (n - n_1 + 1) \cdot (n_1 - i)}{i! \cdot (n_1 - i)! \cdot (m - i + 1)! \cdot (n - n_1 - m + i)!} = \\ &= \frac{(n_1 - 1)! \cdot (n - n_1)!}{i! \cdot (n_1 - i)! \cdot (m - i + 1)! \cdot (n - n_1 - m + i)!} \times \end{aligned}$$

$$\begin{aligned} & \times (n_1 \cdot (n - n_1 - m + i) - (n - n_1 + 1) \cdot (n_1 - i)) = \\ & = \frac{(n_1 - 1)! \cdot (n - n_1)!}{i! \cdot (n_1 - i)! \cdot (m - i + 1)! \cdot (n - n_1 - m + i)!} \cdot (i \cdot (n + 1) - n_1 \cdot (m + 1)). \end{aligned}$$

Зауважимо, що вираз $\frac{(n_1-1)! \cdot (n-n_1)!}{i! \cdot (n_1-i)! \cdot (m-i+1)! \cdot (n-n_1-m+i)!}$ (при $1 \leq i < m$, $n_1 \geq i$ та $n_2 = n - n_1 \geq m - i$) завжди буде більшим нуля. Отже, $N_I > N_{II}$, коли $(i \cdot (n + 1) - n_1 \cdot (m + 1)) > 0$, тобто, $n_1 < \frac{i \cdot (n+1)}{m+1}$. З іншого боку, $N_I < N_{II}$, коли $(i \cdot (n + 1) - n_1 \cdot (m + 1)) < 0$, тобто, $n_1 > \frac{i \cdot (n+1)}{m+1}$. Таким чином, за умови фіксації параметрів m , n та i кількість блокованих векторів спочатку монотонно зростатиме зі збільшенням параметру n_1 , після чого по досягненні $n_1 > \frac{i \cdot (n+1)}{m+1}$ почне монотонно спадати. Отже, найбільша кількість блокованих векторів буде досягнута при найбільшому можливому значенні n_1 , за якого $n_1 \leq \frac{i \cdot (n+1)}{m+1}$, тобто, при $n_1 = \left\lfloor \frac{i \cdot (n+1)}{m+1} \right\rfloor$. ■

На Рис. 2.1–2.4 наведено графіки залежності максимальної кількості векторів, блокованих в результаті модифікації відповідно моделей $K(5, 12)$, $K(5, 11)$, $K(6, 16)$ та $K(6, 30)$, від значення параметру n_1 для різних значень i . Як бачимо, найбільшу кількість векторів дійсно вдається блокувати саме при $n_1 = \left\lfloor \frac{i \cdot (n+1)}{m+1} \right\rfloor$.

Зазначимо, що при модифікації моделі $K(m, n)$ згідно із запропонованим методом випадок $i = m$ не є допустимим. Як можна побачити з графіків, максимальну кількість векторів для кожної з моделей може бути блоковано при $i = 1$, та, згідно з Теоремою 2.3, $n_1 = \left\lfloor \frac{n+1}{m+1} \right\rfloor$. При цьому буде блоковано

$$N = C_{n_1}^i \cdot C_{n-n_1}^{m-i+1} = C_{\left\lfloor \frac{n+1}{m+1} \right\rfloor}^1 \cdot C_{n-\left\lfloor \frac{n+1}{m+1} \right\rfloor}^m = \left\lfloor \frac{n+1}{m+1} \right\rfloor \cdot C_{n-\left\lfloor \frac{n+1}{m+1} \right\rfloor}^m \text{ векторів.}$$

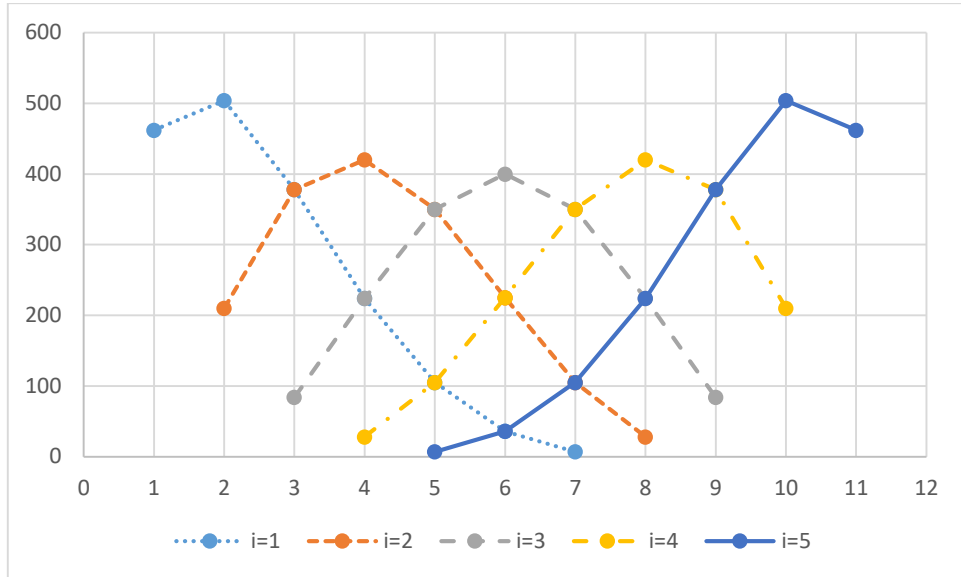


Рис. 2.1. Залежність максимальної кількості векторів, блокованих в результаті модифікації моделі $K(5, 12)$, від значення параметру n_1 для різних значень i .
Оптимальні значення n_1 згідно з Теоремою 2.3: 2 для $i = 1$, 4 для $i = 2$, 6 для $i = 3$, 8 для $i = 4$ та 10 для $i = 5$.

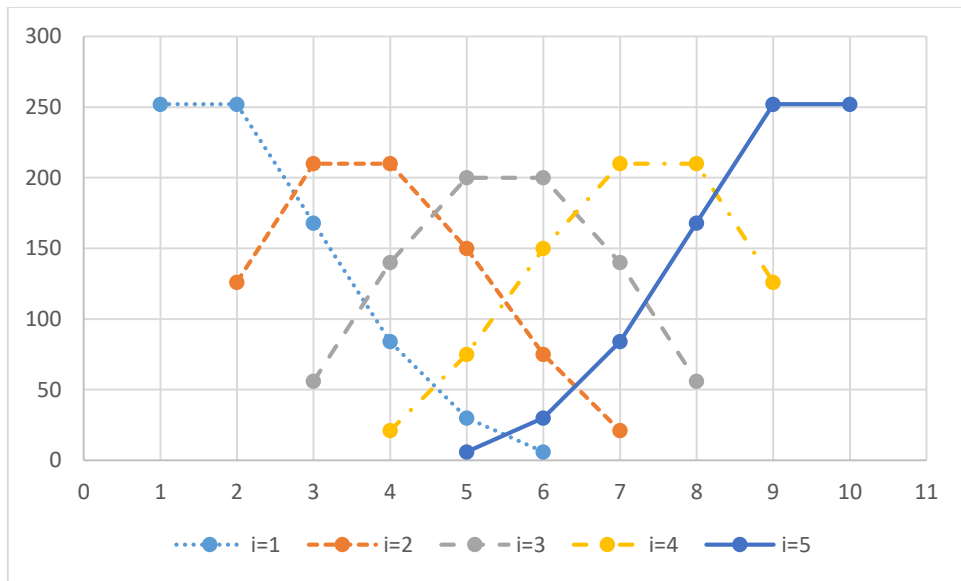


Рис. 2.2. Залежність максимальної кількості векторів, блокованих в результаті модифікації моделі $K(5, 11)$, від значення параметру n_1 для різних значень i .
Оптимальні значення n_1 згідно з Теоремою 2.3: 2 для $i = 1$, 4 для $i = 2$, 6 для $i = 3$, 8 для $i = 4$ та 10 для $i = 5$.

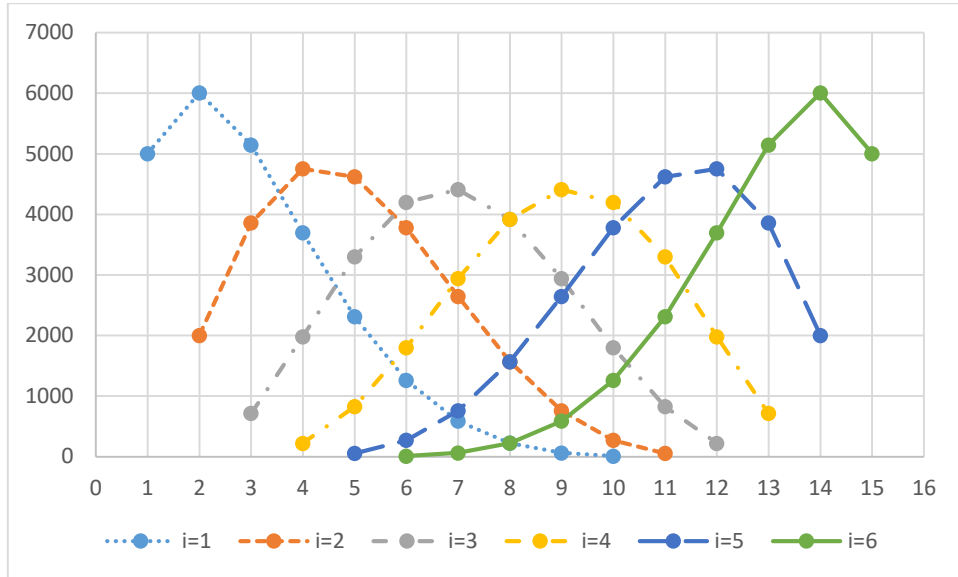


Рис. 2.3. Залежність максимальної кількості векторів, блокованих в результаті модифікації моделі $K(6, 16)$, від значення параметру n_1 для різних значень i .
Оптимальні значення n_1 згідно з Теоремою 2.3: 2 для $i = 1$, 4 для $i = 2$, 7 для $i = 3$, 9 для $i = 4$, 12 для $i = 5$ та 14 для $i = 6$.

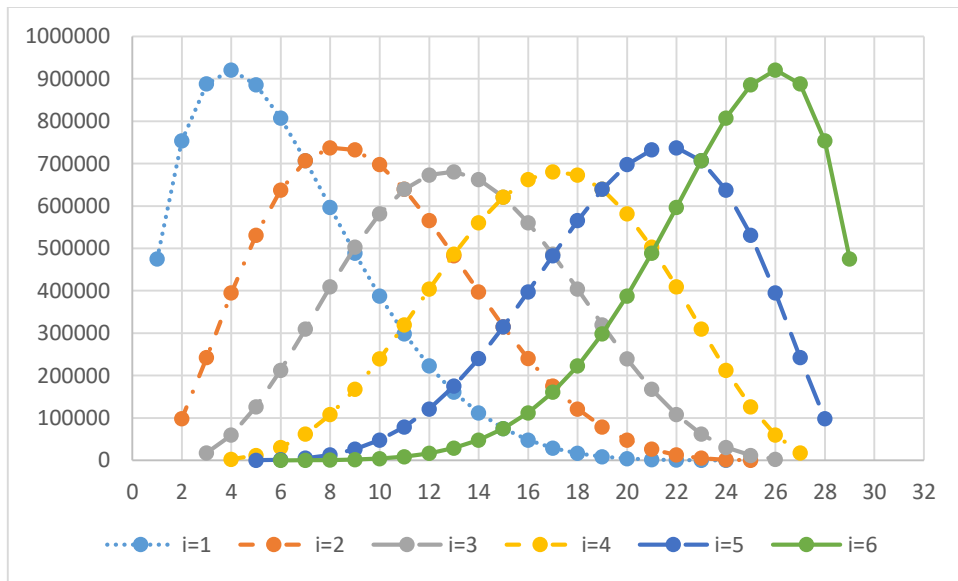


Рис. 2.4. Залежність максимальної кількості векторів, блокованих в результаті модифікації моделі $K(6, 30)$, від значення параметру n_1 для різних значень i .
Оптимальні значення n_1 згідно з Теоремою 2.3: 4 для $i = 1$, 8 для $i = 2$, 13 для $i = 3$, 17 для $i = 4$, 22 для $i = 5$ та 26 для $i = 6$.

2.5. Блокування векторів шляхом послаблення

Розглянемо підхід, що дозволяє модифікувати довільну реберну функцію базової МВР-моделі, так, щоб блокувати шляхом ослаблення деякі з векторів стану системи. На відміну від базової $K(m, n)$ моделі, модифікована модель показуватиме нероботоздатний стан також на деякій множині векторів з m нулями.

Скористаємося позначеннями, представленими в попередніх параграфах. Для модифікації моделі $K(m, n)$ в реберній функції $f_p = \kappa_1(i, n_1) \vee \kappa_2(m - i, n_2)$ деяким чином модифікуємо вираз $\kappa_1(i, n_1)$, отримавши замість нього вираз $\kappa'_1(i, n_1)$, який відрізняється від оригінального тим, що приймає нульове значення також на деякій множині B підвекторів з $l_1 = i - 1$ нулями. Таким чином, в результаті заміни в вихідній моделі функції f_p на іншу функцію $f'_p = \kappa'_1(i, n_1) \vee \kappa_2(m - i, n_2)$, отримаємо модифіковану модель $K'(m, n)$.

Як було показано в параграфі 2.1 поведінка модифікованої моделі відрізнятиметься від поведінки базової моделі зі степінню відмовостійкості m , на основі якої вона була отримана, на деякій множині вхідних векторів, що містять m нулів. На цих векторах граф базової моделі залишатиметься зв'язним, а граф модифікованої моделі втратить зв'язність. Такі вектори, як і в попередніх параграфах, будемо називати блокованими.

Перейдемо до розгляду деяких властивостей отриманої моделі.

Властивість 1

Для запропонованої модифікації множиною блокованих векторів буде об'єднання двох підмножин:

- 1) Всі вектори, у яких v_1 належить множині B , v_2 – будь-який вектор з $m - i$ нулями і v_3 – будь-який вектор з 1 нулем.
- 2) Вектори, у яких v_1 належить множині B , v_2 – будь-який вектор з $m - i + 1$ нулями і v_3 – вектор, що не містить нулів.

Потужність множини блокованих векторів:

$$N = |B| \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}) \quad (2.5)$$

Як вже було сказано, модель $K(m, n)$ може використовуватися в якості службової (проміжної) для побудови однієї з реберних функцій базової моделі, побудованої за [86], яка описує поведінку системи $K_H(M, N)$ в потоці відмов. Тому розглянемо, на яких векторах відрізнятимуться значення виразів $\kappa(m, n)$ та $\kappa'(m, n)$, де $\kappa(m, n)$ – кон'юнкція всіх реберних функцій моделі $K(m, n)$, а $\kappa'(m, n)$ – кон'юнкція всіх реберних функцій моделі $K'(m, n)$.

Властивість 2

Вектори, на яких значення $\kappa(m, n)$ та $\kappa'(m, n)$ відрізнятимуться це вектори, у яких v_1 належить множині B , v_2 – будь-який вектор з $m - i$ нулями і v_3 – вектор, що не містить нулів.

Кількість таких векторів:

$$N = |B| \cdot C_{n_2}^{m-i} \quad (2.6)$$

Перейдемо до доведення справедливості сформульованих положень. Розглянемо всі можливі значення виразів $\kappa_1(i, n_1)$, $\kappa'_1(i, n_1)$ і $\kappa_2(m - i, n_2)$, а також значення функцій f_p' та f_p в залежності від них (Таблиця 2.3). Відзначимо також, що ситуація, коли $\kappa_1(i, n_1) = 0$, а $\kappa'_1(i, n_1) = 1$ в принципі не можлива.

Таблиця 2.3

Значення виразів $\kappa_1(i, n_1)$, $\kappa'_1(i, n_1)$ та $\kappa_2(m - i, n_2)$ і функцій f_p' та f_p

$\kappa_1(i, n_1)$	$\kappa'_1(i, n_1)$	$\kappa_2(m-i, n_2)$	f_p	f_p'
0	0	0	0	0
1	0	0	1	0
1	1	0	1	1
0	0	1	1	1
1	0	1	1	1
1	1	1	1	1

Таблиця показує, що значення функцій f_p і f_p' відрізняються тільки в разі одночасного виконання трьох умов: $\kappa_1(i, n_1) = 1$, $\kappa'_1(i, n_1) = 0$ і $\kappa_2(m - i, n_2) = 0$.

Перші дві умови виконуються тоді і тільки тоді, коли $v_1 \in B$, а третя – коли $l_2 \geq m - i$. Зауважимо, що всі вектори в B містять $i - 1$ нулів, тобто $l_1 = i - 1$.

Проаналізуємо поведінку моделі $K'(m, n)$, отриманої шляхом заміни в моделі $K(m, n)$ реберної функції f_p на f_p' . Розглянемо, для яких векторів зв'язність графів вихідної і модифікованої моделей буде відрізнятися.

Оскільки модифіковано лише одну з реберних функцій, відмінності будуть тільки у випадку, коли вихідна модель втрачає 1 ребро ($f_p = 1$) і граф залишається зв'язним, а модифікована – 2 ребра ($f_p' = 0$) і граф втрачає зв'язність. Протилежна ситуація, коли вихідна і модифікована моделі втрачають відповідно 2 і 1 ребро (тобто $f_p = 0$ і $f_p' = 1$), відповідно до Таблиці 2.3 – неможлива.

Відповідно до (2.2) модель $K(m, n)$ втрачає 1 ребро тільки тоді, коли вхідний вектор містить рівно $l = m$ нулів. При цьому, виходячи з необхідності відмінності значень функцій f_p та f_p' маємо додаткові умови: $v_1 \in B$ ($l_1 = i - 1$) и $l_2 \geq m - i$.

Згадаємо, що $l_1 + l_2 + l_3 = l$. Виходячи з умов: $l = m$, $l_1 = i - 1$, $l_2 \geq m - i$, і враховуючи, що l, l_1, l_2, l_3 – цілі невід'ємні числа, допустимі тільки два варіанти:

- 1) $v_1 \in B$ ($l_1 = i - 1$), $l_2 = m - i$, $l_3 = 1$;
- 2) $v_1 \in B$ ($l_1 = i - 1$), $l_2 = m - i + 1$, $l_3 = 0$.

Сказане стосується векторів, що відповідають *Властивості 1* моделі.

Порахуємо кількість таких векторів:

- 1) Кількість векторів $v_1 \in B$ дорівнює потужності множини B .
- 2) Кількість векторів v_2 з $m - i$ нулями дорівнює $C_{n_2}^{m-i}$.
- 3) Кількість векторів v_2 з $m - i + 1$ нулями дорівнює $C_{n_2}^{m-i+1}$.
- 4) Кількість векторів v_3 з одним нулем дорівнює $C_{n_3}^1 = n_3$.
- 5) Кількість векторів v_3 без нулів дорівнює $C_{n_3}^0 = 1$.

Виконавши нескладні перетворення отримуємо:

$$N = |B| \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}), \text{ що відповідає виразу (2.5).}$$

Визначимо, на яких векторах значення $\kappa(m, n)$ та $\kappa'(m, n)$ відрізнятимуться. Вирази $\kappa(m, n)$ і $\kappa'(m, n)$ приймають значення 0 тоді і тільки тоді, коли відповідно

моделі $K(m, n)$ і $K'(m, n)$ втрачають хоча б одне ребро. Відмінності між значеннями $\kappa(m, n)$ і $\kappa'(m, n)$ будуть тільки у випадку, якщо $K(m, n)$ не втрачає ребер ($f_p = 1$), а $K'(m, n)$ втрачає одне ребро ($f_p' = 0$).

Відповідно до (2.2) модель $K(m, n)$ не втрачає ребер тільки тоді, коли вхідний вектор містить $l < m$ нулів. При цьому, виходячи з необхідності відмінності значень функцій f_p та f_p' маємо додаткові умови: $v_1 \in B$ ($l_1 = i - 1$) і $l_2 \geq m - i$.

Виходячи з умов: $l < m$, $l_1 = i - 1$, $l_2 \geq m - i$, і враховуючи, що l, l_1, l_2, l_3 – цілі невід'ємні числа, допустимий лише один варіант:

$$v_1 \in B \ (l_1 = i - 1), \ l_2 = m - i, \ l_3 = 0.$$

Сказане стосується векторів, що відповідають *Властивості 2* моделі.

В свою чергу, кількість таких векторів: $N = |B| \cdot C_{n_2}^{m-i}$, що відповідає виразу (2.6).

Одним з можливих варіантів модифікації моделі $K_l(i, n_l)$ є її заміна на модель, з меншою на 1 степінню відмовостійкості, тобто $K_l(i - 1, n_l)$. В такому випадку вектор B міститиме всі можливі підвектори v_1 із рівно $i - 1$ нулями.

Відповідно до доведених вище властивостей, вектори, на яких поведінка моделей буде відрізнятися це:

1) Вектори, у яких v_1 – будь-який вектор з $i - 1$ нулями, v_2 – будь-який вектор з $m - i$ нулями і v_3 – будь-який вектор з одним нулем.

2) Вектори, у яких v_1 – будь-який вектор з $i - 1$ нулями, v_2 – будь-який який вектор з $m - i + 1$ нулями і v_3 – вектор, який не містить нулів.

Кількість векторів, на яких поведінка моделей відрізнятиметься:

$$N = C_{n_1}^{i-1} \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}).$$

Відмінності між $\kappa(m, n)$ та $\kappa'(m, n)$ в такому випадку матимуть місце на векторах, у яких v_1 – будь-який вектор з $i - 1$ нулями, v_2 – будь-який вектор з $m - i$ нулями і v_3 – вектор, який не містить нулів.

$$\text{Кількість таких векторів: } N = C_{n_1}^{i-1} \cdot C_{n_2}^{m-i}.$$

Приклад

В моделі $K(3, 11)$, розглянутій в параграфі 2.2 візьмемо, наприклад, функцію f_3 . При побудові вона мала вигляд $f_3 = \kappa_1(1,3) \vee \kappa_2(2,3)$, і при цьому $V_1 = \{x_4, x_5, x_6\}$, $V_2 = \{x_1, x_2, x_3\}$, $V_3 = \{x_7, x_8, x_9, x_{10}, x_{11}\}$. Модифікуємо цю функцію наступним чином: $f'_3 = \kappa_1(1,3) \vee \kappa_2(1,3)$, тобто отримаємо: $f'_3 = x_1 x_2 x_3 \vee x_4 x_5 x_6$.

Експерименти показують, що поведінка моделей відрізняється на наступних 54 векторах: {11011011110, 10111011110, 01111011110, 11010111110, 10110111110, 01110111110, 11001111110, 10101111110, 01101111110, 11011011101, 10111011101, 01111011101, 11010111101, 10110111101, 01110111101, 11001111101, 10101111101, 01101111101, 11011011011, 10111011011, 01111011011, 11010111011, 10110111011, 01110111011, 11001111011, 10101111011, 01101111011, 11011010111, 10111010111, 01111010111, 11010110111, 10110110111, 01110110111, 11001110111, 10101110111, 01101110111, 11011001111, 10111001111, 01111001111, 11010101111, 10110101111, 01110101111, 11001101111, 10101101111, 01101101111, 10011011111, 01011011111, 00111011111, 10010111111, 01010111111, 00110111111, 10001111111, 01001111111, 00101111111}.

Якщо розрахувати кількість блокованих векторів теоретично, керуючись запропонованими вище співвідношеннями (враховуючи, що $i = 2$, $m = 3$, $n_1 = 3$, $n_2 = 3$, $n_3 = 5$), отримаємо:

$$N = C_{n_1}^{i-1} \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}) = C_3^1 \cdot (C_3^1 \cdot 5 + C_3^2) = 3 \cdot (3 \cdot 5 + 3) = 54.$$

Як бачимо приклад підтверджує справедливність доведених вище положень. Порівняємо множину блокованих векторів з передбаченою теоретично:

1) Вектори, у яких підвектор v_1 – будь-який вектор з 1 нулем (тобто {110, 101, 011}), підвектор v_2 – будь-який вектор з 1 нулем (тобто також {110, 101, 011}) і підвектор v_3 – будь-який вектор з 1 нулем (тобто {11110, 11101, 11011, 10111, 01111}).

2) Вектори, у яких підвектор v_1 – будь-який вектор з 1 нулем (тобто $\{110, 101, 011\}$), підвектор v_2 – будь-який вектор з 2 нулями (тобто $\{100, 010, 001\}$) і підвектор v_3 – вектор, який не містить нулів (тобто $\{11111\}$).

Як бачимо, вектори з п.1 відповідають першим 45 з множини отриманих векторів, а вектори з п.2 – останнім 9.

2.6. Граничний випадок блокування векторів шляхом послаблення

Окремо розглянемо ситуацію, коли $i = 1$. У цьому випадку функція, що модифікується має вигляд: $f_p = \kappa_1(1, n_1) \vee \kappa_2(m - 1, n_2)$. Для побудови модифікованої функції, $f'_p = \kappa'_1(1, n_1) \vee \kappa_2(m - 1, n_2)$, відповідно до запропонованого вище методу необхідно було б побудувати модель $K_1(0, n_1)$, проте, це неможливо.

Відповідно до Лема 2.1, вираз $\kappa_1(1, n_1)$ приймає значення 0 на будь-якому векторі, що містить хоча б один нуль. Отже, цей вираз приймає значення 1 тільки в одному випадку: якщо всі n_1 змінних, що входять до множини V_1 мають значення 1. Таким чином, існує лише одне значення вектору v_1 , на якому $\kappa_1(1, n_1) = 1$: це вектор, що складається з n_1 одиниць. Отже, існує єдиний можливий варіант модифікації виразу $\kappa_1(1, n_1)$ запропонованим в параграфі 2.5 методом, при якому множина B складатиметься з одного вектору, всі n_1 елементів якого дорівнюють одиниці. Вираз $\kappa'_1(1, n_1)$ відрізнятиметься від $\kappa_1(1, n_1)$ лише тим, що прийматиме значення, що дорівнює 0 на цьому векторі, і, отже, ніколи не прийматиме значення 1, тобто, $\kappa'_1(1, n_1) \equiv 0$.

Вираз модифікованої функції матиме вигляд: $f'_p = 0 \vee \kappa_2(m - 1, n_2)$, спростивши який отримаємо $f'_p = \kappa_2(m - 1, n_2)$.

З огляду на, що множина B складається з одного вектору, всі n_1 елементів якого дорівнюють одиниці, вектори, на яких поведінка моделей відрізнятиметься це:

1) Вектори, у яких v_1 – вектор, що складається лише з одиниць, v_2 – будь-який вектор з $m - 1$ нулем і v_3 – будь-який вектор з одним нулем.

2) Вектори, у яких v_1 – вектор, що складається тільки з одиниць, v_2 – будь-який вектор з m нулями і v_3 – вектор, який не містить нулів.

Враховуючи те, що $|B| = 1$, кількість векторів, на яких поведінка моделей відрізнятиметься: $N = C_{n_2}^{m-1} \cdot n_3 + C_{n_2}^m$.

Приклад

Побудуємо модель $K(3, 15)$ згідно з [86], вона міститиме наступні реберні функції:

$$f_1 = x_1 \vee x_2 \vee x_3 x_4$$

$$f_2 = x_1 x_2 \vee x_3 \vee x_4$$

$$f_3 = (x_1 \vee x_2)(x_1 x_2 \vee x_3 x_4)(x_3 \vee x_4) \vee x_5 x_6 x_7 x_8$$

$$f_4 = x_1 x_2 x_3 x_4 \vee (x_5 \vee x_6)(x_5 x_6 \vee x_7 x_8)(x_7 \vee x_8)$$

$$f_5 = x_5 \vee x_6 \vee x_7 x_8$$

$$f_6 = x_5 x_6 \vee x_7 \vee x_8$$

$$f_7 = (x_1 \vee x_2)(x_1 x_2 \vee x_3 x_4)(x_3 \vee x_4)(x_1 x_2 x_3 x_4 \vee x_5 x_6 x_7 x_8)(x_5 \vee x_6) \wedge$$

$$\wedge (x_5 x_6 \vee x_7 x_8)(x_7 \vee x_8) \vee x_9 x_{10} x_{11} x_{12} x_{13} x_{14} x_{15}$$

$$f_8 = x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 \vee (x_9 \vee x_{10})(x_9 x_{10} \vee x_{11} x_{12})(x_{11} \vee x_{12}) \wedge$$

$$\wedge (x_9 x_{10} x_{11} x_{12} \vee x_{13} x_{14} x_{15})(x_{13} \vee x_{14})(x_{13} x_{14} \vee x_{15})$$

$$f_9 = x_9 \vee x_{10} \vee x_{11} x_{12}$$

$$f_{10} = x_9 x_{10} \vee x_{11} \vee x_{12}$$

$$f_{11} = (x_9 \vee x_{10})(x_9 x_{10} \vee x_{11} x_{12})(x_{11} \vee x_{12}) \vee x_{13} x_{14} x_{15}$$

$$f_{12} = x_9 x_{10} x_{11} x_{12} \vee (x_{13} \vee x_{14})(x_{13} x_{14} \vee x_{15})$$

$$f_{13} = x_{13} \vee x_{14} \vee x_{15}$$

Модифікуватимемо функцію f_4 . При побудові вона мала вигляд $f_4 = \kappa_1(1,4) \vee \kappa_2(2,4)$, і при цьому $V_1 = \{x_1, x_2, x_3, x_4\}$, $V_2 = \{x_5, x_6, x_7, x_8\}$, $V_3 = \{x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}\}$. В результаті модифікації моделі ми отримаємо нову реберну функцію $f'_4 = \kappa_2(2,4)$, тобто отримаємо $f'_4 = (x_5 \vee x_6)(x_5 x_6 \vee x_7 x_8)(x_7 \vee x_8)$.

Експерименти показують, що поведінка моделей відрізняється на наступних 46 векторах: $\{111111001111110, 111110101111110, 111101101111110,$

111110011111110, 111101011111110, 111100111111110, 111111001111101,
 111110101111101, 111101101111101, 111110011111101, 111101011111101,
 111100111111101, 111111001111011, 111110101111011, 111101101111011,
 111110011111011, 111101011111011, 111100111111011, 111111001110111,
 111110101110111, 111101101110111, 111110011110111, 111101011110111,
 111100111110111, 111111001101111, 111110101101111, 111101101101111,
 111110011101111, 111101011101111, 111100111101111, 111111001011111,
 111110101011111, 111101101011111, 111110011011111, 111101011011111,
 111100111011111, 111111000111111, 111110100111111, 111101100111111,
 111110010111111, 111101010111111, 111100110111111, 111110001111111,
 111101001111111, 111100101111111, 111100011111111}.

Якщо підрахувати кількість векторів теоретично (враховуючи, що $m = 3$, $n_1 = 4$, $n_2 = 4$, $n_3 = 7$), отримаємо:

$$N = C_{n_2}^{m-1} \cdot n_3 + C_{n_2}^m = C_4^2 \cdot 7 + C_4^3 = 6 \cdot 7 + 4 = 46.$$

Порівняємо множину блокованих векторів з передбаченою теоретично:

1) Вектори, у яких підвектор v_1 – вектор, що складається тільки з одиниць (тобто $\{1111\}$), підвектор v_2 – будь-який вектор з 2 нулями (тобто $\{1100, 1010, 0110, 1001, 0101, 0011\}$) і підвектор v_3 – будь-який вектор з 1 нулем (тобто $\{111110, 1111101, 1111011, 1110111, 1101111, 1011111, 0111111\}$).

2) Вектори, у яких підвектор v_1 – вектор, що складається тільки з одиниць (тобто $\{1111\}$), підвектор v_2 – будь-який вектор з 3 нулями (тобто $\{1000, 0100, 0010, 0001\}$) і підвектор v_3 – вектор, який не містить нулів (тобто $\{1111111\}$).

Як бачимо, вектори з п. 1 відповідають першим 42 з множини отриманих векторів, а вектори з п. 2 – останнім 4 з них.

Таким чином, поведінка моделі, отриманої запропонованим вище методом відповідає очікуваній поведінці модифікованої моделі.

2.7. Комбіноване блокування векторів

У попередніх параграфах розглядалися методи модифікації моделі, що дозволяють отримати моделі, що відрізняються від базової $K(m, n)$ тим, що або

показують роботоздатний стан на деякій множині векторів з $m + 1$ нулем, або показують нероботоздатний стан на деякій множині векторів з m нулями. Для узагальнення цього підходу побудуємо модель, яка відрізнятиметься від базової $K(m, n)$ моделі тим, що на деякій множині векторів з $m + 1$ нулем показуватиме роботоздатний стан, і, в той же час, на деякій множині векторів з m нулями показуватиме нероботоздатний стан.

В цьому випадку, для модифікації моделі $K(m, n)$ в реберній функції $f_p = \kappa_1(i, n_1) \vee \kappa_2(m - i, n_2)$ деяким чином модифікуємо вираз $\kappa_1(i, n_1)$, отримавши замість нього вираз $\kappa'_1(i, n_1)$, який відрізняється від оригінального тим, що приймає значення 0 на деякій множині B^0 підвекторів з $l_1 = i - 1$ нулями, а також приймає значення 1 на деякій множині B^1 підвекторів з $l_1 = i$ нулями. В результаті заміни в вихідній моделі функції f_p на іншу функцію $f'_p = \kappa'_{11}(i, n_1) \vee \kappa_2(m - i, n_2)$, отримаємо модифіковану модель $K'(m, n)$.

Як і в параграфах 2.2 і 2.5 розглянемо деякі властивості отриманої моделі. Справедливість цих властивостей легко доводиться за аналогією з доведенням з вищевказаних двох параграфів.

Властивість 1

Для запропонованої модифікації множиною векторів, блокованих шляхом посилення, буде об'єднання двох множин:

1) Усі вектори, у яких v_1 належить множині B^0 , v_2 – будь-який вектор з $m - i$ нулями і v_3 – будь-який вектор з 1 нулем.

2) Вектори, у яких v_1 належить множині B^0 , v_2 – будь-який вектор з $m - i + 1$ нулями і v_3 – вектор, що не містить нулів.

Потужність множини векторів, блокованих шляхом посилення:

$$N_0 = |B^0| \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}) \quad (2.7)$$

Множиною векторів, блокованих шляхом послаблення, буде об'єднання двох множин:

1) Всі вектори, у яких v_1 належить множині B^1 , v_2 – будь-який вектор з $m - i$ нулями і v_3 – будь-який вектор з 1 нулем.

2) Вектори, у яких v_1 належить множині B^1 , v_2 – будь-який вектор з $m - i + 1$ нулями і v_3 – вектор, що не містить нулів.

$$N_1 = |B^1| \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}) \quad (2.8)$$

Очевидно, що множини B^0 і B^1 не перетинаються (оскільки вони містять вектори з різною кількістю нулів), а тому не перетинаються і обидві множини блокованих векторів.

Загальна кількість блокованих векторів:

$$N = N_0 + N_1 = (|B^0| + |B^1|) \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}) \quad (2.9)$$

Для випадку, коли модель $K(m, n)$ використовується в якості проміжної для побудови деякої іншої моделі $K_H(M, N)$, розглянемо, на яких векторах відрізнятимуться значення виразів $\kappa(m, n)$ та $\kappa'(m, n)$, де $\kappa(m, n)$ – кон'юнкція всіх реберних функцій моделі $K(m, n)$, а $\kappa'(m, n)$ – кон'юнкція всіх реберних функцій моделі $K'(m, n)$.

Властивість 2

Вектори, на яких значення $\kappa(m, n)$ та $\kappa'(m, n)$ відрізнятимуться це:

1) Вектори, у яких v_1 належить множині B^0 , v_2 – будь-який вектор з $m - i$ нулями і v_3 – вектор, що не містить нулів. На цих векторах $\kappa(m, n) = 1$, а $\kappa'(m, n) = 0$.

Кількість таких векторів:

$$N_0 = |B^0| \cdot C_{n_2}^{m-i} \quad (2.10)$$

2) Вектори, у яких v_1 належить множині B^1 , v_2 – будь-який вектор з $m - i$ нулями і v_3 – вектор, який не містить нулів. На цих векторах $\kappa(m, n) = 0$, а $\kappa'(m, n) = 1$.

Кількість таких векторів:

$$N_1 = |B^1| \cdot C_{n_2}^{m-i} \quad (2.11)$$

Загальна кількість, на яких значення $\kappa(m, n)$ та $\kappa'(m, n)$ відрізнятимуться:

$$N = N_0 + N_1 = (|B^0| + |B^1|) \cdot C_{n_2}^{m-i} \quad (2.12)$$

2.8. Модифікація обох частин однієї реберної функції

Як було показано в даному розділі, модифікація однієї реберної функції дозволяє будувати моделі, близькі до базових. У попередніх параграфах було розглянуто метод модифікації, що базується на модифікації однієї частини реберної функції, при цьому друга частина залишалася незмінною. Модифікація одночасно обох частин дозволяє досягти більшої гнучкості модифікації моделі.

Для модифікації моделі $K(m, n)$ в реберній функції $f_p = \kappa_1(i, n_1) \vee \kappa_2(m - i, n_2)$ деяким чином модифікуємо вираз $\kappa_1(i, n_1)$, отримавши замість нього вираз $\kappa'_1(i, n_1)$, що відрізняється від оригінального тим, що приймає значення 0 на деякій множині B_1^0 підвекторів з $l_1 = i - 1$ нулями, а також приймає значення 1 на деякій множині B_1^1 підвекторів з $l_1 = i$ нулями. Крім того, модифікуємо також вираз $\kappa_2(m - i, n_2)$, отримавши замість нього вираз $\kappa'_2(m - i, n_2)$, який відрізняється від оригінального тим, що приймає значення 0 на деякій множині B_2^0 підвекторів з $l_2 = m - i - 1$ нулями, а також приймає значення 1 на деякій множині B_2^1 підвекторів з $l_2 = m - i$ нулями. В результаті заміни в вихідній моделі функції f_p на іншу функцію $f'_p = \kappa'_1(i, n_1) \vee \kappa'_2(m - i, n_2)$, отримаємо модифіковану модель $K'(m, n)$.

Розглянемо деякі властивості отриманої моделі.

Властивість 1

Для запропонованої модифікації множиною векторів, блокованих шляхом посилення, буде об'єднання чотирьох множин:

- 1) Вектори, у яких v_1 належить множині B_1^1 , v_2 – будь-який вектор з $m - i$ нулями і v_3 – будь-який вектор з 1 нулем.
- 2) Вектори, у яких v_1 належить множині B_1^1 , v_2 – будь-який вектор з $m - i + 1$ нулями і v_3 – вектор, який не містить нулів.
- 3) Вектори, у яких v_2 належить множині B_2^1 , v_1 – будь-який вектор з i нулями і v_3 – будь-який вектор з 1 нулем.
- 4) Вектори, у яких v_2 належить множині B_2^1 , v_1 – будь-який вектор з $i + 1$ нулями і v_3 – вектор, що не містить нулів.

Потужність множини векторів, блокованих шляхом посилення:

$$N_1 = (|B_1^1| \cdot C_{n_2}^{m-i} + |B_2^1| \cdot C_{n_1}^i - |B_1^1| \cdot |B_2^1|) \cdot n_3 + \\ + |B_1^1| \cdot C_{n_2}^{m-i+1} + |B_2^1| \cdot C_{n_1}^{i+1} \quad (2.13)$$

Множиною векторів, блокованих шляхом послаблення, буде об'єднання п'яти множин:

- 1) Вектори, у яких v_1 належить множині B_1^0 , v_2 – будь-який вектор з $m-i$ нулями, що не належить множині B_2^1 і v_3 – будь-який вектор з 1 нулем.
- 2) Вектори, у яких v_1 належить множині B_1^0 , v_2 – будь-який вектор з $m-i+1$ нулями і v_3 – вектор, який не містить нулів.
- 3) Вектори, у яких v_2 належить множині B_2^0 , v_1 – будь-який вектор з i нулями, що не належить множині B_1^1 і v_3 – будь-який вектор з 1 нулем.
- 4) Вектори, у яких v_2 належить множині B_2^0 , v_1 – будь-який вектор з $i+1$ нулями і v_3 – вектор, що не містить нулів.
- 5) Вектори, у яких v_1 належить множині B_1^0 , v_2 належить множині B_2^0 і v_3 – будь-який вектор з 2 нулями.

Потужність множини векторів, блокованих шляхом послаблення:

$$N_0 = [|B_1^0| \cdot (C_{n_2}^{m-i} - |B_2^1|) + |B_2^0| \cdot (C_{n_1}^i - |B_1^1|)] \cdot n_3 + \\ + |B_1^0| \cdot C_{n_2}^{m-i+1} + |B_2^0| \cdot C_{n_1}^{i+1} + |B_1^0| \cdot |B_2^0| \cdot \frac{n_3(n_3-1)}{2} \quad (2.14)$$

Для випадку, коли модель $K(m, n)$ використовується в якості проміжної для побудови деякої іншої моделі $K_H(M, N)$, розглянемо, на яких векторах будуть відрізнятися значення виразів $\kappa(m, n)$ та $\kappa'(m, n)$, де $\kappa(m, n)$ – кон'юнкція всіх реберних функцій моделі $K(m, n)$, а $\kappa'(m, n)$ – кон'юнкція всіх реберних функцій моделі $K'(m, n)$.

Властивість 2

Множина векторів, на яких $\kappa(m, n) = 0$, а $\kappa'(m, n) = 1$ є об'єднанням двох множин:

- 1) Вектори, у яких v_1 належить множині B_1^1 , v_2 – будь-який вектор з $m-i$ нулями і v_3 – вектор, який не містить нулів.

2) Вектори, у яких v_2 належить множині B_2^1 , v_1 – будь-який вектор з i нулями і v_3 – вектор, який не містить нулів.

Кількість таких векторів:

$$N_1 = |B_1^1| \cdot C_{n_2}^{m-i} + |B_2^1| \cdot C_{n_1}^i - |B_1^1| \cdot |B_2^1| \quad (2.15)$$

Аналогічно, множина векторів, на яких $\kappa(m, n) = 1$, а $\kappa'(m, n) = 0$ є об'єднанням трьох множин:

1) Вектори, у яких v_1 належить множині B_1^0 , v_2 – будь-який вектор з $m-i$ нулями, що не належить множині B_2^1 і v_3 – вектор, який не містить нулів.

2) Вектори, у яких v_2 належить множині B_2^0 , v_1 – будь-який вектор з i нулями, що не належить множині B_1^1 і v_3 – вектор, що не містить нулів.

3) Вектори, у яких v_1 належить множині B_1^0 , v_2 належить множині B_2^0 і v_3 – будь-який вектор з 1 нулем.

Кількість таких векторів:

$$N_0 = |B_1^0| \cdot (C_{n_2}^{m-i} - |B_2^1|) + |B_2^0| \cdot (C_{n_1}^i - |B_1^1|) + |B_1^0| \cdot |B_2^0| \cdot n_3 \quad (2.16)$$

Перейдемо до доведення сформульованих вище властивостей. Розглянемо всі можливі значення виразів $\kappa_1(i, n_1)$, $\kappa_1'(i, n_1)$, $\kappa_2(m-i, n_2)$ та $\kappa_2'(m-i, n_2)$, а також значення функцій f_p і f_p' в залежності від них (Таблиця 2.4).

Також побудуємо таблиці залежностей значень виразів $\kappa_1(i, n_1)$, $\kappa_1'(i, n_1)$ (Таблиця 2.5) і $\kappa_2(m-i, n_2)$, $\kappa_2'(m-i, n_2)$ (таблиця 2.6) від значень змінних з множин V_1 та V_2 відповідно.

Розглянемо, в яких випадках функції будуть приймати неоднакові значення. Як бачимо з Таблиці 2.4, мають місце дві ситуації:

$$1) f_p = 0, \text{ а } f_p' = 1;$$

$$2) f_p = 1, \text{ а } f_p' = 0.$$

Таблиця 2.4

Значення виразів $\kappa_1(i, n_1)$, $\kappa_1'(i, n_1)$, $\kappa_2(m-i, n_2)$ та $\kappa_2'(m-i, n_2)$ і функцій f_p' та f_p

$\kappa_1(i, n_1)$	$\kappa_1'(i, n_1)$	$\kappa_2(m-i, n_2)$	$\kappa_2'(m-i, n_2)$	f_p	f_p'
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	0	1	0
1	1	0	0	1	1
0	0	0	1	0	1
0	1	0	1	0	1
1	0	0	1	1	1
1	1	0	1	1	1
0	0	1	0	1	0
0	1	1	0	1	1
1	0	1	0	1	0
1	1	1	0	1	1
0	0	1	1	1	1
0	1	1	1	1	1
1	0	1	1	1	1
1	1	1	1	1	1

Таблиця 2.5

Значення виразів $\kappa_1(i, n_1)$ та $\kappa_1'(i, n_1)$ в залежності від значень змінних з множини V_1

V_1	$\kappa_1(i, n_1)$	$\kappa_1'(i, n_1)$
$l_1 \geq i, v_1 \notin B_1^1$	0	0
$v_1 \in B_1^1 (l_1 = i)$	0	1
$v_1 \in B_1^0 (l_1 = i - 1)$	1	0
$l_1 < i, v_1 \notin B_1^0$	1	1

Таблиця 2.6

Значення виразів $\kappa_2(m-i, n_2)$ та $\kappa_2'(m-i, n_2)$ в залежності від значень змінних з множини V_2

V_2	$\kappa_2(m-i, n_2)$	$\kappa_2'(m-i, n_2)$
$l_2 \geq m-i, v_2 \notin B_2^1$	0	0
$v_2 \in B_2^1 (l_2 = m-i)$	0	1
$v_2 \in B_2^0 (l_2 = m-i-1)$	1	0
$l_2 < m-i, v_2 \notin B_2^0$	1	1

Оскільки модифікована лише одна з реберних функцій, то відмінності між поведінками моделей будуть тільки у випадках, коли вихідна модель втрачає 2 ребра ($f_p = 0$) і її граф втрачає зв'язність, а модифікована – 1 ребро ($f_p' = 1$) і її граф залишається зв'язним, або коли вихідна модель втрачає 1 ребро ($f_p = 1$) і її граф залишається зв'язним, а модифікована – 2 ребра ($f_p' = 0$) і її граф втрачає зв'язність. Відповідно до (2.2) першій ситуації відповідають вектори, у яких $l = m + 1$, а другій – вектори, у яких $l = m$.

Згадаймо, що $l_1 + l_2 + l_3 = l$, а також, що l, l_1, l_2, l_3 – цілі невід'ємні числа.

Відповідно до Таблиць 2.4-2.6 ситуація, коли $f_p = 0$, а $f_p' = 1$ виникає в наступних випадках:

- 1) $v_1 \in B_1^1 (l_1 = i)$ та $l_2 \geq m-i, v_2 \notin B_2^1$;
- 2) $l_1 \geq i, v_1 \notin B_1^1$ та $v_2 \in B_2^1 (l_2 = m-i)$;
- 3) $v_1 \in B_1^1 (l_1 = i)$ та $v_2 \in B_2^1 (l_2 = m-i)$.

Враховуючи також додаткову умову $l = m + 1$ поведінка моделей відрізнятиметься в наступних випадках:

- 1) $v_1 \in B_1^1 (l_1 = i), l_2 = m-i, v_2 \notin B_2^1, l_3 = 1$;
- 2) $v_1 \in B_1^1 (l_1 = i), l_2 = m-i+1, l_3 = 0$;
- 3) $l_1 = i, v_1 \notin B_1^1, v_2 \in B_2^1 (l_2 = m-i), l_3 = 1$;
- 4) $l_1 = i+1, v_2 \in B_2^1 (l_2 = m-i), l_3 = 0$;
- 5) $v_1 \in B_1^1 (l_1 = i), v_2 \in B_2^1 (l_2 = m-i), l_3 = 1$.

Можемо об'єднати пункти 1 і 5, а також 3 і 5 (зауважимо, що в такому випадку, пункт 5 буде включено два рази), отримаємо:

- 1) $v_1 \in B_1^1$ ($l_1 = i$), $l_2 = m - i$, $l_3 = 1$;
- 2) $v_1 \in B_1^1$ ($l_1 = i$), $l_2 = m - i + 1$, $l_3 = 0$;
- 3) $l_1 = i$, $v_2 \in B_2^1$ ($l_2 = m - i$), $l_3 = 1$;
- 4) $l_1 = i + 1$, $v_2 \in B_2^1$ ($l_2 = m - i$), $l_3 = 0$.

Сказане стосується векторів, блокованих шляхом посилення. Підрахуємо кількість таких векторів і переконаємося, що вона відповідає виразу (2.13):

$$\begin{aligned} N_1 = & [|B_1^1| \cdot (C_{n_2}^{m-i} - |B_2^1|) + |B_2^1| \cdot (C_{n_1}^i - |B_1^1|) + |B_1^1| \cdot |B_2^1|] \cdot n_3 + \\ & + |B_1^1| \cdot C_{n_2}^{m-i+1} + |B_2^1| \cdot C_{n_1}^{i+1} = (|B_1^1| \cdot C_{n_2}^{m-i} + |B_2^1| \cdot C_{n_1}^i - |B_1^1| \cdot |B_2^1|) \cdot n_3 + \\ & + |B_1^1| \cdot C_{n_2}^{m-i+1} + |B_2^1| \cdot C_{n_1}^{i+1}. \end{aligned}$$

Аналогічно, відповідно до Таблиць 2.4-2.6 ситуація, коли $f_p = 1$, а $f_p' = 0$ виникає в наступних випадках:

- 1) $v_1 \in B_1^0$ ($l_1 = i - 1$) та $l_2 \geq m - i$, $v_2 \notin B_2^1$;
- 2) $l_1 \geq i$, $v_1 \notin B_1^1$ та $v_2 \in B_2^0$ ($l_2 = m - i - 1$);
- 3) $v_1 \in B_1^0$ ($l_1 = i - 1$) та $v_2 \in B_2^0$ ($l_2 = m - i - 1$).

Враховуючи також додаткову умову $l = m$ поведінка моделей відрізнятиметься в наступних випадках:

- 1) $v_1 \in B_1^0$ ($l_1 = i - 1$), $l_2 = m - i$, $v_2 \notin B_2^1$, $l_3 = 1$;
- 2) $v_1 \in B_1^0$ ($l_1 = i - 1$), $l_2 = m - i + 1$, $l_3 = 0$;
- 3) $l_1 = i$, $v_1 \notin B_1^1$, $v_2 \in B_2^0$ ($l_2 = m - i - 1$), $l_3 = 1$;
- 4) $l_1 = i + 1$, $v_1 \notin B_1^1$, $v_2 \in B_2^0$ ($l_2 = m - i - 1$), $l_3 = 0$;
- 5) $v_1 \in B_1^0$ ($l_1 = i - 1$), $v_2 \in B_2^0$ ($l_2 = m - i - 1$), $l_3 = 2$.

Сказане стосується векторів, блокованих шляхом послаблення. Підрахуємо кількість таких векторів і переконаємося, що вона відповідає виразу (2.14):

$$\begin{aligned} N_0 = & [|B_1^0| \cdot (C_{n_2}^{m-i} - |B_2^1|) + |B_2^0| \cdot (C_{n_1}^i - |B_1^1|)] \cdot n_3 + |B_1^0| \cdot C_{n_2}^{m-i+1} + \\ & + |B_2^0| \cdot C_{n_1}^{i+1} + |B_1^0| \cdot |B_2^0| \cdot C_{n_3}^2 = [|B_1^0| \cdot (C_{n_2}^{m-i} - |B_2^1|) + |B_2^0| \times \end{aligned}$$

$$\times (C_{n_1}^i - |B_1^1|) \cdot n_3 + |B_1^0| \cdot C_{n_2}^{m-i+1} + |B_2^0| \cdot C_{n_1}^{i+1} + |B_1^0| \cdot |B_2^0| \cdot \frac{n_3(n_3 - 1)}{2}$$

Визначимо, на яких векторах значення $\kappa(m, n)$ і $\kappa'(m, n)$ відрізнятимуться. Вирази $\kappa(m, n)$ та $\kappa'(m, n)$ приймають значення 0 тоді і тільки тоді, коли відповідно моделі $K(m, n)$ та $K'(m, n)$ втрачають хоча б одне ребро. Відмінності між значеннями $\kappa(m, n)$ та $\kappa'(m, n)$ матимуть місце лише у випадку, якщо $K(m, n)$ втрачає одне ребро ($f_p = 0$), а $K'(m, n)$ не втрачає ребер ($f_p' = 1$), або $K(m, n)$ не втрачає ребер ($f_p = 1$), а $K'(m, n)$ втрачає одне ребро ($f_p' = 0$). Відповідно до (2.2) першій ситуації відповідають вектори, у яких $l = m$, а другій – вектори, у яких $l = m - 1$.

Враховуючи вищезазначені умови ситуація, коли $\kappa(m, n) = 0$, а $\kappa'(m, n) = 1$ виникає в наступних випадках:

- 1) $v_1 \in B_1^1$ ($l_1 = i$), $l_2 = m - i$, $v_2 \notin B_2^1$, $l_3 = 0$;
- 2) $l_1 = i$, $v_1 \notin B_1^1$, $v_2 \in B_2^1$ ($l_2 = m - i$), $l_3 = 0$;
- 3) $v_1 \in B_1^1$ ($l_1 = i$), $v_2 \in B_2^1$ ($l_2 = m - i$), $l_3 = 0$.

Можемо об'єднати пункти 1 і 3, а також 2 і 3 (зауважимо, що в такому випадку, пункт 3 буде включений два рази), отримаємо:

- 1) $v_1 \in B_1^1$ ($l_1 = i$), $l_2 = m - i$, $l_3 = 0$;
- 2) $l_1 = i$, $v_2 \in B_2^1$ ($l_2 = m - i$), $l_3 = 0$.

Підрахуємо кількість таких векторів та переконаємося, що вона відповідає виразу (2.15):

$$\begin{aligned} N_1 &= |B_1^1| \cdot (C_{n_2}^{m-i} - |B_2^1|) + |B_2^1| \cdot (C_{n_1}^i - |B_1^1|) + |B_1^1| \cdot |B_2^1| = \\ &= |B_1^1| \cdot C_{n_2}^{m-i} + |B_2^1| \cdot C_{n_1}^i - |B_1^1| \cdot |B_2^1|. \end{aligned}$$

Аналогічно, ситуація, коли $\kappa(m, n) = 1$, а $\kappa'(m, n) = 0$ виникає в наступних випадках:

- 1) $v_1 \in B_1^0$ ($l_1 = i - 1$), $l_2 = m - i$, $v_2 \notin B_2^1$, $l_3 = 0$;
- 2) $l_1 = i$, $v_1 \notin B_1^1$, $v_2 \in B_2^0$ ($l_2 = m - i - 1$), $l_3 = 0$;
- 3) $v_1 \in B_1^0$ ($l_1 = i - 1$), $v_2 \in B_2^0$ ($l_2 = m - i - 1$), $l_3 = 1$.

Підрахуємо кількість таких векторів та переконаємося, що вона відповідає виразу (2.16):

$$N_0 = |B_1^0| \cdot (C_{n_2}^{m-i} - |B_2^1|) + |B_2^0| \cdot (C_{n_1}^i - |B_1^1|) + |B_1^0| \cdot |B_2^0| \cdot n_3.$$

Відзначимо, що методи перетворення моделей, запропоновані в параграфах 2.2, 2.5 та 2.7, які базуються на модифікації тільки однієї частини реберної функції є окремими випадками перетворення, розглянутого в даному параграфі. Дійсно, модифікації, що блокує вектори шляхом посилення (параграф 2.2) відповідає ситуація, коли, $B_1^1 = B$, $B_1^0 = B_2^1 = B_2^0 = \emptyset$, модифікації, що блокує вектори шляхом послаблення (параграф 2.5) відповідає ситуація, коли, $B_1^0 = B$, $B_1^1 = B_2^1 = B_2^0 = \emptyset$, а модифікації, що блокує вектори шляхом і посилення, і послаблення (параграф 2.7) – ситуація, коли, $B_1^1 = B^1$, $B_1^0 = B^0$, $B_2^1 = B_2^0 = \emptyset$.

2.9. Граничні випадки модифікації

В попередніх параграфах вже були розглянуті деякі граничні випадки модифікації ($i = n_1$ для випадку блокування векторів шляхом посилення і $i = 1$ для випадку блокування векторів шляхом послаблення). В загальному випадку подібні граничні випадки існують і при модифікації другої частині виразу реберної функції (тобто $i = m - n_2$ для випадку блокування векторів шляхом посилення і $i = m - 1$ для випадку блокування векторів шляхом ослаблення). Аналіз цих ситуацій – повністю аналогічний викладеному вище.

Існують також інші граничні випадки, при яких множини блокованих векторів будуть дещо відрізнятися від описаних в попередніх розділах. Так, наприклад, коли $n_3 = 0$ (тобто $V_3 = \emptyset$), до опису множини блокованих векторів входитимуть тільки ті підмножини, для яких передбачається, що підвектор v_3 не містить нулів. А у випадках, коли $i = n_1$ і/або $i = m - n_2$ з опису множини блокованих векторів необхідно виключити підмножини, для яких передбачається, що підвектори v_1 та v_2 містять $i + 1$ і $m - i + 1$ нулів відповідно. Очевидно, необхідно вносити відповідну корекцію і в формули для обчислення

кількості блокованих ребер. Зауважимо також, що при поєднанні деяких граничних випадків, модифікація реберної функції може не призвести до блокування векторів стану системи (наприклад, при спробі блокування векторів шляхом посилення у випадку, коли $n_3 = 0$ та $i = m - n_2$).

Розглянемо декілька прикладів.

Приклад 1: $n_3 = 0$.

Побудуємо модель $K(3, 7)$ відповідно до [86], вона міститиме наступні реберні функції:

$$f_1 = x_1 \vee x_2 \vee x_3 x_4$$

$$f_2 = x_1 x_2 \vee x_3 \vee x_4$$

$$f_3 = (x_1 \vee x_2)(x_1 x_2 \vee x_3 x_4)(x_3 \vee x_4) \vee x_5 x_6 x_7$$

$$f_4 = x_1 x_2 x_3 x_4 \vee (x_5 \vee x_6)(x_5 x_6 \vee x_7)$$

$$f_5 = x_5 \vee x_6 \vee x_7$$

Модифікуватимемо функцію f_3 . При побудові вона мала вигляд $f_3 = \kappa_1(2,4) \vee \kappa_2(1,3)$, і при цьому $V_1 = \{x_1, x_2, x_3, x_4\}$, $V_2 = \{x_5, x_6, x_7\}$, $V_3 = \emptyset$. В результаті модифікації моделі ми отримаємо нову реберну функцію $f'_3 = \kappa_1(3,4) \vee \kappa_2(1,3)$, тобто: $f'_3 = (x_1 \vee x_2 \vee x_3 x_4)(x_1 x_2 \vee x_3 \vee x_4) \vee x_5 x_6 x_7$.

Експерименти показують, що поведінка моделей відрізняється на наступних 18 векторах: $\{1100100, 1010100, 0110100, 1001100, 0101100, 0011100, 1100010, 1010010, 0110010, 1001010, 0101010, 0011010, 1100001, 1010001, 0110001, 1001001, 0101001, 0011001\}$.

Підрахувавши кількість блокованих векторів теоретично (враховуючи, що $i = 2$, $m = 3$, $n_1 = 4$, $n_2 = 3$, $n_3 = 0$), отримаємо:

$$N = C_{n_1}^i \cdot \left(C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1} \right) = C_{n_1}^i \cdot C_{n_2}^{m-i+1} = C_4^2 \cdot C_3^2 = 6 \cdot 3 = 18.$$

Порівняємо множину блокованих векторів з передбаченою теоретично (із врахуванням поправок для граничного випадку, коли $n_3 = 0$):

Вектори, у яких підвектор v_1 – вектор, що містить $i = 2$ нулів (тобто $\{1100, 1010, 0110, 1001, 0101, 0011\}$), підвектор v_2 – будь-який вектор з $m - i + 1 = 2$

нулями (тобто $\{100, 010, 001\}$). Як бачимо, дані вектори відповідають множині отриманих векторів.

Приклад 2: $i = m - n_2$.

У моделі $K(3, 7)$, що була побудована в Прикладі 1 модифікуватимемо функцію f_2 . При побудові вона мала вигляд $f_2 = \kappa_1(1,2) \vee \kappa_2(2,2)$, і при цьому $V_1 = \{x_1, x_2\}$, $V_2 = \{x_3, x_4\}$, $V_3 = \{x_5, x_6, x_7\}$. В результаті модифікації моделі отримаємо нову реберну функцію $f'_2 = \kappa_1(2,2) \vee \kappa_2(2,2)$, тобто: $f'_2 = x_1 \vee x_2 \vee x_3 \vee x_4$.

Експерименти показують, що поведінка моделей відрізняється на наступних 6 векторах: $\{1000110, 0100110, 1000101, 0100101, 1000011, 0100011\}$.

Підрахувавши кількість блокованих векторів теоретично (враховуючи, що $i = 1, m = 3, n_1 = 2, n_2 = 2, n_3 = 3$), отримаємо:

$$N = C_{n_1}^i \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}) = C_{n_1}^i \cdot C_{n_2}^{m-i} \cdot n_3 = C_2^1 \cdot C_2^2 \cdot 3 = 2 \cdot 1 \cdot 3 = 6.$$

Порівняємо множину блокованих векторів з передбаченою теоретично (із врахуванням поправок для граничного випадку, коли $i = m - n_2$):

Вектори, у яких підвектор v_1 – вектор, що містить $i = 1$ нулів (тобто $\{10, 01\}$), підвектор v_2 – будь-який вектор з $m - i = 2$ нулями (тобто $\{00\}$) і підвектор v_3 – будь-який вектор з 1 нулем (тобто $\{110, 101, 011\}$). Як бачимо, дані вектори відповідають множині отриманих векторів.

Приклад 3: $n_3 = 0$ и $i = m - n_2$.

Побудуємо модель $K(4, 6)$ відповідно до [86], вона міститиме наступні реберні функції:

$$f_1 = x_1 \vee x_2 \vee x_3 \vee x_4 x_5 x_6$$

$$f_2 = (x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee (x_4 \vee x_5)(x_4 x_5 \vee x_6)$$

$$f_3 = x_1 x_2 x_3 \vee x_4 \vee x_5 \vee x_6$$

Модифікуватимемо функцію f_3 . При побудові вона мала вигляд $f_3 = \kappa_1(1,3) \vee \kappa_2(3,3)$, і при цьому $V_1 = \{x_1, x_2, x_3\}$, $V_2 = \{x_4, x_5, x_6\}$, $V_3 = \emptyset$. В результаті модифікації моделі отримаємо нову

реберну функцію $f'_3 = \kappa_1(2,3) \vee \kappa_2(3,3)$, тобто отримаємо:
 $f'_3 = (x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4 \vee x_5 \vee x_6$.

Експерименти показують, що поведінка моделей не відрізняється на жодному векторі.

Підрахувавши кількість векторів теоретично (враховуючи, що $i = 1$, $m = 4$, $n_1 = 3$, $n_2 = 3$, $n_3 = 0$), отримаємо:

$$N = C_{n_1}^i \cdot (C_{n_2}^{m-i} \cdot n_3 + C_{n_2}^{m-i+1}) = 0.$$

З урахуванням поправок для граничних випадків, коли $n_3 = 0$ та $i = m - n_2$ передбачена теоретично множина блокованих векторів буде порожньою, що співпадає з практично отриманими результатами.

Висновки до розділу 2

В результаті модифікації $K(m, n)$ моделі, її поведінка може відрізнитися від поведінки базової моделі на деякій множині так званих блокованих векторів. Вектори, на яких модифікована модель, на відміну від базової, показує роботоздатний стан системи запропоновано називати блокованими шляхом посилення, а вектори, на яких вона, на відміну від базової показує нероботоздатний стан – блокованими шляхом послаблення.

В результаті модифікації однієї реберної функції базової $K(m, n)$ МВР-моделі можуть бути блоковані шляхом посилення тільки деякі вектори з $m + 1$ нулем, а шляхом послаблення – тільки деякі вектори з m нулями.

Вектори з m нулями, на яких функція, що модифікується приймала значення, рівне нулю, а також вектори з $m + 1$ нулем, на яких вона приймала значення, рівне одиниці, не можуть бути блоковані в результаті цієї модифікації.

У виразі реберної функції базової МВР-моделі можна виділити дві частини. Модифікація функції може бути виконана шляхом зміни як однієї, так і обох її частин, при цьому, в будь-якому випадку, вектори можуть бути блоковані як шляхом посилення, так і шляхом послаблення (причому, можливо блокувати одночасно деякі вектори шляхом посилення, а деякі – шляхом ослаблення).

Модифікована модель може бути використана як допоміжна (проміжна) для побудови іншої моделі (верхнього рівня), тому проаналізовано вплив її модифікації на зміну поведінки моделі верхнього рівня (цей аналіз може бути продовжений на довільну кількість рівнів вглиб).

Для всіх випадків модифікації наведені формули і опис множин блокованих векторів, що дозволяють теоретично (без необхідності проведення експериментів) оцінити результати модифікації і вибрати з множини доступних модифікацій найбільш відповідну.

Розглянуто граничні випадки модифікації, при яких множина блокованих векторів може відрізнитися від звичайної, аж до того, що може бути порожньою. Також в деяких випадках з міркувань оптимізації може бути дещо змінена структура моделі, за рахунок видалення ребра з її графа, який попри це все-одно залишиться циклічним.

РОЗДІЛ 3.

МОДИФІКАЦІЯ ДЕКІЛЬКОХ РЕБЕРНИХ ФУНКЦІЙ МВР-МОДЕЛІ

У попередньому розділі було запропоновано метод модифікації базової моделі, побудованої, на основі циклічного графа (МВР-моделі [86]), шляхом модифікації однієї з її реберних функцій. В результаті цієї модифікації ми отримуємо модель, яка, на відміну від вихідної, показує несправний стан системи на деякій множині векторів з m нулями (вектори, блоковані шляхом послаблення) і/або роботоздатний стан на деякій множині векторів з $m + 1$ нулем (вектори, блоковані шляхом посилення). Також були виділені параметри модифікації, шляхом зміни яких можливо змінювати множину блокованих векторів.

На жаль, шляхом модифікації однієї реберної функції неможливо блокувати довільну множину векторів. Зокрема, як було показано в попередньому розділі, існує обмеження на кількість векторів, блокованих в такий спосіб.

Для вирішення даної проблеми може стати необхідною модифікація одночасно декількох реберних функцій моделі. При цьому передбачається, що блокована буде множина векторів, що дорівнює об'єднанню множин векторів, блокованих в результаті модифікації кожної з функцій окремо. На практиці ж виявляється, що в деяких випадках блокованими виявляються також і деякі з векторів з більш, ніж $m + 1$ нулем (або менше ніж m нулями для випадку блокування векторів шляхом послаблення).

Для побудови коректних моделей важливо розуміти, в яких випадках вищеописаний побічний ефект матиме місце, а в яких ні. Це дозволить або вибрати такий набір модифікацій (реберних функцій), який не призводить до його (побічного ефекту) появи, або вжити заходів для його компенсації. Даній проблемі і присвячено поточний розділ.

Зазначимо, що тут будуть розглянуті модифікації моделі, що призводять до блокування векторів шляхом посилення, зокрема, через припущення, що

такого роду модифікації можуть бути найбільш затребуваними розробниками ОМС. Аналогічний аналіз може бути виконано і для випадків модифікації, що призводять до блокування векторів шляхом послаблення.

3.1. Аналіз загального випадку модифікації

В результаті модифікації, описаної в попередньому розділі деяка реберна функція f_p моделі $K(m, n)$ замінюється на функцію f_p' . При цьому було показано, що f_p' приймає нульове значення на всіх вхідних векторах, на яких f_p теж не дорівнює нулю, а також на деякій додатковій множині векторів, яка залежить від параметрів модифікації.

Нехай модель $K'(m, n)$ була отримана внаслідок заміни деяких M реберних функцій моделі $K(m, n)$. Ці функції $f_{p1}, f_{p2}, \dots, f_{pM}$ були замінені на функції $f_{p1}', f_{p2}', \dots, f_{pM}'$, котрі відрізняються від вихідних тим, що приймають нульове значення на додаткових множинах векторів, які позначимо відповідно L_1, L_2, \dots, L_M .

Нехай $\xi_i(v)$, $i=1,2,\dots,M$ – індикаторна функція, що приймає значення 1, якщо вхідний вектор $v \in L_i$, і 0 – в іншому випадку.

В [95] було доведено, що модель $K(m, n)$ на будь-яких векторах з l нулями втрачає рівно $\varphi(m, l)$ ребер, де:

$$\varphi(m, l) = \begin{cases} 0, & \text{при } l < m \\ l - m + 1, & \text{при } l \geq m \end{cases} \quad (3.1)$$

Позначимо $\Omega(v) \triangleq \sum_{i=1}^M \xi_i(v)$ – кількість функцій модифікованої моделі, значення яких відрізняються від значень відповідних функцій базової моделі на векторі v . Очевидно, що

$$\Omega(v) \leq M \quad (3.2)$$

Тоді модель $K'(m, n)$ на векторі v з l нулями втратить

$$\psi(v) = \varphi(m, l) - \Omega(v) \quad (3.3)$$

ребер, тобто всі ті ребра, які втрачає базова модель, за винятком тих із них, які відповідають функціям, котрі поведуть себе на даному векторі відмінно від відповідних функцій вихідної базової моделі.

Як базова, так і модифікована моделі побудовані на основі циклічного графа, який втрачає зв'язність при втраті будь-яких двох або більше ребер. Блоковані вектори, як було сказано вище, – це такі вектори, на яких базова модель показує нероботоздатний стан (тобто її граф втрачає зв'язність), а модифікована – роботоздатний (тобто її граф залишається зв'язним). Таким чином, вектор v з l нулями буде блокований, якщо $\varphi(m, l) \geq 2$, а $\psi(v) \leq 1$. Відповідно до (3.1) і (3.3) маємо дві умови:

$$\begin{aligned} 1) \quad l &\geq m + 1 \\ 2) \quad \Omega(v) &\geq l - m \end{aligned} \tag{3.4}$$

Таким чином, внаслідок вищеописаної модифікації будуть блоковані такі вектори з $l \geq m + 1$ нулями, на яких щонайменше $l - m$ реберних функцій модифікованої моделі приймають значення, відмінні від значень відповідних функцій базової моделі.

Вектор v з $l = m + 1$ нулем буде блокований, якщо $\Omega(v) \geq 1$, тобто якщо на цьому векторі значення хоч однієї з модифікованих функцій відрізняється від значення відповідної їй вихідної функції. Таким чином, неважко помітити, що множина блокованих векторів з $m + 1$ нулем дійсно буде об'єднанням множин векторів, блокованих внаслідок модифікації кожної з функцій $f_{p1}, f_{p2}, \dots, f_{pM}$ окремо (в попередньому розділі було показано, що при модифікації однієї функції, блокуються виключно вектори з $m + 1$ нулем).

У той же час, якщо існують такі вектори v з $l > m + 1$ нулем, для яких $\Omega(v) \geq l - m$, то вони також будуть блоковані. Однак, відповідно до (3.2) і (3.4) можна побачити, що для цих векторів $l \leq m + M$.

Таким чином, при модифікації одночасно декількох реберних функцій базової моделі можлива поява побічного ефекту який полягає в тому, що крім векторів з $m + 1$ нулем будуть блоковані також і деякі вектори з більшою кількістю нулів (але такою, що не перевищує m на кількість реберних функцій, які були модифіковані).

Для того, щоб даний побічний ефект не з'являвся (якщо це необхідно), відповідно до умов (2.4), достатньо здійснювати модифікацію моделі так, щоб

для будь-якого вектору з $m + t$ нулями, де $2 \leq t \leq M$, від значень відповідних функцій базової моделі відрізнялися значення менш ніж t функцій модифікованої моделі [97], [98]. В протилежному випадку ці вектори також будуть блоковані, що слід мати на увазі при подальшому використанні моделі (наприклад, можливо провести додаткову модифікацію моделі, що компенсуватиме даний побічний ефект, або ж враховувати його наявність при роботі з моделлю).

3.2. Модифікація двох реберних функцій

Забезпечення відсутності вищеописаного побічного ефекту в загальному випадку може виявитися досить складною задачею через необхідність аналізу великої кількості умов (зазвичай їх кількість зростатиме з ростом кількості модифікованих функцій). Тому розробнику ОМС можуть виявитися корисними критерії, що дозволяють визначити такі множини функцій, модифікація яких не призводить до його появи. В силу обмеження обсягу даної роботи, а також виходячи з припущення, що такого роду модифікацій часто буде достатньо, тут будуть сформульовані критерії для випадку модифікації двох реберних функцій МВР моделі, в кожній з яких змінюється лише одна частина. В разі модифікації більшого числа реберних функцій і/або якщо виконуються будь-які інші типи модифікацій реберних функцій, аналіз може бути виконаний за аналогією з міркуваннями з поточного розділу.

Розглянемо модель $K(m, n)$. Відповідно до алгоритму, запропонованого в [86], при її побудові множина вхідних змінних розбивається на 2 підмножини. Далі будуються аналогічні моделі для цих підмножин, для чого, при необхідності, вони також розбиваються. Слід зауважити, що внаслідок розбиття, деякі реберні функції вихідної моделі залежатимуть не від усіх вхідних змінних. У такій моделі реберні функції мають вигляд:

$$f_p = \kappa_1(i, n_1) \vee \kappa_2(m - i, n_2), i \in N, 1 \leq i < m, \quad (3.5)$$

де $\kappa_1(i, n_1)$ – кон'юнкція виразів реберних функцій моделі $K_1(i, n_1)$, побудованої на деякій підмножині вхідних змінних V_1 , потужністю n_1 , а $\kappa_2(m - i, n_2)$ –

кон'юнкція виразів реберних функцій моделі $K_2(m - i, n_2)$, побудованої на деякій підмножині вхідних змінних V_2 , потужністю n_2 і неперетинній з V_1 .

Виберемо деякі дві реберні функції. В загальному випадку вони матимуть вигляд:

$$f_1 = \kappa_{11}(i, n_{11}) \vee \kappa_{12}(m - i, n_{12}), i \in N, 1 \leq i < m, \quad (3.6)$$

та

$$f_2 = \kappa_{21}(j, n_{21}) \vee \kappa_{22}(m - j, n_{22}), j \in N, 1 \leq j < m, \quad (3.7)$$

Множину вхідних змінних V можна розбити на три неперетинних підмножини: V_{11} , V_{12} і V_{13} , де V_{11} та V_{12} відповідають розбиттю вхідної множини для функції f_1 , а V_{13} включає в себе змінні, від яких ця функція не залежить (може бути порожньою). Аналогічно для функції f_2 множина вхідних змінних V може бути розбита на три підмножини V_{21} , V_{22} і V_{23} .

Потужності підмножин позначимо відповідно n_{11} , n_{12} , n_{13} та n_{21} , n_{22} , n_{23} . При цьому

$$n_{11} + n_{12} + n_{13} = n_{21} + n_{22} + n_{23} = n, \quad (3.8)$$

де n – загальна кількість вхідних змінних.

Вираз $\kappa_{11}(i, n_{11})$ відповідає кон'юнкції виразів реберних функцій моделі $K_{11}(i, n_{11})$, побудованої для підмножини V_{11} , $\kappa_{12}(m - i, n_{12})$ – моделі $K_{12}(m - i, n_{12})$, для V_{12} , $\kappa_{21}(j, n_{21})$ – моделі $K_{21}(j, n_{21})$, для V_{21} , а $\kappa_{22}(m - j, n_{22})$ – моделі $K_{22}(m - j, n_{22})$, для V_{22} .

Для будь-якого вхідного вектору v його частини (підвектори), що відповідають підмножинам V_{11} , V_{12} , V_{13} , V_{21} , V_{22} , та V_{23} позначатимемо відповідно $s_{11}(v)$, $s_{12}(v)$, $s_{13}(v)$, $s_{21}(v)$, $s_{22}(v)$ та $s_{23}(v)$. Кількість нулів у векторі v , позначатимемо $l(v)$, а кількість нулів в його частинах, що відповідають множинам V_{11} , V_{12} , V_{13} , V_{21} , V_{22} , и V_{23} – відповідно $l_{11}(v) = l(s_{11}(v))$, $l_{12}(v) = l(s_{12}(v))$, $l_{13}(v) = l(s_{13}(v))$, $l_{21}(v) = l(s_{21}(v))$, $l_{22}(v) = l(s_{22}(v))$ и $l_{23}(v) = l(s_{23}(v))$. Очевидно, що

$$l_{11}(v) + l_{12}(v) + l_{13}(v) = l_{21}(v) + l_{22}(v) + l_{23}(v) = l. \quad (3.9)$$

Кон'юнкція значень реберних функцій моделі приймає значення нуль тоді і тільки тоді, коли хоч одна з них приймає значення, рівне нулю, тобто модель втрачає хоча б одне ребро. Відповідно до Лема 2.1, для вхідного вектору v вирази

$\kappa_{11}(i, n_{11}), \kappa_{12}(m - i, n_{12}), \kappa_{21}(j, n_{21})$ та $\kappa_{22}(m - j, n_{22})$ прийматимуть нульове значення тоді і тільки тоді, коли відповідно $l_{11}(v) \geq i, l_{12}(v) \geq m - i, l_{21}(v) \geq j$ та $l_{22}(v) \geq m - j$.

Нехай модель $K'(m, n)$ отримана з моделі $K(m, n)$ шляхом заміни реберних функцій f_1 на f_1' та f_2 на f_2' . Функції f_1' та f_2' отримані відповідно з f_1 та f_2 способом, описаним в параграфі 2.2, тобто шляхом заміни $\kappa_{11}(i, n_{11})$ на $\kappa_{11}'(i, n_{11})$ та $\kappa_{21}(j, n_{21})$ на $\kappa_{21}'(j, n_{21})$. Ці функції мають вигляд:

$$f_1' = \kappa_{11}'(i, n_{11}) \vee \kappa_{12}(m - i, n_{12}), \quad (3.10)$$

$$f_2' = \kappa_{21}'(j, n_{21}) \vee \kappa_{22}(m - j, n_{22}), \quad (3.11)$$

де $\kappa_{11}'(i, n_{11})$ відрізняється від $\kappa_{11}(i, n_{11})$ тим, що не приймає нульового значення на деякій множині B_1 підвекторів з i нулями, а $\kappa_{21}'(j, n_{21})$ відрізняється від $\kappa_{21}(j, n_{21})$ тим, що не приймає нульового значення на деякій множині B_2 підвекторів з j нулями.

Як було показано в попередньому розділі, значення f_1 відрізнятимуться від значень f_1' тільки на векторах v , таких що $s_{11}(v) \in B_1$ і $l_{12}(v) \geq m - i$, причому $f_1(v) = 0$, а $f_1'(v) = 1$. Аналогічно значення f_2 відрізнятимуться від значень f_2' тільки на векторах v , таких що $s_{21}(v) \in B_2$ і $l_{22}(v) \geq m - j$, причому $f_2(v) = 0$, а $f_2'(v) = 1$.

У Таблиці 3.1 наведено залежність кількості ребер, які втратять моделі $K(m, n)$ та $K'(m, n)$ від кількості нулів у вхідному векторі і значень функцій f_1, f_2, f_1', f_2' . Зауважимо, що, як було сказано в розділі 2, ситуації, коли $f_1 = 1$, а $f_1' = 0$ та $f_2 = 1$, а $f_2' = 0$ виникнути не можуть. Також в таблиці вказано зв'язність графів моделей $K(m, n)$ і $K'(m, n)$.

Для векторів з менш, ніж m нулями модель $K(m, n)$ не втрачає ребер, а тому і f_1 , і f_2 можуть бути рівні тільки 1. Для векторів з m нулями модель $K(m, n)$ втрачає рівно одне ребро, а тому нулю може дорівнювати лише одна з функцій: f_1 , або f_2 . Для векторів з $m + 1$ і більше нулями модель $K(m, n)$ втрачає 2 і більше ребер, отже нулю можуть бути рівні обидві функції: f_1 та f_2 .

В таблиці виділені рядки, що відповідають ситуаціям, коли граф моделі $K(m, n)$ втрачає зв'язність, а граф $K'(m, n)$ залишається зв'язним. При цьому протилежної ситуації, як можна побачити, бути не може.

Таблиця 3.1

Кількість ребер, що втратять моделі $K(m, n)$ та $K'(m, n)$ в залежності від кількості нулів у вхідному векторі і значень функцій f_1, f_2, f_1', f_2'

Кількість нулів у векторі	f_1	f_1'	f_2	f_2'	Кількість ребер, що втрачає $K(m, n)$	Кількість ребер, що втрачає $K'(m, n)$	Зв'язність графу моделі $K(m, n)$	Зв'язність графу моделі $K'(m, n)$
$< m$	1	1	1	1	0	0	+	+
m	1	1	1	1	1	1	+	+
m	0	0	1	1	1	1	+	+
m	0	1	1	1	1	0	+	+
m	1	1	0	0	1	1	+	+
m	1	1	0	1	1	0	+	+
$m+1$	1	1	1	1	2	2	—	—
$m+1$	1	1	0	0	2	2	—	—
$m+1$	1	1	0	1	2	1	—	+
$m+1$	0	0	1	1	2	2	—	—
$m+1$	0	1	1	1	2	1	—	+
$m+1$	0	0	0	0	2	2	—	—
$m+1$	0	0	0	1	2	1	—	+
$m+1$	0	1	0	0	2	1	—	+
$m+1$	0	1	0	1	2	0	—	+
$m+2$	1	1	1	1	3	3	—	—
$m+2$	1	1	0	0	3	3	—	—
$m+2$	1	1	0	1	3	2	—	—
$m+2$	0	0	1	1	3	3	—	—
$m+2$	0	1	1	1	3	2	—	—
$m+2$	0	0	0	0	3	3	—	—
$m+2$	0	0	0	1	3	2	—	—

Кількість нулів у векторі	f_1	f_1'	f_2	f_2'	Кількість ребер, що втрачає $K(m, n)$	Кількість ребер, що втрачає $K'(m, n)$	Зв'язність графу моделі $K(m, n)$	Зв'язність графу моделі $K'(m, n)$
$m+2$	0	1	0	0	3	2	—	—
$m+2$	0	1	0	1	3	1	—	+
$\geq m+3$	1	1	1	1	≥ 4	≥ 4	—	—
$\geq m+3$	1	1	0	0	≥ 4	≥ 4	—	—
$\geq m+3$	1	1	0	1	≥ 4	≥ 3	—	—
$\geq m+3$	0	0	1	1	≥ 4	≥ 4	—	—
$\geq m+3$	0	1	1	1	≥ 4	≥ 3	—	—
$\geq m+3$	0	0	0	0	≥ 4	≥ 4	—	—
$\geq m+3$	0	0	0	1	≥ 4	≥ 3	—	—
$\geq m+3$	0	1	0	0	≥ 4	≥ 3	—	—
$\geq m+3$	0	1	0	1	≥ 4	≥ 2	—	—

Зауважимо, що для векторів з не більш, ніж m нулями поведінка обох моделей співпадає (обидва графа є зв'язними). Для векторів з не менш, ніж $m + 3$ нулями поведінка обох моделей також співпадає (обидва графа втрачають зв'язність).

Розглянемо, які вектори з $m + 1$ нулем будуть блоковані в результаті даної модифікації. Як бачимо з таблиці 2.1, це – вектори, на яких $f_1 = 0$, а $f_1' = 1$ і/або $f_2 = 0$, а $f_2' = 1$. Тобто множина блокованих векторів являтиме собою об'єднання множин векторів, блокованих внаслідок модифікацій тільки функції f_1 і тільки функції f_2 . Ця множина являтиме собою об'єднання чотирьох підмножин (відповідно до результатів з параграфу 2.2):

1) Вектори, у яких $s_{11}(v)$ належить множині B_1 , $s_{12}(v)$ – будь-який вектор з $m - i$ нулями і $s_{13}(v)$ – будь-який вектор з одним нулем.

2) Вектори, у яких $s_{11}(v)$ належить множині B_1 , $s_{12}(v)$ – будь-який вектор з $m - i + 1$ нулями і $s_{13}(v)$ – вектор, що не містить нулів.

3) Вектори, у яких $s_{21}(v)$ належить множині B_2 , $s_{22}(v)$ – будь-який вектор з $m - j$ нулями і $s_{23}(v)$ – будь-який вектор з одним нулем.

4) Вектори, у яких $s_{21}(v)$ належить множині B_2 , $s_{22}(v)$ – будь-який вектор з $m - j + 1$ нулями і $s_{23}(v)$ – вектор, що не містить нулів.

Потужність множини блокованих векторів з $m + 1$ нулем в такому випадку дорівнюватиме

$$N = N_1 + N_2 - N_{12}, \quad (3.12)$$

де N_1 – потужність множини векторів, блокованих внаслідок модифікації функції f_1 , N_2 – потужність множини векторів, блокованих внаслідок модифікації функції f_2 , а N_{12} – потужність перетину вищезазначених множин.

При цьому, як було показано в попередньому розділі:

$$N_1 = |B_1| \cdot (C_{n_{12}}^{m-i} \cdot n_{13} + C_{n_{12}}^{m-i+1}), \quad (3.13)$$

$$N_2 = |B_2| \cdot (C_{n_{22}}^{m-j} \cdot n_{23} + C_{n_{22}}^{m-j+1}). \quad (3.14)$$

Отже, у випадках, коли $N_{12} = 0$, кількість блокованих векторів з $m + 1$ нулем розраховуватиметься за формулою:

$$N = |B_1| \cdot (C_{n_{12}}^{m-i} \cdot n_{13} + C_{n_{12}}^{m-i+1}) + |B_2| \cdot (C_{n_{22}}^{m-j} \cdot n_{23} + C_{n_{22}}^{m-j+1}) \quad (3.15)$$

Окрім векторів з $m + 1$ нулем, в результаті модифікації можуть бути також блоковані вектори з $m + 2$ нулями. У більшості випадків їх наявність є небажаним побічним ефектом. Сформулюємо умови, достатні для того, щоб він не виникав. Крім того, оцінимо, які вектори з $m + 2$ нулями будуть блоковані, а також визначимо, в яких випадках значення N_{12} відрізнятиметься від нуля.

Можна помітити, що відповідно до Таблиці 3.1 будуть блоковані такі вектори з $m + 2$ нулями, на яких $f_1 = 0$, $f_1' = 1$, $f_2 = 0$ і $f_2' = 1$. В свою чергу, значення N_{12} відрізнятиметься від нуля, якщо існують такі вектори з $m + 1$ нулем, які блокуються як внаслідок модифікації f_1 , так і внаслідок модифікації f_2 , тобто на яких також $f_1 = 0$, $f_1' = 1$, $f_2 = 0$ і $f_2' = 1$.

Вектори, на яких $f_1 = 0$, $f_1' = 1$, $f_2 = 0$ і $f_2' = 1$ називатимемо *особливими*. Як було показано в попередньому розділі, $f_1 = 0$ і $f_1' = 1$ на таких векторах v , для яких $s_{11}(v) \in B_1$ і $l_{12}(v) \geq m - i$, а $f_2 = 0$ і $f_2' = 1$ на таких векторах v , для яких $s_{21}(v) \in$

B_2 і $l_{22}(v) \geq m - j$. Сформулюємо умови приналежності вектору v до множини *особливих* векторів:

- 1) $s_{11}(v) \in B_1$,
- 2) $l_{12}(v) \geq m - i$,
- 3) $s_{21}(v) \in B_2$,
- 4) $l_{22}(v) \geq m - j$.

Інтерес становлять лише особливі вектори, що містять $m + 1$ та $m + 2$ нулі: існування перших призводить до ненульового значення N_{12} , а других – до появи вищеописаного побічного ефекту.

Відповідно до побудови, запропонованої в [86], можливі наступні варіанти взаємного відношення між множинами V_{11} , V_{12} , V_{13} та V_{21} , V_{22} , V_{23} :

- 1) $V_{11} = V_{21}$ та $V_{12} = V_{22}$;
- 2) $V_{11} = V_{22}$ та $V_{12} = V_{21}$;
- 3) $V_{11} \subset V_{21}$ та $V_{12} \subset V_{21}$;
- 4) $V_{11} \subset V_{22}$ та $V_{12} \subset V_{22}$;
- 5) $V_{21} \subset V_{11}$ та $V_{22} \subset V_{11}$;
- 6) $V_{21} \subset V_{12}$ та $V_{22} \subset V_{12}$;
- 7) $V_{11} \subset V_{23}$, $V_{12} \subset V_{23}$, $V_{21} \subset V_{13}$ та $V_{22} \subset V_{13}$.

Пункти 1 і 2 відповідають ситуації, коли модифікуються реберні функції побудовані на одному і тому ж розбитті множини вхідних змінних, пункти 3, 4, 5 і 6 – коли одна з них побудована на підрозбитті множини вхідних змінних другої функції, а пункт 7 – коли модифікуються реберні функції побудовані на незалежних підмножинах множини вхідних змінних.

Розглянемо можливість одночасного виконання описаних вище умов для кожної ситуації.

1. $V_{11} = V_{21}$ та $V_{12} = V_{22}$.

В цьому випадку $s_{11}(v) = s_{21}(v)$, $s_{12}(v) = s_{22}(v)$. Позначимо $s_{11}(v)$ та $s_{21}(v)$ як x , а $s_{12}(v)$ та $s_{22}(v)$ як y . Тоді умови приналежності вектору v до множини *особливих* векторів матимуть вигляд:

- 1) $x \in B_1$,
- 2) $l(y) \geq m - i$,
- 3) $x \in B_2$,
- 4) $l(y) \geq m - j$.

Зауважимо, що B_1 та B_2 містять (під)вектори відповідно з i та j нулями. Отже, для виконання умови 1 необхідно, щоб підвектор x містив рівно i нулів, а для виконання умови 3 він повинен містити рівно j нулів. Ця ситуація можлива тільки, якщо $i = j$. Проте в такому випадку f_1 та f_2 – одна й та сама функція (і при цьому модифікована одна і та ж сама її частина), що є неможливим, оскільки за умовою передбачається модифікація двох різних функцій.

Таким чином, в даній ситуації, особливих векторів існувати не буде. Отже, вектори з $m + 2$ нулями не будуть блоковані, а кількість блокованих векторів з $m + 1$ нулем відповідатиме виразу (3.15).

$$2. V_{11} = V_{22} \text{ та } V_{12} = V_{21}.$$

В такому випадку $s_{11}(v) = s_{22}(v)$, $s_{12}(v) = s_{21}(v)$. Позначимо $s_{11}(v)$ та $s_{22}(v)$ як x , а $s_{12}(v)$ та $s_{21}(v)$ як y . Тоді умови приналежності вектору v до множини особливих векторів матимуть вигляд:

- 1) $x \in B_1$,
- 2) $l(y) \geq m - i$,
- 3) $y \in B_2$,
- 4) $l(x) \geq m - j$.

Зауважимо, що B_1 та B_2 містять (під)вектори відповідно з i та j нулями. Для виконання умов 1 і 3 необхідно, щоб x містив рівно i нулів, а y – рівно j нулів. Тобто $l(x) = i$, а $l(y) = j$. Підставивши ці значення в умови 2 і 4, отримаємо: $j \geq m - i$ та $i \geq m - j$. Звідси маємо $i + j \geq m$.

Таким чином, якщо $i + j < m$, особливих векторів не існуватиме, а отже, блоковані будуть лише вектори з $m + 1$ нулем і їх кількість відповідатиме виразу (3.15).

У випадку, якщо $i + j = m$, отримаємо, $j = m - i$, що відповідає тому, що f_1 та f_2 – одна і та ж функція (в якій були модифіковані різні частини), але це неможливо, оскільки за умовою передбачається модифікація двох різних функцій.

В той же час, якщо $i + j > m$ і умови 1 та 3 виконуються, то, відповідно до (3.9) $l(v) = l(s_{11}(v)) + l(s_{12}(v)) + l(s_{13}(v)) \geq l(s_{11}(v)) + l(s_{12}(v)) = i + j$, тобто

$$l(v) \geq i + j \quad (3.16)$$

Отже, вектор v міститиме щонайменше $i + j$ нулів.

Повернувшись до умов 1 і 3 згадаємо, що для *особливого* вектору v буде справедливо $s_{11}(v) \in B_1$ і $s_{12}(v) \in B_2$. Крім того, відповідно до (3.9), $l(s_{13}(v)) = l(v) - l(s_{11}(v)) - l(s_{12}(v))$. Отже,

$$l(s_{13}(v)) = l(v) - l(s_{11}(v)) - l(s_{12}(v)) = l(v) - (i + j). \quad (3.17)$$

Якщо $i + j = m + 1$, *особливими* векторами з $m + 1$ нулем будуть такі вектори v , у яких $s_{11}(v)$ належить множині B_1 , $s_{12}(v)$ належить множині B_2 , а $s_{13}(v)$ не містить нулів (відповідно до (3.17)). Кількість таких векторів:

$$N_{12} = |B_1| \cdot |B_2| \quad (3.18)$$

а кількість блокованих векторів з $m + 1$ нулем, відповідно до (3.12-3.14), дорівнює:

$$\begin{aligned} N = & |B_1| \cdot (C_{n_{12}}^{m-i} \cdot n_{13} + C_{n_{12}}^{m-i+1}) + \\ & + |B_2| \cdot (C_{n_{22}}^{m-j} \cdot n_{23} + C_{n_{22}}^{m-j+1}) - |B_1| \cdot |B_2| \end{aligned} \quad (3.19)$$

В той же час, *особливими* векторами з $m + 2$ нулями будуть такі вектори v , у яких $s_{11}(v)$ належить множині B_1 , $s_{12}(v)$ належить множині B_2 , а $s_{13}(v)$ містить один нуль (відповідно до (3.17)). Кількість таких векторів:

$$N' = |B_1| \cdot |B_2| \cdot n_{13} \quad (3.20)$$

Відзначимо, що у випадку, якщо $n_{13} = n_{23} = 0$, *особливих* векторів з $m + 2$ не існуватиме і, отже, побічний ефект не проявлятиметься.

Якщо $i + j = m + 2$, то, відповідно до (3.16), *особливі* вектори міститимуть щонайменше $m + 2$ нулів. Отже, $N_{12} = 0$ і кількість блокованих векторів з $m + 1$ нулем відповідатиме виразу (3.15).

В той же час, *особливими* векторами з $m + 2$ нулями будуть такі вектори v , у яких $s_{11}(v)$ належить множині B_1 , $s_{12}(v)$ належить множині B_2 , а $s_{13}(v)$ не містить нулів (відповідно до (3.17)). Кількість таких векторів:

$$N' = |B_1| \cdot |B_2| \quad (3.21)$$

В тому ж випадку, коли $i + j > m + 2$, відповідно до (3.16), *особливі* вектори міститимуть більш ніж $m + 2$ нулі, що не призведе ані до блокування векторів з $m + 2$ нулями, ані до ненульового значення N_{12} .

Таким чином, побічний ефект може проявлятися тільки якщо $i + j = m + 1$, або $i + j = m + 2$. У випадках, коли $i + j \leq m$ або $i + j > m + 2$, побічний ефект не проявляється і кількість блокованих векторів розраховується відповідно до (3.15).

$$3. V_{11} \subset V_{21} \text{ та } V_{12} \subset V_{21}.$$

В такому випадку для будь-якого вектору v буде вірно:

$$l(s_{21}(v)) \geq l(s_{11}(v)) + l(s_{12}(v)). \quad (3.22)$$

Розглянемо, в яких випадках виконуватимуться умови приналежності вектору v до множини особливих векторів. Для виконання умови 1 повинно мати місце $s_{11}(v) \in B_1$, отже, $l(s_{11}(v)) = i$, а для виконання умови 2: $l(s_{12}(v)) \geq m - i$. Таким чином, відповідно до (3.22)

$$l(s_{21}(v)) \geq m. \quad (3.23)$$

Для виконання умови 3 повинно мати місце $s_{21}(v) \in B_2$, і, отже,

$$l(s_{21}(v)) = j. \quad (3.24)$$

В такому випадку, відповідно до (3.23) і (2.24) маємо: $j \geq m$. Але, це неможливо, оскільки відповідно до (3.7), $1 \leq j < m$. Отже, виконання умов 1-3 є неможливим, а тому *особливих* векторів не існуватиме, побічний ефект не виникатиме і кількість блокованих векторів розраховуватиметься відповідно до (3.15).

$$4. V_{11} \subset V_{22} \text{ та } V_{12} \subset V_{22}.$$

У такому випадку для будь-якого вектору v буде вірно

$$l(s_{22}(v)) \geq l(s_{11}(v)) + l(s_{12}(v)). \quad (3.25)$$

Розглянемо, в яких випадках виконуватимуться умови приналежності вектору v до множини *особливих* векторів. Для виконання умови 1 повинно мати місце $s_{11}(v) \in B_1$, з чого випливає, що $l(s_{11}(v)) = i$, а для виконання умови 2: $l(s_{12}(v)) \geq m - i$. Отже, відповідно до (2.25)

$$l(s_{22}(v)) \geq m. \quad (3.26)$$

Умова 4 виконуватиметься автоматично. Для виконання умови 3 повинно мати місце $s_{21}(v) \in B_2$, отже $l(s_{21}(v)) = j$.

Відповідно до (3.9) буде справедливо:

$$\begin{aligned} l(v) &= l(s_{21}(v)) + l(s_{22}(v)) + l(s_{23}(v)) \geq \\ &\geq l(s_{21}(v)) + l(s_{22}(v)) = m + j. \end{aligned} \quad (3.27)$$

Отже, вектор v міститиме щонайменше $m + j$ нулів.

Нехай $s_R(v)$ – підвектор вектору v , що відповідає підмножині множини вхідних змінних $V_R = V \setminus (V_{11} \cup V_{12} \cup V_{21})$. Довжина цього вектору

$$n_R = n - (n_{11} + n_{12} + n_{21}), \quad (3.28)$$

а кількість нулів в ньому:

$$l_R(v) = l(s_R(v)) = l(v) - (l_{11}(v) + l_{12}(v) + l_{21}(v)). \quad (3.29)$$

Враховуючи, що $V_{13} = V \setminus (V_{11} \cup V_{12})$ маємо $V_R = V_{13} \setminus V_{21}$, а тому:

$$n_R = n_{13} - n_{21}. \quad (3.30)$$

Якщо $j = 1$, то для *особливого* вектору v з $m + 1$ нулем відповідно до умов 1-3 і внаслідок того, що $l(v) = m + 1$, а $l(s_{12}(v))$ та $l_R(v)$ – цілі невід'ємні числа, допустимий лише один варіант: $s_{11}(v) \in B_1$, $l(s_{12}(v)) = m - i$, $s_{21}(v) \in B_2$, $l_R(v) = 0$. Отже, *особливими* векторами з $m + 1$ нулем будуть такі вектори v , у яких $s_{11}(v)$ належить множині B_1 , $s_{21}(v)$ належить множині B_2 , $s_{12}(v)$ – будь-який вектор з $m - i$ нулями, а решта елементів вектору v нулів не містять. Кількість таких векторів:

$$N_{12} = |B_1| \cdot |B_2| \cdot C_{n_{12}}^{m-i}, \quad (3.31)$$

а кількість блокованих векторів з $m + 1$ нулем, відповідно до (3.12-3.14), дорівнює:

$$\begin{aligned}
N = & |B_1| \cdot (C_{n_{12}}^{m-i} \cdot n_{13} + C_{n_{12}}^{m-i+1}) + \\
& + |B_2| \cdot (C_{n_{22}}^{m-j} \cdot n_{23} + C_{n_{22}}^{m-j+1}) - |B_1| \cdot |B_2| \cdot C_{n_{12}}^{m-i}.
\end{aligned}
\tag{3.32}$$

В той же час, для *особливого* вектору v з $m+2$ нулями відповідно до умов 1-3 і внаслідок того, що $l(v) = m+2$, а $l(s_{12}(v))$ і $l_R(v)$ – цілі невід'ємні числа, допустимі два варіанти:

- 1) $s_{11}(v) \in B_1$, $l(s_{12}(v)) = m-i+1$, $s_{21}(v) \in B_2$, $l_R(v) = 0$;
- 2) $s_{11}(v) \in B_1$, $l(s_{12}(v)) = m-i$, $s_{21}(v) \in B_2$, $l_R(v) = 1$.

Отже, *особливими* векторами з $m+2$ нулями будуть такі вектори v , у яких:

- 1) $s_{11}(v)$ належить множині B_1 , $s_{21}(v)$ належить множині B_2 , $s_{12}(v)$ – будь-який вектор з $m-i+1$ нулями, а решта елементів вектору v нулів не містять.
- 2) $s_{11}(v)$ належить множині B_1 , $s_{21}(v)$ належить множині B_2 , $s_{12}(v)$ – будь-який вектор з $m-i$ нулями, а решта елементів вектору v містять рівно один нуль.

Кількість таких векторів:

$$\begin{aligned}
N' = & |B_1| \cdot |B_2| \cdot (C_{n_{12}}^{m-i+1} + C_{n_{12}}^{m-i} \cdot n_R) = \\
= & |B_1| \cdot |B_2| \cdot (C_{n_{12}}^{m-i+1} + C_{n_{12}}^{m-i} \cdot (n_{13} - n_{21})).
\end{aligned}
\tag{3.33}$$

Зауважимо, що якщо $m-i = n_{12}$, а $V_{21} = V_{13}$, *особливих* векторів з $m+2$ нулями не існуватиме і, отже, побічний ефект не проявлятиметься. Вираз (3.32) для кількості блокованих векторів з $m+1$ нулем в цьому випадку прийме вигляд (3.19).

Якщо $j=2$, то відповідно до (3.27) *особливі* вектори міститимуть щонайменше $m+2$ нулів. Отже, $N_{12} = 0$ і кількість блокованих векторів з $m+1$ нулем відповідатиме виразу (3.15).

В той же час, для *особливого* вектору v з $m+2$ нулями відповідно до умов 1-3 і внаслідок того, що $l(v) = m+2$, а $l(s_{12}(v))$ і $l_R(v)$ – цілі невід'ємні числа, допустимий лише один варіант: $s_{11}(v) \in B_1$, $l(s_{12}(v)) = m-i$, $s_{21}(v) \in B_2$, $l_R(v) = 0$. Отже, *особливими* векторами з $m+2$ нулями будуть такі вектори v , у яких $s_{11}(v)$ належить множині B_1 , $s_{21}(v)$ належить множині B_2 ,

$s_{12}(v)$ – будь-який вектор з $m - i$ нулями, а решта елементів вектору v нулів не містять. Кількість таких векторів:

$$N' = |B_1| \cdot |B_2| \cdot C_{n_{12}}^{m-i}. \quad (3.34)$$

Якщо ж $j > 2$, то відповідно до (3.27) *особливі* вектори міститимуть більш ніж $m + 2$ нулі, що не призведе ані до блокування векторів з $m + 2$ нулями, ані до ненульового значення N_{12} і, отже, кількість блокованих векторів розраховуватиметься відповідно до (3.15).

5. $V_{21} \subset V_{11}$ та $V_{22} \subset V_{11}$.

Доведення є аналогічним до пункту 3. Побічний ефект не виникає, $N_{12} = 0$ і кількість блокованих векторів розраховується відповідно до (3.15).

6. $V_{21} \subset V_{12}$ та $V_{22} \subset V_{12}$.

Доведення є аналогічним до пункту 4.

При $i = 1$ *особливими* векторами з $m + 1$ нулем будуть такі вектори v , у яких $s_{11}(v)$ належить множині B_1 , $s_{21}(v)$ належить множині B_2 , $s_{22}(v)$ – будь-який вектор з $m - j$ нулями, а решта елементів вектору v нулів не містять. Кількість таких векторів:

$$N_{12} = |B_1| \cdot |B_2| \cdot C_{n_{22}}^{m-j}, \quad (3.35)$$

а кількість блокованих векторів з $m + 1$ нулем, відповідно до (3.12-3.14), дорівнює:

$$\begin{aligned} N = & |B_1| \cdot (C_{n_{12}}^{m-i} \cdot n_{13} + C_{n_{12}}^{m-i+1}) + \\ & + |B_2| \cdot (C_{n_{22}}^{m-j} \cdot n_{23} + C_{n_{22}}^{m-j+1}) - |B_1| \cdot |B_2| \cdot C_{n_{22}}^{m-j}. \end{aligned} \quad (3.36)$$

Також будуть блоковані вектори з $m + 2$ нулями, у яких:

- 1) $s_{11}(v)$ належить множині B_1 , $s_{21}(v)$ належить множині B_2 , $s_{22}(v)$ – будь-який вектор з $m - j + 1$ нулями, а решта елементів вектору v нулів не містять.
- 2) $s_{11}(v)$ належить множині B_1 , $s_{21}(v)$ належить множині B_2 , $s_{22}(v)$ – будь-який вектор з $m - j$ нулями, а решта елементів вектору v містять рівно один нуль.

Кількість таких векторів:

$$N' = |B_1| \cdot |B_2| \cdot \left(C_{n_{22}}^{m-j+1} + C_{n_{22}}^{m-j} \cdot (n_{23} - n_{11}) \right). \quad (3.37)$$

При цьому, якщо, $m - j = n_{22}$, а $V_{11} = V_{23}$ *особливих* векторів з $m + 2$ нулями не існуватиме і, отже, побічний ефект не проявлятиметься. Вираз (3.36) для кількості блокованих векторів з $m + 1$ в цьому випадку прийме вигляд (3.19).

При $i = 2$, значення N_{12} дорівнюватиме нулю і кількість блокованих векторів з $m + 1$ нулем відповідатиме виразу (3.15). В той же час будуть блоковані вектори з $m + 2$ нулями, у яких $s_{11}(v)$ належить множині B_1 , $s_{21}(v)$ належить множині B_2 , $s_{22}(v)$ – будь-який вектор з $m - j$ нулями, а решта елементів вектору v нулів не містять. Кількість таких векторів:

$$N' = |B_1| \cdot |B_2| \cdot C_{n_{22}}^{m-j}, \quad (3.38)$$

При $i > 2$ побічний ефект не виникатиме, N_{12} дорівнюватиме нулю і кількість блокованих векторів розраховуватиметься відповідно до (3.15).

$$7. V_{11} \subset V_{23}, V_{12} \subset V_{23}, V_{21} \subset V_{13} \text{ та } V_{22} \subset V_{13}.$$

В такому випадку для будь-якого вектору v буде вірно

$$l(s_{13}(v)) \geq l(s_{21}(v)) + l(s_{22}(v)). \quad (3.39)$$

Розглянемо, в яких випадках виконуватимуться умови приналежності вектору v до множини *особливих* векторів. Для виконання умови 3 повинно мати місце $s_{21}(v) \in B_2$, отже $l(s_{21}(v)) = j$, а для виконання умови 4: $l(s_{22}(v)) \geq m - j$. Отже, відповідно до (3.39), $l(s_{13}(v)) \geq m$. З іншого боку, для виконання умови 1 повинно мати місце $s_{11}(v) \in B_1$, отже $l(s_{11}(v)) = i$, а для виконання умови 2: $l(s_{12}(v)) \geq m - i$.

Відповідно до (3.9) $l(v) = l(s_{11}(v)) + l(s_{12}(v)) + l(s_{13}(v))$, отже

$$l(v) \geq 2m. \quad (3.40)$$

Таким чином, вектор v міститиме щонайменше $2m$ нулів.

У випадку $m = 1$ ми маємо справу з моделлю $K(1, n)$. Відповідно до [86] вона міститиме n реберних функцій виду $f_i = x_i$. Єдиним варіантом модифікації такої функції (яка призводить до блокування векторів шляхом посилення) є заміна її на функцію $f_i' = 1$. В свою чергу, ребро, відповідне цій функції завжди буде присутнім в графі моделі і, отже, не зможе привести до втрати його

зв'язності, а тому може бути просто видалене. Інцидентні йому вершини при цьому будуть об'єднані.

Після видаленні ребра, що відповідає модифікації функції f_i , поведінка нової моделі не залежатиме від значення i -го елементу вхідного вектору. Модель показуватиме роботоздатний стан системи на всіх векторах, що містять не більше, ніж один нуль серед всіх елементів, без урахування i -го. Таким чином, блоковані будуть такі вектори з $m + 1 = 2$ нулями, у яких один нуль знаходиться на i -й, а другий – на будь-який інший позиції. Зауважимо, що кількість таких позицій $n - 1$, де n – кількість елементів вхідного вектору. Отже, кількість блокованих векторів:

$$N_1 = C_{n-1}^1 = n - 1 \quad (3.41)$$

Аналогічно, після видаленні ребра, що відповідає модифікації функції $f_j = x_j$, поведінка нової моделі не залежатиме від значення j -го елементу вхідного вектору, і блоковані будуть такі вектори з $m + 1 = 2$ нулями, у яких один нуль знаходиться на j -й, а другий – на будь-який інший позиції. Кількість таких векторів:

$$N_2 = C_{n-1}^1 = n - 1 \quad (3.42)$$

При модифікації двох функцій $f_i = x_i$ та $f_j = x_j$ видалені будуть вже два ребра, що призведе до того, що поведінка нової моделі не залежатиме від значень i -го та j -го елементів вхідного вектору. Таким чином, будуть блоковані такі вектори з $m + 1 = 2$ нулями, у яких один нуль знаходиться на i -й або j -й, а другий – на будь-який інший позиції, причому, один з них, з нулями і на i -й, на j -й позиції, буде *особливим*, на ньому модель не втратить жодного ребра. Зауважимо, що цей вектор входить як до множини векторів, блокованих в результаті модифікації функції f_i , так і до множини векторів, блокованих в результаті модифікації функції f_j . Таким чином, $N_{12} = 1$ і, відповідно до (3.12, 3.41 та 3.42) кількість блокованих векторів з $m + 1 = 2$ нулями:

$$N = N_1 + N_2 - N_{12} = (n - 1) + (n - 1) - 1 = 2n - 3, \quad (3.43)$$

де N_1 та N_2 – кількість векторів, блокованих в результаті модифікації відповідно тільки i -ї і тільки j -ї функції, а N_{12} – кількість *особливих* векторів з $m + 1 = 2$ нулями.

Крім того, також будуть блоковані вектори з $m + 2 = 3$ нулями, що містять по одному нулю на i -й і j -й позиції, а також один нуль на будь-який інший позиції. Зауважимо, що кількість таких позицій $n - 2$, де n – кількість елементів вхідного вектору. Отже, кількість особливих векторів з $m + 2 = 3$ нулями:

$$N' = C_{n-2}^1 = n - 2. \quad (3.44)$$

При $m = 2$, відповідно до (3.40) *особливі* вектори міститимуть щонайменше $2m = m + 2$ нуля. Отже, $N_{12} = 0$ і кількість блокованих векторів з $m + 1$ нулем відповідатиме виразу (2.15).

Згадаємо, що відповідно до (3.6-3.7), $1 \leq i < m$ і $1 \leq j < m$. Отже, для $m = 2$ існує єдиний доступний варіант: $i = j = 1$. Таким чином, відповідно до умов 1-4 *особливими* будуть такі вектори v з $m + 2 = 4$ нулями, у яких: $s_{11}(v) \in B_1$, $s_{12}(v)$ містить $m - i = 1$ нуль, $s_{21}(v) \in B_2$, $s_{22}(v)$ містить $m - j = 1$ нуль, а всі інші частини вектору (якщо вони є) нулів не містять.

Кількість таких векторів:

$$N' = |B_1| \cdot |B_2| \cdot n_{12} \cdot n_{22}. \quad (3.45)$$

Якщо ж $m > 2$, відповідно до (3.40) *особливі* вектори міститимуть більш ніж $m + 2$ нулі, що не призведе ані до блокування векторів з $m + 2$ нулями, ані до ненульового значення N_{12} і, отже, кількість блокованих векторів розраховуватиметься відповідно до (3.15).

Таким чином, побічний ефект, що полягає в блокуванні векторів з $m + 2$ нулями не виникатиме в наступних випадках [99]:

- 1) $V_{11} = V_{21}$ та $V_{12} = V_{22}$;
- 2) $V_{11} = V_{22}$, $V_{12} = V_{21}$ та $i + j \leq m$, або $i + j > m + 2$;
- 3) $V_{11} \subset V_{21}$ та $V_{12} \subset V_{21}$;
- 4) $V_{11} \subset V_{22}$, $V_{12} \subset V_{22}$ та $j > 2$;
- 5) $V_{21} \subset V_{11}$ та $V_{22} \subset V_{11}$;

6) $V_{21} \subset V_{12}, V_{22} \subset V_{12}$ та $i > 2$;

7) $V_{11} \subset V_{23}, V_{12} \subset V_{23}, V_{21} \subset V_{13}, V_{22} \subset V_{13}$ та $m > 2$.

При цьому потужність множини блокованих векторів дорівнюватиме:

$$N = |B_1| \cdot (C_{n_{12}}^{m-i} \cdot n_{13} + C_{n_{12}}^{m-i+1}) + |B_2| \cdot (C_{n_{22}}^{m-j} \cdot n_{23} + C_{n_{22}}^{m-j+1})$$

Побічний ефект не виникатиме також в наступних випадках:

1) $V_{11} = V_{22}, V_{12} = V_{21}, i + j = m + 1$ та $n_{13} = n_{23} = 0$;

2) $V_{11} \subset V_{22}, V_{12} \subset V_{22}, j = 1, V_{21} = V_{13}$ та $m - i = n_{12}$;

3) $V_{21} \subset V_{12}, V_{22} \subset V_{12}, i = 1, V_{11} = V_{23}$ та $m - j = n_{22}$.

В цих випадках потужність множини блокованих векторів дорівнюватиме:

$$N = N_1 + N_2 - N_{12} = \\ = |B_1| \cdot (C_{n_{12}}^{m-i} \cdot n_{13} + C_{n_{12}}^{m-i+1}) + |B_2| \cdot (C_{n_{22}}^{m-j} \cdot n_{23} + C_{n_{22}}^{m-j+1}) - |B_1| \cdot |B_2|.$$

В Додатку А наведено приклад, що наочно показує результат впливу побічного ефекту для різних комбінацій модифікованих функцій.

Висновки до розділу 3

Внаслідок наявності ряду обмежень для множини векторів, блокованих в результаті модифікації однієї реберної функції базової GL-моделі, такої зміни може бути недостатньо для забезпечення бажаного її перетворення. В таких випадках корисною є можливість одночасної модифікації декількох функцій. Однак, іноді це призводить до появи побічного ефекту: крім векторів з $m + 1$ нулем також блокуються вектори з більшою кількістю нулів. Користувачу моделі доводиться або вживати заходів для його компенсації, або виконувати модифікацію так, щоб він не з'являвся. В зв'язку з цим, корисними будуть критерії, що дозволяють вибирати такі функції, модифікація яких не призводить до появи побічного ефекту.

У розділі було показано, що в результаті модифікації M реберних функцій, блокованими можуть бути тільки вектори з не більш ніж $m + M$ нулями. Також сформульовано умову, достатню для того, щоб вищеописаний побічний ефект не проявлявся.

Сформульована умова в загальному випадку вимагає глибокого аналізу використовуваних модифікацій. Тому для випадку модифікації двох реберних функцій способом, описаним в попередньому розділі, сформульовані прості критерії, що дозволяють уникнути прояву побічного ефекту шляхом вибору допустимих пар функцій, що модифікуються. Для кожної з можливих пар функцій, що модифікуються, представлені формули, котрі дозволяють розрахувати кількість блокованих векторів з $m + 1$ нулем, а для тих з них, модифікація яких призводить до прояву побічного ефекту – і з $m + 2$ нулями.

РОЗДІЛ 4.

ВСТАНОВЛЕННЯ СТАНІВ ПРОЦЕСОРІВ ВБС В ПОТОЦІ ВІДМОВ ЗА РЕЗУЛЬТАТАМИ ЇХ ВЗАЄМНИХ ТЕСТОВИХ ПЕРЕВІРОК

Важливою задачею при розробці ВБС є організація процесу її самотестування з метою визначення стану в потоці відмов (роботоздатний/відмовив) кожного з її процесорів. Звичайно, можливо організувати відмовостійкість системи і без використання інформації про поточний стан кожного з її процесорів [100], [101]. В той же час, інформація про поточний стан кожного з процесорів, або хоча б деяких із них, може дозволити більш гнучко і ефективно використовувати наявні резерви та виконувати реконфігурацію системи задля забезпечення продовження її функціонування [102], [103].

Процесори можуть тестуватися шляхом виконання на них деякого набору тестів і порівняння їх результатів з еталоном. Якщо для деякого процесора результати не співпали, то він вважається несправним. Тестування процесорів може проводитися як окремим спеціалізованим вузлом, так і в результаті їх взаємного тестування, при якому тестуючий процесор визначає стан тестованого процесору. Відповідно до моделі, запропонованої в [104], яка називається моделлю Препарати-Метце-Чена (ПМЧ) результатом тестування процесора x справним процесором y буде 1, якщо в результаті тестування були виявлені помилки або 0, якщо помилок виявлено не було. Якщо ж тестуючий процесор y в момент тестування був несправний, то в результаті може бути отримано як значення 0, так і значення 1 незалежно від стану тестованого процесора x . Як і в більшості робіт тут припускається, що канали зв'язку між процесорами є справними, оскільки ймовірність виходу їх з ладу вважається відносно низькою.

Топологія реальних систем далеко не завжди дозволяє організувати тестування кожного процесору будь-яким іншим процесором. Ці обмеження можуть бути викликані, зокрема, відсутністю безпосередніх зв'язків між процесорами, обмеженням обсягу пам'яті для зберігання тестових програм і

еталонних значень результатів тестів (якщо в системі знаходяться процесори різних типів) і т.д. Можливість тестування процесорів одне одним можна представити у вигляді орієнтованого графа G , кожна з вершин якого відповідає деякому процесору, а наявність дуги з деякої вершини a в деяку вершину b означає можливість тестування процесора, що відповідає вершині b процесором, відповідним вершині a . Такий граф фактично відображає можливості системи в сенсі тестування.

Для будь-якої ВБС має місце t -діагностованість [105]. Система є t -діагностованою, якщо, знаючи результати взаємного тестування процесорів, можна встановити стан всіх її процесорів в разі виходу з ладу не більше t будь-яких з них.

На виконання самотестування система витрачає певні ресурси, тому бажано організувати цей процес так, щоб проводилася мінімальна кількість тестових випробувань. Це дозволяє зменшити загальний час тестування, а чим частіше буде повторюватися процес тестування, тим швидше нова відмова буде виявлена і тим швидше система може бути реконфігурована. Іншими словами, мінімізація кількості проведених тестів є важливою проблемою [106], [107] так само, як це має місце і для інших задач тестування [108] – [110].

Будемо вважати, що процесор може в один і той же момент часу або тестувати інший процесор, або бути тестованим іншим процесором. В роботі [111] було запропоновано алгоритм, що дозволяє визначити стан системи шляхом виконання не більше, ніж $N + 2p$ тестових перевірок, де N – кількість процесорів в системі, а p – число реально несправних процесорів. Необхідна умова застосування цього алгоритму: оргграф, що відображає можливості взаємотестування процесорів ВБС, має бути повним. Реальні ж структури далеко не завжди відповідають повному графу.

В [104] визначена теоретична границя величини t і доведено, що повне діагностування системи, тобто визначення справності всіх процесорів системи, є завжди можливим, якщо допустима кількість несправних процесорів не перевищує величини $[(N - 1) / 2]$, де N – кількість процесорів системи. В [112]

показано, що практично завжди це можливо і при більшій кількості несправностей.

У багатьох випадках визначення стану процесорів системи на основі результатів виконаних взаємних тестових перевірок є досить складною і трудомісткою процедурою. У даному розділі вирішується задача формалізації процедури встановлення стану всіх процесорів після виконання певної кількості тестових випробувань (не обов'язково всіх можливих) за будь-якої топології зв'язків (не обов'язково повний граф) системи. Варто зазначити, що задача побудови повного тесту в межах даної роботи не розглядається, тобто припускається, що деякий набір тестових перевірок заданий розробником системи.

4.1. Формальне представлення результатів окремих тестових перевірок

Нехай M -діагностована система складається з N процесорів і стану кожного з них відповідає деяка булева змінна x_i ($i = 1, 2, \dots, N$), що приймає значення 1, якщо процесор є справним і 0, якщо він вийшов з ладу. В деякий момент часу частина процесорів є справними і відповідні їм змінні рівні 1. Множину цих змінних позначимо як W . Решта процесорів – несправні, відповідні їм змінні рівні 0, і множину цих змінних позначимо F . Очевидно, що:

$$\left(\bigwedge_{x_i \in W} x_i \right) \wedge \left(\bigwedge_{x_j \in F} \bar{x}_j \right) = 1. \quad (4.1)$$

Зазначимо, що, якщо прибрати, додати, або переставити хоча б одну інверсію в лівій частині рівняння (4.1), воно прийме нульове значення.

Побудуємо таблицю значень результатів тестування j -го процесора i -м процесором в залежності від станів цих процесорів відповідно до моделі ПМЧ (таблиця 4.1).

Таблиця 4.1

Значення результатів тестування j -го процесора i -м процесором в залежності від станів цих процесорів

x_i	x_j	Результат тестування	
1	1	0	
1	0	1	
0	1	*	
0	0	*	

Символом "*" тут позначено довільне значення: 0 або 1.

Відповідно до рівняння (4.1) і таблиці 4.1 результат тестування *може дорівнювати 0* лише в двох випадках: якщо обидва процесора є справними або тестуючий процесор є несправним, тобто коли справедливо:

$$x_i x_j \vee \bar{x}_i \equiv x_j \vee \bar{x}_i = 1. \quad (4.2)$$

В той же час, результат тестування *може дорівнювати 1* також лише в двох випадках: якщо тестуючий процесор є справним, а тестований – несправним, або якщо тестуючий процесор несправний, тобто коли справедливо:

$$x_i \bar{x}_j \vee \bar{x}_i \equiv \bar{x}_j \vee \bar{x}_i = 1. \quad (4.3)$$

Нехай після проведення $k - 1$ тестових перевірок побудоване деяке рівняння виду $V_{k-1}(x_1, x_2, \dots, x_N) = 1$. І нехай k -й тест полягає в тестуванні j -го процесора i -м процесором. В результаті цього тесту отримано значення r_k . Можемо записати:

$$R_k(x_i, x_j) \triangleq \begin{cases} x_i x_j \vee \bar{x}_i, & \text{якщо } r_k = 0 \\ x_i \bar{x}_j \vee \bar{x}_i, & \text{якщо } r_k = 1 \end{cases}, \quad (4.4)$$

або в спрощеній формі

$$R_k(x_i, x_j) \triangleq \begin{cases} x_j \vee \bar{x}_i, & \text{якщо } r_k = 0 \\ \bar{x}_j \vee \bar{x}_i, & \text{якщо } r_k = 1 \end{cases}, \quad (4.5)$$

а також

$$V_k(x_1, x_2, \dots, x_N) \triangleq V_{k-1}(x_1, x_2, \dots, x_N) \wedge R_k(x_i, x_j). \quad (4.6)$$

Відповідно до (4.2-4.5) бачимо, що $R_k(x_i, x_j) = 1$, отже,

$$V_k(x_1, x_2, \dots, x_N) = 1. \quad (4.7)$$

Далі, використовуючи результати наступного тестового випробування, отримаємо вираз R_{k+1} і, відповідно, V_{k+1} . Продовжуємо цей процес доти, поки не будуть враховані результати всіх виконаних тестових перевірок [113].

В якості початкового виразу використовуватимемо $V_0(x_1, x_2, \dots, x_N) \triangleq 1$, і, отже, $V_1 \triangleq R_1$. Після проведення n тестів в результаті ітеративного застосування формули (4.6) вираз V_n набуде вигляду:

$$V_n \triangleq R_1 \wedge R_2 \wedge \dots \wedge R_n. \quad (4.8)$$

Перетворивши вираз (4.8) в досконалиу диз'юнктивну нормальну форму (ДДНФ), отримаємо:

$$V_n \equiv C_1 \vee C_2 \vee \dots \vee C_K. \quad (4.9)$$

Таким чином, маємо рівняння:

$$C_1 \vee C_2 \vee \dots \vee C_K = 1. \quad (4.10)$$

Кожен з виразів C_i є конституентною одиницею – елементарною кон'юнкцією, що містить всі змінні x_i ($i = 1, 2, \dots, N$). Очевидно, що тільки одна з конституент дорівнює 1, і відповідно до (4.1) вона ідентифікує стан системи (тобто представляє стан кожного з її процесорів). Доведемо, що вона буде присутня в (4.10).

Пригадаємо, що початковий (вироджений) варіант рівняння (до виконання будь-яких перевірок) має вигляд $1 = 1$, де ліва (або права) частина може бути виражена як диз'юнкція всіх можливих конституент одиниці, серед яких, безумовно, буде і та, що відповідає актуальному стану системи. Далі, відповідно до результатів тестів формуються вирази R_i , що дозволяють виключити деякі з конституент (причому, часто групами), наближаючи нас до єдиного рішення.

Розглянемо вираз V_n , отриманий після проведення n тестів, представлений в ДДНФ (4.10). Припустимо, що даний вираз дійсно містить конституенту, що відповідає реальному стану системи, яку позначимо як C_R . Нехай на основі результатів $(n + 1)$ -го тесту було сформовано вираз R_{n+1} . Тоді відповідно до (4.6) та (4.10)

$$\begin{aligned}
 V_{n+1} &\triangleq V_n \wedge R_{n+1} \equiv (C_1 \vee C_2 \vee \dots \vee C_K) \wedge R_{n+1} \equiv \\
 &\equiv C_1 \wedge R_{n+1} \vee C_2 \wedge R_{n+1} \vee \dots \vee C_K \wedge R_{n+1}.
 \end{aligned}
 \tag{4.11}$$

Очевидно, що для конститuent C_i можливі тільки два варіанти:

1. $C_i \wedge R_{n+1} \equiv C_i$, що відповідає ситуації, коли R_{n+1} не виключає стану системи, відповідного C_i .
2. $C_i \wedge R_{n+1} \equiv 0$, що відповідає ситуації, коли R_{n+1} такий стан виключає.

Покажемо, що конститuenta C_R , яка відповідає реальному стану системи виключена бути не може. Відзначимо, що відповідно до (4.1) справедливо

$$C_R = 1. \tag{4.12}$$

Припустимо, що вираз R_{n+1} виключає стан, відповідний C_R , тобто

$$C_R \wedge R_{n+1} \equiv 0. \tag{4.13}$$

Згадаємо, що відповідно до (4.2-4.5) справедливо

$$R_{n+1} = 1. \tag{4.14}$$

Об'єднавши (4.12) та (4.14) отримаємо $C_R \wedge R_{n+1} = 1$. В сукупності з (4.13) це призводить до протиріччя виду $0 = 1$, тобто висловлене припущення є невірним. Узагальнюючи даний висновок для будь-якого V_i , можна стверджувати, що жодне тестове випробування не призведе до виключення конститuentи C_R , яка відповідає реальному стану системи, і, отже, $C_R \wedge R_{n+1} \equiv C_R$, і вираз V_{n+1} в ДДНФ також міститиме конститuentу C_R .

Далі звернемо увагу на наступне: система є M -діагностованою, а тому ми можемо визначити її стан, якщо в ній вийшло з ладу не більше ніж M процесорів. Припустимо, що це так, тобто в ній дійсно вийшло з ладу M або менше процесорів. Тоді відповідно до (4.1) будь-яка C_j , що містить більш ніж M інверсій дорівнюватиме нулю. Іншими словами, можна виключати з V_n всі C_j , що містять більше, ніж M інверсій. Зазначимо, що конститuenta C_R , що відповідає реальному стану системи на даному етапі також не буде виключена (якщо тільки вона не містить більш ніж M інверсій, що відповідає ситуації, коли в системі вийшло з ладу більше ніж M процесорів).

В результаті після такого перетворення, рівняння (4.10) прийме одну з трьох форм:

$$1) C_i = 1;$$

$$2) C_{i_1} \vee C_{i_1} \vee \dots \vee C_{i_L} = 1;$$

$$3) 0 = 1.$$

У першому випадку ми отримали в лівій частині рівняння рівно одну конституенту одиниці, тобто елементарну кон'юнкцію, яка містить всі змінні x_i ($i = 1, 2, \dots, N$). В цьому випадку, якщо в системі дійсно вийшло з ладу не більше ніж M процесорів (тобто якщо кон'юнкції з більшою кількістю інверсій були виключені правомірно), відповідно до (4.1) можливо визначити стан кожного з процесорів, а саме: справними є всі ті процесори, яким відповідають змінні, що входять в кон'юнкцію без інверсій, і несправними є всі ті процесори, яким відповідають змінні, що входять в кон'юнкцію з інверсіями.

В другому випадку в виразі залишилося більше однієї елементарної кон'юнкції. Якщо в системі дійсно вийшло з ладу не більше ніж M процесорів, то можна лише сказати, що одна з цих кон'юнкцій дійсно відповідає реальному стану системи (як уже було показано, вона не може бути виключена відповідно до результатів жодного із тестів). Іншими словами, має місце певна невизначеність, тобто тест, обраний розробником є недостатнім для повного діагнозу станів процесорів) та для визначення цього стану необхідно провести додаткові тести. Якщо ж вже були виконані всі тести, можливі відповідно до топології системи, то даний результат, очевидно, означає, що система не є M -діагностованою.

Третій випадок, коли з виразу виключаються всі елементарні кон'юнкції, може настати лише тоді, коли в системі насправді вийшло з ладу більш ніж M процесорів (отже, кон'юнкція, що відповідає реальному стану системи, була виключена неправомірно, що, в свою чергу, призвело до відсутності кон'юнкцій в лівій частині рівняння). Відзначимо, що зворотне, взагалі кажучи, не вірно: в разі виходу з ладу більш ніж M процесорів рівняння може приймати будь-яку з вищенаведених форм.

В Додатку Б наведено ряд прикладів застосування даного методу.

4.2. Проведення додаткових тестів

Як було показано в попередньому параграфі, проведення деякого набору тестів може бути недостатньо для однозначного визначення стану всіх процесорів системи. Дана ситуація характерна тим, що в виразі V_n залишається більше однієї елементарної кон'юнкції. В цьому випадку необхідно проводити додаткові тести до тих пір, поки всі, окрім однієї кон'юнкції не будуть виключені. В той же час, деякі тести можуть виявитися марними та проводити їх немає необхідності. Інтерес представляє можливість вибирати і проводити лише корисні тести, тобто такі, на основі результатів яких можуть бути виключені ті чи інші кон'юнкції, присутні в поточному виразі.

Розглянемо вираз V_n після проведення деякого набору тестів та приведення до ДДНФ. Він містить L елементарних кон'юнкцій та має вигляд:

$$C_{b_1} \vee C_{b_2} \vee \dots \vee C_{b_L} = 1. \quad (4.15)$$

Розглянемо деякий тест, який полягає в тестуванні j -го процесора i -м процесором. В результаті проведення цього тесту і в залежності від отриманого результату r_{n+1} , отримаємо вираз R_{n+1} . Відповідно до (4.4, 4.6), вираз V_{n+1} матиме вигляд:

$$V_{n+1} = (C_{b_1} \vee C_{b_2} \vee \dots \vee C_{b_L}) \wedge (x_i x_j \vee \bar{x}_i), \quad (4.15)$$

якщо $r_{n+1} = 0$, і

$$V_{n+1} = (C_{b_1} \vee C_{b_2} \vee \dots \vee C_{b_L}) \wedge (x_i \bar{x}_j \vee \bar{x}_i), \quad (4.16)$$

якщо $r_{n+1} = 1$.

Розкривши дужки у виразі (4.15) отримаємо:

$$V_{n+1} = (C_{b_1} x_i x_j \vee C_{b_2} x_i x_j \vee \dots \vee C_{b_L} x_i x_j) \vee (C_{b_1} \bar{x}_i \vee C_{b_2} \bar{x}_i \vee \dots \vee C_{b_L} \bar{x}_i). \quad (4.17)$$

Аналогічно, із виразу (4.16) отримаємо:

$$V_{n+1} = (C_{b_1} x_i \bar{x}_j \vee C_{b_2} x_i \bar{x}_j \vee \dots \vee C_{b_L} x_i \bar{x}_j) \vee (C_{b_1} \bar{x}_i \vee C_{b_2} \bar{x}_i \vee \dots \vee C_{b_L} \bar{x}_i). \quad (4.18)$$

Якщо ж замість виразів з (4.4) використовувати спрощені вирази з (4.5), то в результаті застосування (4.6) отримаємо

$$V_{n+1} = (C_{b_1} \vee C_{b_2} \vee \dots \vee C_{b_L}) \wedge (x_j \vee \bar{x}_i), \quad (4.19)$$

якщо $r_{n+1} = 0$, і

$$V_{n+1} = (C_{b_1} \vee C_{b_2} \vee \dots \vee C_{b_L}) \wedge (\bar{x}_j \vee \bar{x}_i), \quad (4.20)$$

якщо $r_{n+1} = 1$, а після розкриття дужок в (4.19) та (4.20) відповідно

$$V_{n+1} = (C_{b_1} x_j \vee C_{b_2} x_j \vee \dots \vee C_{b_L} x_j) \vee (C_{b_1} \bar{x}_i \vee C_{b_2} \bar{x}_i \vee \dots \vee C_{b_L} \bar{x}_i). \quad (4.21)$$

та

$$V_{n+1} = (C_{b_1} \bar{x}_j \vee C_{b_2} \bar{x}_j \vee \dots \vee C_{b_L} \bar{x}_j) \vee (C_{b_1} \bar{x}_i \vee C_{b_2} \bar{x}_i \vee \dots \vee C_{b_L} \bar{x}_i). \quad (4.22)$$

Легко показати, що в результаті спрощення (зокрема, видалення повторюваних членів) вирази (4.21) та (4.22) набудуть вигляду відповідно виразів (4.15) та (4.16). При цьому, варто зазначити: було помічено, що використання саме неспрощених виразів з (4.4) дозволяє в подальшому дещо спростити процес їх перетворення.

Розіб'ємо множину кон'юнкцій, $\{C_{b_1}, C_{b_2}, \dots, C_{b_L}\}$ на три підмножини: Q_S , Q_F та Q_U , в які входять кон'юнкції, що містять відповідно x_i та x_j , x_i та \bar{x}_j , а також \bar{x}_i . Легко помітити, що ці множини не перетинаються між собою, а також, що будь-яку C_{b_k} можна віднести до однієї з цих множин. Таким чином, $|Q_S| + |Q_F| + |Q_U| = L$.

Легко помітити, що $\forall C_{b_k} \in Q_S$ вірно $C_{b_k} x_i x_j \equiv C_{b_k}$, $C_{b_k} x_i \bar{x}_j \equiv 0$ та $C_{b_k} \bar{x}_i \equiv 0$. Аналогічно, $\forall C_{b_l} \in Q_F$ вірно $C_{b_l} x_i x_j \equiv 0$, $C_{b_l} x_i \bar{x}_j \equiv C_{b_l}$ та $C_{b_l} \bar{x}_i \equiv 0$, а також $\forall C_{b_h} \in Q_U$ вірно $C_{b_h} x_i x_j \equiv 0$, $C_{b_h} x_i \bar{x}_j \equiv 0$ та $C_{b_h} \bar{x}_i \equiv C_{b_h}$.

Таким чином, якщо $r_{n+1} = 0$,

$$V_{n+1} \equiv \left(\bigvee_{C_{b_k} \in Q_S} C_{b_k} \right) \vee \left(\bigvee_{C_{b_h} \in Q_U} C_{b_h} \right) \equiv \bigvee_{C_{b_g} \in (Q_S \cup Q_U)} C_{b_g} \quad (4.23)$$

Отже, V_{n+1} міститиме $|Q_S| + |Q_U| \leq L$ елементарних кон'юнкцій, що на $|Q_F|$ менше, ніж кількість елементарних кон'юнкцій, що містить V_n .

В той же час, якщо $r_{n+1} = 1$,

$$V_{n+1} \equiv \left(\bigvee_{C_{b_l} \in Q_F} C_{b_l} \right) \vee \left(\bigvee_{C_{b_h} \in Q_U} C_{b_h} \right) \equiv \bigvee_{C_{b_g} \in (Q_F \cup Q_U)} C_{b_g} \quad (4.24)$$

Відтак, V_{n+1} міститиме $|Q_F| + |Q_U| \leq L$ елементарних кон'юнкцій, що на $|Q_S|$ менше, ніж кількість елементарних кон'юнкцій, що містить V_n .

Таким чином, після проведення такого тесту з рівняння буде виключено в гіршому випадку $d = \min(|Q_S|, |Q_F|)$, а в кращому – $D = \max(|Q_S|, |Q_F|)$ елементарних кон'юнкцій. Очевидно, якщо $D = 0$, проводити такий тест не має сенсу. Відзначимо також, що кон'юнкції з множини Q_U не будуть виключені незалежно від результату тесту.

Варто зазначити, що розбиття множини кон'юнкцій на підмножини Q_S , Q_F та Q_U залежить від обраного тесту, тобто, номерів i та j тестуючого та тестованого процесорів. На базі викладеного вище аналізу можуть будуватися стратегії вибору оптимальної послідовності тестів. Наприклад, в якості наступного тестового випробування можна вибирати таке, яке дає найбільшу гарантовану кількість кон'юнкцій, що виключаються (максимізувати значення d), найбільшу очікувану кількість кон'юнкцій, що виключаються (найбільше значення $\frac{d+D}{2}$), а також намагатися уникнути фрагментації (коли після проведення деякого набору тестових випробувань, для будь-якого можливого наступного тесту значення $|Q_U|$ є дуже близьким до L) тощо.

При послідовному тестуванні корисним може бути (якщо це можливо) виконувати пошук (перерахунок) наступного тесту безпосередньо після отримання результату чергового тестового випробування і на основі цього значення. При паралельному тестуванні це зробити неможливо, тому що тести проводяться паралельно і їх результати є ще невідомими. В такому випадку для отримання оптимальних тестових наборів може знадобитися проведення більш глибокого аналізу. Логічно припустити, що бажано вибирати тести так, щоб якомога більша частина з них не виявлялася безглуздою після проведення інших, незалежно від їх результату (тобто уникати ситуацій, коли після проведення частини тестових випробувань із тестового набору, незалежно від того, якими

були їх результати, решта тестів вже жодним чином не дозволять виключити будь-які елементарні кон'юнкції).

Цікавою також є можливість визначення стану системи навіть для деяких випадків виходу з ладу більш, ніж M процесорів. В такій ситуації стратегія може полягати в проведенні всіх можливих тестів і виборі в рівнянні-результаті кон'юнкції з найменшим числом інверсій. Звичайно, при цьому з виразу також слід не виключати кон'юнкції, що містять більш ніж M інверсій. Зазначимо, що у випадку наявності т.з. «плаваючих» несправностей (коли результат виконання певного тестового випробування несправним процесором може відрізнитися від разу до разу), може стати корисним також повторне виконання вже виконаних раніше тестів.

Аналіз кожного з цих випадків виходить за рамки даної роботи, але може стати основою для проведення подальших досліджень.

4.3. Оцінка ефективності методу для заданого набору тестів

Визначимо за яких умов запропонований метод дозволяє визначити стан системи за заданим набором тестів для випадку виходу з ладу певної кількості будь-яких її процесорів. Вважатимемо, що ця кількість (максимальне її значення) відповідатиме максимальному значенню ступені діагностованості системи, що може використовуватися при застосуванні методу.

Для представлення набору тестових перевірок скористаємося однією з моделей діагностичного графа [114], [115]. Діагностичний граф G є орієнтованим графом, вершини якого відповідають процесорам системи, а дуги – тестовим перевіркам. При цьому вершина v_i відповідає i -му процесору, а дуга з вершини v_j в вершину v_k відповідає тестуванню k -го процесора j -м процесором.

Нехай діагностування системи, що містить N процесорів відбувається згідно з моделлю ПМЧ та відповідає діагностичному графу G . Нехай при виході з ладу будь-яких M процесорів, в результаті проведення деякого набору тестів, можливо однозначно встановити стан (роботоздатний/несправний) кожного з процесорів.

Твердження 1.

$$M < \frac{N}{2}$$

Доведення:

Припустимо, що $M \geq \frac{N}{2}$. Довільним чином розіб'ємо множину процесорів системи на дві підмножини: A та B , що мають потужності відповідно M та $L = N - M$. Оскільки $M \geq \frac{N}{2}$, то $L = N - M \leq \frac{N}{2}$, а отже, $L \leq M$.

Розглянемо два можливих випадки.

Випадок 1:

Всі процесори з множини A є роботоздатними, а всі процесори з множини B – несправними. При цьому несправності в цих процесорах є такими, що завжди призводять до інверсії результатів тестування цими процесорами інших процесорів (тобто, якщо такий процесор тестуватиме інший несправний процесор, результатом тестування буде 0, а якщо ж він тестуватиме роботоздатний процесор, то 1). Таким чином, в даній ситуації матимемо $L \leq M$ несправних процесорів.

В результаті проведення тестування процесору y процесором x отримаємо наступні результати:

- 1) 0, якщо $x \in A$ та $y \in A$;
- 2) 1, якщо $x \in A$ та $y \in B$;
- 3) 1, якщо $x \in B$ та $y \in A$;
- 4) 0, якщо $x \in B$ та $y \in B$.

Випадок 2:

Ситуація є протилежною: всі процесори з A – несправні (причому, як і в попередньому випадку, результати тестів інвертуються), а всі процесори з B є роботоздатними. В даному випадку матимемо M несправних процесорів.

В результаті проведення тестування процесору y процесором x отримаємо наступні результати:

- 1) 0, якщо $x \in A$ та $y \in A$;

- 2) 1, якщо $x \in A$ та $y \in B$;
- 3) 1, якщо $x \in B$ та $y \in A$;
- 4) 0, якщо $x \in B$ та $y \in B$.

Як бачимо, у обох випадках результати всіх можливих тестів співпадатимуть. Тобто, має місце неоднозначність, в результаті якої визначити достовірний стан системи за результатами тестів не вдається.

Таким чином, при $M \geq \frac{N}{2}$, не завжди є можливим однозначно визначити стан системи. ■

Твердження 2.

Запропонованим методом достовірний стан системи у випадку виходу з ладу не більш ніж M її довільних процесорів при заданому наборі тестових перевірок, що відповідають діагностичному графу G , можливо встановити тоді і тільки тоді, коли для будь-якої множини F вершин графа G , потужністю K існує $T > M - \left\lfloor \frac{K}{2} \right\rfloor$ вершин графа G , які не містяться в F , таких, що з кожної з них існує дуга хоча б в одну з вершин із множини F .

Відзначимо також, що значення ступені діагностованості системи, що використовується при застосуванні методу не має перевищувати M .

Доведення:

Розглянемо деяку множину F вершин графа G , потужністю K . Їй відповідатиме множина процесорів F_p .

Нехай множина E містить всі вершини графа G , що не містяться в F , та з кожної з яких існує дуга хоча б в одну з вершин із множини F . T – потужність множини E . Цій множині відповідає множина процесорів E_p .

Доведемо необхідність:

Припустимо, що $T \leq M - \left\lfloor \frac{K}{2} \right\rfloor$. Нехай в системі вийшли з ладу всі процесори з E_p (T штук), а також $\left\lfloor \frac{K}{2} \right\rfloor$ процесорів з F_p . Тобто, кількість несправних процесорів в системі: $T + \left\lfloor \frac{K}{2} \right\rfloor \leq M$.

Оскільки $\left\lceil \frac{K}{2} \right\rceil \geq \frac{K}{2}$, то, виходячи з *Твердження 1*, тестування процесорів з F_p лише процесорами з F_p (тобто лише власними силами) не дозволить у всіх випадках однозначно встановити стан всіх з них.

Інші тестові перевірки процесорів з F_p можуть проводитися лише процесорами з E_p (оскільки дуги до вершин з множини F є лише у вершин з множини E), але всі ці процесори є несправними, тобто результати тестів можуть бути будь-якими, незалежно від станів тестованих процесорів, а тому вони не допоможуть встановити стан процесорів з F_p .

Таким чином, якщо $T \leq M - \left\lceil \frac{K}{2} \right\rceil$, то не завжди є можливим однозначно встановити стан всіх процесорів системи.

Доведемо достатність.

Нехай після проведення деякої множини тестових перевірок, згідно з методом, описаним в параграфі 4.1, було отримане рівняння виду:

$$C_1 \vee C_2 \vee \dots \vee C_S = 1. \quad (4.25)$$

Вважатимемо, що при застосуванні методу було використане значення ступені діагностованості системи, рівне M . Таким чином, кількість змінних з інверсіями (що відповідають несправним процесорам) в кожній з кон'юнкцій C_i у виразі (4.25) не може перевищувати M (оскільки такі кон'юнкції виключаються з рівняння).

Нехай C_r відповідає реальному стану процесорів системи. Доведемо, що для кожної C_i , $i \neq r$, існуватиме тест, що дозволить її виключити.

Вирази C_i та C_r можна представити в наступному вигляді:

$$C_r = A_i \wedge B_i, \quad (4.26)$$

$$C_i = A_i \wedge \tilde{B}_i, \quad (4.27)$$

де A_i – кон'юнкція всіх змінних, що співпадають в C_i та C_r (тобто, в обох випадках є неінвертованими або в обох випадках є інвертованими), а B_i та \tilde{B}_i – кон'юнкції всіх змінних, що відрізняються в них, причому, очевидно, що всі змінні в \tilde{B}_i будуть інвертовані відносно відповідних змінних в B_i .

Нехай $A_i = A_i^w \wedge A_i^f$, де A_i^w містить лише змінні без інверсій, а A_i^f – лише змінні з інверсіями. Окрім того, кількість змінних в A_i^f позначимо як L .

Нехай $B_i = B_i^w \wedge B_i^f$, де B_i^w містить лише змінні без інверсій, а B_i^f – лише змінні з інверсіями. Окрім того, кількість змінних в B_i^w та B_i^f позначимо відповідно як P та Q .

Нехай також $\tilde{B}_i = \tilde{B}_i^w \wedge \tilde{B}_i^f$. Причому, \tilde{B}_i^w містить лише змінні без інверсій. Це всі ті змінні, що входять до B_i^f , а тому їх кількість – Q . В той же час, \tilde{B}_i^f містить лише змінні з інверсіями, а це є всі ті змінні, що входять до B_i^w , тому їх кількість – P .

Нехай F_p – множина всіх процесорів, котрим відповідають змінні з B_i та \tilde{B}_i . Її потужність: $K = Q + P$. Цій множині відповідає деяка множина F вершин графа G . Нехай E – множина вершин графа, що не містяться в F та таких, що з кожної з них існує дуга хоча б в одну з вершин із множини F . Потужність цієї множини позначимо як T . Цим вершинам відповідатиме множина процесорів E_p . Очевидно, що множини E_p та F_p не перетинаються, і тому процесорам з множини E_p відповідатимуть змінні лише з A_i .

Відзначимо, що L , Q та P – число інверсій відповідно в A_i , B_i та \tilde{B}_i , а отже, кількість інверсій в C_r : $L + Q$, а в C_i : $L + P$. Згадаємо, що кількість інверсій в C_r , як і в C_i не може перевищувати M , тобто, $L + Q \leq M$ та $L + P \leq M$, а отже, $Q \leq M - L$ та $P \leq M - L$. Тоді: $K = Q + P \leq 2 \cdot (M - L)$, тобто, $2L \leq 2M - K$, відповідно $L \leq M - \frac{K}{2}$, а враховуючи що L та M є цілими, справедливо і $L \leq M - \left\lceil \frac{K}{2} \right\rceil$.

Якщо $T > M - \left\lceil \frac{K}{2} \right\rceil \geq L$, то існує хоча б один процесор з E_p , що не відповідає ні одній зі змінних з A_i^f , а отже він відповідає змінній з A_i^w , тобто такий, що є неінвертованою як в C_r , так і в C_i . Нехай цьому процесору, який називатимемо j -м, відповідає вершина $v_j \in E$ графа G , з якої існує дуга в деяку вершину $v_k \in F$, що відповідає процесору, який називатимемо k -м. Таким чином,

можна провести тестування k -го процесора j -м процесором, в результаті чого буде отримано вираз: $R = \tilde{x}_k \vee \bar{x}_j$, де x_k є інвертованою або неінвертованою в залежності від результату тесту. Оскільки, як вже було сказано, x_i є неінвертованою як в C_r , так і в C_i , то $R \wedge C_r \equiv \tilde{x}_k \wedge C_r$, а $R \wedge C_i \equiv \tilde{x}_k \wedge C_i$. З іншого боку, оскільки k -й процесор відповідає вершині з множини F , то x_k входить до B_i (з інверсією або без) і, відповідно, до \tilde{B}_i (але в протилежному стані). Таким чином, за результатами тесту одну з кон'юнкцій буде виключено. Очевидно, це буде саме C_i , оскільки C_r відповідає реальному стану системи і як було показано в параграфі 4.1 виключена бути не може).

Відзначимо, що аналогічним чином можна довести існування тесту, що виключатиме C_i для будь-якого $i \neq r$. Отже, всі C_i окрім C_r будуть виключені і, таким чином, буде встановлено реальний стан системи. ■

Висновки до розділу 4

Запропоновано формальний метод, що дозволяє встановити стан процесорів M -діагностованої ВБС, який зводиться до вирішення булевого рівняння, сформованого на основі результатів деякої множини тестових перевірок. Змінні цього рівняння відповідають станам кожного з процесорів. Метод передбачає формальне перетворення булевого виразу із виключенням всіх кон'юнкцій, які містять більш ніж M інверсій. В результаті має бути отриманий вираз, що містить рівно одну кон'юнкцію, змінні якої фактично визначають стани процесорів. Якщо цього не досягнуто, необхідно виконати додаткові тестові перевірки. Якщо ж додаткові перевірки, відповідно до топології системи, є неможливими, то, очевидно, система насправді не є M -діагностованою.

Справедливість сказаного вище має місце лише для ситуації, коли в системі дійсно вийшло з ладу не більше, ніж M процесорів. В протилежному випадку отриманий вираз може містити як рівно одну кон'юнкцію (що, очевидно, не відповідатиме реальному стану системи), так і декілька кон'юнкцій (жодна з

яких не відповідатиме реальному стану системи), або ж і не містити жодної кон'юнкції взагалі.

Виконано оцінку ефективності запропонованого методу відповідно до заданого набору тестових перевірок. Сформульовано критерії для діагностичного графу, що дозволяють визначити максимальну кратність відмов, при яких застосування методу є можливим.

РОЗДІЛ 5.

МОДЕЛІ ІНШИХ НЕБАЗОВИХ СИСТЕМ

Окрім модифікації реберних функцій, описаної в розділах 2 та 3 можливим є ряд інших модифікацій MBR-моделей, що дозволяють будувати моделі небазових систем різних типів. Зокрема, варто відзначити так звані зважені системи, системи, які мають більше двох станів роботоздатності, а також системи, що складаються з кількох окремих підсистем. В загальному випадку такі системи не є базовими.

Даний розділ присвячено методам побудови GL-моделей систем розглянутих вище типів на базі модифікації MBR-моделей. Відзначимо, що реальні системи можуть поєднувати в собі властивості систем різних типів, а тому запропоновані в цьому і інших розділах підходи можуть бути комбіновані для побудови їх моделей.

5.1. Системи з більш ніж двома станами роботоздатності

У попередніх роботах, присвячених GL-моделям передбачалося, що система, як і її компоненти можуть перебувати лише в двох станах: роботоздатному та несправному. В той же час, деякі реальні системи організовані так, що кількість їх станів може бути більшою, тобто вводяться так звані стани часткової роботоздатності. При цьому, в загальному випадку як для системи в цілому, так і для кожного з її компонентів, число різних станів може відрізнятися.

Так, наприклад, при виході з ладу деяких процесорів, система може втратити можливість виконувати всі необхідні для її повноцінної роботи обчислення, але все ще зможе вирішувати деякі, найбільш важливі задачі, що вимагають меншої продуктивності. Підсистеми ієрархічної системи також можуть перебувати в більш, ніж двох станах. При цьому, стан системи залежатиме від станів її підсистем.

Відзначимо, що навіть ВБС, яка має лише два стани, може складатися з компонентів і підсистем з більш ніж двома станами роботоздатності. Наявність більше ніж двох станів дозволяє більш раціонально використовувати ресурси частково роботоздатних компонентів, в той час як у випадку двох станів, такі компоненти були б виключені повністю. Особливо актуальною така організація виявляється для ієрархічних систем, оскільки вихід з ладу компонентів підсистеми часто призводить лише до зниження її продуктивності, і далеко не завжди вимагає її повного відключення. В результаті, при достатньому рівні надмірності (відмовостійкості), система могла б залишатися повністю роботоздатною навіть коли більшість (або навіть всі) її підсистем знаходяться в частково роботоздатних станах. З іншого боку, система керування, в разі потрапляння в стан часткової роботоздатності, може перейти до аварійного режиму (живучості), виконуючи при цьому менш трудомісткі задачі, націлені на забезпечення безпечного завершення роботи системи, або підтримки безпечного стану об'єкта керування.

Для вищеописаних систем так само як і для систем з двома станами необхідно розраховувати різні показники надійності, наприклад, ймовірності перебування в тому чи іншому стані. В [11] розглянуто ряд підходів, що використовуються для вирішення цих задач. Відзначимо, що графо-логічні моделі після деякої модифікації також можуть бути використані для роботи з системами, що мають більш ніж два стани роботоздатності.

5.1.1 GL-моделі систем з більш ніж двома станами роботоздатності

Для побудови моделей вищеописаних систем пропонується модифікувати графо-логічні моделі. В якості вектору стану системи використовуватиметься вектор, що складається з n елементів, де n – кількість компонентів системи, а i -й елемент цього вектору приймає значення з множини $\{0, 1, \dots, S_i - 1\}$, що відповідає стану i -го компонента системи, де S_i – кількість можливих станів цього компонента. При цьому значення 0 відповідає повністю несправному стану, $S_i - 1$ – повністю роботоздатному, а всі інші – станам з частковою

роботоздатністю (більше значення відповідає більш роботоздатному стану). Реберні функції представлятимуть собою функції багатозначної логіки [116], що отримують в якості аргументів елементи цього вектору та приймають значення з множини $\{0, 1, \dots, S - 1\}$, де S – кількість станів системи. В якості критерію, що свідчить про перебування системи в деякому стані t (а точніше, не меншому, ніж t) може використовуватися наявність зв'язків між усіма вершинами графа за допомогою лише таких ребер, значення реберних функцій яких є не меншим ніж t [117]. Цей критерій у випадку $S = 2$ відповідає зв'язності графа.

Розглянемо описаний вище критерій більш докладно. Очевидно, він зводиться до наявності в графі кістяка, все ребра якого мають вагу не меншу ніж t . Таким чином, для визначення стану системи достатньо знайти в графі моделі такий кістяк, у якого вага ребра з мінімальною вагою буде найбільшою. Значення цієї ваги і відповідатиме стану системи.

Задачу пошуку такого кістяка можна звести до задачі пошуку мінімального кістякового дерева, для вирішення якої можна скористатися одним з відомих алгоритмів [118]. Далі будуть запропоновані модифіковані версії двох з них: алгоритму Прима [119] та алгоритму Крускала [120].

Модифікований алгоритм Прима:

Початково множина ребер T та множина вершин U є порожніми. Шуканий стан $W = S - 1$, де S – кількість станів системи.

1. До множини вершин U додаємо довільну вершину графа.
2. Серед всіх ребер графа, що є інцидентними одній вершині з множини U та одній, що не входить до U обираємо таке, вага якого є максимальною.
3. Це ребро додаємо до множини T .
4. Якщо вага цього ребра w є меншою за поточне значення W , то присвоюємо $W = w$.
5. Вершину, інцидентну цьому ребру та таку, що не входить до U додаємо до даної множини.

6. Якщо множина U ще не містить всіх вершин графа, то переходимо до п. 2.
7. Кінець.

Модифікований алгоритм Крускала:

Початково множина ребер T є порожньою. Шуканий стан $W = S - 1$, де S – кількість станів системи.

1. З усіх ребер графу, додавання яких до множини T не призведе до появи циклу, обираємо таке, вага якого є максимальною. Якщо таких ребер немає, переходимо до п. 5.
2. Додаємо це ребро до множини T .
3. Якщо вага цього ребра w є меншою за поточне значення W , то присвоюємо $W = w$.
4. Переходимо до п. 1.
5. Кінець.

Після виконання будь-якого з цих алгоритмів множина T міститиме шуканий кістяк, а W – мінімальну вагу ребра з цього кістяка, і це значення відповідатиме стану модельованої системи.

5.1.2 Побудова моделі простої системи з S станами роботоздатності

Побудуємо модель системи, що складається з n елементів. Елементи цієї системи, як і сама система в цілому можуть перебувати в одному з S станів. Причому, система буде перебувати в стані не меншому ніж t , тільки якщо не більше, ніж m її елементів знаходяться в стані меншому ніж t . Така система розглядається в [11]. Будемо таку систему і відповідну їй модель називати багатозначною (або S -значною) базовою та позначати $K^S(m, n)$.

Для побудови моделі такої системи за основу візьмемо базову $K(m, n)$ МВР-модель, та замінімо у виразах її реберних функцій всі кон'юнкції на операцію взяття мінімуму, а диз'юнкції – на операції взяття максимуму. Доведемо адекватність побудованої таким способом моделі.

Розглянемо довільну реберну функцію вихідної моделі $f_p(x_1, x_2, \dots, x_n)$ та відповідну їй функцію нової моделі $g_p(y_1, y_2, \dots, y_n)$.

Нехай

$$\xi_k(y) = \begin{cases} 0, & \text{при } y < k \\ 1, & \text{при } y \geq k \end{cases} \quad (5.1)$$

Доведемо що $\forall k, y_1, y_2, \dots, y_n \in \{0, 1, \dots, S-1\}$ справедливо

$$\xi_k(g_p(y_1, y_2, \dots, y_n)) = f_p(\xi_k(y_1), \xi_k(y_2), \dots, \xi_k(y_n)).$$

Для доведення скористаємося методом математичної індукції. В якості базису індукції візьмемо: $f_p(x_1, x_2, \dots, x_n) = x_i$, а $g_p(y_1, y_2, \dots, y_n) = y_i$, де $i \in \{1, 2, \dots, n\}$. В такому випадку $\xi_k(g_p(y_1, y_2, \dots, y_n)) = \xi_k(y_i)$, а $f_p(\xi_k(y_1), \xi_k(y_2), \dots, \xi_k(y_n)) = \xi_k(y_i)$, отже, твердження для базису виконується.

Перейдемо до індукційного переходу. Функція f_p містить лише кон'юнкції та диз'юнкції, а g_p – операції \min та \max . Таким чином, матиме місце один з двох випадків:

$$1) f_p = f_1 \wedge f_2, \text{ а } g_p = \min(g_1, g_2).$$

$$2) f_p = f_1 \vee f_2, \text{ а } g_p = \max(g_1, g_2).$$

Причому вважатимемо, що $\xi_k(g_1(y_1, y_2, \dots, y_n)) = f_1(\xi_k(y_1), \xi_k(y_2), \dots, \xi_k(y_n))$, а $\xi_k(g_2(y_1, y_2, \dots, y_n)) = f_2(\xi_k(y_1), \xi_k(y_2), \dots, \xi_k(y_n))$.

Доведемо справедливість твердження для першого випадку. У відповідності з усім вищевикладеним, побудуємо таблицю значень функцій $f_p(\xi_k(y_1), \xi_k(y_2), \dots, \xi_k(y_n))$, $g_p(y_1, y_2, \dots, y_n)$ та $\xi_k(g_p(y_1, y_2, \dots, y_n))$ в залежності від значень $f_1(\xi_k(y_1), \xi_k(y_2), \dots, \xi_k(y_n))$, $f_2(\xi_k(y_1), \xi_k(y_2), \dots, \xi_k(y_n))$ та $g_1(y_1, y_2, \dots, y_n)$, $g_2(y_1, y_2, \dots, y_n)$ (таблиця 5.1).

Відповідно до таблиці 5.1, $\xi_k(g_p) = f_p$, а отже твердження в даному випадку виконується.

Доведемо справедливість твердження для другого випадку. Аналогічно доведенню для першого випадку, побудуємо таблицю значень функцій

$f_p(\xi_k(y_1), \xi_k(y_2), \dots, \xi_k(y_n)), g_p(y_1, y_2, \dots, y_n)$ та $\xi_k(g_p(y_1, y_2, \dots, y_n))$ в залежності від значень $f_1(\xi_k(y_1), \xi_k(y_2), \dots, \xi_k(y_n)), f_2(\xi_k(y_1), \xi_k(y_2), \dots, \xi_k(y_n))$ та $g_1(y_1, y_2, \dots, y_n), g_2(y_1, y_2, \dots, y_n)$ (таблиця 5.2).

Таблиця 5.1

Значення f_p, g_p та $\xi_k(g_p)$ в залежності від значень f_1, f_2, g_1 та g_2 для випадку

$$f_p = f_1 \wedge f_2 \text{ та } g_p = \min(g_1, g_2)$$

f_1	f_2	g_1	g_2	$f_p = f_1 \wedge f_2$	$g_p = \min(g_1, g_2)$	$\xi_k(g_p)$
0	0	$< k$	$< k$	0	$< k$	0
0	1	$< k$	$\geq k$	0	$< k$	0
1	0	$\geq k$	$< k$	0	$< k$	0
1	1	$\geq k$	$\geq k$	1	$\geq k$	1

Таблиця 5.2

Значення f_p, g_p та $\xi_k(g_p)$ в залежності від значень f_1, f_2, g_1 та g_2 для випадку

$$f_p = f_1 \vee f_2 \text{ та } g_p = \max(g_1, g_2)$$

f_1	f_2	g_1	g_2	$f_p = f_1 \vee f_2$	$g_p = \max(g_1, g_2)$	$\xi_k(g_p)$
0	0	$< k$	$< k$	0	$< k$	0
0	1	$< k$	$\geq k$	1	$\geq k$	1
1	0	$\geq k$	$< k$	1	$\geq k$	1
1	1	$\geq k$	$\geq k$	1	$\geq k$	1

Відповідно до таблиці 5.2, $\xi_k(g_p) = f_p$, а отже твердження в даному випадку також виконується.

Таким чином, твердження є справедливим для обох випадків. Отже, індукційний перехід має місце. ■

Відзначимо, що модель $K^S(m, n)$, як і модель $K(m, n)$ буде побудована на основі циклічного графа. Як було доведено в [95] кількість таких ребер (і відповідних реберних функцій) $N = n - m + 1$.

Нехай $\psi_k(m, l_k)$ – кількість ребер з вагою, меншою за k , які міститиме граф моделі $K^S(m, n)$ на векторі, що містить l_k елементів зі значенням, меншим ніж k ($k \in \{0, 1, \dots, S-1\}$). Доведемо що:

$$\psi_k(m, l_k) = \begin{cases} 0, & \text{при } l_k < m \\ l_k - m + 1, & \text{при } l_k \geq m \end{cases} \quad (5.2)$$

Розглянемо довільний вектор $Y = \langle y_1, y_2, \dots, y_n \rangle$, де $y_i \in \{0, 1, \dots, S-1\}$, для $i \in \{1, 2, \dots, n\}$. Нехай двійковий вектор $X = \langle x_1, x_2, \dots, x_n \rangle = \langle \xi_k(y_1), \xi_k(y_2), \dots, \xi_k(y_n) \rangle$, де k – довільне число з множини $\{0, 1, \dots, S-1\}$. Нехай l_k – кількість елементів вектору Y , значення яких є меншим k . Очевидно, що згідно з (5.1) кількість нульових елементів у векторі X також дорівнюватиме l_k . Нехай f_1, f_2, \dots, f_N – реберні функції моделі $K(m, n)$, а g_1, g_2, \dots, g_N – реберні функції моделі $K^S(m, n)$. Нехай вектор $R = \langle r_1, r_2, \dots, r_n \rangle$, де $r_j = g_j(y_1, y_2, \dots, y_n)$ для $j \in \{1, 2, \dots, N\}$, а двійковий вектор $Q = \langle q_1, q_2, \dots, q_n \rangle$, де $q_j = f_j(x_1, x_2, \dots, x_n)$ для $j \in \{1, 2, \dots, N\}$. Згідно з доведеним вище твердженням $\forall j \in \{1, 2, \dots, N\}$ справедливо $f_j(x_1, x_2, \dots, x_n) = \xi_k(g_j(y_1, y_2, \dots, y_n))$, отже

$$Q = \langle \xi_k(r_1), \xi_k(r_2), \dots, \xi_k(r_N) \rangle. \quad (5.3)$$

Вектор Q містить значення всіх реберних функцій моделі $K(m, n)$, отже, відповідно до (2.2) кількість нульових елементів в ньому – $\varphi(m, l_k)$. Згідно з (5.1) та (5.3) елементи зі значеннями, меншими ніж k , в векторі R відповідатимуть нульовим елементам в векторі Q . Отже, кількість таких елементів також дорівнюватиме $\varphi(m, l_k)$. Вектор R містить значення всіх реберних функцій моделі $K^S(m, n)$. Відповідно, кількість реберних функцій цієї моделі, що приймуть значення менші ніж k дорівнюватимуть $\psi_k(m, l_k) = \varphi(m, l_k)$, що відповідає (5.2). ■

Як вже було сказано, наявність в графі кістяка, всі ребра якого мають вагу, не меншу, ніж t свідчить про те, що модельована система знаходиться в стані, не меншому ніж t . Оскільки граф моделі – циклічний, то його кістяк може бути отриманий тільки шляхом видалення одного (причому довільного) з його ребер. Таким чином, не більш ніж одне ребро повинно мати вагу, меншу ніж k , тобто:

$$\psi_k(m, l_k) \leq 1. \quad (5.4)$$

Отже, відповідно до (5.2) та (5.4), модель показуватиме стан не менший ніж t тільки в разі, якщо $l_k \leq t$, що і відповідає поведінці модельованої системи.

Приклад. Побудуємо модель системи $K^5(3, 9)$.

Модель будуватимемо за методом, описаним в [86], проте всі операції кон'юнкції замінимо операціями взяття мінімуму, а всі диз'юнкції – операціями взяття максимуму. Таким чином, реберні функції моделі матимуть наступний вигляд:

$$f_1 = \max(x_1, x_2, x_3)$$

$$f_2 = \max[\min(\max(x_1, x_2), \max(\min(x_1, x_2), x_3)), \min(x_4, x_5)]$$

$$f_3 = \max[\min(x_1, x_2, x_3), x_4, x_5]$$

$$f_4 = \max \left[\min \left(\begin{array}{c} \max(x_1, x_2) \\ \max(\min(x_1, x_2), x_3) \\ \max(\min(x_1, x_2, x_3), \min(x_4, x_5)) \\ \max(x_4, x_5) \end{array} \right), \min(x_6, x_7, x_8, x_9) \right]$$

$$f_5 = \max \left[\min(x_1, x_2, x_3, x_4, x_5), \min \left(\begin{array}{c} \max(x_6, x_7) \\ \max(\min(x_6, x_7), \min(x_8, x_9)) \\ \max(x_8, x_9) \end{array} \right) \right]$$

$$f_6 = \max[x_6, x_7, \min(x_8, x_9)]$$

$$f_7 = \max[\min(x_6, x_7), x_8, x_9]$$

Граф моделі матиме вигляд, представлений на рис. 5.1.

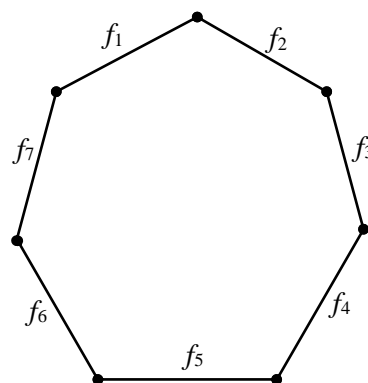


Рис. 5.1. Граф моделі системи $K^5(3, 9)$.

В якості прикладу розглянемо три вектори стану системи:
 $V_1 = \langle 0, 3, 4, 0, 2, 1, 3, 0, 0 \rangle$, $V_2 = \langle 1, 2, 3, 1, 1, 4, 3, 2, 3 \rangle$ та

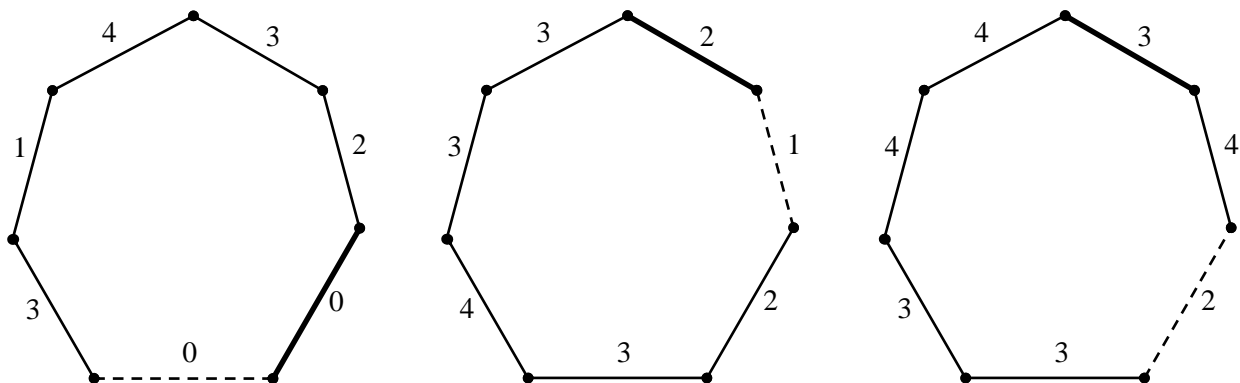
$V_3 = \langle 3, 2, 4, 0, 4, 3, 1, 3, 4 \rangle$. Розрахуємо значення реберних функцій для кожного з цих векторів та результати занесемо до таблиці 5.3.

Таблиця 5.3

Значення реберних функцій моделі $K^5(3, 9)$

	f_1	f_2	f_3	f_4	f_5	f_6	f_7
V_1	4	3	2	0	0	3	1
V_2	3	2	1	2	3	4	3
V_3	4	3	4	2	3	3	4

На рис. 5.2 представлені графи моделі з відповідними вагами для векторів V_1 , V_2 та V_3 . Пунктиром відмічено ребро, що не входить до кістяка, який використовується для визначення стану системи, а більш жирним відмічено ребро з мінімальною вагою з цього кістяка. Вага цього ребра відповідає стану модельованої системи.

Рис. 5.2. Графи моделі системи $K^5(3, 9)$ для векторів V_1 , V_2 та V_3

Таким чином, на векторі V_1 модель показуватиме стан 0, на векторі V_2 – стан 2, а на векторі V_3 – стан 3, що відповідає станам модельованої системи (очікуваним станам) на відповідних векторах.

5.2. Зважені системи

Раніше, розглядаючи базові системи ми припускали, що всі їх елементи є рівноцінними. Однак, на практиці це не завжди так. Наприклад, деякі з процесорів ВБС можуть мати більш високу продуктивність, ніж інші і тому вихід їх з ладу більш суттєво впливає на втрату її роботоздатності. Звісно, така система вже не буде базовою, але іноді її все ж можна представити за допомогою базової моделі.

Нехай кожному елементу системи, що складається з n компонентів відповідає деяка вага w_i , $i \in \{1, 2, \dots, n\}$. Система залишається роботоздатною рівно до тих пір, поки сума ваг всіх її роботоздатних елементів є не меншою за деяке порогове значення t . Такі системи називаються зваженими, вони розглядаються зокрема в [11], [82], [121]. Очевидно, якщо всі ваги рівні 1, така система буде еквівалентною базовій $K(m, n)$ системі.

Звісно, зважені системи можуть бути досить складними і мати нетривіальні умови виходу з ладу. Крім того, в результаті модифікація існуюча система може стати зваженою (наприклад, в результаті заміни деяких її процесорів більш продуктивними).

Розробникам ВБС, які звикли користуватися графо-логічними моделями було б зручно застосовувати цей же інструмент і для вищеописаних систем. На щастя, це виявляється можливим: в наступних параграфах будуть запропоновані способи побудови моделей зважених систем на основі GL-моделей. Відзначимо, що в разі необхідності отримані моделі можуть бути додатково модифіковані вже відомими способами модифікації GL-моделей.

5.2.1 Моделі систем з цілими вагами

Розглянемо зважену систему, всі ваги w_i якої, як і порогове значення t , представлені цілими невід'ємними числами. Очевидно, сума ваг всіх елементів такої системи $w = \sum_{i=1}^n w_i$ – також ціле невід'ємне число. Система залишається роботоздатною рівно доти, доки сума ваг всіх її роботоздатних елементів є не

меншою ніж t . Ця умова рівнозначна тому, що сума ваг всіх несправних елементів системи є не більшою ніж $w - t$.

Нехай вектор $v = \langle x_1, x_2, \dots, x_n \rangle$ – вектор стану системи, а вектор v_e складається зі значень x_i , $i \in \{1, 2, \dots, n\}$, кожне з яких повторюється w_i разів. Таким чином, довжина вектору v_e дорівнює w . Сума ваг всіх роботоздатних компонентів системи відповідатиме кількості одиниць у векторі v_e , а сума ваг всіх нероботоздатних компонентів – кількості нулів в ньому. Отже, система залишається роботоздатною рівно доти, доки кількість нулів в векторі v_e не перевищує $w - t$, що відповідає моделі $K(w - t, w)$.

Таким чином, зваженій системі з цілими невід'ємними вагами w_i і цілим невід'ємним значенням порога t відповідає базова модель $K(w - t, w)$, вхідний вектор якої побудований на основі вектору стану системи шляхом повторення кожного його елемента кількість разів, рівну значенню ваги відповідного компонента системи (w – сума ваг всіх компонентів системи) [122].

Приклад

Побудуємо модель системи, що складається з 5 елементів, з вагами 1, 1, 2, 3 та 4, яка залишається роботоздатною рівно доти, доки сума ваг всіх її роботоздатних компонентів є не меншою за 7.

Сума ваг всіх компонентів системи: $1+1+2+3+4=11$. Вектор стану системи: $v = \langle x_1, x_2, x_3, x_4, x_5 \rangle$. Таким чином, системі відповідатиме модель $K(11 - 7, 11)$, тобто $K(4, 11)$, вхідний вектор якої матиме вигляд: $v_e = \langle x_1, x_2, x_3, x_3, x_4, x_4, x_4, x_5, x_5, x_5, x_5 \rangle$.

Побудуємо реберні функції цієї моделі:

$$f_1 = x_1 \vee x_2 \vee x_3 \vee x_3 x_4 x_4$$

$$f_2 = (x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee (x_3 \vee x_4)(x_3 x_4 \vee x_4)$$

$$f_3 = x_1 x_2 x_3 \vee x_3 \vee x_4 \vee x_4$$

$$f_4 = (x_1 \vee x_2 \vee x_3)((x_1 \vee x_2)(x_1 \vee x_2 x_3) \vee x_3 x_4 x_4) \wedge$$

$$\wedge (x_1 x_2 x_3 \vee (x_3 \vee x_4)(x_3 x_4 \vee x_4))(x_3 \vee x_4 \vee x_4) \vee x_4 x_5 x_5 x_5 x_5$$

$$f_5 = (x_1 \vee x_2)(x_1 x_2 \vee x_3)(x_1 x_2 x_3 \vee x_3 x_4 x_4)(x_3 \vee x_4)(x_3 x_4 \vee x_4) \vee$$

$$\vee (x_4 \vee x_5)(x_4 x_5 \vee x_5)(x_4 x_5 x_5 \vee x_5 x_5)(x_5 \vee x_5)$$

$$f_6 = x_1 x_2 x_3 x_3 x_4 x_4 \vee$$

$$\vee (x_4 \vee x_5 \vee x_5)((x_4 \vee x_5)(x_4 x_5 \vee x_5) \vee x_5 x_5)(x_4 x_5 x_5 \vee x_5 \vee x_5)$$

$$f_7 = x_4 \vee x_5 \vee x_5 \vee x_5 x_5$$

$$f_8 = (x_4 \vee x_5)(x_4 x_5 \vee x_5) \vee x_5 \vee x_5$$

Після спрощення ці функції матимуть наступний вигляд:

$$f_1 = x_1 \vee x_2 \vee x_3$$

$$f_2 = (x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4$$

$$f_3 = x_3 \vee x_4$$

$$f_4 = x_1 x_2 x_3 \vee x_4(x_1 x_2 \vee x_3 \vee x_5)$$

$$f_5 = (x_1 \vee x_2)x_3 x_4 \vee x_5$$

$$f_6 = x_1 x_2 x_3 x_4 \vee x_5$$

$$f_7 = x_4 \vee x_5$$

$$f_8 = x_5$$

В якості прикладу розглянемо п'ять векторів стану системи: $V_1 = \langle 0, 1, 0, 1, 1 \rangle$, $V_2 = \langle 1, 1, 1, 1, 0 \rangle$, $V_3 = \langle 1, 0, 1, 1, 0 \rangle$, $V_4 = \langle 0, 0, 0, 1, 1 \rangle$ та $V_5 = \langle 1, 1, 0, 0, 1 \rangle$. Розрахуємо значення реберних функцій на кожному з векторів, результати занесемо до таблиці 5.4.

Таблиця 5.4

Значення реберних функцій моделі зваженої системи

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
V_1	1	1	1	1	1	1	1	1
V_2	1	1	1	1	1	1	1	0
V_3	1	1	1	1	1	0	1	0
V_4	0	1	1	1	1	1	1	1
V_5	1	1	0	0	1	1	1	1

Як бачимо, на векторах V_1 , V_2 та V_4 модель втрачає не більше одного ребра, що відповідає роботоздатному стану системи. Дійсно, в цих випадках сума ваг всіх її роботоздатних компонентів дорівнює відповідно 8, 7 та 7, що є не меншим за порогове значення. Вектори V_3 та V_5 відповідають нероботоздатному стану системи. Дійсно, сума ваг всіх її роботоздатних компонентів в обох цих випадках дорівнює 6, що менше порогового значення. Модель в цих випадках втрачає по 2 ребра, що призводить до втрати зв'язності її графом. Таким чином, результат експерименту підтверджує адекватність моделі.

5.2.2 Випадок систем, що містять компоненти з від'ємними вагами

В попередньому параграфі був запропонований метод побудови моделі зваженої системи, у якій значення ваг всіх компонентів – невід'ємні. Це припущення є цілком природним. У разі ж, якщо з якоїсь причини необхідно, щоб вага деяких компонентів була від'ємною (зокрема, можна припустити, що це може бути корисним при побудові моделей деяких ієрархічних систем), можна скористатися наступним підходом.

Нехай вага i -го компонента системи – від'ємна і дорівнює w_i . Цьому компоненту відповідає індикаторна змінна (або ж елемент вектору системи) x_i , що дорівнює 1, якщо він знаходиться в роботоздатному стані і 0, якщо він є несправним. Суму ваг всіх роботоздатних компонентів, окрім i -го позначимо як s . Нехай t -значення порогу, тоді в разі справності i -го компонента, система буде роботоздатною, якщо:

$$s + w_i \geq t. \quad (5.5)$$

Позначимо $t' = t - w_i$, тоді умова (5.5) матиме вигляд: $s \geq t'$. У випадку несправності i -го компонента, система буде роботоздатною, якщо $s \geq t$ або ж $s - w_i \geq t'$. Тобто, поведінка системи буде такою ж, як і в системи, поріг якої дорівнює t' , всі компоненти якої, окрім i -го є ідентичними відповідним компонентам вихідної системи, вага i -го компонента дорівнює « $-w_i$ », тобто є додатною, а його стану відповідає значення \bar{x}_i (тобто, його стан є інвертованим відносно стану відповідного компонента вихідної системи). Зазначимо, що

подібні перетворення можуть далі бути виконані для наступних компонентів системи, доки це буде необхідно.

Таким чином, якщо деякі компоненти системи мають від'ємні ваги, достатньо значення порога збільшити на суму абсолютних значень усіх цих ваг, змінити знак у кожній з них, а відповідні елементи вектору стану системи інвертувати. В результаті буде отримана «допоміжна» система, всі ваги у якої є невід'ємними.

5.2.3 Зважені системи з раціональними вагами

Вище були розглянуті системи, значення ваг всіх елементів яких є цілими. На жаль, в реальних системах це не завжди так. Розглянемо більш загальний випадок, коли значення ваг та порогу можуть бути представлені раціональними числами. В такому випадку $w_i = \frac{q_i}{r_i}$, для $i \in \{1, 2, \dots, n\}$, де q_i є цілим, а r_i – натуральним числом. Значення порогу матиме вигляд: $t = \frac{e}{d}$, де e – ціле, а d – натуральне число. Нехай z – найменший спільний знаменник значень всіх ваг та порогу, тобто $z = \text{НСК}(r_1, r_2, \dots, r_n, d)$. Помножимо значення всіх ваг та порогу на значення z та отримаємо систему з цілими значеннями ваг та порогу. Модель такої системи може бути побудована вищезапропонованим способом.

Відзначимо, що довільні дійсні числа можуть бути як завгодно точно наближені за допомогою раціональних чисел. В той же час, збільшення точності може привести до істотного ускладнення моделі, тому при вирішенні даної задачі потрібен компроміс між точністю та складністю розрахунків.

5.3. Ієрархічні системи

Великі системи зазвичай вирішують складні задачі, які часто можуть бути розбиті на декілька окремих підзадач. Кожна така підзадача, будучи в свою чергу також досить складною, може вирішуватися окремою групою процесорів – підсистемою. Складні підсистеми, в свою чергу, також можуть складатися з підсистем нижчого рівня і т.д.

Таким чином, система розбивається на кілька рівнів організації (ієрархії): від найбільш високого – система в цілому, до найнижчого – окремі процесори. Завдяки такому підходу можна значно спростити як побудову, так і обслуговування складних систем. Ці особливості позначаються і на побудові графо-логічних моделей таких систем, чому й будуть присвячені наступні параграфи роботи.

5.3.1 Побудова моделей ієрархічних систем

Розглянемо систему, що має декілька рівнів ієрархії. Кожна з підсистем такої системи складається з підсистем наступних рівнів ієрархії і/або безпосередньо процесорів та може бути стійкою до відмови деяких з них. Така система в загальному випадку є небазовою і, звісно, для неї може бути побудована відповідна модель, наприклад, за допомогою одного з відомих методів. Проте, така модель може бути досить складною, зокрема, оскільки загальна кількість процесорів в системі може бути дуже великою.

З іншого боку, знаючи стан компонентів підсистеми в потоці відмов може бути побудований вектор стану цієї підсистеми, а для визначення її поведінки може бути побудована відповідна GL-модель. Побудувавши моделі для всіх підсистем всіх рівнів, ми в кінці-кінців отримаємо модель, що описує стан системи в цілому.

Розрахунок значень для моделей підсистем не може виконуватися в довільному порядку, оскільки не для кожної з них одразу може бути визначений вектор стану. Спочатку необхідно визначити стан підсистем, що складаються лише з процесорів, для них вектори стану є визначеним. Далі стають можливими розрахунки для підсистем, що містять процесори і/або підсистеми, стани яких вже відомі. Таким чином, поступово будуть визначені стани всіх підсистем і системи в цілому.

Модель описаної вище ієрархічної системи є також ієрархічною і складається з підмоделей. Як компоненти вхідного вектору для підмоделі вищого рівня окрім значень, що відповідають станам процесорів

використовуються також значення, розраховані за допомогою підмоделей нижчого рівня ієрархії, які відповідають станам окремих підсистем. Розрахунки починаються з підмоделей найнижчого рівня ієрархії і закінчуються підмоделлю найвищого рівня. Розбиття за рівнями ієрархії системи і моделі в загальному випадку можуть різнитися між собою.

Таким чином, запропонований підхід базується на принципі розбиття складної задачі на сукупність простіших, що має ряд переваг. Окрім спрощення процесу побудови моделі системи в цілому, може спроститися також процес її модифікації в разі подальшого доопрацювання системи. Так, у разі зміни окремих її вузлів, досить виконати модифікацію тільки відповідних їм підмоделей.

5.3.2 Системи, що містять процесори лише на нижніх рівнях ієрархії

Розглянемо ієрархічну систему, кожна з підсистем якої містить лише підсистеми наступного рівня ієрархії і є стійкою до відмови деяких з них. Використовуючи в якості вхідних векторів значення, що відповідають станам підсистем наступного рівня, ми можемо побудувати моделі цих систем (базові або небазові). Кількість рівнів ієрархії в моделі і в системі буде однаковим [123], [124].

Приклад. Система складається з п'яти підсистем і стійка до відмови двох з них (рис. 5.3). Перша підсистема містить 11 процесорів і є стійкою до відмови 4 з них, друга підсистема містить 9 процесорів і стійка до відмови 3 з них, третя – 12 процесорів і стійка до відмови 5 з них, четверта – 7 процесорів і стійка до відмови 2 з них, а п'ята – 15 процесорів та стійка до відмови 4 з них. Тобто всі підсистеми в системі є базовими.

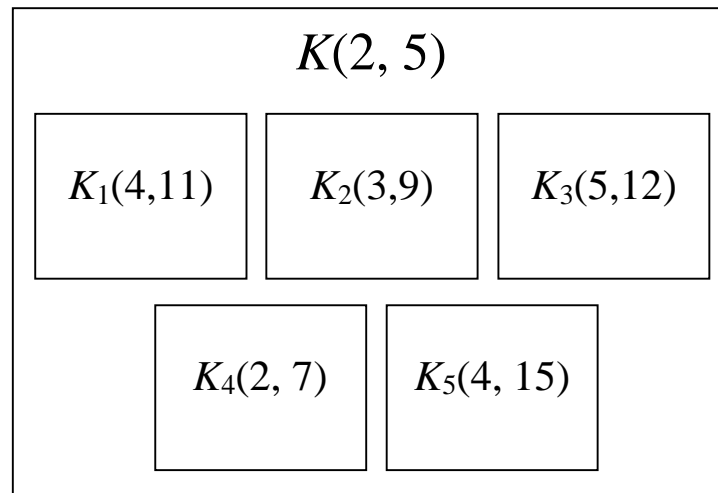


Рис. 5.3. Структура ієрархічної системи

Побудуємо модель даної системи. Вважатимемо, що станам процесорів першої підсистеми відповідають змінні x_1, x_2, \dots, x_{11} , другої – змінні $x_{12}, x_{13}, \dots, x_{20}$, третьої – $x_{21}, x_{22}, \dots, x_{32}$, четвертої – $x_{33}, x_{34}, \dots, x_{39}$, а п'ятої – $x_{40}, x_{41}, \dots, x_{54}$. Для початку побудуємо моделі підсистем нижнього рівня. Моделі будуватимемо відповідно до [86], тобто використовуватимемо МВР-моделі. Першій підсистемі відповідатиме модель $K_1(4, 11)$, що містить 8 реберних функцій; другій – $K_2(3, 9)$ з 7 реберними функціями, третій – $K_3(5, 12)$ із 8 реберних функцій, четвертій – $K_4(2, 7)$, яка містить 6 реберних функцій і п'ятій – $K_5(4, 15)$ з 12 реберних функцій. Реберні функції моделей позначатимемо f_i^j , де i – номер функції, а j – номер підсистеми.

Відповідно до станів підсистем, розрахованих за допомогою описаних вище моделей формується вектор стану системи: $\langle y_1, y_2, y_3, y_4, y_5 \rangle$. Системі відповідатиме модель $K(2, 5)$ з 4 реберними функціями, які позначимо f_k , де k – номер функції.

Таким чином, модель системи має два рівні ієрархії та складається з шести підмоделей. Вирази реберних функцій підмоделей та структура моделі наведені в Додатку В.

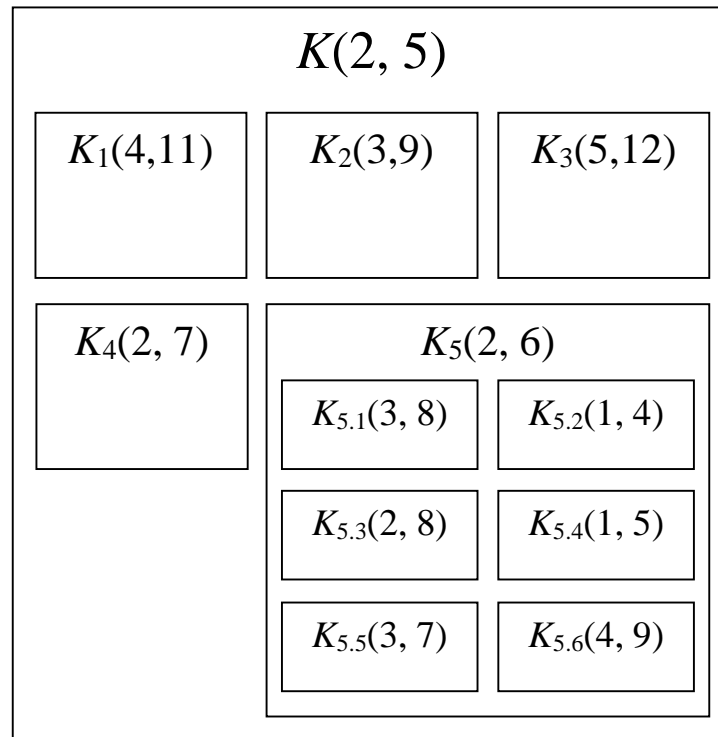


Рис. 5.4. Структура ієрархічної системи з додатковим рівнем ієрархії

Відзначимо, що для спрощення прикладів тут і далі передбачається, що всі підсистеми є базовими. В той же час, якщо деякі з підсистем є небазовими, відповідні їм моделі можуть бути модифіковані за допомогою тих чи інших відомих методів.

Кількість рівнів ієрархії не обов'язково має співпадати для всіх підсистем. Так, нехай п'ята підсистема розглянутої вище системи складається з шести підсистем і є стійкою до виходу з ладу двох з них (рис. 5.4). Ці підсистеми складаються з 8, 4, 8, 5, 7 і 9 процесорів, і стійкі до відмови відповідно 3, 1, 2, 1, 3 і 4 з них.

Побудуємо моделі підсистем нижнього рівня. Вважатимемо, що станам процесорів підсистеми 5.1 відповідають змінні $x_{40}, x_{41}, \dots, x_{47}$, підсистеми 5.2 – змінні $x_{48}, x_{49}, x_{50}, x_{51}$, підсистеми 5.3 – $x_{52}, x_{53}, \dots, x_{59}$, підсистеми 5.4 – $x_{60}, x_{61}, \dots, x_{64}$, підсистеми 5.5 – $x_{65}, x_{66}, \dots, x_{71}$ та підсистеми 5.6 – $x_{72}, x_{73}, \dots, x_{80}$.

Підсистемі 5.1 відповідатиме модель $K_{5.1}(3, 8)$ з 6 реберними функціями, підсистемі 5.2 – модель $K_{5.2}(1, 4)$ з 4 реберними функціями, підсистемі 5.3 – модель $K_{5.3}(2, 8)$ з 7 реберними функціями, підсистемі 5.4 – модель $K_{5.4}(1, 5)$ з 5

реберними функціями, підсистемі 5.5 – модель $K_{5.5}(3, 7)$ з 5 реберними функціями та підсистемі 5.6 – модель $K_{5.6}(4, 9)$ з 6 реберними функціями.

Відповідно до станів підсистем, розрахованих за допомогою вищеописаних моделей, формується вектор стану п'ятої підсистеми верхнього рівня: $\langle z_1, z_2, z_3, z_4, z_5, z_6 \rangle$. Цій підсистемі відповідатиме модель $K_5(2, 6)$ з 5 реберними функціями.

Модель системи має три рівні ієрархії та складається з дванадцяти підмоделей. Вирази реберних функцій підмоделей та структура моделі наведені в Додатку Г.

5.3.3 Системи, що містять процесори на всіх рівнях ієрархії

В попередньому параграфі були розглянуті системи, що містять процесори лише в підсистемах нижнього рівня ієрархії. У той же час, (під)система може містити не тільки підсистеми нижчих рівнів ієрархії, а й безпосередньо процесори. При цьому вона буде стійкою з одного боку до відмови деяких своїх підсистем, а з іншого – до відмови деяких своїх процесорів. Процесори в даному випадку можуть виконувати, наприклад, функції пост-обробки даних, отриманий від підсистем нижнього рівня.

Звісно, можливо об'єднати стани як підсистем, так і процесорів в один вектор стану, і побудувати, ймовірно, досить складну небазову модель системи. Як вже було сказано, процесори і підсистеми, скоріш за все, вирішуватимуть абсолютно різні задачі (зокрема і за масштабом), тобто неможливим буде замінити підсистему, що вийшла з ладу роботоздатним процесором та навпаки. В такому випадку корисним може виявитися розбиття моделі на дві підмоделі. Перша підмодель оцінюватиме можливість повноцінної роботи системи виходячи зі станів її процесорів, а друга – виходячи з станів її підсистем. Звісно, для того, щоб система була роботоздатною необхідно, щоб обидві підмоделі показували роботоздатний стан. Тобто ці результати можна об'єднати за допомогою кон'юнкції або ж моделі $K(0, 2)$.

В загальному випадку вектор стану системи може розбиватися й іншим чином. Наприклад, деяка група процесорів системи може мати певні особливості (наприклад, специфічний набір команд), що не дозволяють їм взяти на себе задачі інших процесорів, або ж навпаки бути замінені ними в разі виходу з ладу. Подібних груп може бути декілька. В такому випадку, корисною може виявитися побудова для них окремих підмоделей. Таким чином навіть неієрархічній системі може відповідати ієрархічна модель. В даному випадку можна припустити, що відповідне розбиття на підсистеми, хоч і не явно, але все ж має місце. Те саме стосується і підсистем, коли не всі з них можуть взяти на себе задачі одне одного. В таких випадках замість однієї складної моделі достатньо побудувати N більш простих підмоделей та об'єднати їх результати за допомогою кон'юнкції або моделі $K(0, N)$.

Іншими словами, процесори та/або підсистеми розбиваються на декілька груп, які розглядаються як підсистеми більш високого рівня. При цьому система вважається нестійкою до відмов цих підсистем, що відповідає моделі $K(0, N)$. Далі модель системи може бути побудована відповідно до описаного в попередньому параграфі способу [125], [126].

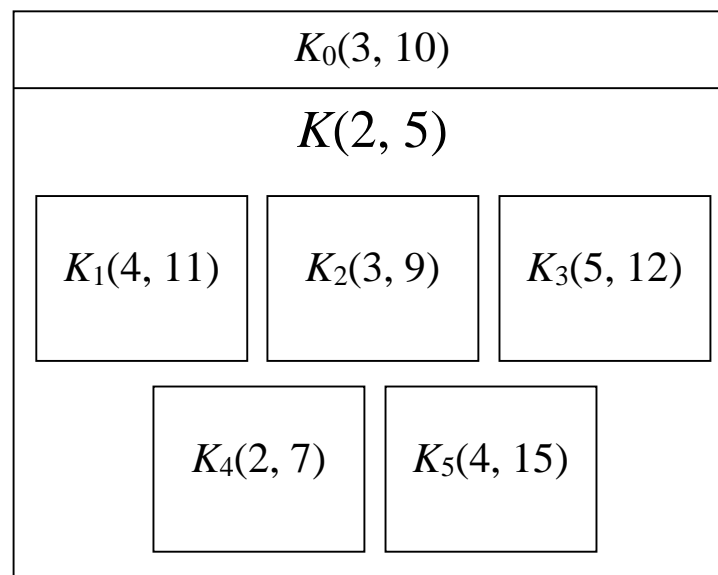


Рис. 5.5. Структура ієрархічної системи з процесорами
на всіх рівнях ієрархії

Розглянемо приклад. Нехай в системі, що була розглянута вище підсистема верхнього рівня містить 10 процесорів і є стійкою до відмови 3 з них (рис. 5.5). як і раніше всі підсистеми в системі є базовими.

Представимо підсистему верхнього рівня ієрархії у вигляді двох підсистем, які об'єднуюватимуться підсистемою $K(0, 2)$ (рис. 5.6).

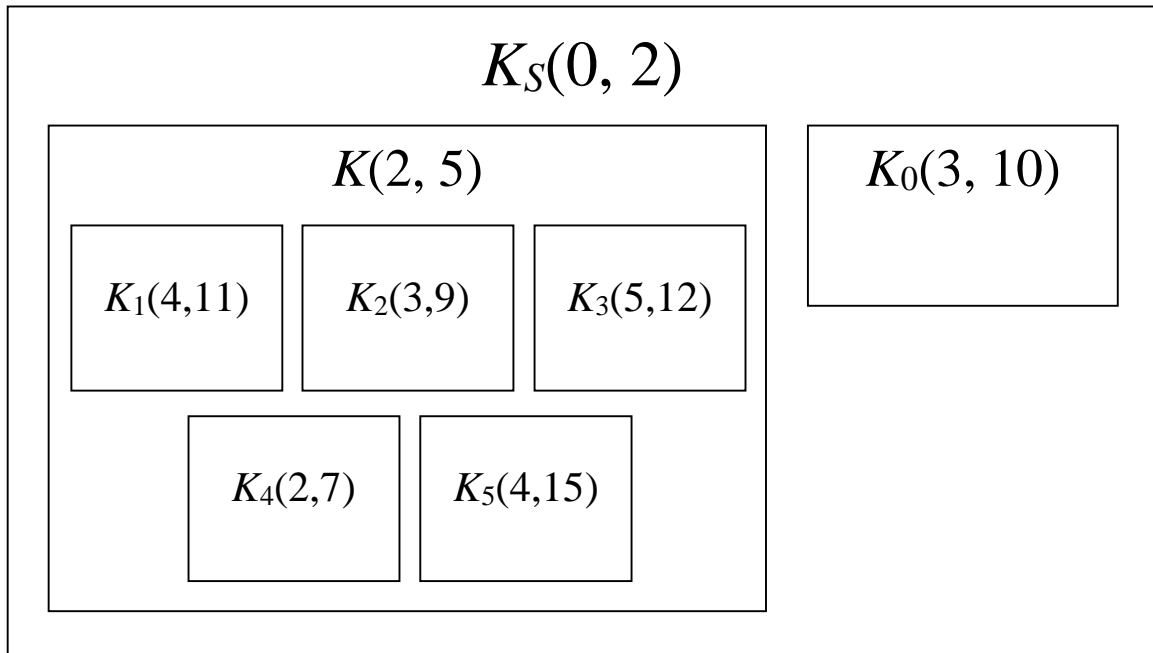


Рис. 5.6. Структура ієрархічної системи, підсистема верхнього рівня якої представлена у вигляді двох підсистем

Нехай результату, розрахованому за допомогою моделі $K(2, 5)$, побудованої для системи, що містить процесори тільки в підсистемах нижнього рівня, відповідає змінна w_1 . Також важатимемо, що станам процесорів підсистеми верхнього рівня ієрархії відповідають змінні $x_{55}, x_{56}, \dots, x_{64}$.

Для процесорів підсистеми верхнього рівня побудуємо модель $K_0(3, 10)$, що міститиме 8 реберних функцій. Результату, отриманому за допомогою цієї підмоделі відповідатиме змінна w_2 .

Вектор $\langle w_1, w_2 \rangle$ використовуватимемо як вхідний для моделі $K_S(0, 2)$. Проте, побудувати модель $K(0, n)$ методом, описаним в [86] – неможливо. В той же час, легко помітити, що дана модель буде еквівалентною моделі $K(1, 3)$ із вхідним вектором $\langle w_1, w_2, 0 \rangle$. Реберними функціями цієї моделі будуть:

$$F_1 = w_1$$

$$F_2 = w_2$$

$$F_3 = 0$$

Як бачимо, ребро графа моделі, що відповідає функції F_3 може бути виключене, оскільки його реберна функція завжди дорівнює нулю. Таким чином, модель системи має три рівні ієрархії та складається з восьми підмоделей. Вирази реберних функцій підмоделі $K_0(3, 10)$ та структура моделі наведені в Додатку Д.

Зазначимо, що модель $K(0, n)$ для довільного значення n може бути побудована аналогічним чином. В результаті буде отримана модель, що базується на лінійному графі, що містить n ребер (або циклічному, що містить $n + 1$ ребро і одна з реберних функцій якого завжди дорівнює «0») та реберні функції якого мають вигляд $F_i = v_i$, $i \in \{1, 2, \dots, n\}$, де v_i – відповідний елемент вхідного вектору моделі.

5.4. Системи з ковзним резервом

Для забезпечення відмовостійкості багатопроцесорної системи, її підсистеми будуються з тим чи іншим видом надлишковості, що дозволяє їм залишатися роботоздатними навіть при виході з ладу деяких своїх процесорів. В той же час, для забезпечення необхідного рівня надійності системи, даних заходів може виявитися недостатньо. Очевидним рішенням здається збільшення рівня надмірності в деяких (або всіх) підсистемах, тобто фактично додавання в них додаткових процесорів. Однак, цей підхід не завжди може бути використаний, внаслідок обмежень, що накладаються на систему (наприклад, обмеження вартості, маси, енергоспоживання тощо).

Іншим підходом є використання так званого ковзного резерву – набору процесорів, що є спільними для кількох підсистем, та кожний з яких в результаті реконфігурування можуть бути залучені до вирішення задач будь-якої з них. Хоч цей підхід, вочевидь, і вимагає більш складної організації системи, проте, в той же час кількість додаткових процесорів необхідних для забезпечення заданого рівня надійності може виявитися меншою.

В наступних параграфах будуть запропоновані підходи, що дозволяють будувати графо-логічні моделі для деяких випадків організації таких систем. Застосування цих підходів може спростити побудову моделей в порівнянні з методами побудови небазових моделей, що застосовуються в загальних випадках.

5.4.1 Модель системи з ковзним резервом

Нехай система складається з M підсистем, кожна з яких є базовою, містить n_i процесорів і є стійкою до виходу з ладу m_i з них, де $i = 1, 2, \dots, M$ – номер підсистеми. Також система містить r процесорів, спільних для всіх підсистем, які формують ковзний резерв. Для роботоздатності системи необхідно, щоб роботоздатними були всі її підсистеми.

Розглянемо i -у підсистему. Вона складається з n_i процесорів і є стійкою до відмови m_i з них. Нехай l_i – кількість реальних відмов процесорів в даній підсистемі. Очевидно, якщо $l_i > m_i$, то для відновлення її роботоздатності потрібно залучити $l_i - m_i$ процесорів ковзного резерву. Нехай λ_i – кількість процесорів ковзного резерву, необхідних для відновлення роботоздатності i -ї підсистеми. тобто:

$$\lambda_i = \begin{cases} 0, & \text{при } l_i \leq m_i \\ l_i - m_i, & \text{при } l_i > m_i \end{cases} \quad (5.6)$$

Відзначимо, що відповідно до [95], модель $K(m_i + 1, n_i)$, що прийматиме на вхід вектор стану i -ї підсистеми (вектор, кожний елемент якого відповідає стану в потоці відмов одного з процесорів підсистеми), втрачатиме $\varphi(m_i + 1, l_i)$ ребер, де

$$\varphi(m_i + 1, l_i) = \begin{cases} 0, & \text{при } l_i < m_i + 1 \\ l_i - m_i, & \text{при } l_i \geq m_i + 1 \end{cases} \quad (5.7)$$

Відповідно до [94], ця модель матиме $\theta_i = n_i - m_i + 1$ реберних функцій, які позначимо $f_1^i, f_2^i, \dots, f_{\theta_i}^i$. Порівнявши вирази (5.6) та (5.7) та враховуючи, що m_i та l_i є цілими числами, можемо зробити висновок, що $\lambda_i = \varphi(m_i + 1, l_i)$, тобто вказана

вище модель на векторах з l_i втрачатиме рівно λ_i ребер, або, що є тим самим, серед значень функцій f_j^i ($j = 1, 2, \dots, \theta_i$) буде рівно λ_i нулів.

Нехай

$$\Lambda = \sum_{i=1}^M \lambda_i. \quad (5.8)$$

Тобто Λ – кількість процесорів ковзного резерву, необхідних для відновлення роботоздатності всіх підсистем, або, іншими словами, всієї системи. В ковзному резерві є всього r процесорів, нехай s з них вийшли з ладу, тобто роботоздатних залишилося $r - s$. Отже, система буде роботоздатною тільки в разі, якщо $\Lambda \leq r - s$, що еквівалентно умові:

$$s + \Lambda \leq r. \quad (5.9)$$

Нехай вектор v містить значення, що відповідають станам всіх процесорів ковзного резерву, а також значення всіх функцій f_j^i ($i = 1, 2, \dots, M, j = 1, 2, \dots, \theta_i$). Він міститиме $s + \sum_{i=1}^M \varphi(m_i + 1, l_i) = s + \sum_{i=1}^M \lambda_i = s + \Lambda$ нулів, а довжина його дорівнюватиме

$$\theta = r + \sum_{i=1}^M \theta_i = r + \sum_{i=1}^M (n_i - m_i + 1) = r + M + \sum_{i=1}^M n_i - \sum_{j=1}^M m_j \quad (5.10)$$

Легко помітити, що відповідно до (5.8), поведінці системи відповідатиме модель $K(r, \Theta)$, що приймає на вхід вектор v .

Таким чином, для побудови моделі, що складається з M підсистем, кожна з яких є базовою, містить n_i процесорів та є стійкою до відмови m_i з них, де $i = 1, 2, \dots, M$ – номер підсистеми; а також яка має ковзний резерв, що складається з r процесорів достатньо [127], [128]:

1. Для кожної з підсистем побудувати реберні функції $f_1^i, f_2^i, \dots, f_{\theta_i}^i$ моделі $K(m_i+1, n_i)$, яка приймає на вхід вектор стану підсистеми (відповідну частину вектору стану системи, що відповідає процесорам підсистеми), де $\theta_i = n_i - m_i + 1$, а $i = 1, 2, \dots, M$ – номер підсистеми.

2. Побудувати вектор v , що містить значення елементів вектору системи, які відповідають станам процесорів ковзного резерву, а також значення всіх функцій f_j^i (для $i = 1, 2, \dots, M, j = 1, 2, \dots, \theta_i$).
3. Побудувати модель $K(r, \Theta)$, що прийматиме на вхід вектор v , котра і буде моделлю системи, де Θ – довжина вектору v , що може бути розрахована відповідно до (5.10).

Розглянемо приклад.

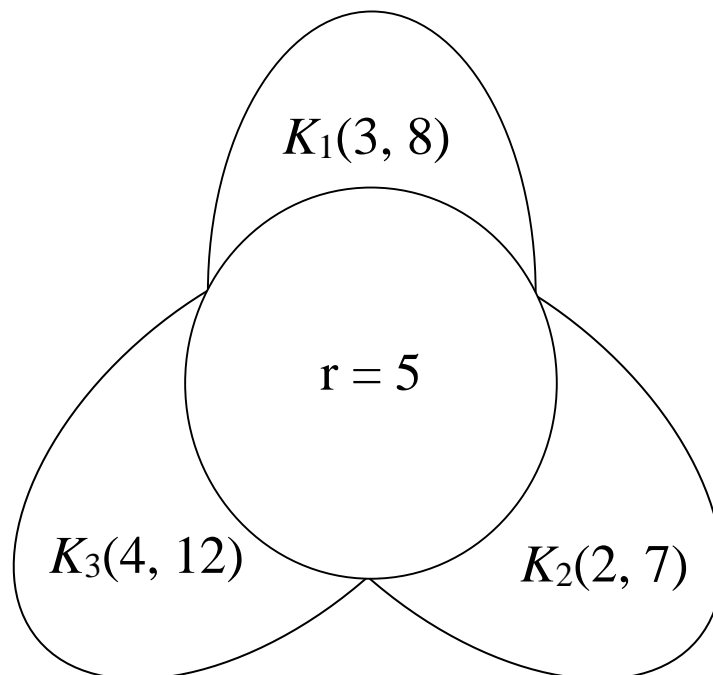


Рис. 5.7. Структура системи з ковзним резервом

Нехай система складається з трьох підсистем, які містять 8, 7 та 12 процесорів та є стійкими до відмови відповідно не більш ніж 3, 2 та 4 з них. Ковзний резерв системи складається з 5 процесорів (рис. 5.7).

Вважатимемо, що станам процесорів першої підсистеми відповідають змінні x_1, x_2, \dots, x_8 , другої – змінні $x_9, x_{10}, \dots, x_{15}$, третьої – змінні $x_{16}, x_{17}, \dots, x_{27}$, а процесорам ковзного резерву – змінні $x_{28}, x_{29}, \dots, x_{32}$.

Для першої підсистеми побудуємо модель $K_1(4, 8)$, що міститиме 5 реберних функцій, для другої – модель $K_2(3, 7)$ з 5 реберними функціями, а для третьої – модель $K_3(15, 2)$ із 8 реберними функціями.

Сформуємо вектор v , що матиме вигляд:

$$v = \langle x_{28}, x_{29}, x_{30}, x_{31}, x_{32}, f_1^1, f_2^1, f_3^1, f_4^1, f_5^1, f_1^2, f_2^2, f_3^2, f_4^2, f_5^2, f_1^3, f_2^3, f_3^3, f_4^3, f_5^3, f_6^3, f_7^3, f_8^3 \rangle.$$

Системі відповідатиме модель $K(5, 23)$ з 19 реберними функціями, що приймає на вхід вектор v .

Вирази реберних функцій та структура моделі наведені в Додатку Е.

5.4.2 Модель систем з «асиметричним» ковзним резервом

В попередньому параграфі припускалося, що всі підсистеми мають однаковий пріоритет у використанні ресурсів ковзного резерву. У реальній системі це може бути не так. Наприклад, в силу тих чи інших причин, деяка група процесорів може використовуватися для забезпечення відмовостійкості в першу чергу однієї підсистеми, а вже потім, в разі, якщо надлишкових ресурсів достатньо, надавати підтримку і іншим підсистемам.

З іншого боку цю ситуацію можна трактувати так, що підсистема використовує наявні у неї надлишкові ресурси для відновлення роботоздатності інших підсистем, у яких власних ресурсів вже недостатньо, приймаючи на себе частину їх задач. Таким чином, на відміну від розглянутого раніше випадку, ковзний резерв використовується «асиметрично».

Нехай підсистема містить n процесорів і є стійкою до відмови будь-яких m з них. Припустимо, що l її процесорів вийшло з ладу ($l < m$). Нехай підсистема була реконфігурована так, що k ($k \leq m - l$) її роботоздатних процесорів виконують сторонні задачі (наприклад, задіяні у відновленні роботоздатності інших підсистем). З точки зору даної підсистеми, ці процесори можна також вважати неробочими, проте, вона не втратить своєї роботоздатності, оскільки $l + k \leq m$.

Відзначимо, що для виконання сторонніх задач не обов'язково мають виділятися окремі процесори. Вони можуть виконуватися і разом з власними задачами підсистеми. Проте, для зручності вважатимемо їх окремими "віртуальними" процесорами еквівалентної продуктивності.

Для подальшої побудови моделі всієї системи корисним буде отримати вектор, що містить рівно стільки одиниць, скільки процесорів підсистеми може бути задіяно в вирішенні сторонніх задач. Для вищезгаданої підсистеми це число може бути розраховане за формулою:

$$\mu(m, l) = \begin{cases} 0, & \text{при } l \geq m \\ m - l, & \text{при } l < m \end{cases} \quad (5.11)$$

Дану задачу дозволяє вирішити так звана двоїста модель, описана в наступному параграфі.

5.4.3 Двоїста модель

Як вже було сказано, на векторах із k нулями модель $K(m, n)$, побудована відповідно до [86] втрачає рівно $\varphi(m, k)$ ребер [95], тобто рівно стільки її реберних функцій приймають нульове значення, де

$$\varphi(m, k) = \begin{cases} 0, & \text{при } k < m \\ k - m + 1, & \text{при } k \geq m \end{cases} \quad (5.12)$$

Нехай модель $K'(m, n)$ містить реберні функції, двоїсті по відношенню до функцій моделі $K(m, n)$. Таку модель називатимемо двоїстою моделлю. Використовуючи правила де Моргана легко показати, що вирази реберних функцій двоїстої моделі можуть бути отримані з виразів реберних функцій базової моделі шляхом заміни в них кон'юнкцій на диз'юнкції, а диз'юнкцій на кон'юнкції. Очевидно, що у двоїстої моделі $K'(m, n)$ на векторах з k одиницями рівно $\varphi(m, k)$ функцій прийматимуть значення, рівне одиниці.

Нехай в базовій m -відмовостійкій системі, що складається з n процесорів, вийшло з ладу l з них. Відповідно, кількість роботоздатних процесорів дорівнює $n - l$. У двоїстої моделі $K'(n - m + 1, n)$, що приймає на вхід вектор стану системи рівно $\varphi(n - m + 1, n - l)$ функцій прийме значення, що дорівнює одиниці. Згідно з (5.11) та (5.12) та враховуючи що n , m та l є цілими числами, можна зробити висновок, що $\varphi(n - m + 1, n - l) = \mu(m, l)$. Таким чином, вектор, що складається зі значень реберних функцій моделі $K'(n - m + 1, n)$ міститиме рівно $\mu(m, l)$ одиниць. Крім того, відповідно до [94] такий вектор матиме довжину, рівну m .

5.4.4 Модель пари підсистем «донор»-«реципієнт»

Розглянемо ситуацію, коли «надлишкові» процесори однієї підсистеми можуть бути використані для відновлення роботоздатності іншої підсистеми. Тобто можна сказати, що перша підсистема є «донором», а друга – «реципієнтом».

Нехай перша з підсистем містить n_1 процесорів і є стійкою до відмови будь-яких m_1 з них, а друга – n_2 процесорів і є стійкою до відмови m_2 з них. При цьому «надлишкові» ресурси першої підсистеми можуть бути використані для вирішення частини задач другої підсистеми, в разі, якщо її власних ресурсів недостатньо.

Для першої підсистеми достатньо побудувати звичайну базову модель $K(m_1, n_1)$, що і відображатиме її поведінку в потоці відмов. Крім того, побудуємо двоїсту модель $K'(n_1 - m_1 + 1, n_1)$, що приймає на вхід вектор стану першої підсистеми, і за допомогою якої буде отримано допоміжний вектор довжини m_1 . Об'єднавши цей вектор (шляхом конкатенації) з вектором стану другої підсистеми, ми отримаємо вектор (котрий позначимо v), що містить стільки одиниць, скільки процесорів може бути використано для вирішення задач другої підсистеми. Його довжина дорівнюватиме $m_1 + n_2$.

Згадаємо, що друга підсистема є стійкою до виходу з ладу не більш ніж m_2 з n_2 своїх процесорів, тобто, для вирішення її задач необхідно не менш ніж $n_2 - m_2$ роботоздатних процесорів. Враховуючи процесори надані першою підсистемою (яких може бути до m_1 штук) можна сказати, що система є стійкою до відмови не більш ніж $m_1 + m_2$ з $m_1 + n_2$ процесорів. Таким чином, другий підсистемі відповідатиме модель $K(m_1 + m_2, m_1 + n_2)$, що отримує на вхід вищеописаний вектор v .

Розглянемо приклад. Нехай система складається з двох підсистем. Перша підсистема містить 10 процесорів і є стійкою до виходу з ладу не більш ніж 5 з них, а друга підсистема містить 7 процесорів і стійкою до відмови не більш ніж 3 з них. При цьому «надлишкові» процесори першої підсистеми можуть

використовуватися для відновлення роботоздатності другої. Структура цієї системи представлена на рис. 5.8.

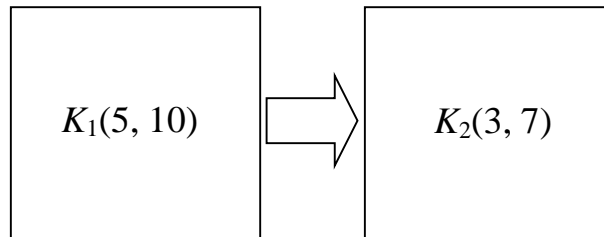


Рис. 5.8. Структура системи «донор»-«реципієнт»

Побудуємо моделі кожної з підсистем. Нехай станам процесорів першої підсистеми відповідають змінні x_1, x_2, \dots, x_{10} , а другої – змінні $x_{11}, x_{12}, \dots, x_{17}$.

Першій підсистемі відповідатиме модель $K_1(5, 10)$ з 6 реберними функціями, які позначимо $f_1^1, f_2^1, \dots, f_6^1$. Також побудуємо допоміжну модель $K_1'(6, 10)$ із 5 реберними функціями, які позначимо $g_1^1, g_2^1, \dots, g_5^1$.

Сформуємо допоміжний вектор v_1 , що міститиме значення елементів вектору стану системи, котрі відповідають станам процесорів другої підсистеми, а також значення реберних функцій допоміжної моделі:

$$v_1 = \langle x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, g_1^1, g_2^1, g_3^1, g_4^1, g_5^1 \rangle.$$

Другій підсистемі відповідатиме модель $K_2(8, 12)$, яка приймає v_1 в якості вхідного вектору. Ця модель матиме 5 реберних функцій.

Вирази реберних функцій та структура моделі наведені в Додатку Ж.

5.4.5 Модель «симетричної» пари підсистем «донор»-«реципієнт»

В попередньому параграфі було розглянуто випадок, коли одна підсистема може сприяти у відновленні роботоздатності другої підсистеми. В свою чергу, можлива така організація системи, що і друга підсистема може також сприяти в відновленні роботоздатності першої. Тобто, в залежності від ситуації кожна з підсистем може стати як «донором», так і «реципієнтом».

В такому випадку, для побудови моделі першої підсистеми достатньо за аналогією з попереднім параграфом, побудувати двоїсту модель $K'(n_2 - m_2 + 1, n_2)$, що приймає на вхід вектор стану другої підсистеми. За

допомогою даної моделі буде отримано допоміжний вектор, довжиною m_2 . Об'єднавши цей вектор (шляхом конкатенації) з вектором стану першої підсистеми, ми отримаємо вектор, що містить стільки одиниць, скільки процесорів може бути використано для вирішення задач першої підсистеми. Його довжина дорівнюватиме $m_2 + n_1$. Першій підсистемі відповідатиме модель $K(m_2 + m_1, m_2 + n_1)$, що приймає на вхід цей вектор. Модель другої підсистеми будується так само, як і в попередньому випадку.

Відзначимо, що для обох двоїстих моделей використовуються вектори стану, що містять елементи, відповідні процесорам, які містяться тільки в тій чи іншій підсистемі. Дійсно, ці моделі відображають кількість «надлишкових» ресурсів безпосередньо у підсистем, без урахування ресурсів, що можуть бути отримані від інших підсистем.

Також відзначимо, що в простому випадку (зокрема, який розглянуто в даному параграфі) поведінка такої системи відповідає базовій моделі $K(m_1 + m_2, n_1 + n_2)$. Проте, запропонований підхід може бути використаний і для більш складних випадків, що не можуть бути зведені до базових моделей, зокрема, коли систем більше ніж дві та взаємозв'язки між ними не є тривіальними: деякі з підсистем не можуть виступати «донорами» для інших або ж кількість процесорів, що можуть брати участь у відновленні роботоздатності підсистеми є обмеженою (що розглянуто в наступному параграфі). Також запропонована модель може бути корисною у випадках, коли необхідними є окремі моделі для кожної з підсистем.

В якості прикладу розглянемо систему подібну розглянутій в попередньому параграфі, але в якій друга підсистема також може використовувати свої «надлишкові» ресурси для відновлення роботоздатності першої підсистеми. Структура системи представлена на рис. 5.9.

Відзначимо, що модель другої підсистеми буде точно такою ж, як і в попередньому прикладі, тому обмежимося побудовою моделі першої підсистеми.

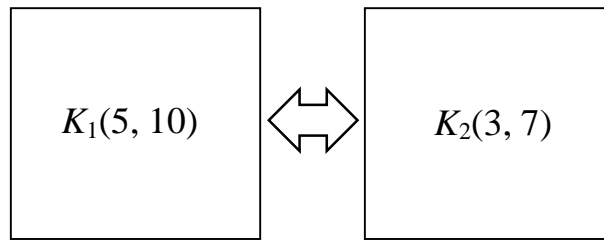


Рис. 5.9. Структура «симетричної» пари підсистем «донор»-«реципієнт»

Побудуємо допоміжну двоїсту модель $K_2'(5, 7)$, що міститиме 3 реберні функції: g_1^2, g_2^2, g_3^2 . Сформуємо допоміжний вектор v_2 , що міститиме значення елементів вектору стану системи, котрі відповідають станам процесорів першої підсистеми, а також значення реберних функцій допоміжної моделі:

$$v_2 = \langle x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, g_1^2, g_2^2, g_3^2 \rangle.$$

Першій підсистемі відповідатиме модель $K_1(8, 13)$, що приймає вектор v_2 в якості вхідного вектору. Ця модель міститиме 6 реберних функцій.

Вирази реберних функцій та структура моделі наведені в Додатку И.

5.4.6 Обмеження кількості «донорських» процесорів

В реальній системі можливості використання «надлишкових» ресурсів однієї підсистеми для відновлення роботоздатності іншою можуть бути обмежені, наприклад, пропускнуою здатністю шин. В такому випадку, з доступних «надлишкових» процесорів може бути використано не більше ніж певна кількість. Ця особливість також має бути відображена в моделі.

Нехай ϵ вектор, довжиною n , що містить l одиниць. Потрібно отримати вектор, довжиною h , $h < n$, що містить $k = \min(l, h)$ одиниць. Очевидно, що розглянутий вище вектор містить $n - l$ нулів. Модель $K(n - h + 1, n)$, що прийматиме його на вхід втратить $\varphi(n - h + 1, n - l)$ ребер, тобто рівно така кількість її реберних функцій приймуть нульове значення. Загальна кількість реберних функцій даної моделі відповідно до [94]:

$$n - (n - h + 1) + 1 = h. \quad (5.13)$$

Таким чином, вектор, що складається зі значень реберних функцій даної моделі матиме довжину h , міститиме $\varphi(n - h + 1, n - l)$ нулів і, відповідно, $\rho(l) = h - \varphi(n - h + 1, n - l)$ одиниць. Відповідно до (5.12):

$$\rho(l) = \begin{cases} h, & \text{при } l > h - 1 \\ l, & \text{при } l \leq h - 1 \end{cases} \quad (5.14)$$

Враховуючи що l та h – цілі числа, $\rho(l) = \min(l, h)$.

Таким чином, у випадку, якщо кількість процесорів підсистеми, які можуть використовуватися для вирішення сторонніх задач обмежена деяким числом h , достатньо допоміжний вектор, отриманий за допомогою двоїстої моделі подати на модель $K(n - h + 1, n)$, де n – довжина даного вектора. Побудований на основі значень реберних функцій даної моделі вектор і міститиме кількість одиниць, що дорівнює кількості «надлишкових» процесорів підсистеми, з урахуванням вищевказаного обмеження. Даний вектор може бути використаний таким же чином, як в попередніх випадках використовувався вектор, отриманий за допомогою допоміжної моделі.

Для прикладу розглянемо систему, подібну розглянутій в параграфі 5.4.4, але в якій кількість «надлишкових» процесорів першої підсистеми, що можуть бути використані для відновлення роботоздатності другої підсистеми обмежена трьома. Структура системи представлена на рис. 5.10.

Відзначимо, що в даному випадку зміниться тільки модель другої підсистеми. При цьому допоміжна двоїста модель залишиться такою ж як і в попередньому прикладі.

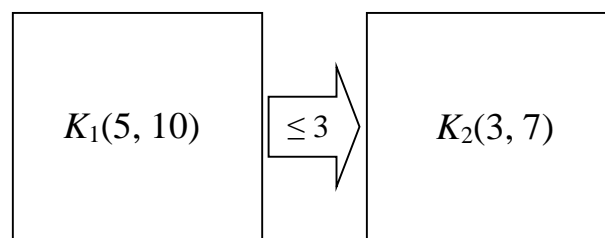


Рис. 5.10. Структура системи «донор»-«реципієнт» з обмеженнями

Побудуємо допоміжну модель, що обмежує кількість використовуваних «надлишкових» процесорів. Ця модель $K_1''(3, 5)$, що прийматиме на вхід вектор v_1 міститиме три реберні функції: h_1^1 , h_2^1 та h_3^1 .

Сформуємо допоміжний вектор w_1 , що міститиме значення елементів вектору стану системи, котрі відповідають станам процесорів другої підсистеми, а також значення реберних функцій сформованої вище допоміжної моделі:

$$w_1 = \langle x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, h_1^1, h_2^1, h_3^1 \rangle.$$

Другій підсистемі відповідатиме модель $K_2(6, 10)$, яка приймає w_1 в якості вхідного вектору. Ця модель матиме 5 реберних функцій.

Вирази реберних функцій та структура моделі надані в Додатку К.

5.4.7 Загальний випадок систем з ковзним резервом

В попередніх параграфах були розглянуті лише найбільш прості випадки взаємозв'язку між підсистемами. Реальні системи можуть бути організовані значно складніше: наприклад, «надлишкові» процесори можуть надаватися одночасно декількома підсистемами і/або бути використані для відновлення роботоздатності одночасно декількох підсистем. В такому випадку для побудови моделі розглянуті підходи можуть бути комбіновані.

Розглянемо приклад. Система складається з 5 підсистем. Дві з них, що містять 7 і 9 процесорів є стійкими до відмови відповідно 3 і 4 з них, а також дозволяють використовувати свої «надлишкові» ресурси для відновлення роботоздатності інших трьох підсистем, що містять 7, 8 та 5 процесорів та є стійкими до відмови відповідно 2, 3 і 2 з них. Структура системи представлена на рис. 5.11.

Нехай станам першої підсистеми відповідають змінні x_1, x_2, \dots, x_7 , другої — x_8, x_9, \dots, x_{17} , третьої — $x_{18}, x_{19}, \dots, x_{23}$, четвертої — $x_{24}, x_{25}, \dots, x_{31}$, а п'ятій — $x_{32}, x_{33}, \dots, x_{36}$.

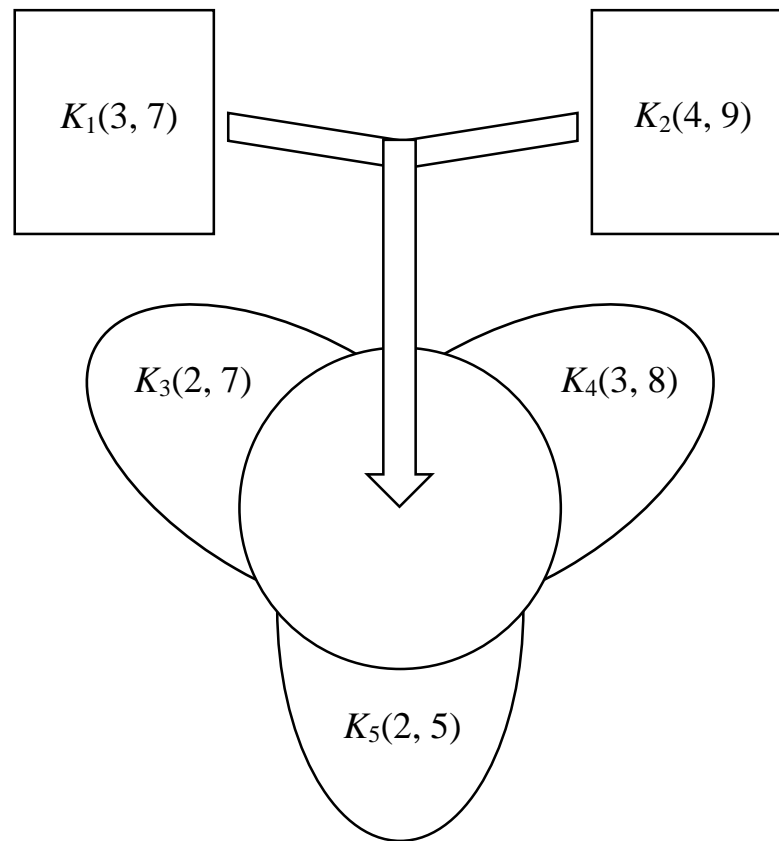


Рис. 5.11. Структура складної системи з ковзним резервом

Першій підсистемі відповідатиме модель $K_1(3, 7)$ із 5 реберними функціями. Другій підсистемі відповідатиме модель $K_2(4, 9)$, що містить 6 реберних функцій.

Окрім того побудуємо допоміжні двоїсті моделі $K_1'(5, 7)$ з реберними функціями g_1^1, g_2^1, g_3^1 та $K_2'(6, 9)$ із реберними функціями g_1^2, g_2^2, g_3^2 і g_4^2 .

Сформуємо вектор стану «надлишкових» процесорів r , що складатиметься зі значень реберних функцій обох допоміжних моделей:

$$r = \langle g_1^1, g_2^1, g_3^1, g_1^2, g_2^2, g_3^2, g_4^2 \rangle.$$

Наявні «надлишкові» процесори використовуватимуться як звичайний ковзний резерв для відновлення роботоздатності 3-ї, 4-ї та 5-ї підсистем (аналогічно до викладеного в параграфу 5.4.1). Для побудови моделі цієї частини системи відповідно до алгоритму побудуємо допоміжні моделі для кожної з підсистем.

Допоміжна модель третьої підсистеми $K_3(3, 7)$ міститиме 5 реберних функцій: $f_1^3, f_2^3, \dots, f_5^3$. Допоміжна модель четвертої підсистеми $K_4(4, 8)$ також

міститиме 5 реберних функцій: $f_1^4, f_2^4, \dots, f_5^4$. Допоміжна модель п'ятої підсистеми $K_5(3, 5)$ міститиме 3 реберні функції: f_1^5, f_2^5, f_3^5 .

Сформуємо вхідний вектор v , що міститиме значення реберних функцій допоміжних моделей підсистем 3, 4, 5 та двоїстих допоміжних моделей підсистем 1 та 2 (значення елементів вищеописаного вектору r):

$$v = \langle g_1^1, g_2^1, g_3^1, g_1^2, g_2^2, g_3^2, g_4^2, f_1^3, f_2^3, f_3^3, f_4^3, f_5^3, f_1^4, f_2^4, f_3^4, f_4^4, f_5^4, f_1^5, f_2^5, f_3^5 \rangle.$$

Поведінці підсистем 3, 4 та 5 відповідатиме модель $K(7, 20)$ із 14 реберними функціями.

Вирази реберних функцій та структура моделі надані в Додатку Л.

Відзначимо, що аналогічний підхід міг бути застосований і для прикладів, розглянутих в параграфів 5.4.4-5.4.6. В такому випадку для кожної з підсистем-«реципієнтів» було б побудовано по дві (але більш прості) моделі, одна з яких враховувала процесори власне підсистеми, а інша – ресурси, отримані від «донорів».

Для прикладу побудуємо модель системи, розглянутої в параграфі 5.4.6. Для другої підсистеми, що містить 7 процесорів та є стійкою до виходу з ладу 3 з них побудуємо допоміжну модель $K_2''(4, 7)$ з реберними функціями e_1^2, e_2^2, e_3^2 та e_4^2 .

Далі сформуємо вхідний вектор $\langle e_1^2, e_2^2, e_3^2, e_4^2, h_1^1, h_2^1, h_3^1 \rangle$ для моделі $K_2(3, 7)$, де 3 – максимально можлива кількість процесорів, що можуть бути отримані від «донора», котрі тут інтерпретуються як процесори ковзного резерву. Дана модель і відповідатиме поведінці другої підсистеми в потоці відмов.

Вирази реберних функцій та структура моделі надані в Додатку М.

Висновки до розділу 5

Системи та/або їх компоненти можуть перебувати в більш ніж двох станах роботоздатності. Поведінка таких систем в потоці відмов може моделюватися GL-моделями після їх деякого розширення. В таких моделях в якості реберних

функцій використовуються функції багатозначної логіки, а визначення стану системи зводиться до задачі пошуку кістяка максимальної ваги в графі. Запропоновано модифікації алгоритмів Прима та Крускала, що дозволяють визначати стан системи в потоці відмов.

В простих випадках GL-моделі таких систем можуть бути отримані з базових шляхом заміни у виразах їх реберних функцій операцій кон'юнкції та диз'юнкції відповідно на операції взяття мінімуму та максимуму.

Різні компоненти (процесори) системи можуть відігравати різну роль в роботі системи: зокрема, вихід з ладу деяких з них може в більшій мірі знижувати її роботоздатність. В цьому випадку кожному процесору може бути співставлена деяка вага, а система є роботоздатною доти, доки вага її роботоздатних компонентів є не меншою за деяке порогове значення. Для побудови GL-моделі такої системи вектор стану модифікується так, що кожний з його елементів може входити до нього по кілька разів (це число має бути кратним значенню ваги відповідного процесора). В разі, якщо ваги деяких компонентів є від'ємними, побудувати модель можна перейшовши спочатку до допоміжної системи, ваги всіх компонентів якої є додатними. Запропоновано також рішення, що дозволяє будувати моделі систем з раціональними вагами, які, в свою чергу, можуть з заданою точністю наближати загальний випадок систем із довільними (дійсними) значеннями ваг.

Великі системи, що вирішують складні задачі часто можуть бути розділені на окремі підсистеми, кожна з яких вирішує свою підзадачу. Таким чином, система може також бути поділена на декілька рівнів ієрархії. Модель такої системи може бути корисно розглядати як ієрархію більш простих моделей. При цьому передбачається, що моделі більш високих рівнів ієрархії в якості компонентів векторів станів можуть використовувати результати, отримані за допомогою моделей нижчих рівнів. Залежно від особливостей системи кількість рівнів ієрархії, а також об'єктів на кожному з них для моделі і системи можуть відрізнятися.

Окрім того, запропоновано ряд підходів для побудови моделей систем, підсистеми яких можуть використовувати спільні процесори для відновлення роботоздатності. Запропоновані підходи можуть бути комбіновані для побудови моделей більш складних систем.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

В роботі запропоновано методи перетворення базових МВР-моделей шляхом модифікації їх реберних функцій з метою адекватного відображення поведінки небазових відмовостійких багатопроцесорних систем керування складними об'єктами в потоці відмов, а також метод визначення вектору стану таких систем із результатів взаємного тестування пар їх процесорів.

Основні результати роботи полягають в наступному.

1. Проведено аналіз відомих методів розрахунку показників надійності відмовостійких багатопроцесорних систем. Встановлено, що при застосуванні до ВБС керування об'єктами критичного застосування, які відрізняються складною поведінкою в потоці відмов, найбільш ефективними є методи, що базуються на проведенні статистичних експериментів з моделями.

2. Запропоновано метод модифікації реберних функцій МВР-моделі. Сформульовані вирази, що дозволяють теоретично передбачити множину та кількість векторів, блокованих в результаті модифікації однієї реберної функції моделі в залежності від виразу цієї функції (а точніше значень параметрів, використаних для її створення згідно з алгоритмом побудови МВР-моделей) та виду модифікації. З одного боку метод дозволяє будувати GL-моделі для ВБС, що по різному ведуть себе при появі відмов однієї кратності, а з іншого боку зменшує час виконання одного експерименту з моделлю (на 5-18 відсотків для різних ВБС).

3. Виконано аналіз поведінки моделі при модифікації кількох реберних функцій. Виявлено, що в цьому випадку можуть бути блоковані вектори, кількість нулів в яких перевищує значення відмовостійкості відповідної базової системи але не більш ніж на число функцій, що були модифіковані. З одного боку це визначає мінімальну кількість реберних функцій, які необхідно модифікувати для блокування тих чи інших векторів стану системи, а з іншого вказує на можливість виникнення іноді небажаного побічного ефекту у вигляді блокування векторів із підвищеною кратністю відмов.

4. Сформульовано критерії вибору пар реберних функцій МВР-моделі, модифікація яких не призводить до появи вищезгаданого побічного ефекту тобто, дозволяє блокувати лише вектори з кількістю нулів рівно на один більшою, ніж значення відмовостійкості відповідної базової системи. Даний сценарій модифікації вбачається найбільш імовірним з практичної точки зору. Для випадку одночасної модифікації більш, ніж двох реберних функцій, критерії, хоч і не наведені в роботі (зокрема, через їх значну громіздкість), проте можуть бути отримані аналогічним чином.

5. Розроблено аналітичний апарат для аналізу результатів взаємного тестування пар процесорів ВБС з довільною топологією зв'язків та формального отримання достовірного вектору стану системи (згідно з моделлю Препарата-Метса-Чена), що базується на спрощенні булевого виразу у символічному вигляді та представленні його у досконалій диз'юнктивній нормальній формі (ДДНФ). Даний апарат може також бути використаний для пошуку оптимальних або квазіоптимальних (за кількістю необхідних тестів) стратегій вибору наступних тестів в залежності від результатів попередніх. Виконано оцінку ефективності запропонованого методу відповідно до заданого набору тестових перевірок. Сформульовано критерії для діагностичного графу, що визначають максимальну кратність відмов, при яких застосування методу є можливим.

6. Запропоновано метод формування моделей так званих зважених систем, компоненти яких мають різні ваги, що характеризують їх вклад в процес розрахунку. Метод базується на дублюванні відповідних елементів у векторі стану системи. Відзначено, що можливе ускладнення виразів реберних функцій компенсується спрощенням за рахунок того, що деякі змінні входять до виразів із повтореннями.

7. Запропоновано узагальнення GL-моделі із використанням функцій багатозначної логіки, що дозволяє будувати моделі систем, які, як і їх компоненти, мають більш ніж два стани роботоздатності. Замість критерію зв'язності в таких моделях запропоновано використовувати критерій наявності в графі кістякового дерева, всі ребра якого мають вагу не меншу ніж задана. З

метою зменшення складності пошуку таких дерев виконано модифікацію алгоритмів Прима та Крускала.

8. Запропоновано метод побудови моделей ієрархічних систем, що базується на композиції GL-моделей, при якій значення результатів моделей нижнього рівня ієрархії формують вхідні вектори моделей верхнього рівня. При цьому передбачено можливість формування підсистем верхніх рівнів ієрархії з комбінації підсистем нижніх рівнів та безпосередньо процесорів із реалізацією відмовостійкості. Варто зазначити, що кількості та структура рівнів ієрархії системи та моделі в загальному випадку можуть відрізнятися.

9. Запропоновано метод побудови моделей систем із ковзним резервом, що базується на композиції МВР-моделей, а також модифікована двоїста МВР модель, яка спрощує побудову таких моделей. При цьому враховується, що підсистеми ВБС можуть бути самі по собі відмовостійкими.

10. Розроблено програмне забезпечення, за допомогою якого отримані експериментальні дані, що підтверджують коректність та ефективність теоретично отриманих результатів.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. ГОСТ 27.002-89. Надежность в технике. Основные понятия. Термины и определения. — М.: ИПК Издательство стандартов, 2002. — 24 с.
2. Авиженис А. Отказоустойчивость — свойство, обеспечивающее постоянную работоспособность цифровых схем // ТИИЭР. — 1978. — Т. 7, № 10. — С.5-25.
3. Авиженис А., Лапри Ж.-К. Гарантоспособные вычисления: от идей до реализации // ТИИЭР. — 1986. — Т.74. №5. — С.8-21.
4. Каравай М. Д., Согомонян Е. С., Шагаев И. В. Отказоустойчивые вычислительные системы // Итоги науки и техники. Сер. Техническая кибернетика. — 1989. — Т. 28. — С.77-118.
5. Палагин А.В., Опанасенко В.Н. Реконфигурируемые вычислительные системы. — Киев: Просвіта. — 2006. — 295 с.
6. Лученко О.А., Гавриленко О.И., Кулик А.С. Обеспечение активной отказоустойчивости технических систем // Радіоелектронні і комп'ютерні системи. — 2006. — №5. — С.41-47.
7. Юрченко Ю.Б. Задачи построения архитектуры отказоустойчивых информационно-вычислительных комплексов реального времени с компонентами «система-на-кристалле» // Радіоелектронні і комп'ютерні системи. — 2009. — №6. — С.227-230.
8. Викторова В.С., Степанянц А.С. Модели и методы расчета надежности технических систем. — Ленанд. М.: 2016. — 256 с.
9. Rosenthal. A. Series-parallel reduction for Difficult Measures of Network Reliability // Networks — Vol. 11, 1981 — Pp.323-334.
10. Cancela H., El Khadiri M. Series-parallel reductions in Monte Carlo network-reliability evaluation // IEEE Transactions on Reliability — Vol. 47, Issue:2, 1998 — Pp.159-164.
11. Kuo W., Zuo M. Optimal Reliability Modeling — John Willey & Sons, 2002. — 560 p.

- 12.Коваленко И.Н. Расчет вероятностных характеристик систем — К.:Техника, 1982. — 96 с.
- 13.Hu B., Seiler P. Pivotal decomposition for reliability analysis of fault tolerant control systems on unmanned aerial vehicles // Reliability Engineering & System Safety – Vol.140, 2015 — Pp.130-141.
- 14.Yuan. J. Pivotal Decomposition to Find Availability and Failure-Frequency of Systems with Common-Cause Failures // IEEE Transactions on Reliability — Vol.:R-36, Issue:1, 1987 — Pp.48-53.
- 15.Беляев Ю. К. Надежность технических систем: справочник / Ю. К. Беляев, В. А. Богатырев, В. В. Болотин [и др.]: под ред. И. А. Ушакова. — М. : Радио и связь, 1985. — 608 с.
- 16.Nakazawa H. A decomposition method for computing system reliability by a Boolean expression // IEEE Transactions on Reliability. — Vol.:R-26, Issue:4, 1977 — Pp.250-252.
- 17.Billinton R.E., Allan R.N. Reliability Evaluation of Engineering Systems: Concepts and Techniques (2nd edition) — Plenum, New York, 1992 — 456 p.
- 18.Aggarwal K.K., Gupta J.S., Misra K.B. A new method for system reliability evaluation // Microelectronics and Reliability — Vol.12, 1973 — Pp.435-440.
- 19.Fratta L., Montanari U.G. A Boolean algebra method for computing the terminal reliability in a communication network // IEEE Transactions on Circuit Theory — Vol.CT-20, 1973 — Pp.203-211.
- 20.Abraham J.A. An improved method for network reliability // IEEE Transactions on Reliability — Vol.R-28, Issue:1, 1979 — Pp.58-61.
- 21.Locks M.O. Recursive disjoint products, inclusion-exclusion, and min-cut approximations // IEEE Transactions on Reliability — Vol.R-29, Issue:5, 1980 — Pp.368-371.
- 22.Locks M.O. Recursive disjoint products: A review of three algorithms // IEEE Transactions on Reliability — Vol.R-31, Issue:1, 1982 — Pp.33-35.

- 23.Locks M.O. Comments on: Improved method of inclusion-exclusion applied to k -out of- n systems // IEEE Transactions on Reliability — Vol.R-33, Issue:4, 1984 — Pp.321-323.
- 24.Locks M.O. A minimizing algorithm for sum of disjoint products // IEEE Transactions on Reliability — Vol.R-36, 1987 — Pp.445-453.
- 25.Wilson J.M. An improved minimizing algorithm for sum of disjoint products // IEEE Transactions on Reliability — Vol.R-39, Issue:1, 1990 — Pp.42-45.
- 26.Chao M.T., Lin G.D. Economical design of large consecutive- k -out-of- n :F systems // IEEE Transactions on Reliability — Vol.R-33, Issue:5, 1984 — Pp.411-413.
- 27.Fu J.C., Hu B. On reliability of a large consecutive- k -out-of- n :F system with $(k - 1)$ -step Markov dependence // IEEE Transactions on Reliability — Vol.R-36, Issue:1, 1987 — Pp.75-77.
- 28.Chao M.T., Fu J.C. A limit theorem of certain repairable systems // Annals of the Institute of Statistical Mathematics — Vol.41(4), 1989 — Pp.809-818.
- 29.Chao M.T., Fu J.C. The reliability of large series systems under Markov structure // Advances in Applied Probability — Vol.23, 1991 — Pp.894-908.
- 30.Fu J.C., Lou W.Y. On reliabilities of certain large linearly connected engineering systems // Statistics and Probability Letters — Vol.12, 1991 — Pp.291-296.
- 31.Koutras M.V. On a Markov chain approach for the study of reliability structures // Journal of Applied Probability — Vol.33(2), 1996 — Pp.357-367.
- 32.Rosenthal A. Note on closed form solutions for delta-star and star-delta conversion of reliability networks // IEEE Transactions on Reliability — Vol.R-27, Issue:2, 1978 — Pp.110-111.
- 33.Rosenthal A., Frisque D. Transformations for simplifying network reliability calculations // Network — Vol.7, No.1, 1977 — Pp.97-111, 382.
- 34.Wang S.D., Sun C.H. Transformations of star-delta and delta-star reliability networks // IEEE Transactions on Reliability — Vol.R-45, Issue:1, 1996 — Pp.120-126.

35. Ушаков, И.А. Курс теории надежности систем [Текст]: учебное пособие / И. А. Ушаков. — М.: Дрофа. — 2008. — 239 с.
36. Сотсков Б. С. Основы теории и расчета надежности элементов и устройств автоматики и вычислительной техники. — М., 1970. — 270 с.
37. Ушаков И.А. Анализ надежности структурно-сложных систем — М.: Знание, 1989. — 214с.
38. Barlow R.E., Proschan F. Statistical Theory of Reliability and Life Testing: Probability Models — New York: Holt, Rinehart and Winston, 1975. — 223 p.
39. Hoang P.. Handbook of reliability engineering — Piscataway, New Jersey, USA: «Springer», 2003. — 626 p.
40. Азарсков В. Н. Надежность систем управления и автоматики: Учеб. Пособие / В. Н. Азарсков, В. П. Стрельников. — К.: НАУ, 2004. — 164 с.
41. Бабешко Е.В., Ильяшенко О.А., Харченко В.С. Многоэтапный анализ надежности и безопасности информационно-управляющих систем // Радиоэлектронні і комп'ютерні системи. — № 7(48), 2010 — С.283-288.
42. Гнеденко Б. В. Надёжность и эффективность в технике / Б. В. Гнеденко. — М. : «Машиностроение». — 1987. — Т. 2. — 208 с.
43. Говорущенко Т.О., Поморова О.В. Метод оцінки достатності інформації для визначення складності та якості програмного забезпечення на основі порівняльного аналізу онтологій // Радиоэлектронні і комп'ютерні системи — №6, 2016 — С.59-81.
44. Кортунув В.И., Илюшко В.М., Мазуренко А.В. Проблемы надежности программно-аппаратных средств беспилотных систем // Радиоэлектронні і комп'ютерні системи. — № 7(59), 2012 — С.307-313.
45. Костолни Й., Зайцева О. Діаграми рішень і логічні направлені похідні для аналізу надійності систем з декількома рівнями надійності(англ. мовою) // Радиоэлектронні і комп'ютерні системи — № 5(64), 2013 — С.276-282.
46. Одарущенко О.Н., Живилю С.В., Харченко В.С., Одарущенко Е.Б. Анализ архитектур отказоустойчивых серверов для оценки их надежности // Радиоэлектронні і комп'ютерні системи. — № 7(59), 2012 — С. 60-68.

47. Райншке К., Ушаков И.А. Оценка надежности систем с использованием графов / Под ред. И.А. Ушакова. — М.: Радио и связь, 1988. — 208с.
48. Рыжкин А.А. Основы теории надежности: Учеб. Пособие / А. А. Рыжкин, Б. Н. Слюсарь, К. Г. Шучев. — Ростов-на-Дону: Издательский центр ДГТУ, 2002. — 182 с.
49. Скарга-Бандурова И.С., Нестеров М.В. Реализация архитектуры высокой готовности для медицинской информационной системы (на англ. яз.) // Радиоелектронні і комп'ютерні системи — №5, 2012 — С.23-27.
50. Скопинцев В.А. Качество электроэнергетических систем: надёжность, безопасность, экономичность, живучесть. — М.: Энергоатомиздат., 2009. — 322с.
51. Щербовських С.В. Математичні моделі та методи для визначення характеристик надійності багатотермінальних систем із урахуванням перерозподілу навантаження: монографія [Текст] / С. В. Щербовських. — Львів : Вид-во Львівської політехніки, 2012. — 296 с.
52. Яремчук С.А. Проблемы априорной оценки показателей надежности гарантоспособных информационных систем критического назначения // Радиоелектронні і комп'ютерні системи — № 5(64), 2013 — С.414-421.
53. Esary J.D., Proschan F. A reliability bound for systems of maintained, interdependent components // Journal of American Statistical Association — Vol.65(329), 1970 — Pp.329-338.
54. Distefano S., Xing L. A new approach to modeling the system reliability: dynamic reliability block diagrams // Reliability and Maintainability Symposium — IEEE, 2006 — Pp.189-195.
55. Ahmed W., Hasan O., Tahar S. Formalization of reliability block diagrams in higher-order logic. // Journal of Applied Logic — Vol.18, 2016 — Pp.19-41.
56. Xu H., Xing L., Robidoux R. Drbd: dynamic reliability block diagrams for system reliability modelling // International Journal of Computers and Applications — Vol.31, Issue:2, 2009 — Pp.132-141.

- 57.Hasan O., Ahmed W., Tahar S., Hamdi M.S. Reliability block diagrams based analysis: a survey // International Conference of Numerical Analysis and Applied Mathematics — Vol.1648, 2015 — Pp.850129.1-4.
- 58.Lee W.S., Grosh D.L., Tillman F.A., Lie C.H. Fault tree analysis, methods and applications: a review // IEEE Transactions on Reliability — Vol.34, Issue:3, 1985 — Pp.194-203.
- 59.Pukite J., Pukite P. Modeling for Reliability Analysis: Markov Modeling for Reliability, Maintainability, Safety, and Supportability Analyses of Complex Systems — Wiley-IEEE Press, 1998 — 278p.
- 60.Armin Z. Reliability modeling and evaluation of dynamic systems with Stochastic Petri Nets // Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools — 2013 — Pp.325-264.
- 61.Eryilmaz S. Consecutive k -within- m -out-of- n :F System with Nonidentical Components // Mathematical Problems in Engineering. — 2012. — Pp.1-8.
- 62.Griffith W.S. On consecutive k -out-of- n failure systems and their generalizations // Reliability and Quality Control — 1986 — Pp.157-165.
- 63.Sfakianakis M, Kounias S., Hillaris A. Reliability of a consecutive k -out-of- r -from- n :F system // IEEE Transactions on Reliability — Vol.41, Issue:3, 1992 — Pp.442-447.
- 64.Habib A., Al-Seedy R.O., Radwan T. Reliability evaluation of multi-state consecutive k -out-of- r -from- n :G system // Applied Mathematical Modelling — Vol.31, Issue:11, 2007 — Pp.2412-2423.
- 65.Eryilmaz S., Kan C., Akici F. Consecutive k -Within- m -Out-of- n :F System with Exchangeable Components // Naval Research Logistics. — Vol.56, 2009 — Pp. 503-510.
- 66.Tung S.S. Combinatorial analysis in determining reliability // Proceedings. Annual Reliability and Maintainability Symposium —1982 — Pp.262-266.
- 67.Hwang, F.K. Simplified reliabilities for consecutive k -out-of- n systems // SIAM. J. on Algebraic and Discrete Methods — Vol.7(2), 1986 — Pp.258–264.

- 68.Chang G.J., Cui L., Hwang F.K. Reliabilities for (n, f, k) systems // Statistics & Probability Letters. — Vol. 43, 1999 — Pp.237-242.
- 69.Wu J.S., Chen R.J. An algorithm for computing the reliability of a weighted- k -out-of- n system // IEEE Transactions on Reliability. — Vol.R-43, 1994 — Pp.327-328.
- 70.Block H.W., Savits T.H. Continuous multi-state structure functions // Operations Research — Vol.32(3), 1982 — Pp.703-714.
- 71.El-Newehi E., Proschan F., Sethuraman J. Multi-state coherent system // Journal of Applied Probability — Vol.15, 1978 — Pp.675-688.
- 72.Huang J., Zuo M., Kuo. W. Multi-state k -out-of- n systems. In H. Pham (Ed.). Reliability Engineering Handbook — Springer-Verlag, London, 2002.
- 73.Boedigheimer R.A., Kapur K.C. Customer-driven reliability models for multistate coherent systems // IEEE Transactions on Reliability — Vol.R-43, Issue:1, 1994 — Pp.46-50.
- 74.Huang J., Zuo M.J., Wu Y.H. Generalized multi-state k -out-of- n :G systems // IEEE Transactions on Reliability — Vol.R-49, Issue:1, 2000 — Pp.105-111.
- 75.Shooman M.L. Reliability of Computer Systems and Networks. — New York: “John Wiley & Sons, Inc.”, INC., 2002. — 528p.
- 76.Belfore L.A. An $O(n(\log_2(n))^2)$ algorithm for computing the reliability of k -out-of- n :G & k -to- l -out-of- n :G systems // IEEE Transaction on Reliability. — Vol.R-44, No.1., 1995 — Pp.132-136.
- 77.Rushdi A.M. Utilization of symmetric switching function in the computation of k -out-of- n system reliability // Microelectronics and Reliability. — Vol.26, No.5., 1986 — Pp.973-987.
- 78.Rushdi A.M. Partially-Redundant Systems: Examples, Reliability, and Life Expectancy // IMACST — Vol.1, No.1, 2010 — Pp.1-13.
- 79.Heidtmann K.D. Improved method of inclusion-exclusion applied to k -out-of- n :F system // Proceedings of the Annual Reliability and Maintainability Symposium — 1991 — Pp.582-586.

80. McGrady P.W. The availability of a k -out-of- n :G network // IEEE Transactions on Reliability — Vol.R-34, No.5, 1985 — Pp.451-452.
81. Barlow R.E., Heidtmann K. D. Computing k -out-of- n system reliability // IEEE Transaction on Reliability — Vol.R-33, 1984 — Pp.322-323.
82. Ding Y., Zio E., Li Y., Cheng L., Wu Q. Definition of Multi-state Weighted k -out-of- n :F System // International Journal of Performability Engineering — Vol.8, No.2, 2011 — Pp.217-219.
83. Li W., Zuo M.J., Moghaddass R. Optimal design of a multi-state weighted series-parallel system using physical programming and genetic algorithms // Asia-Pacific Journal of Operational Research — Vol.28, No.4, 2011 — Pp.543-562
84. Романкевич А.М., Карачун Л.Ф., Романкевич Л.Ф. Графо-логические модели для анализа сложных отказоустойчивых вычислительных систем // Электронное моделирование. — 2001. — Т. 23, № 1. — С. 102-111.
85. Романкевич В.А., Рабах Мох'д Ахмад Ал Шбул, Назаренко В.В. О минимизации базовых циклических GL-моделей // Вісник ТУП, ч.1, т.2 "Технічні науки" — Хмельницький 2004 — С.42-46.
86. Романкевич В.А., Потапова Е.Р., Бахтари Хедаятоллах, Назаренко В.В. GL-модель поведения отказоустойчивых многопроцессорных систем с минимальным числом теряемых рёбер // Вісник НТУУ "КПІ" — Інформатика, управління та ОТ — 2006. — №45. — С.93-100.
87. Романкевич А.М., Романкевич В.А., Кононова А.А., Рабах Ал Шбул О некоторых особенностях GL-моделей $K(2, n)$ // Вісник НТУУ "КПІ". — Інформатика, управління та ОТ — 2004. — №41. — С.85-92.
88. Романкевич А.М., Романкевич В.А., Кононова А.А. Граничные характеристики графо-логической модели 2-отказоустойчивой многопроцессорной системы // Вісник НТУУ "КПІ". — Інформатика, управління та ОТ — 2004. — №42. — С.28-39.
89. Романкевич В.О., Морозов К.В. Базова $K(3, n)$ GL-модель, яка будується методом зсуву змінних // Прикладна математика та комп'ютеринг

- (ПМК-2012). Збірник тез доповідей. — Київ, НТУУ "КПІ". — 2012. — С.230-235.
90. Гроль В.В., Морозов К.В., Романкевич В.А. Упрощенная 3-отказоустойчивая GL-модель для отказоустойчивых многопроцессорных систем // Інформаційно-керуючі системи на залізничному транспорті. Тези доповідей. — 2012. — №5. — С. 90.
 91. Романкевич В.А., Кононова А.А. Об одном методе преобразования GL-моделей поведения отказоустойчивых многопроцессорных систем в потоке отказов // Радіоелектронні і комп'ютерні системи. — №7, 2007. — С.49-56.
 92. Романкевич В.А., Ефремова А.А., Гаврилюк А.С. Об одном алгоритме преобразования GL-моделей // Радіоелектронні і комп'ютерні системи. — №5, 2010. — С.186-191.
 93. Романкевич В.А., Морозов К.В., Фесенюк А.П. Об одном методе модификации рёберных функций GL-моделей // Радіоелектронні і комп'ютерні системи. — №6, 2014. — С.95-99.
 94. Романкевич А.М., Романкевич В.А., Майданюк И.В. Граничные оценки числа рёбер GL-моделей поведения отказоустойчивых многопроцессорных систем в потоке отказов // Электронное моделирование. — №1, т.30. — 2008. — С.59-70.
 95. Майданюк И.В., Морозов К.В., Потапова Е.Р., Шурига А.В. Об одном свойстве GL-модели с минимальным количеством теряемых ребер // Науковий вісник Чернівецького університету. Серія: Комп'ютерні системи та компоненти. — Т.1, №2. — 2010. — С. 31-34.
 96. Романкевич А.М., Сапсай Т.Г., Морозов К.В. О модификации реберных функций GL-моделей // Інформаційно-керуючі системи на залізничному транспорті. Тези доповідей. — 2013. — №4 (додаток). — С. 36.
 97. Морозов К.В., Романкевич А.М., Романкевич В.А. О характере влияния модификации рёберных функций GL-модели на её поведение в потоке

- отказов // Радіоелектронні і комп'ютерні системи. — 2016. — № 6. — С.108-112.
98. Романкевич А.М., Романкевич В.А., Морозов К.В. Об одном условии модификации рёберных функций GL-модели // Системный анализ и информационные технологии: материалы 17-й Международной научно-практической конференции SAIT 2015. — К.: УНК «ИПСА» НТУУ «КПИ», 2015. — С. 106.
 99. Морозов К.В., Романкевич В.О., Потапова К.Р. Про модифікацію графологічної моделі // Інформаційні технології та комп'ютерна інженерія. — Том 3, № 34. — 2015. — С.42-49.
 100. Фон Нейман Дж. Вероятностная логика и синтез надежных организмов из ненадежных компонент // Автоматы. — М.: Изд-во иностр. лит., 1956. — С.68-139.
 101. Тюрин С.Ф. Обеспечение надежности технических средств путем их троирования и расчетверения // Надежность. — Том 19, № 1, 2019 — С.4-9.
 102. Ведешенков В.А. Организация диагностирования цифровых систем со структурой симметричного двудольного графа // Пробл. управления. — 2009. № 6. — С.59–67.
 103. Каравай М.Ф., Подлазов В.С. Расширенный обобщенный гиперкуб как отказоустойчивая системная сеть для многопроцессорных систем // УБС. — 2013. № 45. — С.344-371.
 104. Preparata F.P., Metze G., Chien R.T. On the Connection Assignment Problem of Diagnosable Systems // IEEE Trans. Electron. Comput. — 1967. — ES-16. No. 6. — Pp.848-854.
 105. Hakimi S.L., Amin A.T. Characterization of Connection Assignment of Diagnosable Systems // IEEE Trans. Comput. — 1974. — C-23. No. 1. — Pp.86-88.
 106. Романкевич В.А. Самотестирование многопроцессорных систем с регулярными диагностическими связями // АиТ. 2017. № 2. С.115-127.

- Romankevich V. A. Self-testing of Multiprocessor Systems with Regular Diagnostic Connections // Autom. Remote Control. 2017. V. 78. No. 2. P.289-299.
107. Романкевич В.А., Романкевич А.В, Ахмедова Д.Н. Метод уменьшения количества взаимопроверок при самотестировании многопроцессорных систем // Радиоэлектронные и компьютерные системы. 2018. №4. С.61-66.
 108. Drozd J., Drozd A., Al-dhabi M. A resource approach to on-line testing of computing circuits // Proc. IEEE East-West Design & Test Symposium. Batumi, Georgia, 2015. P. 276–281. DOI: 10.1109/EWDTS.2015.7493122
 109. Drozd A., Drozd J., Antoshchuk S., Nikul V., Al-dhabi M. Objects and Methods of On-Line Testing: Main Requirements and Perspectives of Development // Proc. IEEE East-West Design & Test Symposium. Yerevan, Armenia, 2016. P. 72–76. DOI: 10.1109/EWDTS.2016.7807750
 110. Димитриев Ю.К. О t-диагностируемости мультипроцессорных систем с симметричной циркулянтной структурой // АиТ. 2013. № 1. С.135-145.
Dimitriev Y.K. On t-diagnosability of Multicore Systems with Symmetric Circulant Structure // Autom. Remote Control. 2013. V. 74. No. 1. P. 105–112.
 111. Белявский В.Е., Валуйский В.Н., Романкевич А.М. и др. Самодиагностируемые многомодульные системы: некоторые оценки тестирования // АиТ. — 1999. — №8. — С.148-153.
Belyavskii V.E., Valuiskii V.N., Romankevich A.M. and Romankevich V.A. Self-Diagnosable Multimodular Systems: Some Estimates of Testing // Autom. Remote Control. 1999. V. 60. No. 8. P. 1179–1183.
 112. Романкевич А.М., Романкевич В.А. О диагностировании многопроцессорных систем при отказе более половины процессоров // АиТ. — 2017. — №9. — С.84-90.
Romankevich A. M., Romankevich V. A. Diagnosis of Multiprocessor Systems

- Under Failure of More Than Half Processors // Autom. Remote Control. 2017. V. 78. No. 9. P. 1614–1618.
113. Морозов К.В., Потапова Е.Р. О самотестировании многопроцессорных систем // Системный анализ и информационные технологии: материалы 18-й Международной научно-практической конференции SAIT 2016. — К.: УНК «ИПСА» НТУУ «КПИ», 2016. — С.396.
 114. Баранов В. Г., Гладков В. В., Махалин Б. Н. Математические модели для системного уровня диагностики неисправностей в мультипроцессорных системах // Обзоры по электронной технике. Сер. 8. Управление качеством, стандартизация, метрология, испытания. — 1991. — Вып. 2 (1601). — 58 с.
 115. Корячко В.П., Скворцов С.В. Шувиков В.И. Характеризация диагностических графов для симметричной модели дешифрации синдрома // Машиностроение и компьютерные технологии. — №4, 2006 — 11с.
 116. Финн В., Бочвар Д. Многозначные логики и их применения: Логические исчисления, алгебры и функциональные свойства. Том 1. — М.: УРСС, 2008. — 416 с.
 117. Морозов К.В., Романкевич В.А., Сапсай Т.Г. О модификации графологической модели для систем и их компонентов с множественными состояниями // Системный анализ и информационные технологии: материалы 19 й Международной научно-практической конференции SAIT 2017. — К.: УНК «ИПСА» КПИ им. Игоря Сикорского, 2017. — С.94.
 118. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, Third Edition — MIT Press, 2009 — 1292p.
 119. Prim R.C. Shortest connection networks and some generalizations // Bell System Technical Journal — Vol.36, 1957 — Pp.1389-1401.
 120. Kruskal J.B. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. // Proc. AMS. — 1956. — Vol7, No.1. — С.48-50.

121. Chen Y., Yang. Q. Reliability of Two-Stage Weighted- k -out-of- n Systems With Components in Common // IEEE Transactions on Reliability. — 2005. — Vol.54, No.3. — P.431-440.
122. Морозов К.В., Романкевич В.А., Олейник В.В. О графо-логических моделях взвешенных систем // Інформаційні технології в освіті, техніці та промисловості. Матеріали II всеукраїнської науково-практичної конференції молодих учених і студентів. — Івано-Франківськ — 2015. — С. 206-207.
123. Romankevich A.M., Morozov K.V., Romankevich V.A. Graph-Logic Models of Hierarchical Fault-Tolerant Multiprocessor Systems // IJCSNS International Journal of Computer Science and Network Security. — 2019. — Vol.19, No.7. — Pp.151-156.
124. Потапова К.Р., Морозов К.В., Трошина М.І. Формування GL-моделей для систем з ієрархічною структурою // Прикладна математика та комп'ютинг (ПМК-2015). Збірник тез доповідей. — Київ, НТУУ "КПІ". — 2015. — С.156-160.
125. Romankevich A.M., Morozov K.V., Romankevich V.A. Hierarchical Graph-Logical Models of Multiprocessor Systems Based on Grouping of Their Components // IJCSNS International Journal of Computer Science and Network Security. — 2019. — Vol.19, No.10. — Pp.138-143.
126. Морозов К.В., Потапова Е.Р., Кичигин Е.Р. GL-модель иерархической системы с процессорами на всех уровнях иерархии // Комп'ютерні системи і проектування технологічних процесів та обладнання: Матеріали всеукраїнської науково-практичної конференції (17-19 лютого 2016р). — Чернівці: ЧФ НТУ «ХП», 2016. — С.35-37.
127. Романкевич А.М., Романкевич В.А., Морозов К.В. Об одной GL-модели системы со скользящим резервом // Радіоелектронні і комп'ютерні системи. — №5, 2013. — С.333-336.
128. Романкевич В.А., Мораведж Милад, Морозов К.В. Построение GL-моделей поведения в потоке отказов для систем со скользящим

резервом // Тези доповідей Третьої міжнародної науково-практичної конференції «Методи та засоби кодування, захисту й ущільнення інформації». — Вінниця, ВНТУ. — 2011. — С.36-37.

Додаток А

Виникнення побічного ефекту в результаті модифікації різних пар функцій GL-моделі

Для прикладу розглянемо модель $K(5, 13)$. Вона містить 9 реберних функцій:

$$f_1 = \kappa_{11}(3,3) \vee \kappa_{12}(2,3)$$

$$f_2 = \kappa_{11}(2,3) \vee \kappa_{12}(3,3)$$

$$f_3 = \kappa_1(4,6) \vee \kappa_2(1,7)$$

$$f_4 = \kappa_1(3,6) \vee \kappa_2(2,7)$$

$$f_5 = \kappa_1(2,6) \vee \kappa_2(3,7)$$

$$f_6 = \kappa_1(1,6) \vee \kappa_2(4,7)$$

$$f_7 = \kappa_{21}(3,3) \vee \kappa_{22}(2,4)$$

$$f_8 = \kappa_{21}(2,3) \vee \kappa_{22}(3,4)$$

$$f_9 = \kappa_{21}(1,3) \vee \kappa_{22}(4,4)$$

Модифікацію виконуватимемо відповідно до того, як було запропоновано в розділі 2, тобто шляхом заміни виразів виду $\kappa(i, n)$ на $\kappa(i + 1, n)$. Кількість векторів, з 6 нулями, блокованих в результаті модифікації однієї функції, в залежності від номера цієї функції і від того, яка (ліва або права) частина функції модифікується приведено в таблиці 1.

Як бачимо, для розглянутої моделі можливе виконання 18 різних модифікацій однієї реберної функції.

В таблиці 2 наведено кількість векторів з 6 і 7 нулями, блокованих в результаті модифікації двох різних функцій моделі. У таблиці літерами "П" і "Л" позначені відповідно модифікація правої і лівої частини реберної функції.

До кожного поля таблиці записується кількість векторів, блокованих в результаті відповідної модифікації. Причому, якщо блокуються тільки вектори з 6 нулями, то в таблицю записується тільки одне число. Якщо ж крім них блокуються вектори з 7 нулями, то в таблицю записується два числа, перше з

яких відповідає кількості векторів з 6 нулями, а друге (в дужках) – кількості векторів з 7 нулями. Зауважимо, що поля таблиці, які знаходяться на перетині рядків і стовпців, відповідних модифікації однієї і тієї ж функції залишаються порожніми, оскільки розглядається випадок модифікації двох різних функцій.

В таблиці виділені ситуації, коли кількість векторів з 6 нулями, блокованих в результаті модифікації двох функцій відрізняється від суми числа векторів блокованих внаслідок модифікації відповідних функцій окремо. Також виділені ситуації, в яких блокуються вектори з 7 нулями.

У таблиці 3 наведено аналіз всіх цих ситуацій та вказані формули, за якими у відповідності до матеріалів розділу 3 має розраховуватися кількість блокованих векторів в кожному окремому випадку. Як бачимо, у всіх випадках кількість блокованих векторів відповідає теоретично розрахованій.

Таблиця 1

Кількість векторів, блокованих в результаті модифікації
функцій моделі $K(5, 13)$

	Ліва	Права
f_1	22	21
f_2	21	22
f_3	315	42
f_4	700	315
f_5	525	700
f_6	126	525
f_7	40	36
f_8	75	76
f_9	18	21

Таблиця 2 Кількість векторів, блокованих в результаті модифікації пар функцій моделі $K(5, 13)$

	$f_1 \text{ Л}$	$f_1 \text{ П}$	$f_2 \text{ Л}$	$f_2 \text{ П}$	$f_3 \text{ Л}$	$f_3 \text{ П}$	$f_4 \text{ Л}$	$f_4 \text{ П}$	$f_5 \text{ Л}$	$f_5 \text{ П}$	$f_6 \text{ Л}$	$f_6 \text{ П}$	$f_7 \text{ Л}$	$f_7 \text{ П}$	$f_8 \text{ Л}$	$f_8 \text{ П}$	$f_9 \text{ Л}$	$f_9 \text{ П}$
$f_1 \text{ Л}$	–	–	43	43 (7)	337	43 (7)	722	337 (63)	547	722	148	547	62	58	97	98	40	43
$f_1 \text{ П}$	–	–	42	43	336	42	721	336 (63)	546	721	147	546	61	57	96	97	39	42
$f_2 \text{ Л}$	43	42	–	–	336	42	721	336 (63)	546	721	147	546	61	57	96	97	39	42
$f_2 \text{ П}$	43 (7)	43	–	–	337	43 (7)	722	337 (63)	547	722	148	547	62	58	97	98	40	43
$f_3 \text{ Л}$	337	336	336	337	–	–	1015	315	840	1015 (525)	441	840	355	351	390	391	333	336
$f_3 \text{ П}$	43 (7)	42	42	43 (7)	–	–	742	357	567	742	168	567	82	78	117	118	60	63
$f_4 \text{ Л}$	722	721	721	722	1015	742	–	–	1225	700	826	1225 (700)	740	736	775	776	718	721
$f_4 \text{ П}$	337 (63)	336 (63)	336 (63)	337 (63)	315	357	–	–	840	1015	441	840	355	351	390	391	333	336
$f_5 \text{ Л}$	547	546	546	547	840	567	1225	840	–	–	651	525	565 (90)	561 (90)	600 (180)	601 (180)	543 (45)	546 (45)
$f_5 \text{ П}$	722	721	721	722	1015 (525)	742	700	1015	–	–	826	1225	740	736	775	776	718	721
$f_6 \text{ Л}$	148	147	147	148	441	168	826	441	651	826	–	–	130 (24)	126	129 (18)	130 (24)	126	129 (18)
$f_6 \text{ П}$	547	546	546	547	840	567	1225 (700)	840	525	1225	–	–	565	561	600	601	543	546
$f_7 \text{ Л}$	62	61	61	62	355	82	740	355	565 (90)	740	130 (24)	565	–	–	115	112 (24)	58	61 (1)
$f_7 \text{ П}$	58	57	57	58	351	78	736	351	561 (90)	736	126	561	–	–	111	112	54	57
$f_8 \text{ Л}$	97	96	96	97	390	117	775	390	600 (180)	775	129 (18)	600	115	111	–	–	93	93 (18)
$f_8 \text{ П}$	98	97	97	98	391	118	776	391	601 (180)	776	130 (24)	601	112 (24)	112	–	–	94	97
$f_9 \text{ Л}$	40	39	39	40	333	60	718	333	543 (45)	718	126	543	58	54	93	94	–	–
$f_9 \text{ П}$	43	42	42	43	336	63	721	336	546 (45)	721	129 (18)	546	61 (1)	57	93 (18)	97	–	–

Таблиця 3

Аналіз випадків виникнення побічного ефекту

Функції, що модифікуються		Відношення підмножин вхідних змінних	Причина	Номер випадку відповідно до Розділу 2	Формула розрахунку кількості особливих векторів	
					6 нулів	7 нулів
$f_3 \text{ Л}$	$f_4 \text{ П}$	$V_{11} = V_{22},$ $V_{12} = V_{21}$	$n_{13} = n_{23} = 0,$ $i + j = m + 1$	2	(3.19)	0
$f_4 \text{ Л}$	$f_5 \text{ П}$					
$f_5 \text{ Л}$	$f_6 \text{ П}$					
$f_1 \text{ Л}$	$f_2 \text{ П}$		$n_{13} \neq 0,$ $n_{23} \neq 0,$ $i + j = m + 1$		(3.19)	(3.20)
$f_7 \text{ Л}$	$f_8 \text{ П}$					
$f_8 \text{ Л}$	$f_9 \text{ П}$					
$f_3 \text{ Л}$	$f_5 \text{ П}$		$i + j = m + 2$		(3.15)	(3.21)
$f_4 \text{ Л}$	$f_6 \text{ П}$					
$f_7 \text{ Л}$	$f_9 \text{ П}$					
$f_1 \text{ П}$	$f_3 \text{ П}$	$V_{11} \subset V_{22},$ $V_{12} \subset V_{22}$	$j = 1,$ $V_{21} = V_{13},$ $m - i = n_{12}$	4	(3.19)	0
$f_2 \text{ Л}$	$f_3 \text{ П}$					
$f_1 \text{ Л}$	$f_3 \text{ П}$					
$f_2 \text{ П}$	$f_3 \text{ П}$		$j = 1,$ $V_{21} = V_{13},$ $m - i \neq n_{12}$		(3.32)	(3.33)
$f_1 \text{ Л}$	$f_4 \text{ П}$					
$f_1 \text{ П}$	$f_4 \text{ П}$					
$f_2 \text{ Л}$	$f_4 \text{ П}$		$j = 2$		(3.15)	(3.34)
$f_2 \text{ П}$	$f_4 \text{ П}$					
$f_6 \text{ Л}$	$f_7 \text{ П}$					
$f_6 \text{ Л}$	$f_9 \text{ Л}$	$V_{21} \subset V_{12},$ $V_{22} \subset V_{12}$	$i = 1,$ $V_{11} = V_{23},$ $m - j = n_{22}$	6	(3.19)	0
$f_6 \text{ Л}$	$f_7 \text{ Л}$					
$f_6 \text{ Л}$	$f_8 \text{ Л}$					
$f_6 \text{ Л}$	$f_8 \text{ П}$		$i = 1,$ $V_{11} = V_{23},$ $m - j \neq n_{22}$		(3.36)	(3.37)
$f_6 \text{ Л}$	$f_9 \text{ П}$					
$f_5 \text{ Л}$	$f_7 \text{ Л}$					
$f_5 \text{ Л}$	$f_7 \text{ П}$		$i = 2$		(3.15)	(3.38)
$f_5 \text{ Л}$	$f_8 \text{ Л}$					
$f_5 \text{ Л}$	$f_8 \text{ П}$					
$f_5 \text{ Л}$	$f_9 \text{ Л}$					
$f_5 \text{ Л}$	$f_9 \text{ П}$					

Додаток Б

Приклади використання формального методу визначення стану процесорів багатопроцесорної системи при її тестуванні

Приклад 1

Розглянемо приклад: 3-діагностована система складається з 8 процесорів (позначимо їх як a, b, c, d, e, f, g, h). Граф, який представляє топологію зв'язків, представлений на рис. 1. Нехай вийшли з ладу 3 процесори: a, f і h .

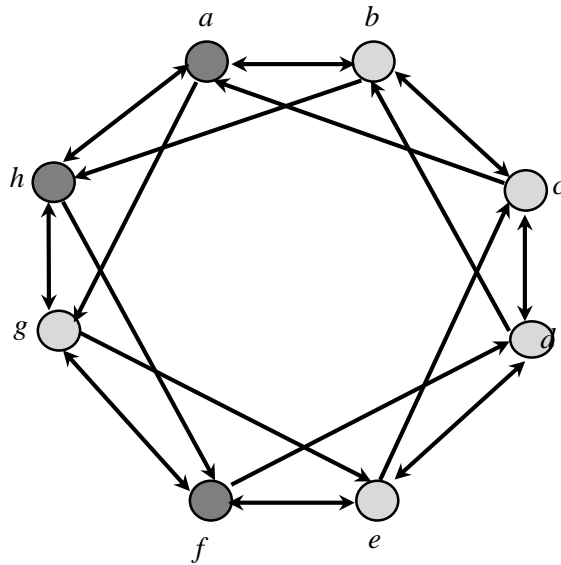


Рис.1. Граф топології системи для прикладу 1

Нехай були проведені наступні тестові перевірки: $a \rightarrow b, c \rightarrow d, e \rightarrow f, g \rightarrow h, b \rightarrow c, d \rightarrow e, f \rightarrow g, h \rightarrow a, c \rightarrow a, d \rightarrow b, g \rightarrow e, h \rightarrow f$ (тестуючий процесор вказано ліворуч від стрілки, а тестований – праворуч), і нехай отримано результати, що наведені в таблиці 1.

Можемо записати: $V_{12} \triangleq R_1 \wedge R_2 \wedge \dots \wedge R_{12} = 1$. Виразимо V_{12} в формі ДДНФ, виключаючи при цьому всі кон'юнкції, що містять більш ніж 3 інверсії:

$$\begin{aligned}
 V_{12} &\triangleq R_1 \wedge R_2 \wedge \dots \wedge R_{12} \triangleq (b \vee \bar{a})(d \vee \bar{c})(\bar{f} \vee \bar{e})(\bar{h} \vee \bar{g})(c \vee \bar{b}) \wedge \\
 &\wedge (e \vee \bar{d})(\bar{g} \vee \bar{f})(a \vee \bar{h})(\bar{a} \vee \bar{c})(b \vee \bar{d})(e \vee \bar{g})(f \vee \bar{h}) = \\
 &= ((b \vee \bar{a})(a \vee \bar{h})(\bar{a} \vee \bar{c}))((d \vee \bar{c})(c \vee \bar{b})(b \vee \bar{d})) \wedge \\
 &\wedge ((\bar{f} \vee \bar{e})(e \vee \bar{d})(e \vee \bar{g}))((\bar{h} \vee \bar{g})(\bar{g} \vee \bar{f})(f \vee \bar{h})) =
 \end{aligned}$$

$$\begin{aligned}
&= (ab\bar{c} \vee b\bar{a}\bar{h} \vee b\bar{c}\bar{h} \vee \bar{a}\bar{h} \vee \bar{a}\bar{c}\bar{h})(bcd \vee \bar{b}\bar{c}\bar{d}) \wedge \\
&\wedge (e\bar{f} \vee e\bar{f}\bar{g} \vee \bar{d}e\bar{f} \vee \bar{d}\bar{f}\bar{g} \vee \bar{d}\bar{e}\bar{g})(\bar{g}\bar{h} \vee f\bar{g}\bar{h} \vee \bar{f}\bar{h} \vee f\bar{g} \vee \bar{f}\bar{g}\bar{h}) = \\
&= (ab\bar{c} \vee \bar{a}\bar{h})(bcd \vee \bar{b}\bar{c}\bar{d})(e\bar{f} \vee \bar{d}\bar{e}\bar{g})(\bar{f}\bar{h} \vee f\bar{g}) = \\
&= (\bar{a}bcd\bar{h} \vee \bar{a}\bar{b}\bar{c}\bar{d}\bar{h})(e\bar{f}\bar{h} \vee \bar{d}\bar{e}f\bar{g} \vee \bar{d}\bar{e}\bar{g}\bar{h}) = \bar{a}bcdef\bar{g}\bar{h} = 1.
\end{aligned}$$

Таблиця 1

Результати проведених тестових перевірок для прикладу 1

№ тесту, i	Тестуючий процесор	Тестований процесор	Результат тесту, r_i	Вираз R_i
1	a	b	0	$R_1 \triangleq b \vee \bar{a}$
2	c	d	0	$R_2 \triangleq d \vee \bar{c}$
3	e	f	1	$R_3 \triangleq \bar{f} \vee \bar{e}$
4	g	h	1	$R_4 \triangleq \bar{h} \vee \bar{g}$
5	b	c	0	$R_5 \triangleq c \vee \bar{b}$
6	d	e	0	$R_6 \triangleq e \vee \bar{d}$
7	f	g	1	$R_7 \triangleq \bar{g} \vee \bar{f}$
8	h	a	0	$R_8 \triangleq a \vee \bar{h}$
9	c	a	1	$R_9 \triangleq \bar{a} \vee \bar{c}$
10	d	b	0	$R_{10} \triangleq b \vee \bar{d}$
11	g	e	0	$R_{11} \triangleq e \vee \bar{g}$
12	h	f	0	$R_{12} \triangleq f \vee \bar{h}$

В результаті отримана рівно одна конституента одиниці, яка відповідає наступному стану системи: процесори b, c, d, e, g – справні, а процесори a, f, h – несправні. Знайдене рішення відповідає дійсному стану системи.

Відзначимо, що, як це і було зроблено в прикладі, виключення елементарних кон'юнкцій, що містять більш ніж M інверсій припустимо виконувати на будь-якому етапі перетворення. Дійсно, в результаті розкриття дужок елементарна кон'юнкція породжує вирази з не меншим, ніж був ще у ній, числом інверсій. Даний прийом часто дозволяє дещо спростити процес перетворення.

Варто також відзначити, що розглянутий для даного прикладу набір тестових перевірок є достатнім для отримання однозначної відповіді лише для

представленої комбінації результатів тестів. Так, у випадку, якщо б результати 8-ї та 12-ї перевірок відрізнялися від представлених (тобто $r_8 = 1$ та $r_{12} = 1$, що також відповідає моделі ПМЧ для розглянутого стану системи), то мало б місце $R_8 \triangleq \bar{a} \vee \bar{h}$ та $R_{12} \triangleq \bar{f} \vee \bar{h}$. Тоді перетворення V_{12} було б наступним:

$$\begin{aligned}
 V_{12} &\triangleq R_1 \wedge R_2 \wedge \dots \wedge R_{12} \triangleq (b \vee \bar{a})(d \vee \bar{c})(\bar{f} \vee \bar{e})(\bar{h} \vee \bar{g})(c \vee \bar{b}) \wedge \\
 &\wedge (e \vee \bar{d})(\bar{g} \vee \bar{f})(\bar{a} \vee \bar{h})(\bar{a} \vee \bar{c})(b \vee \bar{d})(e \vee \bar{g})(\bar{f} \vee \bar{h}) = \\
 &= ((b \vee \bar{a})(\bar{a} \vee \bar{h})(\bar{a} \vee \bar{c}))((d \vee \bar{c})(c \vee \bar{b})(b \vee \bar{d})) \wedge \\
 &\wedge ((\bar{f} \vee \bar{e})(e \vee \bar{d})(e \vee \bar{g}))((\bar{h} \vee \bar{g})(\bar{g} \vee \bar{f})(\bar{f} \vee \bar{h})) = \\
 &= (\bar{a}b \vee \bar{a}b\bar{c} \vee \bar{a}b\bar{h} \vee b\bar{c}\bar{h} \vee \bar{a} \vee \bar{a}\bar{c} \vee \bar{a}\bar{h} \vee \bar{a}\bar{c}\bar{h})(bcd \vee \bar{b}\bar{c}\bar{d}) \wedge \\
 &\wedge (e\bar{f} \vee e\bar{f}\bar{g} \vee \bar{d}e\bar{f} \vee \bar{d}\bar{f}\bar{g} \vee \bar{d}\bar{e}\bar{g})(\bar{f}\bar{g}\bar{h} \vee \bar{g}\bar{h} \vee \bar{f}\bar{h} \vee \bar{f}\bar{g}) = \\
 &= (b\bar{c}\bar{h} \vee \bar{a})(bcd \vee \bar{b}\bar{c}\bar{d})(e\bar{f} \vee \bar{d}\bar{e}\bar{g})(\bar{g}\bar{h} \vee \bar{f}\bar{h} \vee \bar{f}\bar{g}) = \\
 &= (\bar{a}bcd \vee \bar{a}\bar{b}\bar{c}\bar{d})(e\bar{f}\bar{h} \vee e\bar{f}\bar{g} \vee \bar{d}\bar{e}\bar{g}\bar{h} \vee \bar{d}\bar{e}\bar{f}\bar{g}) = \\
 &= \bar{a}bcdef\bar{h} \vee \bar{a}bcdef\bar{g} = \bar{a}bcdef\bar{g}\bar{h} \vee \bar{a}bcdef\bar{g}h \vee \bar{a}bcdef\bar{g}\bar{h} = \\
 &= \bar{a}bcdef\bar{g}\bar{h} \vee \bar{a}bcdef\bar{g}h = 1
 \end{aligned}$$

Як бачимо, в даному випадку отримано дві конституенти одиниці, одна з яких $(\bar{a}bcdef\bar{g}\bar{h})$ відповідає дійсному стану системи. Це означає, що необхідні додаткові тестові перевірки. Втім, навіть отриманий частковий результат дозволяє зробити деякі висновки. Так, в обох конституентах змінні a та f записані з інверсіями, а b , c , d і e – без інверсій, з чого однозначно випливає, що відповідні першим змінним процесори є несправними, а другим – справними. Стан же процесорів g і h – не визначено.

Розглянутий приклад дозволяє дати рекомендації загального характеру: для вирішення невизначеності досить виконати перевірки «підозрюваних» процесорів будь-яким із точно справних. Для розглянутої в прикладі топології зв'язків системи достатньо буде перевірки $b \rightarrow h$. Дійсно, буде отримано результат $r_{13} = 1$, і вираз $R_{13} \triangleq \bar{h} \vee \bar{b}$. Тоді, відповідно до (4.6)

$$V_{13} \triangleq V_{12} \wedge R_{13} = (\bar{a}bcdef\bar{g}\bar{h} \vee \bar{a}bcdef\bar{g}h)(\bar{h} \vee \bar{b}) = \bar{a}bcdef\bar{g}\bar{h} = 1.$$

Таким чином отримане однозначне рішення, що відповідає наступному стану системи: процесори b, c, d, e, g – справні, а процесори a, f, h – несправні. Як і в попередньому випадку, рішення відповідає реальному стану системи.

Приклад 2

Розглянемо ще один приклад (рис. 2). Представлена система, що складається з 8 процесорів, також є 3-діагностованою. Нехай з ладу вийшли процесори b та f .

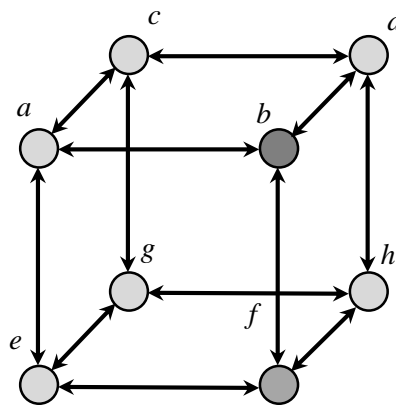


Рис. 2. Граф топології системи для прикладу 2

Нехай були проведені наступні тести: $a \rightarrow b, a \rightarrow c, a \rightarrow e, e \rightarrow a, e \rightarrow f, e \rightarrow g, g \rightarrow c, g \rightarrow e, g \rightarrow h, b \rightarrow f, b \rightarrow a, b \rightarrow d, f \rightarrow b, f \rightarrow e, f \rightarrow h, d \rightarrow c, d \rightarrow h, h \rightarrow d$ та $c \rightarrow d$.

Результати тестів наведені в таблиці 2.

Запишемо, відповідно до (4.8), $V_{19} \triangleq R_1 \wedge R_2 \wedge \dots \wedge R_{19} = 1$ та виразимо його у формі ДДНФ, виключаючи при цьому все кон'юнкції, що містять більш ніж 3 інверсії:

$$\begin{aligned}
 V_{19} &\triangleq R_1 \wedge R_2 \wedge \dots \wedge R_{19} \triangleq (a\bar{b} \vee \bar{a})(ac \vee \bar{a})(ae \vee \bar{a})(ea \vee \bar{e})(e\bar{f} \vee \bar{e}) \wedge \\
 &\wedge (eg \vee \bar{e})(gc \vee \bar{g})(ge \vee \bar{g})(gh \vee \bar{g})(bf \vee \bar{b})(b\bar{a} \vee \bar{b})(bd \vee \bar{b}) \wedge \\
 &\wedge (fb \vee \bar{f})(f\bar{e} \vee \bar{f})(fh \vee \bar{f})(dc \vee \bar{d})(dh \vee \bar{d})(hd \vee \bar{h})(cd \vee \bar{c}) = \\
 &= (a\bar{b}ce \vee \bar{a})(eaf\bar{g} \vee \bar{e})(gceh \vee \bar{g})(b\bar{a}df \vee \bar{b})(fb\bar{e}h \vee \bar{f})(dch \vee \bar{d}) \wedge \\
 &\wedge (hd \vee \bar{h})(cd \vee \bar{c}) = (a\bar{b}cef\bar{g} \vee \bar{a}\bar{e})(cegh \vee \bar{g})(\bar{a}bd\bar{e}fh \vee \bar{b}\bar{f})(cdh \vee \bar{d}\bar{h}) \wedge \\
 &\wedge (cd \vee \bar{c}) = (a\bar{b}cef\bar{g}h \vee \bar{a}\bar{e}\bar{g})(\bar{a}bd\bar{e}fh \vee \bar{b}\bar{f})(cdh \vee \bar{c}\bar{d}\bar{h}) = \\
 &= (a\bar{b}cef\bar{g}h \vee \bar{a}bd\bar{e}f\bar{g}h \vee \bar{a}\bar{b}\bar{e}\bar{g}\bar{f}) (cdh \vee \bar{c}\bar{d}\bar{h}) = a\bar{b}cdef\bar{g}h \vee \bar{a}bcd\bar{e}f\bar{g}h = 1
 \end{aligned}$$

Отриманий вираз складається з двох елементарних кон'юнкцій. Це означає, що можливими є наступні стани системи:

- 1) Справними є всі процесори, окрім b і f .
- 2) Справними є всі процесори, окрім a , e та g .

Таблиця 2

Результати проведених тестових перевірок для прикладу 2

№ тесту, i	Тестуючий процесор	Тестований процесор	Результат тесту, r_i	Вираз R_i
1	a	b	1	$R_1 \triangleq a\bar{b} \vee \bar{a}$
2	a	c	0	$R_2 \triangleq ac \vee \bar{a}$
3	a	e	0	$R_3 \triangleq ae \vee \bar{a}$
4	e	a	0	$R_4 \triangleq ea \vee \bar{e}$
5	e	f	1	$R_5 \triangleq ef \vee \bar{e}$
6	e	g	0	$R_6 \triangleq eg \vee \bar{e}$
7	g	c	0	$R_7 \triangleq gc \vee \bar{g}$
8	g	e	0	$R_8 \triangleq ge \vee \bar{g}$
9	g	h	0	$R_9 \triangleq gh \vee \bar{g}$
10	b	f	0	$R_{10} \triangleq bf \vee \bar{b}$
11	b	a	1	$R_{11} \triangleq b\bar{a} \vee \bar{b}$
12	b	d	0	$R_{12} \triangleq bd \vee \bar{b}$
13	f	b	0	$R_{13} \triangleq fb \vee \bar{f}$
14	f	e	1	$R_{14} \triangleq f\bar{e} \vee \bar{f}$
15	f	h	0	$R_{15} \triangleq fh \vee \bar{f}$
16	d	c	0	$R_{16} \triangleq dc \vee \bar{d}$
17	d	h	0	$R_{17} \triangleq dh \vee \bar{d}$
18	h	d	0	$R_{18} \triangleq hd \vee \bar{h}$
19	c	d	0	$R_{19} \triangleq cd \vee \bar{c}$

Як бачимо, реальний стан системи дійсно знаходиться серед отриманих рішень, однак однозначно визначити його поки не видається можливим. Для вирішення даної невизначеності проведемо додатковий тест: $c \rightarrow a$. Очевидно (оскільки обидва процесора насправді є справними), результат тесту $r_{20} = 0$, і, отже, $R_{20} \triangleq ca \vee \bar{c}$. Згадаємо що $V_{20} \triangleq V_{19} \wedge R_{20}$, тобто

$$V_{20} \triangleq (a\bar{b}cdef\bar{g}h \vee \bar{a}bcd\bar{e}f\bar{g}h)(ca \vee \bar{c}) = a\bar{b}cdef\bar{g}h = 1.$$

Таким чином, після проведення додаткового тесту ми отримали однозначне рішення: справними є всі процесори, окрім b і f , що відповідає реальному стану системи.

Приклад 3

Розглянемо ще один приклад. На рис. 3 представлений граф 2-діагностованої системи, що складається з 5 процесорів. Припустимо, що з ладу вийшли 3 процесора: a , c та d .

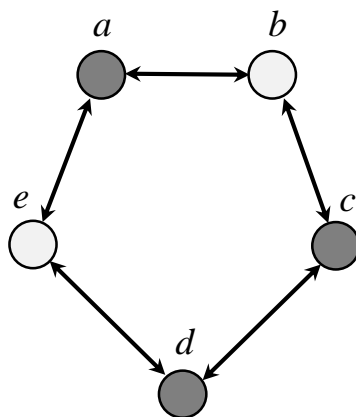


Рис. 3. Граф топології системи для прикладу 3

Одна з можливих комбінацій значень результатів тестів для даної конфігурації системи наведена в таблиці 3.

Таблиця 3

Одна з можливих комбінацій значень результатів тестів для прикладу 3

№ тесту, i	Тестуючий процесор	Тестований процесор	Результат тесту, r_i	Вираз R_i
1	a	b	1	$R_{ab} \triangleq a\bar{b} \vee \bar{a}$
2	b	c	1	$R_{bc} \triangleq b\bar{c} \vee \bar{b}$
3	c	d	0	$R_{cd} \triangleq cd \vee \bar{c}$
4	d	e	1	$R_{de} \triangleq d\bar{e} \vee \bar{d}$
5	e	a	1	$R_{ea} \triangleq e\bar{a} \vee \bar{e}$
6	e	d	1	$R_{ed} \triangleq e\bar{d} \vee \bar{e}$
7	d	c	1	$R_{dc} \triangleq d\bar{c} \vee \bar{d}$
8	c	b	1	$R_{cb} \triangleq c\bar{b} \vee \bar{c}$
9	b	a	1	$R_{ba} \triangleq b\bar{a} \vee \bar{b}$
10	a	e	1	$R_{ae} \triangleq a\bar{e} \vee \bar{a}$

Нехай були проведені наступні 5 тестів: $a \rightarrow b$, $b \rightarrow a$, $c \rightarrow d$, $d \rightarrow e$ та $e \rightarrow d$. Запишемо: $V_5 \triangleq R_{ab} \wedge R_{ba} \wedge R_{cd} \wedge R_{de} \wedge R_{ed} = 1$. Виразимо V_5 в формі ДДНФ, виключаючи при цьому всі кон'юнкції, що містять більш ніж 2 інверсії:

$$\begin{aligned} V_5 &\triangleq R_{ab} \wedge R_{ba} \wedge R_{cd} \wedge R_{de} \wedge R_{ed} \triangleq (a\bar{b} \vee \bar{a}b)(b\bar{a} \vee \bar{b}a)(cd \vee \bar{c}\bar{d})(d\bar{e} \vee \bar{d}\bar{e}) \wedge \\ &\wedge (e\bar{d} \vee \bar{e}d) = (a\bar{b} \vee b\bar{a} \vee \bar{a}\bar{b})(cd \vee \bar{c})(d\bar{e} \vee \bar{d}e \vee \bar{d}\bar{e}) = (a\bar{b} \vee \bar{a}b \vee \bar{a}\bar{b}) \wedge \\ &\wedge (cd\bar{e} \vee \bar{c}d\bar{e} \vee \bar{c}\bar{d}\bar{e} \vee \bar{c}\bar{d}e) = a\bar{b}cd\bar{e} \vee \bar{a}\bar{b}\bar{c}d\bar{e} \vee \bar{a}\bar{b}\bar{c}de \vee \bar{a}b\bar{c}d\bar{e} \vee \bar{a}b\bar{c}de \vee \\ &\vee \bar{a}\bar{b}cde \vee \bar{a}\bar{b}cd\bar{e} \vee \bar{a}\bar{b}\bar{c}de = a\bar{b}cd\bar{e} \vee \bar{a}b\bar{c}d\bar{e} = 1. \end{aligned}$$

Як бачимо, в даній ситуації отриманий вираз складається з двох елементарних кон'юнкцій, що мало б відповідати двом можливим станам системи:

- 1) Справними є всі процесори, окрім b та e .
- 2) Справними є всі процесори, окрім a та e .

Очевидно, жодне з отриманих рішень не відображає реального стану системи.

Якщо ж були проведені інші 5 тестів: $a \rightarrow b$, $b \rightarrow c$, $c \rightarrow d$, $d \rightarrow e$ та $e \rightarrow a$, то рівняння набуде вигляду: $V_5 \triangleq R_{ab} \wedge R_{bc} \wedge R_{cd} \wedge R_{de} \wedge R_{ea} = 1$. Виразимо V_5 в формі ДДНФ, виключаючи при цьому всі кон'юнкції, що містять більш, ніж 2 інверсії:

$$\begin{aligned} V_5 &\triangleq R_{ab} \wedge R_{bc} \wedge R_{cd} \wedge R_{de} \wedge R_{ea} \triangleq (a\bar{b} \vee \bar{a}b)(b\bar{c} \vee \bar{b}c)(cd \vee \bar{c}\bar{d})(d\bar{e} \vee \bar{d}\bar{e}) \wedge \\ &\wedge (e\bar{a} \vee \bar{e}a) = (a\bar{b} \vee \bar{a}b\bar{c} \vee \bar{a}\bar{b}c)(cd \vee \bar{c})(d\bar{e} \vee \bar{d}e \vee \bar{d}\bar{e}) = (a\bar{b} \vee \bar{a}b\bar{c} \vee \bar{a}\bar{b}c) \wedge \\ &\wedge (cd\bar{e} \vee \bar{c}d\bar{e} \vee \bar{c}\bar{d}\bar{e} \vee \bar{c}\bar{d}e) = a\bar{b}cd\bar{e} \vee \bar{a}\bar{b}\bar{c}d\bar{e} \vee \bar{a}\bar{b}\bar{c}de \vee \bar{a}b\bar{c}d\bar{e} \vee \bar{a}b\bar{c}de \vee \\ &\vee \bar{a}\bar{b}cde \vee \bar{a}\bar{b}cd\bar{e} \vee \bar{a}\bar{b}\bar{c}de = a\bar{b}cd\bar{e} = 1. \end{aligned}$$

В даному випадку отриманий вираз містить єдину елементарну кон'юнкцію, що мало б однозначно визначати стан системи: справні всі процесори, крім b та e . Проте, очевидно, отримане рішення не відповідає реальному стану системи.

Тепер розглянемо ще один набір з 5 тестів: $a \rightarrow e$, $b \rightarrow a$, $c \rightarrow b$, $d \rightarrow c$ та $e \rightarrow d$. В цьому випадку рівняння прийме наступний вигляд: $V_5 \triangleq R_{ae} \wedge$

$R_{ba} \wedge R_{cb} \wedge R_{dc} \wedge R_{ed} = 1$. Виразимо V_5 в формі ДДНФ, виключаючи при цьому всі кон'юнкції, що містять більш, ніж 2 інверсії:

$$\begin{aligned} V_5 &\triangleq R_{ae} \wedge R_{ba} \wedge R_{cb} \wedge R_{dc} \wedge R_{ed} \triangleq (a\bar{e} \vee \bar{a})(b\bar{a} \vee \bar{b})(c\bar{b} \vee \bar{c})(d\bar{c} \vee \bar{d}) \wedge \\ &\wedge (e\bar{d} \vee \bar{e}) = (a\bar{b}\bar{e} \vee \bar{a}b \vee \bar{a}\bar{b})(\bar{b}c\bar{d} \vee \bar{c}d \vee \bar{c}\bar{d})(e\bar{d} \vee \bar{e}) = (a\bar{b}\bar{e} \vee \bar{a}b \vee \bar{a}\bar{b}) \wedge \\ &\wedge (\bar{b}c\bar{d}e \vee \bar{b}c\bar{d}\bar{e} \vee \bar{c}d\bar{e} \vee \bar{c}\bar{d}e \vee \bar{e}\bar{d}\bar{e}) = a\bar{b}\bar{c}d\bar{e} \vee \bar{a}b\bar{c}d\bar{e} \vee \bar{a}b\bar{c}\bar{d}e \vee \bar{a}\bar{b}c\bar{d}e \vee \\ &\vee \bar{a}\bar{b}\bar{c}d\bar{e} \vee \bar{a}\bar{b}\bar{c}\bar{d}e = 0 = 1. \end{aligned}$$

В даному випадку не існує жодного рішення, що відповідає реальній ситуації, коли кількість відмов перевищує допустиму (тобто, для даної системи – більш ніж 2 відмови, оскільки система є 2-діагностованою і ми, відповідно, виключали всі кон'юнкції, що містили більш ніж 2 інверсії).

Як бачимо, в залежності від вибору набору тестів та їх результатів були отримані рівняння кожного з описаних в параграфі 4.1 видів. Варто зазначити, що для даної комбінації значень результатів тестів проведення додаткових тестових випробувань (аж до всіх можливих) дозволить дійти до рівняння виду $0 = 1$, що однозначно говорить про те, що кількість реально несправних процесорів перевищує передбачуване максимальне значення. В той же час, це справедливо далеко не для всіх випадків. Зокрема, комбінація значень результатів тестів, наведена в таблиці 4, яка також можлива для даного набору станів процесорів системи, не дозволить коректно встановити реальний стан системи.

Припустимо, що були проведені всі можливі тести, і рівняння (4.8) матиме такий вигляд: $V_{10} \triangleq R_1 \wedge R_2 \wedge \dots \wedge R_{10} = 1$. Виразимо V_{10} в формі ДДНФ, виключаючи при цьому всі кон'юнкції, які містять більш, ніж 2 інверсії:

$$\begin{aligned} V_{10} &\triangleq R_1 \wedge R_2 \wedge \dots \wedge R_{10} \triangleq \\ &\triangleq (a\bar{b} \vee \bar{a})(b\bar{c} \vee \bar{b})(c\bar{d} \vee \bar{c})(d\bar{e} \vee \bar{d})(e\bar{a} \vee \bar{e})(e\bar{d} \vee \bar{e}) \wedge \\ &\wedge (d\bar{c} \vee \bar{d})(c\bar{b} \vee \bar{c})(b\bar{a} \vee \bar{b})(a\bar{e} \vee \bar{a}) = \\ &= (a\bar{b}\bar{e} \vee \bar{a})(\bar{a}b\bar{c} \vee \bar{b})(\bar{b}c\bar{d} \vee \bar{c})(c\bar{d}\bar{e} \vee \bar{d}) \wedge \\ &\wedge (\bar{a}\bar{d}e \vee \bar{e}) = (a\bar{b}\bar{e} \vee \bar{a}b\bar{c} \vee \bar{a}\bar{b})(\bar{b}c\bar{d}\bar{e} \vee \bar{c}\bar{d})(\bar{a}\bar{d}e \vee \bar{e}) = \end{aligned}$$

$$= (\bar{a}\bar{b}\bar{e} \vee \bar{a}b\bar{c}\bar{d}\bar{e} \vee \bar{a}b\bar{c}\bar{e} \vee \bar{a}\bar{b}\bar{d}\bar{e} \vee \bar{a}\bar{b}\bar{e})(\bar{b}cd\bar{e} \vee \bar{c}\bar{d}) = \bar{a}\bar{b}cd\bar{e} \vee \bar{a}\bar{b}\bar{c}\bar{d}\bar{e} = \\ = \bar{a}\bar{b}cd\bar{e} = 1.$$

Таблиця 4

Інша можлива комбінація значень результатів тестів для прикладу 3

№ тесту, i	Тестуючий процесор	Тестований процесор	Результат тесту, r_i	Вираз R_i
1	a	b	1	$R_1 \triangleq a\bar{b} \vee \bar{a}$
2	b	c	1	$R_2 \triangleq b\bar{c} \vee \bar{b}$
3	c	d	0	$R_3 \triangleq cd \vee \bar{c}$
4	d	e	1	$R_4 \triangleq d\bar{e} \vee \bar{d}$
5	e	a	1	$R_5 \triangleq e\bar{a} \vee \bar{e}$
6	e	d	1	$R_6 \triangleq ed \vee \bar{e}$
7	d	c	0	$R_7 \triangleq dc \vee \bar{d}$
8	c	b	1	$R_8 \triangleq c\bar{b} \vee \bar{c}$
9	b	a	1	$R_9 \triangleq b\bar{a} \vee \bar{b}$
10	a	e	1	$R_{10} \triangleq a\bar{e} \vee \bar{a}$

Як бачимо, навіть після проведення всіх можливих відповідно до топології системи тестових перевірок, рішення рівняння не відповідає реальному стану системи. Причина цього одна – число реальних несправностей перевищило допустиме значення.

Додаток В

Модель ієрархічної системи

Структура системи представлена на рис. 1. Вважатимемо, що станам процесорів першої підсистеми відповідають змінні x_1, x_2, \dots, x_{11} , другої – змінні $x_{12}, x_{13}, \dots, x_{20}$, третьої – $x_{21}, x_{22}, \dots, x_{32}$, четвертої – $x_{33}, x_{34}, \dots, x_{39}$, а п'ятої – $x_{40}, x_{41}, \dots, x_{54}$.

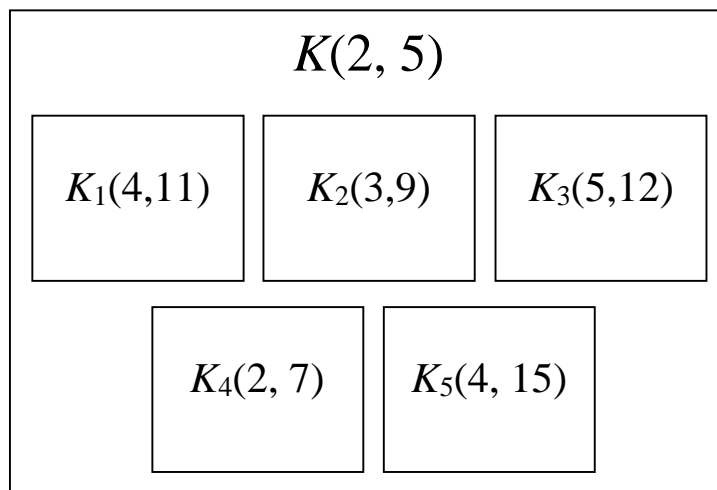


Рис. 1. Структура системи

Модель першої підсистеми $K_1(4, 11)$ міститиме наступні реберні функції:

$$f_1^1 = x_1 \vee x_2 \vee x_3 \vee x_4 x_5 x_6$$

$$f_2^1 = (x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee (x_4 \vee x_5)(x_4 x_5 \vee x_6)$$

$$f_3^1 = x_1 x_2 x_3 \vee x_4 \vee x_5 \vee x_6$$

$$f_4^1 = (x_1 \vee x_2 \vee x_3)((x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4 x_5 x_6) \wedge$$

$$\wedge (x_1 x_2 x_3 \vee (x_4 \vee x_5)(x_4 x_5 \vee x_6))(x_4 \vee x_5 \vee x_6) \vee x_7 x_8 x_9 x_{10} x_{11}$$

$$f_5^1 = (x_1 \vee x_2)(x_1 x_2 \vee x_3)(x_1 x_2 x_3 \vee x_4 x_5 x_6)(x_4 \vee x_5)(x_4 x_5 \vee x_6) \vee$$

$$\vee (x_7 \vee x_8)(x_7 x_8 \vee x_9)(x_7 x_8 x_9 \vee x_{10} x_{11})(x_{10} \vee x_{11})$$

$$f_6^1 = x_1 x_2 x_3 x_4 x_5 x_6 \vee (x_7 \vee x_8 \vee x_9)((x_7 \vee x_8)(x_7 x_8 \vee x_9) \vee x_{10} x_{11}) \wedge$$

$$\wedge (x_7 x_8 x_9 \vee x_{10} \vee x_{11})$$

$$f_7^1 = x_7 \vee x_8 \vee x_9 \vee x_{10} x_{11}$$

$$f_8^1 = (x_7 \vee x_8)(x_7 x_8 \vee x_9) \vee x_{10} \vee x_{11}$$

Другій підсистемі відповідатиме модель $K_2(3, 9)$ з реберними функціями:

$$f_1^2 = x_{12} \vee x_{13} \vee x_{14}$$

$$f_2^2 = (x_{12} \vee x_{13})(x_{12}x_{13} \vee x_{14}) \vee x_{15}x_{16}$$

$$f_3^2 = x_{12}x_{13}x_{14} \vee x_{15} \vee x_{16}$$

$$f_4^2 = (x_{12} \vee x_{13})(x_{12}x_{13} \vee x_{14})(x_{12}x_{13}x_{14} \vee x_{15}x_{16})(x_{15} \vee x_{16}) \vee x_{17}x_{18}x_{19}x_{20}$$

$$f_5^2 = x_{12}x_{13}x_{14}x_{15}x_{16} \vee (x_{17} \vee x_{18})(x_{19} \vee x_{20})$$

$$f_6^2 = x_{17} \vee x_{18} \vee x_{19}x_{20}$$

$$f_7^2 = x_{17}x_{18} \vee x_{19} \vee x_{20}$$

Третій підсистемі відповідатиме модель $K_3(5, 12)$ з реберними функціями:

$$f_1^3 = x_{21} \vee x_{22} \vee x_{23} \vee (x_{24} \vee x_{25})(x_{24}x_{25} \vee x_{26})$$

$$f_2^3 = (x_{21} \vee x_{22})(x_{21}x_{22} \vee x_{23}) \vee x_{24} \vee x_{25} \vee x_{26}$$

$$f_3^3 = (x_{21} \vee x_{22} \vee x_{23} \vee x_{24}x_{25}x_{26}) \wedge$$

$$\wedge ((x_{21} \vee x_{22})(x_{21}x_{22} \vee x_{23}) \vee (x_{24} \vee x_{25})(x_{24}x_{25} \vee x_{26})) \wedge$$

$$\wedge (x_{21}x_{22}x_{23} \vee x_{24} \vee x_{25} \vee x_{26}) \vee x_{27}x_{28}x_{29}x_{30}x_{32}x_{32}$$

$$f_4^3 = (x_{21} \vee x_{22} \vee x_{23})((x_{21} \vee x_{22})(x_{21}x_{22} \vee x_{23}) \vee x_{24}x_{25}x_{26}) \wedge$$

$$\wedge (x_{21}x_{22}x_{23} \vee (x_{24} \vee x_{25})(x_{24}x_{25} \vee x_{26}))(x_{24} \vee x_{25} \vee x_{26}) \vee$$

$$\vee (x_{27} \vee x_{28})(x_{27}x_{28} \vee x_{29})(x_{27}x_{28}x_{29} \vee x_{30}x_{32}x_{32})(x_{30} \vee x_{31})(x_{30}x_{31} \vee x_{32})$$

$$f_5^3 = (x_{21} \vee x_{22})(x_{21}x_{22} \vee x_{23})(x_{21}x_{22}x_{23} \vee x_{24}x_{25}x_{26})(x_{24} \vee x_{25}) \wedge$$

$$\wedge (x_{24}x_{25} \vee x_{26}) \vee (x_{27} \vee x_{28} \vee x_{29})((x_{27} \vee x_{28})(x_{27}x_{28} \vee x_{29}) \vee x_{30}x_{32}x_{32}) \wedge$$

$$\wedge (x_{27}x_{28}x_{29} \vee (x_{30} \vee x_{31})(x_{30}x_{31} \vee x_{32}))(x_{30} \vee x_{31} \vee x_{32})$$

$$f_6^3 = x_{21}x_{22}x_{23}x_{24}x_{25}x_{26} \vee (x_{27} \vee x_{28} \vee x_{29} \vee x_{30}x_{32}x_{32}) \wedge$$

$$\wedge ((x_{27} \vee x_{28})(x_{27}x_{28} \vee x_{29}) \vee (x_{30} \vee x_{31})(x_{30}x_{31} \vee x_{32})) \wedge$$

$$\wedge (x_{27}x_{28}x_{29} \vee x_{30} \vee x_{31} \vee x_{32})$$

$$f_7^3 = x_{27} \vee x_{28} \vee x_{29} \vee (x_{30} \vee x_{31})(x_{30}x_{31} \vee x_{32})$$

$$f_8^3 = (x_{27} \vee x_{28})(x_{27}x_{28} \vee x_{29}) \vee x_{30} \vee x_{31} \vee x_{32}$$

Четвертій підсистемі відповідатиме модель $K_4(2, 7)$ з реберними функціями:

$$f_1^4 = x_{33} \vee x_{34}$$

$$f_2^4 = x_{33}x_{34} \vee x_{35}x_{36}$$

$$f_3^4 = x_{35} \vee x_{36}$$

$$f_4^4 = x_{33}x_{34}x_{35}x_{36} \vee x_{37}x_{38}x_{39}$$

$$f_5^4 = x_{37} \vee x_{38}$$

$$f_6^4 = x_{37}x_{38} \vee x_{39}$$

П'ятій підсистемі відповідатиме модель $K_5(4, 15)$ з реберними функціями:

$$f_1^5 = x_{40} \vee x_{41} \vee x_{42} \vee x_{43}$$

$$f_2^5 = (x_{40} \vee x_{41} \vee x_{42}x_{43})(x_{40}x_{41} \vee x_{42} \vee x_{43}) \vee x_{44}x_{45}x_{46}x_{47}$$

$$f_3^5 = (x_{40} \vee x_{41})(x_{40}x_{41} \vee x_{42}x_{43})(x_{42} \vee x_{43}) \vee$$

$$\vee (x_{44} \vee x_{45})(x_{44}x_{45} \vee x_{46}x_{47})(x_{46} \vee x_{47})$$

$$f_4^5 = x_{40}x_{41}x_{42}x_{43} \vee (x_{44} \vee x_{45} \vee x_{46}x_{47})(x_{44}x_{45} \vee x_{46} \vee x_{47})$$

$$f_5^5 = x_{44} \vee x_{45} \vee x_{46} \vee x_{47}$$

$$f_6^5 = (x_{40} \vee x_{41} \vee x_{42}x_{43})(x_{40}x_{41} \vee x_{42} \vee x_{43}) \wedge$$

$$\wedge ((x_{40} \vee x_{41})(x_{40}x_{41} \vee x_{42}x_{43})(x_{42} \vee x_{43}) \vee x_{44}x_{45}x_{46}x_{47}) \wedge$$

$$\wedge (x_{40}x_{41}x_{42}x_{43} \vee (x_{44} \vee x_{45})(x_{44}x_{45} \vee x_{46}x_{47})(x_{46} \vee x_{47})) \wedge$$

$$\wedge (x_{44} \vee x_{45} \vee x_{46}x_{47})(x_{44}x_{45} \vee x_{46} \vee x_{47}) \vee x_{48}x_{49}x_{50}x_{51}x_{52}x_{53}x_{54}$$

$$f_7^5 = (x_{40} \vee x_{41})(x_{40}x_{41} \vee x_{42}x_{43})(x_{42} \vee x_{43})(x_{40}x_{41}x_{42}x_{43} \vee x_{44}x_{45}x_{46}x_{47}) \wedge$$

$$\wedge (x_{44} \vee x_{45})(x_{44}x_{45} \vee x_{46}x_{47})(x_{46} \vee x_{47}) \vee (x_{48} \vee x_{49})(x_{48}x_{49} \vee x_{50}x_{51}) \wedge$$

$$\wedge (x_{50} \vee x_{51})(x_{48}x_{49}x_{50}x_{51} \vee x_{52}x_{53}x_{54})(x_{52} \vee x_{53})(x_{52}x_{53} \vee x_{54})$$

$$f_8^5 = x_{40}x_{41}x_{42}x_{43}x_{44}x_{45}x_{46}x_{47} \vee (x_{48} \vee x_{49} \vee x_{50}x_{51})(x_{48}x_{49} \vee x_{50} \vee x_{51}) \wedge$$

$$\wedge ((x_{48} \vee x_{49})(x_{48}x_{49} \vee x_{50}x_{51})(x_{50} \vee x_{51}) \vee x_{52}x_{53}x_{54}) \wedge$$

$$\wedge (x_{48}x_{49}x_{50}x_{51} \vee (x_{52} \vee x_{53})(x_{52}x_{53} \vee x_{54}))(x_{52} \vee x_{53} \vee x_{54})$$

$$f_9^5 = x_{48} \vee x_{49} \vee x_{50} \vee x_{51}$$

$$f_{10}^5 = (x_{48} \vee x_{49} \vee x_{50}x_{51})(x_{48}x_{49} \vee x_{50} \vee x_{51}) \vee x_{52}x_{53}x_{54}$$

$$f_{11}^5 = (x_{48} \vee x_{49})(x_{48}x_{49} \vee x_{50}x_{51})(x_{50} \vee x_{51}) \vee (x_{52} \vee x_{53})(x_{52}x_{53} \vee x_{54})$$

$$f_{12}^5 = x_{48}x_{49}x_{50}x_{51} \vee x_{52} \vee x_{53} \vee x_{54}$$

Системі в цілому відповідатиме модель $K(2, 5)$ з реберними функціями:

$$f_1 = y_1 \vee y_2$$

$$f_2 = y_1y_2 \vee y_3$$

$$f_3 = y_1y_2y_3 \vee y_4y_5$$

$$f_4 = y_4 \vee y_5$$

Отримана модель представлена на рис. 2.

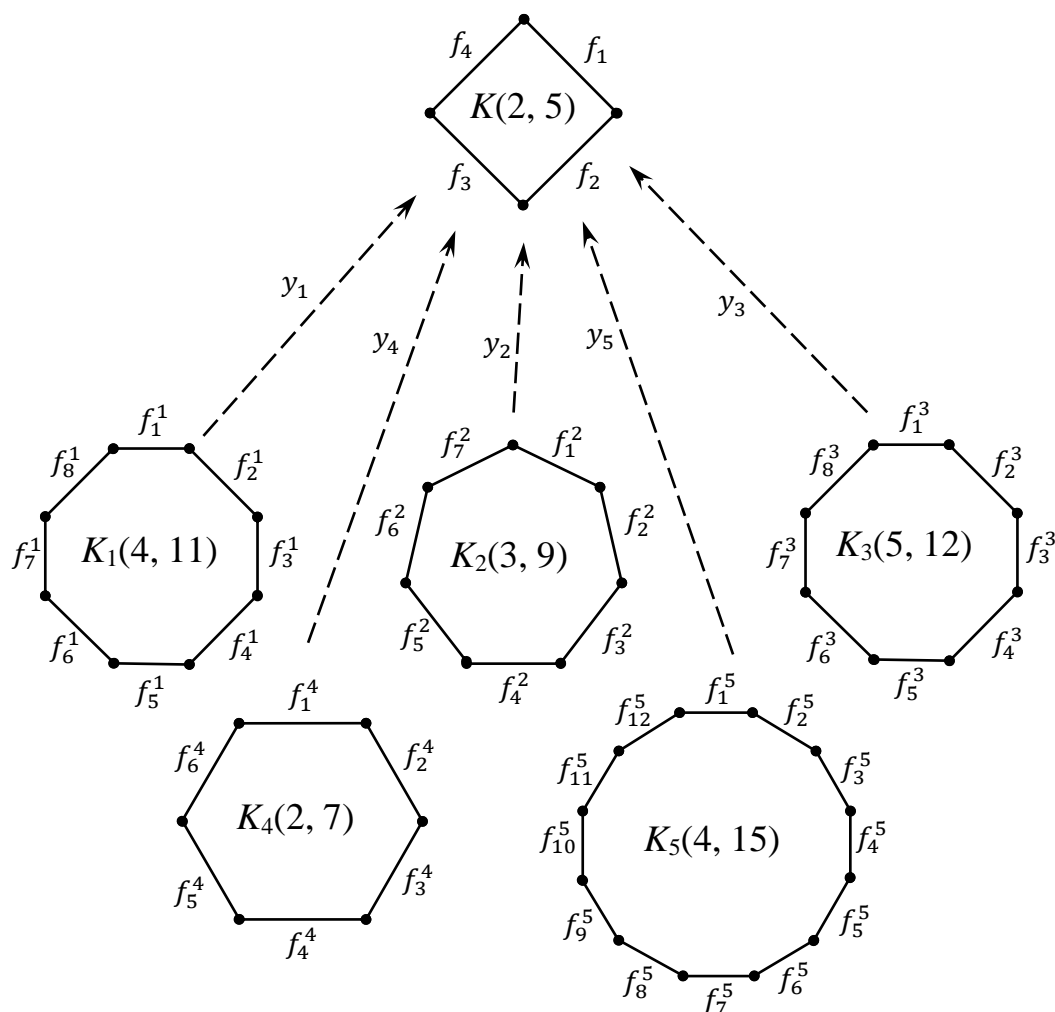


Рис. 2. Модель ієрархічної системи

Додаток Г

Модель ієрархічної системи з додатковим рівнем ієрархії

Структура системи представлена на рис. 1. Вважатимемо, що станам процесорів підсистеми 5.1 відповідають змінні $x_{40}, x_{41}, \dots, x_{47}$, підсистеми 5.2 – змінні $x_{48}, x_{49}, x_{50}, x_{51}$, підсистеми 5.3 – $x_{52}, x_{53}, \dots, x_{59}$, підсистеми 5.4 – $x_{60}, x_{61}, \dots, x_{64}$, підсистеми 5.5 – $x_{65}, x_{66}, \dots, x_{71}$ та підсистеми 5.6 – $x_{72}, x_{73}, \dots, x_{80}$.

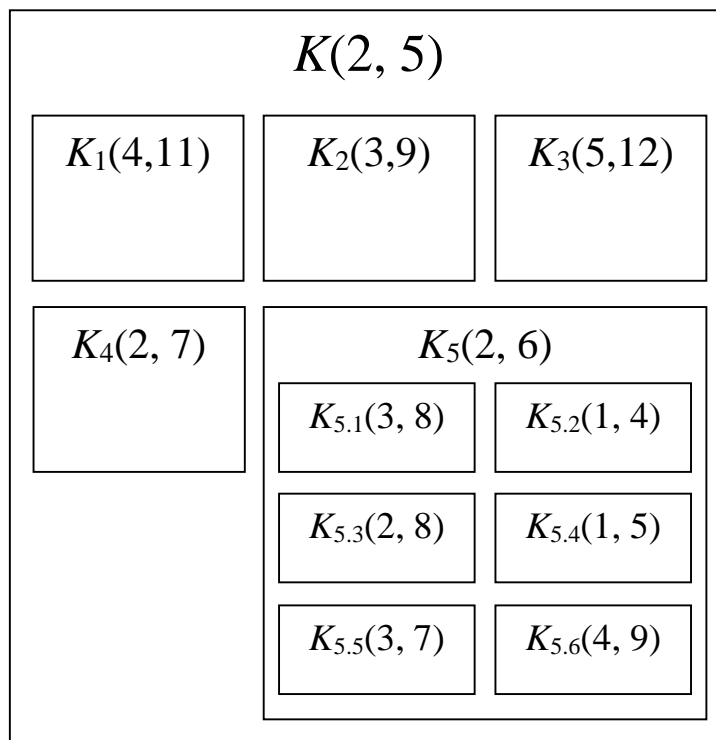


Рис. 1. Структура ієрархічної системи з додатковим рівнем ієрархії

Модель $K_{5.1}(3, 8)$ підсистеми 5.1 міститиме наступні реберні функції:

$$f_1^{5.1} = x_{40} \vee x_{41} \vee x_{42}x_{43}$$

$$f_2^{5.1} = x_{40}x_{41} \vee x_{42} \vee x_{43}$$

$$f_3^{5.1} = (x_{40} \vee x_{41})(x_{40}x_{41} \vee x_{42}x_{43})(x_{42} \vee x_{43}) \vee x_{44}x_{45}x_{46}x_{47}$$

$$f_4^{5.1} = x_{40}x_{41}x_{42}x_{43} \vee (x_{44} \vee x_{45})(x_{44}x_{45} \vee x_{46}x_{47})(x_{46} \vee x_{47})$$

$$f_5^{5.1} = x_{44} \vee x_{45} \vee x_{46}x_{47}$$

$$f_6^{5.1} = x_{44}x_{45} \vee x_{46} \vee x_{47}$$

Модель $K_{5.2}(1, 4)$ підсистеми 5.2 міститиме наступні реберні функції:

$$f_1^{5.2} = x_{48}$$

$$f_2^{5.2} = x_{49}$$

$$f_3^{5.2} = x_{50}$$

$$f_4^{5.2} = x_{51}$$

Модель $K_{5.3}(2, 8)$ підсистеми 5.3 міститиме наступні реберні функції:

$$f_1^{5.3} = x_{52} \vee x_{53}$$

$$f_2^{5.3} = x_{52}x_{53} \vee x_{54}x_{55}$$

$$f_3^{5.3} = x_{54} \vee x_{55}$$

$$f_4^{5.3} = x_{52}x_{53}x_{54}x_{55} \vee x_{56}x_{57}x_{58}x_{59}$$

$$f_5^{5.3} = x_{56} \vee x_{57}$$

$$f_6^{5.3} = x_{56}x_{57} \vee x_{58}x_{59}$$

$$f_7^{5.3} = x_{58} \vee x_{59}$$

Модель $K_{5.4}(1, 5)$ підсистеми 5.4 міститиме наступні реберні функції:

$$f_1^{5.4} = x_{60}$$

$$f_2^{5.4} = x_{61}$$

$$f_3^{5.4} = x_{62}$$

$$f_4^{5.4} = x_{63}$$

$$f_5^{5.4} = x_{64}$$

Модель $K_{5.5}(3, 7)$ підсистеми 5.5 міститиме наступні реберні функції:

$$f_1^{5.5} = x_{65} \vee x_{66} \vee x_{67}x_{68}$$

$$f_2^{5.5} = x_{65}x_{66} \vee x_{67} \vee x_{68}$$

$$f_3^{5.5} = (x_{65} \vee x_{66})(x_{65}x_{66} \vee x_{67}x_{68})(x_{67} \vee x_{68}) \vee x_{69}x_{70}x_{71}$$

$$f_4^{5.5} = x_{65}x_{66}x_{67}x_{68} \vee (x_{69} \vee x_{70})(x_{69}x_{70} \vee x_{71})$$

$$f_5^{5.5} = x_{69} \vee x_{70} \vee x_{71}$$

Модель $K_{5.6}(4, 9)$ підсистеми 5.6 міститиме наступні реберні функції:

$$f_1^{5.6} = x_{72} \vee x_{73} \vee x_{74} \vee x_{75}x_{76}$$

$$f_2^{5.6} = (x_{72} \vee x_{73})(x_{72}x_{73} \vee x_{74}) \vee x_{75} \vee x_{76}$$

$$f_3^{5.6} = (x_{72} \vee x_{73} \vee x_{74})((x_{72} \vee x_{73})(x_{72}x_{73} \vee x_{74}) \vee x_{75}x_{76}) \wedge$$

$$\wedge (x_{72}x_{73}x_{74} \vee x_{75} \vee x_{76}) \vee x_{77}x_{78}x_{79}x_{80}$$

$$f_4^{5.6} = (x_{72} \vee x_{73})(x_{72}x_{73} \vee x_{74})(x_{72}x_{73}x_{74} \vee x_{75}x_{76})(x_{75} \vee x_{76}) \vee$$

$$\vee (x_{77} \vee x_{78})(x_{77}x_{78} \vee x_{79}x_{80})(x_{79} \vee x_{80})$$

$$f_5^{5.6} = x_{72}x_{73}x_{74}x_{75}x_{76} \vee (x_{77} \vee x_{78} \vee x_{79}x_{80})(x_{77}x_{78} \vee x_{79} \vee x_{80})$$

$$f_6^{5.6} = x_{77} \vee x_{78} \vee x_{79} \vee x_{80}$$

Модель $K_5(2, 6)$ підсистеми 5 міститиме наступні реберні функції:

$$f_1^5 = z_1 \vee z_2$$

$$f_2^5 = z_1z_2 \vee z_3$$

$$f_3^5 = z_1z_2z_3 \vee z_4z_5z_6$$

$$f_4^5 = z_4 \vee z_5$$

$$f_5^5 = z_4z_5 \vee z_6$$

Отримана модель представлена на рис. 2.

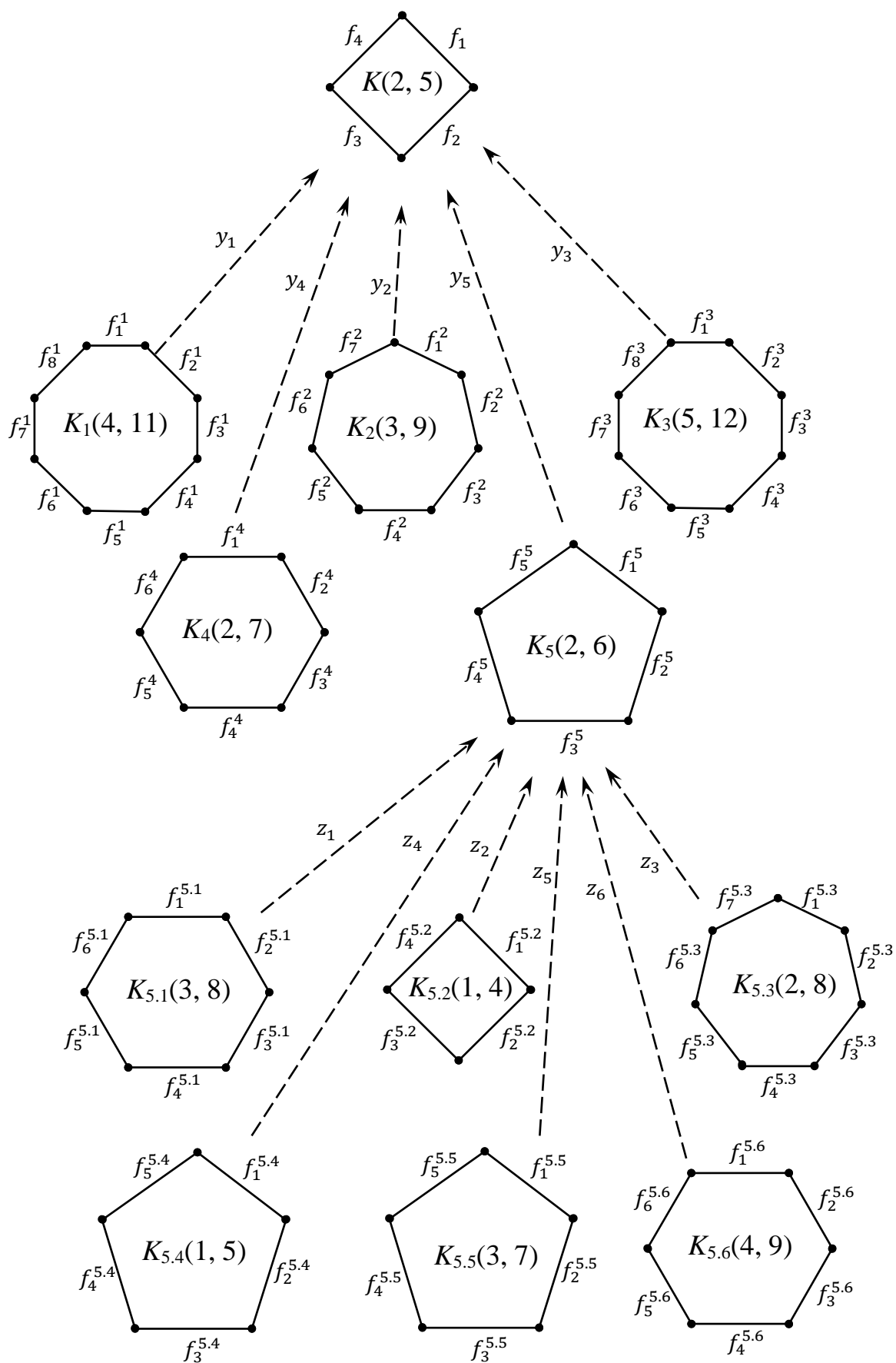


Рис. 2. Модель ієрархічної системи з додатковим рівнем ієрархії

Додаток Д

Модель ієрархічної системи з процесорами на всіх рівнях ієрархії

Структура системи наведена на рис. 1. Для побудови моделі її підсистема верхнього рівня представляється у вигляді двох підсистем (рис. 2). Вважатимемо, що станам процесорів підсистеми верхнього рівня ієрархії відповідають змінні $x_{55}, x_{56}, \dots, x_{64}$.

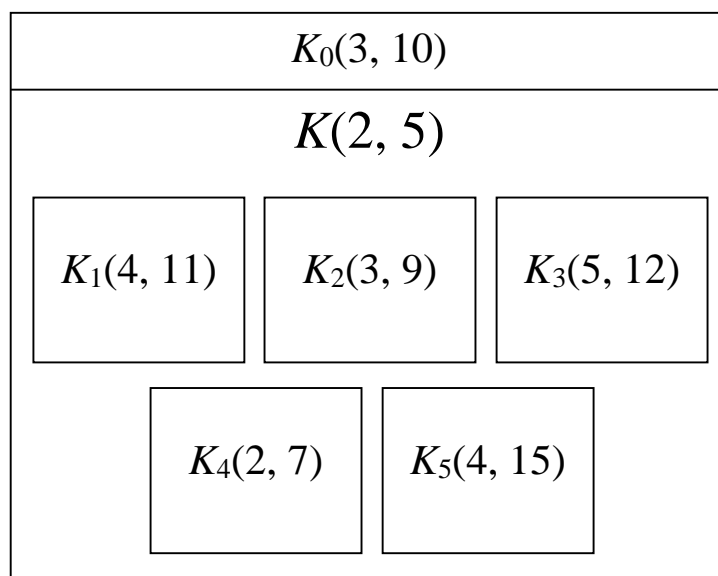


Рис. 1. Структура ієрархічної системи з процесорами
на всіх рівнях ієрархії

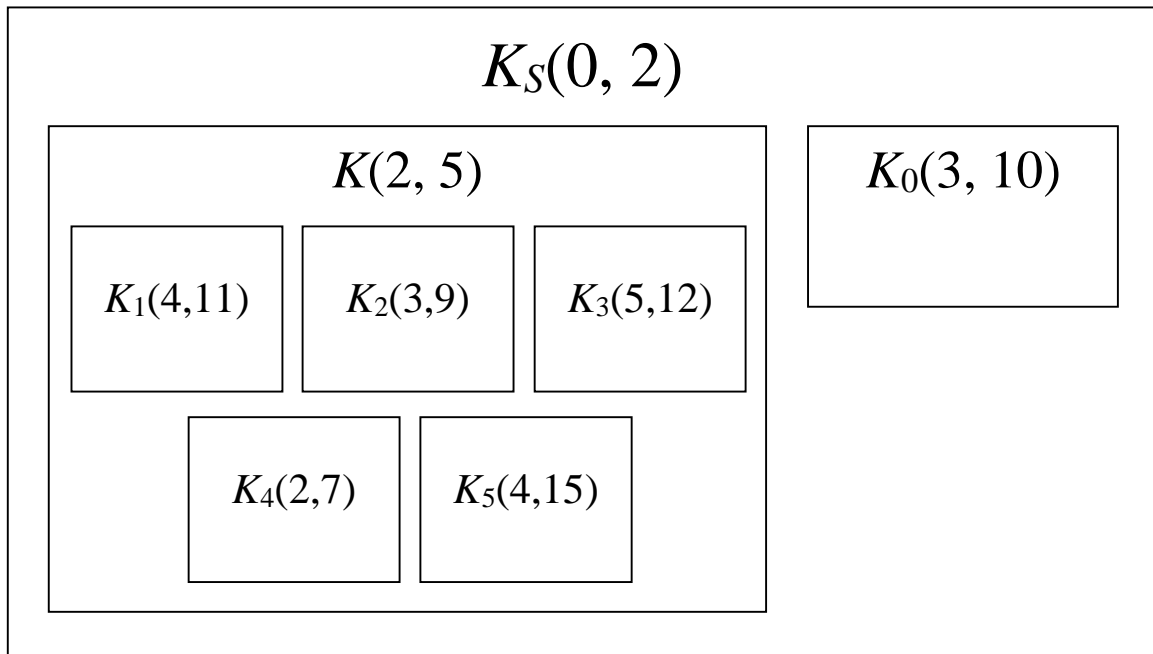


Рис. 2. Структура ієрархічної системи, підсистема верхнього рівня якої представлена у вигляді двох підсистем

Модель $K_0(3, 10)$ міститиме наступні реберні функції:

$$f_1^0 = x_{55} \vee x_{56} \vee x_{57}$$

$$f_2^0 = (x_{55} \vee x_{56})(x_{55}x_{56} \vee x_{57}) \vee x_{58}x_{59}$$

$$f_3^0 = x_{55}x_{56}x_{57} \vee x_{58} \vee x_{59}$$

$$f_4^0 = (x_{55} \vee x_{56})(x_{55}x_{56} \vee x_{57})(x_{55}x_{56}x_{57} \vee x_{58}x_{59})(x_{58} \vee x_{59}) \vee x_{60}x_{61}x_{62}x_{63}x_{64}$$

$$f_5^0 = x_{55}x_{56}x_{57}x_{58}x_{59} \vee$$

$$\vee (x_{60} \vee x_{61})(x_{60}x_{61} \vee x_{62})(x_{60}x_{61}x_{62} \vee x_{63}x_{64})(x_{63} \vee x_{64})$$

$$f_6^0 = x_{60} \vee x_{61} \vee x_{62}$$

$$f_7^0 = (x_{60} \vee x_{61})(x_{60}x_{61} \vee x_{62}) \vee x_{63}x_{64}$$

$$f_8^0 = x_{60}x_{61}x_{62} \vee x_{63} \vee x_{64}$$

Отримана модель представлена на рис. 3. Виключене ребро, що відповідає функції $F_3 = 0$ зображене за допомогою пунктирної лінії.

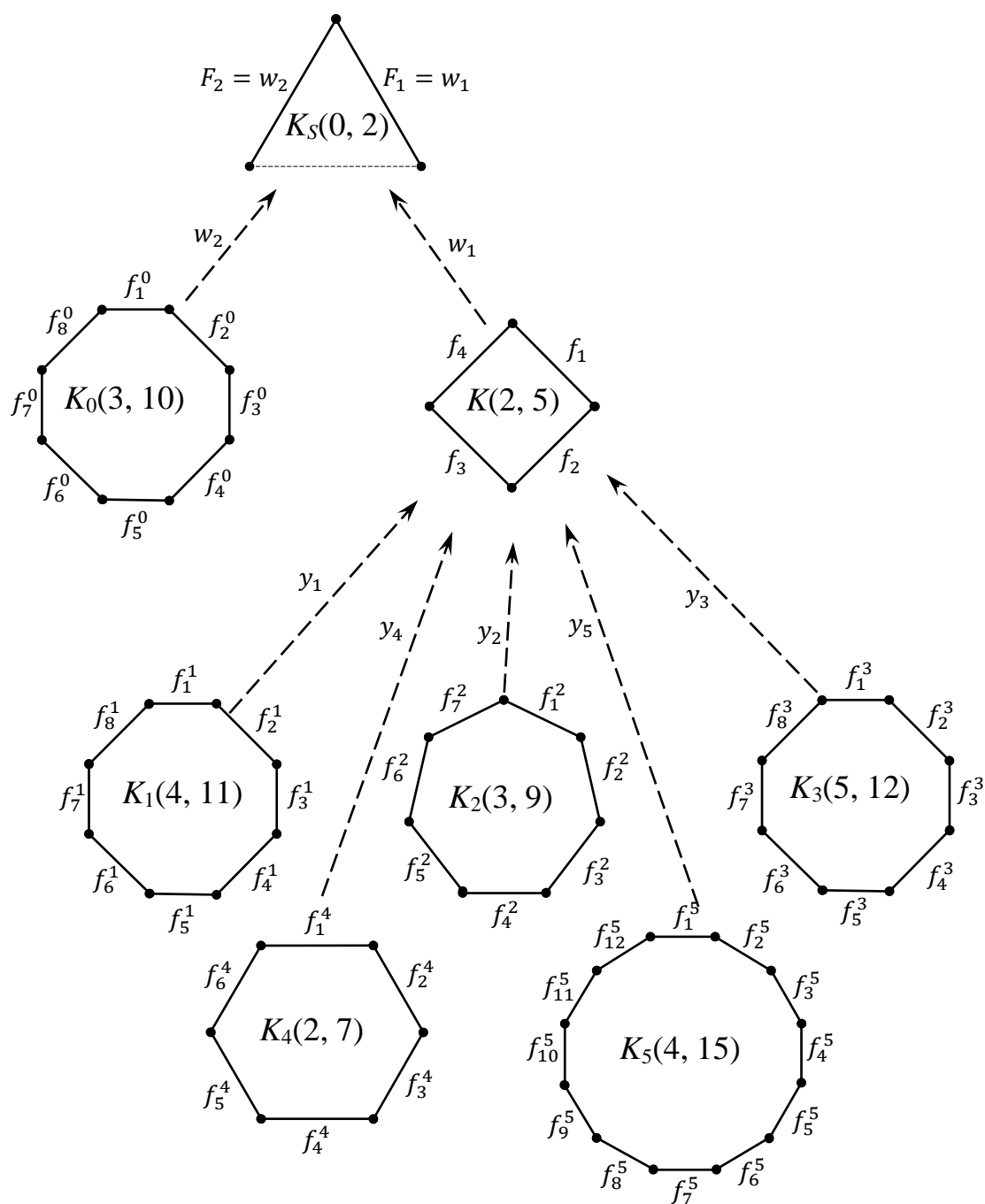


Рис. 3. Модель ієрархічної системи з процесорами на всіх рівнях ієрархії

Додаток Е

Модель системи з ковзним резервом

Структура системи наведена на рис. 1. Вважатимемо, що станам процесорів першої підсистеми відповідають змінні x_1, x_2, \dots, x_8 , другої – змінні $x_9, x_{10}, \dots, x_{15}$, третьої – змінні $x_{16}, x_{17}, \dots, x_{27}$, а процесорам ковзного резерву – змінні $x_{28}, x_{29}, \dots, x_{32}$.

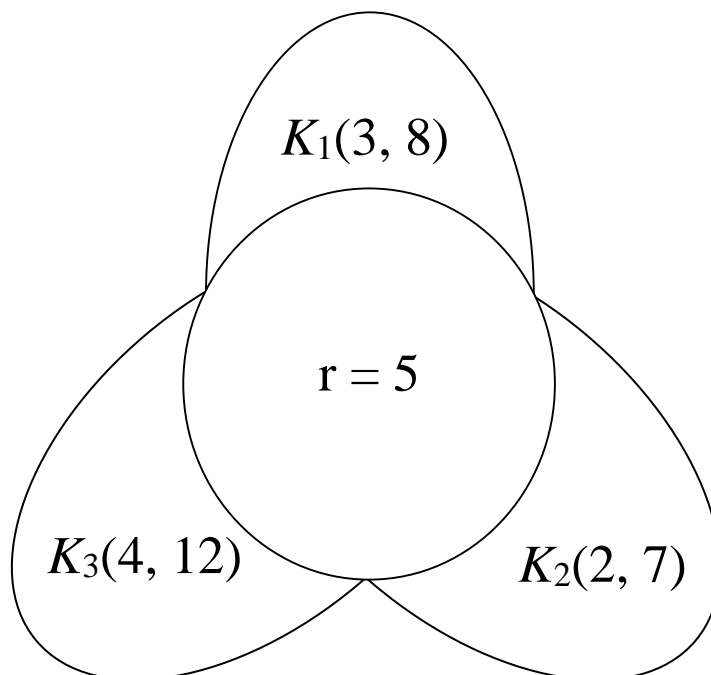


Рис. 1. Структура системи з ковзним резервом

Першій підсистемі відповідає модель $K_1(4, 8)$ з реберними функціями:

$$f_1^1 = x_1 \vee x_2 \vee x_3 \vee x_4$$

$$f_2^1 = (x_1 \vee x_2 \vee x_3 x_4)(x_1 x_2 \vee x_3 \vee x_4) \vee x_5 x_6 x_7 x_8$$

$$f_3^1 = (x_1 \vee x_2)(x_1 x_2 \vee x_3 x_4)(x_3 \vee x_4) \vee (x_5 \vee x_6)(x_5 x_6 \vee x_7 x_8)(x_7 \vee x_8)$$

$$f_4^1 = x_1 x_2 x_3 x_4 \vee (x_5 \vee x_6 \vee x_7 x_8)(x_5 x_6 \vee x_7 \vee x_8)$$

$$f_5^1 = x_5 \vee x_6 \vee x_7 \vee x_8$$

Другій підсистемі відповідає модель $K_2(3, 7)$ з реберними функціями:

$$f_1^2 = x_9 \vee x_{10} \vee x_{11} x_{12}$$

$$f_2^2 = x_9 x_{10} \vee x_{11} \vee x_{12}$$

$$f_3^2 = (x_9 \vee x_{10})(x_9 x_{10} \vee x_{11} x_{12})(x_{11} \vee x_{12}) \vee x_{13} x_{14} x_{15}$$

$$f_4^2 = x_9 x_{10} x_{11} x_{12} \vee (x_{13} \vee x_{14})(x_{13} x_{14} \vee x_{15})$$

$$f_5^2 = x_{13} \vee x_{14} \vee x_{15}$$

Третій підсистемі відповідає модель $K_3(5, 12)$ з реберними функціями:

$$f_1^3 = x_{16} \vee x_{17} \vee x_{18} \vee (x_{19} \vee x_{20})(x_{19} x_{20} \vee x_{21})$$

$$f_2^3 = (x_{16} \vee x_{17})(x_{16} x_{17} \vee x_{18}) \vee x_{19} \vee x_{20} \vee x_{21}$$

$$f_3^3 = (x_{16} \vee x_{17} \vee x_{18} \vee x_{19} x_{20} x_{21}) \wedge$$

$$\wedge ((x_{16} \vee x_{17})(x_{16} x_{17} \vee x_{18}) \vee (x_{19} \vee x_{20})(x_{19} x_{20} \vee x_{21})) \wedge$$

$$\wedge (x_{16} x_{17} x_{18} \vee x_{19} \vee x_{20} \vee x_{21}) \vee x_{22} x_{23} x_{24} x_{25} x_{26} x_{27}$$

$$f_4^3 = (x_{16} \vee x_{17} \vee x_{18})((x_{16} \vee x_{17})(x_{16} x_{17} \vee x_{18}) \vee x_{19} x_{20} x_{21}) \wedge$$

$$\wedge (x_{16} x_{17} x_{18} \vee (x_{19} \vee x_{20})(x_{19} x_{20} \vee x_{21}))(x_{19} \vee x_{20} \vee x_{21}) \vee$$

$$\vee (x_{22} \vee x_{23})(x_{22} x_{23} \vee x_{24})(x_{22} x_{23} x_{24} \vee x_{25} x_{26} x_{27})(x_{25} \vee x_{26})(x_{25} x_{26} \vee x_{27})$$

$$f_5^3 = (x_{16} \vee x_{17})(x_{16} x_{17} \vee x_{18})(x_{16} x_{17} x_{18} \vee x_{19} x_{20} x_{21})(x_{19} \vee x_{20}) \wedge$$

$$\wedge (x_{19} x_{20} \vee x_{21}) \vee (x_{22} \vee x_{23} \vee x_{24})((x_{22} \vee x_{23})(x_{22} x_{23} \vee x_{24}) \vee x_{25} x_{26} x_{27}) \wedge$$

$$\wedge (x_{22} x_{23} x_{24} \vee (x_{25} \vee x_{26})(x_{25} x_{26} \vee x_{27}))(x_{25} \vee x_{26} \vee x_{27})$$

$$f_6^3 = x_{16} x_{17} x_{18} x_{19} x_{20} x_{21} \vee (x_{22} \vee x_{23} \vee x_{24} \vee x_{25} x_{26} x_{27}) \wedge$$

$$\wedge ((x_{22} \vee x_{23})(x_{22} x_{23} \vee x_{24}) \vee (x_{25} \vee x_{26})(x_{25} x_{26} \vee x_{27})) \wedge$$

$$\wedge (x_{22} x_{23} x_{24} \vee x_{25} \vee x_{26} \vee x_{27})$$

$$f_7^3 = x_{22} \vee x_{23} \vee x_{24} \vee (x_{25} \vee x_{26})(x_{25} x_{26} \vee x_{27})$$

$$f_8^3 = (x_{22} \vee x_{23})(x_{22} x_{23} \vee x_{24}) \vee x_{25} \vee x_{26} \vee x_{27}$$

На базі елементів вектору стану системи, що відповідають процесорам ковзного резерву, а також значень реберних функцій побудованих вище моделей сформовано вектор v :

$$v = \langle x_{28}, x_{29}, x_{30}, x_{31}, x_{32}, f_1^1, f_2^1, f_3^1, f_4^1, f_5^1, f_1^2, f_2^2, f_3^2, f_4^2, f_5^2, f_1^3, f_2^3, f_3^3, f_4^3, f_5^3, f_6^3, f_7^3, f_8^3 \rangle$$

Системі відповідає модель $K(5, 23)$ з реберними функціями:

$$f_1 = x_{28} \vee x_{29} \vee x_{30} \vee (x_{31} \vee x_{32})(x_{31} x_{32} \vee f_1^1)$$

$$f_2 = (x_{28} \vee x_{29})(x_{28} x_{29} \vee x_{30}) \vee x_{31} \vee x_{32} \vee f_1^1$$

$$f_3 = (x_{28} \vee x_{29} \vee x_{30} \vee x_{31} x_{32} f_1^1) \wedge$$

$$\begin{aligned}
& \wedge ((x_{28} \vee x_{29})(x_{28}x_{29} \vee x_{30}) \vee (x_{31} \vee x_{32})(x_{31}x_{32} \vee f_1^1)) \wedge \\
& \wedge (x_{28}x_{29}x_{30} \vee x_{31} \vee x_{32} \vee f_1^1) \vee f_2^1 f_3^1 f_4^1 f_5^1 f_1^2 f_2^2 \\
f_4 = & (x_{28} \vee x_{29} \vee x_{30})((x_{28} \vee x_{29})(x_{28}x_{29} \vee x_{30}) \vee x_{31}x_{32}f_1^1) \wedge \\
& \wedge (x_{28}x_{29}x_{30} \vee (x_{31} \vee x_{32})(x_{31}x_{32} \vee f_1^1))(x_{31} \vee x_{32} \vee f_1^1) \vee \\
& \vee (f_2^1 \vee f_3^1)(f_2^1 f_3^1 \vee f_4^1)(f_2^1 f_3^1 f_4^1 \vee f_5^1 f_1^2 f_2^2)(f_5^1 \vee f_1^2)(f_5^1 f_1^2 \vee f_2^2) \\
f_5 = & (x_{28} \vee x_{29})(x_{28}x_{29} \vee x_{30})(x_{28}x_{29}x_{30} \vee x_{31}x_{32}f_1^1)(x_{31} \vee x_{32}) \wedge \\
& \wedge (x_{31}x_{32} \vee f_1^1) \vee (f_2^1 \vee f_3^1 \vee f_4^1)((f_2^1 \vee f_3^1)(f_2^1 f_3^1 \vee f_4^1) \vee f_5^1 f_1^2 f_2^2) \wedge \\
& \wedge (f_2^1 f_3^1 f_4^1 \vee (f_5^1 \vee f_1^2)(f_5^1 f_1^2 \vee f_2^2))(f_5^1 \vee f_1^2 \vee f_2^2) \\
f_6 = & x_{28}x_{29}x_{30}x_{31}x_{32}f_1^1 \vee (f_2^1 \vee f_3^1 \vee f_4^1 \vee f_5^1 f_1^2 f_2^2) \wedge \\
& \wedge ((f_2^1 \vee f_3^1)(f_2^1 f_3^1 \vee f_4^1) \vee (f_5^1 \vee f_1^2)(f_5^1 f_1^2 \vee f_2^2))(f_2^1 f_3^1 f_4^1 \vee f_5^1 \vee f_1^2 \vee f_2^2) \\
f_7 = & f_2^1 \vee f_3^1 \vee f_4^1 \vee (f_5^1 \vee f_1^2)(f_5^1 f_1^2 \vee f_2^2) \\
f_8 = & (f_2^1 \vee f_3^1)(f_2^1 f_3^1 \vee f_4^1) \vee f_5^1 \vee f_1^2 \vee f_2^2 \\
f_9 = & (x_{28} \vee x_{29} \vee x_{30} \vee x_{31}x_{32}f_1^1) \wedge \\
& \wedge ((x_{28} \vee x_{29})(x_{28}x_{29} \vee x_{30}) \vee (x_{31} \vee x_{32})(x_{31}x_{32} \vee f_1^1)) \wedge \\
& \wedge (x_{28}x_{29}x_{30} \vee x_{31} \vee x_{32} \vee f_1^1) \wedge \\
& \wedge \left(\begin{aligned} & (x_{28} \vee x_{29} \vee x_{30})((x_{28} \vee x_{29})(x_{28}x_{29} \vee x_{30}) \vee x_{31}x_{32}f_1^1) \wedge \\ & \wedge (x_{28}x_{29}x_{30} \vee (x_{31} \vee x_{32})(x_{31}x_{32} \vee f_1^1))(x_{31} \vee x_{32} \vee f_1^1) \vee \\ & \vee f_2^1 f_3^1 f_4^1 f_5^1 f_1^2 f_2^2 \end{aligned} \right) \wedge \\
& \wedge \left(\begin{aligned} & (x_{28} \vee x_{29})(x_{28}x_{29} \vee x_{30})(x_{28}x_{29}x_{30} \vee x_{31}x_{32}f_1^1)(x_{31} \vee x_{32}) \wedge \\ & \wedge (x_{31}x_{32} \vee f_1^1) \vee (f_2^1 \vee f_3^1)(f_2^1 f_3^1 \vee f_4^1)(f_2^1 f_3^1 f_4^1 \vee f_5^1 f_1^2 f_2^2) \wedge \\ & \wedge (f_5^1 \vee f_1^2)(f_5^1 f_1^2 \vee f_2^2) \end{aligned} \right) \wedge \\
& \wedge \left(\begin{aligned} & x_{28}x_{29}x_{30}x_{31}x_{32}f_1^1 \vee (f_2^1 \vee f_3^1 \vee f_4^1) \wedge \\ & \wedge ((f_2^1 \vee f_3^1)(f_2^1 f_3^1 \vee f_4^1) \vee f_5^1 f_1^2 f_2^2) \wedge \\ & \wedge (f_2^1 f_3^1 f_4^1 \vee (f_5^1 \vee f_1^2)(f_5^1 f_1^2 \vee f_2^2))(f_5^1 \vee f_1^2 \vee f_2^2) \end{aligned} \right) \wedge \\
& \wedge (f_2^1 \vee f_3^1 \vee f_4^1 \vee f_5^1 f_1^2 f_2^2)((f_2^1 \vee f_3^1)(f_2^1 f_3^1 \vee f_4^1) \vee (f_5^1 \vee f_1^2)(f_5^1 f_1^2 \vee f_2^2)) \wedge \\
& \wedge (f_2^1 f_3^1 f_4^1 \vee f_5^1 \vee f_1^2 \vee f_2^2) \vee f_3^2 f_4^2 f_5^2 f_1^3 f_2^3 f_3^3 f_4^3 f_5^3 f_6^3 f_7^3 f_8^3 \\
f_{10} = & (x_{28} \vee x_{29} \vee x_{30})((x_{28} \vee x_{29})(x_{28}x_{29} \vee x_{30}) \vee x_{31}x_{32}f_1^1) \wedge
\end{aligned}$$

$$\begin{aligned}
& \wedge (x_{28}x_{29}x_{30} \vee (x_{31} \vee x_{32})(x_{31}x_{32} \vee f_1^1))(x_{31} \vee x_{32} \vee f_1^1) \wedge \\
& \wedge \left(\begin{array}{c} (x_{28} \vee x_{29})(x_{28}x_{29} \vee x_{30})(x_{28}x_{29}x_{30} \vee x_{31}x_{32}f_1^1) \wedge \\ \wedge (x_{31} \vee x_{32})(x_{31}x_{32} \vee f_1^1) \vee f_2^1f_3^1f_4^1f_5^1f_1^2f_2^2 \end{array} \right) \wedge \\
& \wedge \left(\begin{array}{c} x_{28}x_{29}x_{30}x_{31}x_{32}f_1^1 \vee (f_2^1 \vee f_3^1)(f_2^1f_3^1 \vee f_4^1) \wedge \\ \wedge (f_2^1f_3^1f_4^1 \vee f_5^1f_1^2f_2^2)(f_5^1 \vee f_1^2)(f_5^1f_1^2 \vee f_2^2) \end{array} \right) \wedge \\
& \wedge (f_2^1 \vee f_3^1 \vee f_4^1)((f_2^1 \vee f_3^1)(f_2^1f_3^1 \vee f_4^1) \vee f_5^1f_1^2f_2^2) \wedge \\
& \wedge (f_2^1f_3^1f_4^1 \vee (f_5^1 \vee f_1^2)(f_5^1f_1^2 \vee f_2^2))(f_5^1 \vee f_1^2 \vee f_2^2) \vee \\
& \vee (f_3^2 \vee f_4^2)(f_3^2f_4^2 \vee f_5^2)(f_3^2f_4^2f_5^2 \vee f_1^3f_2^3f_3^3)(f_1^3 \vee f_2^3)(f_1^3f_2^3 \vee f_3^3) \wedge \\
& \wedge (f_3^2f_4^2f_5^2f_1^3f_2^3f_3^3 \vee f_4^3f_5^3f_6^3f_7^3f_8^3)(f_4^3 \vee f_5^3)(f_4^3f_5^3 \vee f_6^3) \wedge \\
& \wedge (f_4^3f_5^3f_6^3 \vee f_7^3f_8^3)(f_7^3 \vee f_8^3) \\
f_{11} &= (x_{28} \vee x_{29})(x_{28}x_{29} \vee x_{30})(x_{28}x_{29}x_{30} \vee x_{31}x_{32}f_1^1)(x_{31} \vee x_{32}) \wedge \\
& \wedge (x_{31}x_{32} \vee f_1^1)(x_{28}x_{29}x_{30}x_{31}x_{32}f_1^1 \vee f_2^1f_3^1f_4^1f_5^1f_1^2f_2^2)(f_2^1 \vee f_3^1) \wedge \\
& \wedge (f_2^1f_3^1 \vee f_4^1)(f_2^1f_3^1f_4^1 \vee f_5^1f_1^2f_2^2)(f_5^1 \vee f_1^2)(f_5^1f_1^2 \vee f_2^2) \vee \\
& \vee (f_3^2 \vee f_4^2 \vee f_5^2)((f_3^2 \vee f_4^2)(f_3^2f_4^2 \vee f_5^2) \vee f_1^3f_2^3f_3^3) \wedge \\
& \wedge (f_3^2f_4^2f_5^2 \vee (f_1^3 \vee f_2^3)(f_1^3f_2^3 \vee f_3^3))(f_1^3 \vee f_2^3 \vee f_3^3) \wedge \\
& \wedge \left(\begin{array}{c} (f_3^2 \vee f_4^2)(f_3^2f_4^2 \vee f_5^2)(f_3^2f_4^2f_5^2 \vee f_1^3f_2^3f_3^3) \\ (f_1^3 \vee f_2^3)(f_1^3f_2^3 \vee f_3^3) \vee f_4^3f_5^3f_6^3f_7^3f_8^3 \end{array} \right) \wedge \\
& \wedge \left(\begin{array}{c} f_3^2f_4^2f_5^2f_1^3f_2^3f_3^3 \vee (f_4^3 \vee f_5^3)(f_4^3f_5^3 \vee f_6^3) \wedge \\ \wedge (f_4^3f_5^3f_6^3 \vee f_7^3f_8^3)(f_7^3 \vee f_8^3) \end{array} \right) (f_4^3 \vee f_5^3 \vee f_6^3) \wedge \\
& \wedge ((f_4^3 \vee f_5^3)(f_4^3f_5^3 \vee f_6^3) \vee f_7^3f_8^3)(f_4^3f_5^3f_6^3 \vee f_7^3 \vee f_8^3) \\
f_{12} &= x_{28}x_{29}x_{30}x_{31}x_{32}f_1^1f_2^1f_3^1f_4^1f_5^1f_1^2f_2^2 \vee (f_3^2 \vee f_4^2 \vee f_5^2 \vee f_1^3f_2^3f_3^3) \wedge \\
& \wedge ((f_3^2 \vee f_4^2)(f_3^2f_4^2 \vee f_5^2) \vee (f_1^3 \vee f_2^3)(f_1^3f_2^3 \vee f_3^3))(f_3^2f_4^2f_5^2 \vee f_1^3 \vee f_2^3 \vee f_3^3) \wedge \\
& \wedge \left(\begin{array}{c} (f_3^2 \vee f_4^2 \vee f_5^2)((f_3^2 \vee f_4^2)(f_3^2f_4^2 \vee f_5^2) \vee f_1^3f_2^3f_3^3) \wedge \\ \wedge (f_3^2f_4^2f_5^2 \vee (f_1^3 \vee f_2^3)(f_1^3f_2^3 \vee f_3^3))(f_1^3 \vee f_2^3 \vee f_3^3) \vee \\ \vee f_4^3f_5^3f_6^3f_7^3f_8^3 \end{array} \right) \wedge \\
& \wedge \left(\begin{array}{c} (f_3^2 \vee f_4^2)(f_3^2f_4^2 \vee f_5^2)(f_3^2f_4^2f_5^2 \vee f_1^3f_2^3f_3^3)(f_1^3 \vee f_2^3)(f_1^3f_2^3 \vee f_3^3) \vee \\ \vee (f_4^3 \vee f_5^3)(f_4^3f_5^3 \vee f_6^3)(f_4^3f_5^3f_6^3 \vee f_7^3f_8^3)(f_7^3 \vee f_8^3) \end{array} \right) \wedge
\end{aligned}$$

$$\begin{aligned}
& \wedge \left(\wedge \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_6^3) \vee f_7^3 f_8^3 \right) (f_4^3 f_5^3 f_6^3 \vee f_7^3 \vee f_8^3) \right) \wedge \\
& \wedge (f_4^3 \vee f_5^3 \vee f_6^3 \vee f_7^3 f_8^3) \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_6^3) \vee f_7^3 \vee f_8^3 \right) \\
f_{13} &= f_3^2 \vee f_4^2 \vee f_5^2 \vee (f_1^3 \vee f_2^3)(f_1^3 f_2^3 \vee f_3^3) \\
f_{14} &= (f_3^2 \vee f_4^2)(f_3^2 f_4^2 \vee f_5^2) \vee f_1^3 \vee f_2^3 \vee f_3^3 \\
f_{15} &= (f_3^2 \vee f_4^2 \vee f_5^2 \vee f_1^3 f_2^3 f_3^3) \wedge \\
& \wedge \left((f_3^2 \vee f_4^2)(f_3^2 f_4^2 \vee f_5^2) \vee (f_1^3 \vee f_2^3)(f_1^3 f_2^3 \vee f_3^3) \right) \wedge \\
& \wedge (f_3^2 f_4^2 f_5^2 \vee f_1^3 \vee f_2^3 \vee f_3^3) \vee f_4^3 f_5^3 f_6^3 f_7^3 f_8^3 \\
f_{16} &= (f_3^2 \vee f_4^2 \vee f_5^2) \left((f_3^2 \vee f_4^2)(f_3^2 f_4^2 \vee f_5^2) \vee f_1^3 f_2^3 f_3^3 \right) \wedge \\
& \wedge \left(f_3^2 f_4^2 f_5^2 \vee (f_1^3 \vee f_2^3)(f_1^3 f_2^3 \vee f_3^3) \right) (f_1^3 \vee f_2^3 \vee f_3^3) \vee \\
& \vee (f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_6^3)(f_4^3 f_5^3 f_6^3 \vee f_7^3 f_8^3)(f_7^3 \vee f_8^3) \\
f_{17} &= (f_3^2 \vee f_4^2)(f_3^2 f_4^2 \vee f_5^2)(f_3^2 f_4^2 f_5^2 \vee f_1^3 f_2^3 f_3^3)(f_1^3 \vee f_2^3)(f_1^3 f_2^3 \vee f_3^3) \vee \\
& \vee (f_4^3 \vee f_5^3 \vee f_6^3) \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_6^3) \vee f_7^3 f_8^3 \right) (f_4^3 f_5^3 f_6^3 \vee f_7^3 \vee f_8^3) \\
f_{18} &= f_3^2 f_4^2 f_5^2 f_1^3 f_2^3 f_3^3 \vee \\
& \vee (f_4^3 \vee f_5^3 \vee f_6^3 \vee f_7^3 f_8^3) \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_6^3) \vee f_7^3 \vee f_8^3 \right) \\
f_{19} &= f_4^3 \vee f_5^3 \vee f_6^3 \vee f_7^3 \vee f_8^3
\end{aligned}$$

Структуру моделі системи наведено на рис. 2.

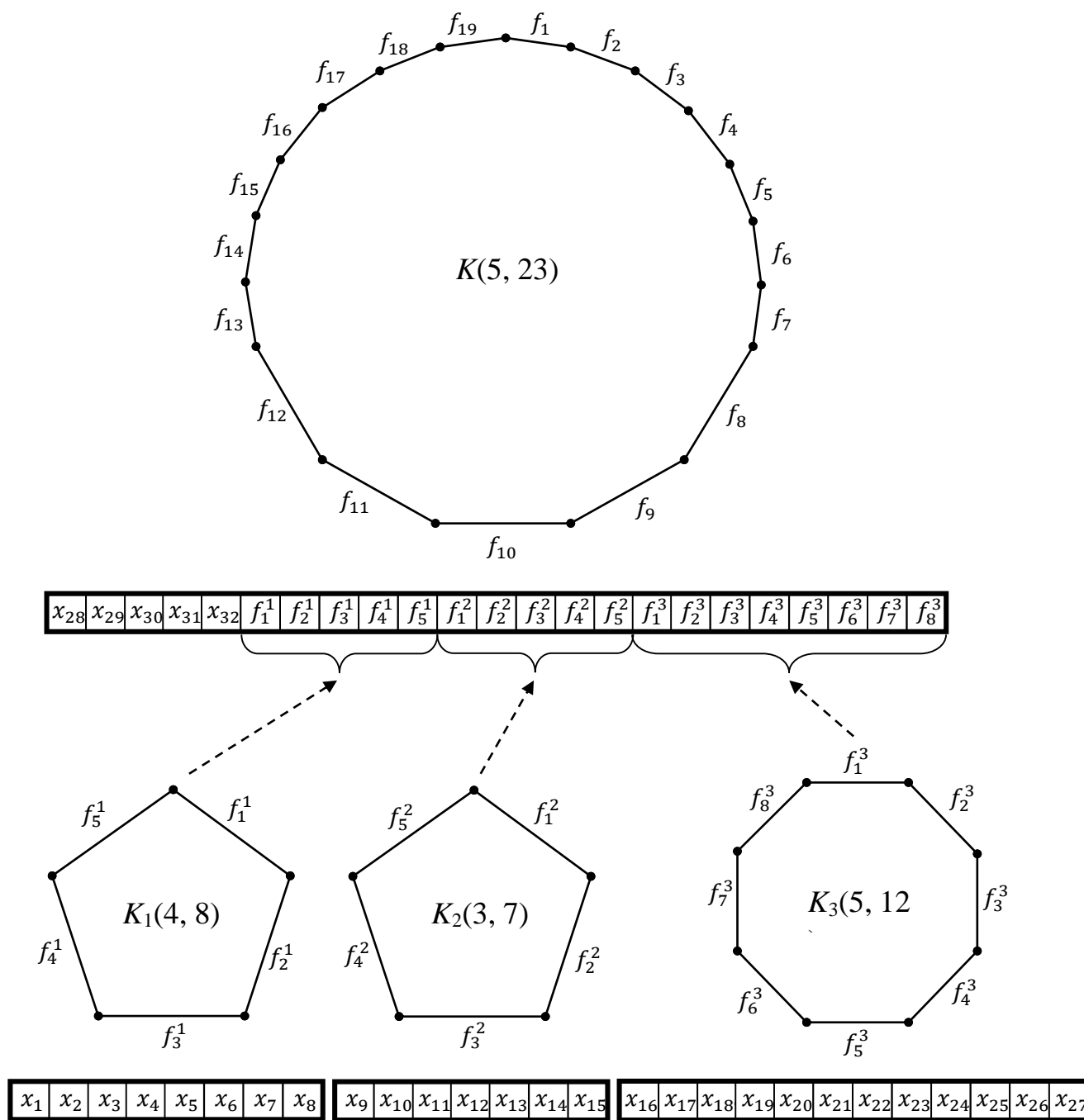


Рис. 2. Модель системи з ковзним резервом

Додаток Ж

Модель системи типу «донор»-«реципієнт»

Структура системи наведена на рис. 1. Вважатимемо, що станам процесорів першої підсистеми відповідають змінні x_1, x_2, \dots, x_{10} , а другої – змінні $x_{11}, x_{12}, \dots, x_{17}$.

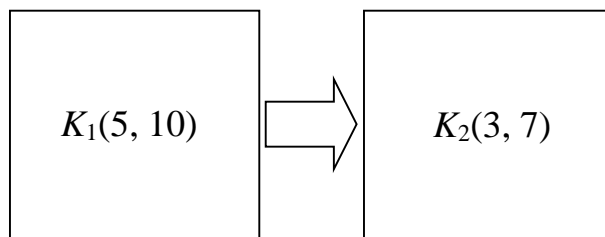


Рис. 1. Структура системи «донор»-«реципієнт»

Першій підсистемі відповідатиме модель $K_1(5, 10)$ з наступними реберними функціями:

$$f_1^1 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5$$

$$f_2^1 = (x_1 \vee x_2 \vee x_3 \vee x_4 x_5) ((x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4 \vee x_5) \vee x_6 x_7 x_8 x_9 x_{10}$$

$$f_3^1 = (x_1 \vee x_2 \vee x_3) ((x_1 \vee x_2)(x_1 x_2 \vee x_3) \vee x_4 x_5) (x_1 x_2 x_3 \vee x_4 \vee x_5) \vee \\ \vee (x_6 \vee x_7)(x_6 x_7 \vee x_8)(x_6 x_7 x_8 \vee x_9 x_{10})(x_9 \vee x_{10})$$

$$f_4^1 = (x_1 \vee x_2)(x_1 x_2 \vee x_3)(x_1 x_2 x_3 \vee x_4 x_5)(x_4 \vee x_5) \vee$$

$$\vee (x_6 \vee x_7 \vee x_8) ((x_6 \vee x_7)(x_6 x_7 \vee x_8) \vee x_9 x_{10})(x_6 x_7 x_8 \vee x_9 \vee x_{10})$$

$$f_5^1 = x_1 x_2 x_3 x_4 x_5 \vee (x_6 \vee x_7 \vee x_8 \vee x_9 x_{10}) ((x_6 \vee x_7)(x_6 x_7 \vee x_8) \vee x_9 \vee x_{10})$$

$$f_6^1 = x_6 \vee x_7 \vee x_8 \vee x_9 \vee x_{10}$$

Побудуємо допоміжну модель $K_1'(6, 10)$ із реберними функціями:

$$g_1^1 = x_1 x_2 x_3 x_4 x_5 (x_6 \vee x_7 \vee x_8 \vee x_9 \vee x_{10})$$

$$g_2^1 = (x_1 x_2 x_3 (x_4 \vee x_5) \vee (x_1 x_2 \vee (x_1 \vee x_2) x_3) x_4 x_5) \wedge$$

$$\wedge (x_6 x_7 \vee (x_6 \vee x_7) x_8 \vee (x_6 \vee x_7 \vee x_8)(x_9 \vee x_{10}) \vee x_9 x_{10})$$

$$g_3^1 = (x_1 x_2 x_3 \vee (x_1 x_2 \vee (x_1 \vee x_2) x_3)(x_4 \vee x_5) \vee (x_1 \vee x_2 \vee x_3) x_4 x_5) \wedge$$

$$\wedge (x_6 x_7 x_8 \vee (x_6 x_7 \vee (x_6 \vee x_7) x_8)(x_9 \vee x_{10}) \vee (x_6 \vee x_7 \vee x_8) x_9 x_{10})$$

$$g_4^1 = (x_1 x_2 \vee (x_1 \vee x_2) x_3 \vee (x_1 \vee x_2 \vee x_3)(x_4 \vee x_5) \vee x_4 x_5) \wedge$$

$$\wedge (x_6 x_7 x_8 (x_9 \vee x_{10}) \vee (x_6 x_7 \vee (x_6 \vee x_7) x_8) x_9 x_{10})$$

$$g_5^1 = (x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5)x_6x_7x_8x_9x_{10}$$

Допоміжний вектор v_1 матиме наступний вигляд:

$$v_1 = \langle x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, g_1^1, g_2^1, g_3^1, g_4^1, g_5^1 \rangle$$

Другій підсистемі відповідатиме модель $K_2(8, 12)$, яка приймає v_1 в якості вхідного вектору. Ця модель матиме наступні реберні функції:

$$f_1^2 = x_{11} \vee x_{12} \vee x_{13} \vee x_{14} \vee x_{15} \vee x_{16} \vee (x_{17} \vee g_1^1)(x_{17}g_1^1 \vee g_2^1) \wedge \\ \wedge (x_{17}g_1^1g_2^1 \vee g_3^1g_4^1g_5^1)(g_3^1 \vee g_4^1)(g_3^1g_4^1 \vee g_5^1)$$

$$f_2^2 = (x_{11} \vee x_{12} \vee x_{13} \vee (x_{14} \vee x_{15})(x_{14}x_{15} \vee x_{16})) \wedge \\ \wedge ((x_{11} \vee x_{12})(x_{11}x_{12} \vee x_{13}) \vee x_{14} \vee x_{15} \vee x_{16}) \vee (x_{17} \vee g_1^1 \vee g_2^1) \wedge \\ \wedge ((x_{17} \vee g_1^1)(x_{17}g_1^1 \vee g_2^1) \vee g_3^1g_4^1g_5^1)(x_{17}g_1^1g_2^1 \vee (g_3^1 \vee g_4^1)(g_3^1g_4^1 \vee g_5^1)) \wedge \\ \wedge (g_3^1 \vee g_4^1 \vee g_5^1)$$

$$f_3^2 = (x_{11} \vee x_{12} \vee x_{13} \vee x_{14}x_{15}x_{16}) \wedge \\ \wedge ((x_{11} \vee x_{12})(x_{11}x_{12} \vee x_{13}) \vee (x_{14} \vee x_{15})(x_{14}x_{15} \vee x_{16})) \wedge \\ \wedge (x_{11}x_{12}x_{13} \vee x_{14} \vee x_{15} \vee x_{16}) \vee (x_{17} \vee g_1^1 \vee g_2^1 \vee g_3^1g_4^1g_5^1) \wedge \\ \wedge ((x_{17} \vee g_1^1)(x_{17}g_1^1 \vee g_2^1) \vee (g_3^1 \vee g_4^1)(g_3^1g_4^1 \vee g_5^1))(x_{17}g_1^1g_2^1 \vee g_3^1 \vee g_4^1 \vee g_5^1)$$

$$f_4^2 = (x_{11} \vee x_{12} \vee x_{13})(x_{11} \vee x_{12})(x_{11}x_{12} \vee x_{13}) \vee x_{14}x_{15}x_{16}) \wedge \\ \wedge (x_{11}x_{12}x_{13} \vee (x_{14} \vee x_{15})(x_{14}x_{15} \vee x_{16}))(x_{14} \vee x_{15} \vee x_{16}) \vee \\ \vee (x_{17} \vee g_1^1 \vee g_2^1 \vee (g_3^1 \vee g_4^1)(g_3^1g_4^1 \vee g_5^1)) \wedge \\ \wedge ((x_{17} \vee g_1^1)(x_{17}g_1^1 \vee g_2^1) \vee g_3^1 \vee g_4^1 \vee g_5^1)$$

$$f_5^2 = (x_{11} \vee x_{12})(x_{11}x_{12} \vee x_{13})(x_{11}x_{12}x_{13} \vee x_{14}x_{15}x_{16})(x_{14} \vee x_{15}) \wedge \\ \wedge (x_{14}x_{15} \vee x_{16}) \vee x_{17} \vee g_1^1 \vee g_2^1 \vee g_3^1 \vee g_4^1 \vee g_5^1$$

Структуру моделі системи наведено на рис. 2.

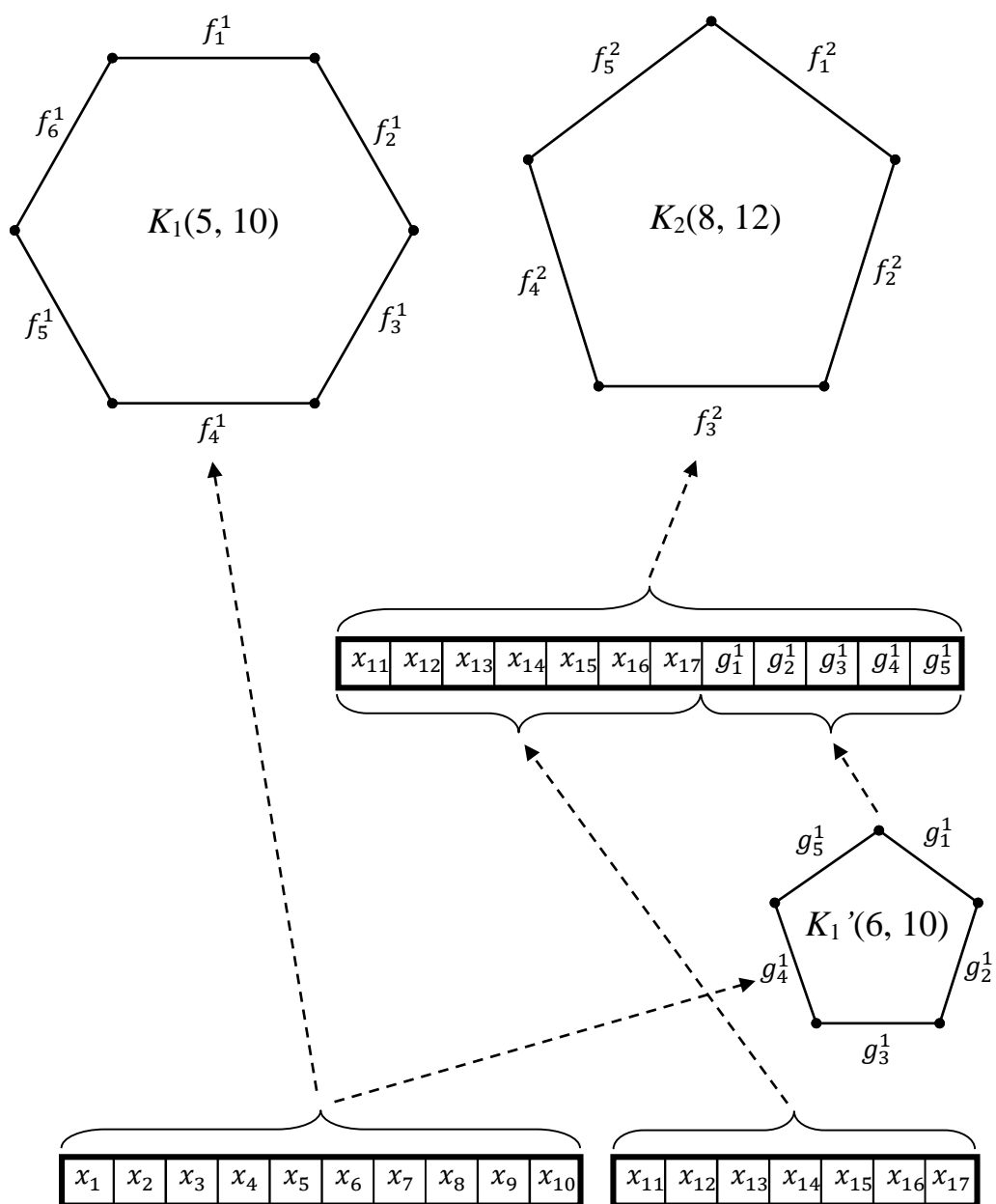


Рис. 2. Модель системы «донор»-«реципиент»

Додаток И

Модель «симетричної» пари підсистем типу «донор»-«реципієнт»

Структура системи наведена на рис. 1. Вважатимемо, що станам процесорів першої підсистеми відповідають змінні x_1, x_2, \dots, x_{10} , а другої – змінні $x_{11}, x_{12}, \dots, x_{17}$.

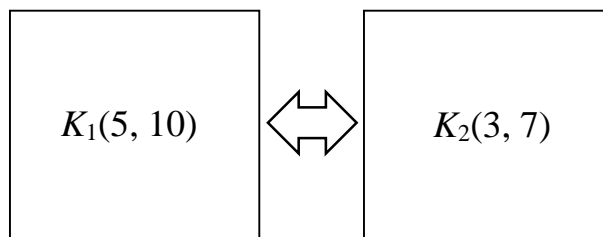


Рис. 1. Структура «симетричної» пари підсистем «донор»-«реципієнт»

Реберні функції g_i^1 та f_j^2 моделей $K_1'(6, 10)$ та $K_2(8, 12)$, що відповідають другій підсистемі, будуть ідентичними до наведених в Додатку Е, тому обмежимося побудовою моделі першої підсистеми.

Побудуємо допоміжну двоїсту модель $K_2'(5, 7)$, що міститиме наступні реберні функції:

$$g_1^2 = x_{11}x_{12}x_{13}x_{14}(x_{15} \vee x_{16} \vee x_{17})$$

$$g_2^2 = (x_{11}x_{12}(x_{13} \vee x_{14}) \vee (x_{11} \vee x_{12})x_{13}x_{14})(x_{15}x_{16} \vee (x_{15} \vee x_{16})x_{17})$$

$$g_3^2 = (x_{11}x_{12} \vee (x_{11} \vee x_{12})(x_{13} \vee x_{14}) \vee x_{13}x_{14})x_{15}x_{16}x_{17}$$

Сформуємо допоміжний вектор v_2 , що міститиме значення елементів вектору стану системи, котрі відповідають станам процесорів першої підсистеми, а також значення реберних функцій допоміжної моделі:

$$v_2 = \langle x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, g_1^2, g_2^2, g_3^2 \rangle$$

Першій підсистемі відповідатиме модель $K_1(8, 13)$, що приймає вектор v_2 в якості вхідного вектору. Ця модель міститиме наступні реберні функції:

$$f_1^1 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \vee x_7 \vee x_8x_9x_{10}g_1^2g_2^2g_3^2$$

$$f_2^1 = (x_1 \vee x_2 \vee x_3 \vee x_4 \vee (x_5 \vee x_6)(x_5x_6 \vee x_7)) \wedge$$

$$\wedge ((x_1 \vee x_2 \vee x_3x_4)(x_1x_2 \vee x_3 \vee x_4) \vee x_5 \vee x_6 \vee x_7) \vee$$

$$\vee (x_8 \vee x_9)(x_8x_9 \vee x_{10})(x_8x_9x_{10} \vee g_1^2g_2^2g_3^2)(g_1^2 \vee g_2^2)(g_1^2g_2^2 \vee g_3^2)$$

$$\begin{aligned}
f_3^1 &= (x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 x_6 x_7) \wedge \\
&\wedge ((x_1 \vee x_2 \vee x_3 x_4)(x_1 x_2 \vee x_3 \vee x_4) \vee (x_5 \vee x_6)(x_5 x_6 \vee x_7)) \wedge \\
&\wedge ((x_1 \vee x_2)(x_1 x_2 \vee x_3 x_4)(x_3 \vee x_4) \vee x_5 \vee x_6 \vee x_7) \vee \\
&\vee (x_8 \vee x_9 \vee x_{10})((x_8 \vee x_9)(x_8 x_9 \vee x_{10}) \vee g_1^2 g_2^2 g_3^2) \wedge \\
&\wedge (x_8 x_9 x_{10} \vee (g_1^2 \vee g_2^2)(g_1^2 g_2^2 \vee g_3^2))(g_1^2 \vee g_2^2 \vee g_3^2) \\
f_4^1 &= (x_1 \vee x_2 \vee x_3 \vee x_4)((x_1 \vee x_2 \vee x_3 x_4)(x_1 x_2 \vee x_3 \vee x_4) \vee x_5 x_6 x_7) \wedge \\
&\wedge ((x_1 \vee x_2)(x_1 x_2 \vee x_3 x_4)(x_3 \vee x_4) \vee (x_5 \vee x_6)(x_5 x_6 \vee x_7)) \wedge \\
&\wedge (x_1 x_2 x_3 x_4 \vee x_5 \vee x_6 \vee x_7) \vee (x_8 \vee x_9 \vee x_{10} \vee g_1^2 g_2^2 g_3^2) \wedge \\
&\wedge ((x_8 \vee x_9)(x_8 x_9 \vee x_{10}) \vee (g_1^2 \vee g_2^2)(g_1^2 g_2^2 \vee g_3^2))(x_8 x_9 x_{10} \vee g_1^2 \vee g_2^2 \vee g_3^2) \\
f_5^1 &= (x_1 \vee x_2 \vee x_3 x_4)(x_1 x_2 \vee x_3 \vee x_4) \wedge \\
&\wedge ((x_1 \vee x_2)(x_1 x_2 \vee x_3 x_4)(x_3 \vee x_4) \vee x_5 x_6 x_7) \wedge \\
&\wedge (x_1 x_2 x_3 x_4 \vee (x_5 \vee x_6)(x_5 x_6 \vee x_7)) \wedge \\
&\wedge (x_5 \vee x_6 \vee x_7) \vee (x_8 \vee x_9 \vee x_{10} \vee (g_1^2 \vee g_2^2)(g_1^2 g_2^2 \vee g_3^2)) \wedge \\
&\wedge ((x_8 \vee x_9)(x_8 x_9 \vee x_{10}) \vee g_1^2 \vee g_2^2 \vee g_3^2) \\
f_6^1 &= (x_1 \vee x_2)(x_1 x_2 \vee x_3 x_4)(x_3 \vee x_4)(x_1 x_2 x_3 x_4 \vee x_5 x_6 x_7)(x_5 \vee x_6) \wedge \\
&\wedge (x_5 x_6 \vee x_7) \vee x_8 \vee x_9 \vee x_{10} \vee g_1^2 \vee g_2^2 \vee g_3^2
\end{aligned}$$

Структуру моделі системи наведено на рис. 2.

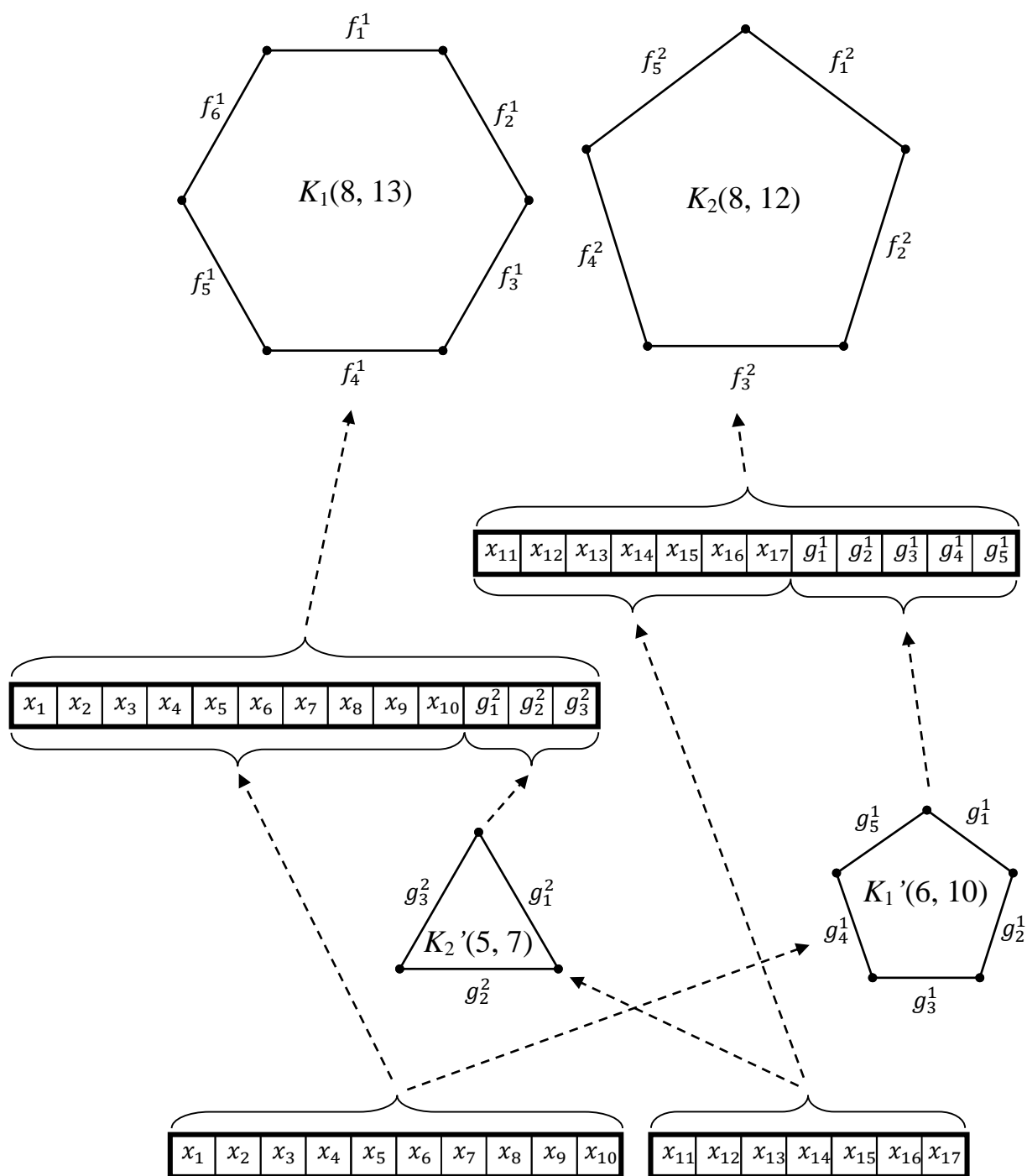


Рис. 2. Модель «симетричної» пари підсистем «донор»-«реципієнт»

Додаток К

Модель системи типу «донор»-«реципієнт» з обмеженнями

Структура системи наведена на рис. 1. Вважатимемо, що станам процесорів першої підсистеми відповідають змінні x_1, x_2, \dots, x_{10} , а другої – змінні $x_{11}, x_{12}, \dots, x_{17}$.

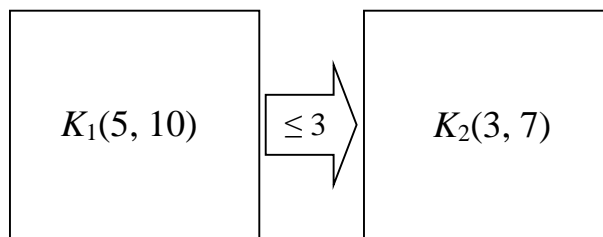


Рис. 1. Структура системи «донор»-«реципієнт» з обмеженнями

Побудуємо допоміжну модель, що обмежує кількість використовуваних «надлишкових» процесорів. Ця модель $K_2''(3, 5)$, що прийматиме на вхід вектор v_1 міститиме наступні функції:

$$h_1^1 = g_1^1 \vee g_2^1 \vee g_3^1$$

$$h_2^1 = (g_1^1 \vee g_2^1)(g_1^1 g_2^1 \vee g_3^1) \vee g_4^1 g_5^1$$

$$h_2^1 = g_1^1 g_2^1 g_3^1 \vee g_4^1 \vee g_5^1$$

Сформуємо допоміжний вектор w_1 , що міститиме значення елементів вектору стану системи, котрі відповідають станам процесорів другої підсистеми, а також значення реберних функцій сформованої вище допоміжної моделі:

$$w_1 = \langle x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, h_1^1, h_2^1, h_3^1 \rangle$$

Другій підсистемі відповідатиме модель $K_1(6, 10)$, яка приймає w_1 в якості вхідного вектору із наступними реберними функціями:

$$f_1^2 = x_{11} \vee x_{12} \vee x_{13} \vee x_{14} \vee x_{15} \vee x_{16} x_{17} h_1^1 h_2^1 h_3^1$$

$$f_2^2 = (x_{11} \vee x_{12} \vee x_{13} \vee x_{14} x_{15}) ((x_{11} \vee x_{12})(x_{11} x_{12} \vee x_{13}) \vee x_{14} \vee x_{15}) \vee (x_{16} \vee x_{17})(x_{16} x_{17} \vee h_1^1)(x_{16} x_{17} h_1^1 \vee h_2^1 h_3^1)(h_2^1 \vee h_3^1)$$

$$f_3^2 = (x_{11} \vee x_{12} \vee x_{13}) ((x_{11} \vee x_{12})(x_{11} x_{12} \vee x_{13}) \vee x_{14} x_{15}) \wedge$$

$$\wedge (x_{11} x_{12} x_{13} \vee x_{14} \vee x_{15}) \vee (x_{16} \vee x_{17} \vee h_1^1) ((x_{16} \vee x_{17})(x_{16} x_{17} \vee h_1^1) \vee h_2^1 h_3^1) \wedge$$

$$\wedge (x_{16}x_{17}h_1^1 \vee h_2^1 \vee h_3^1)$$

$$f_4^2 = (x_{11} \vee x_{12})(x_{11}x_{12} \vee x_{13})(x_{11}x_{12}x_{13} \vee x_{14}x_{15})(x_{14} \vee x_{15}) \vee \\ \vee (x_{16} \vee x_{17} \vee h_1^1 \vee h_2^1 h_3^1)((x_{16} \vee x_{17})(x_{16}x_{17} \vee h_1^1) \vee h_2^1 \vee h_3^1)$$

$$f_5^2 = x_{11}x_{12}x_{13}x_{14}x_{15} \vee x_{16} \vee x_{17} \vee h_1^1 \vee h_2^1 \vee h_3^1$$

Структуру моделі системи наведено на рис. 2.

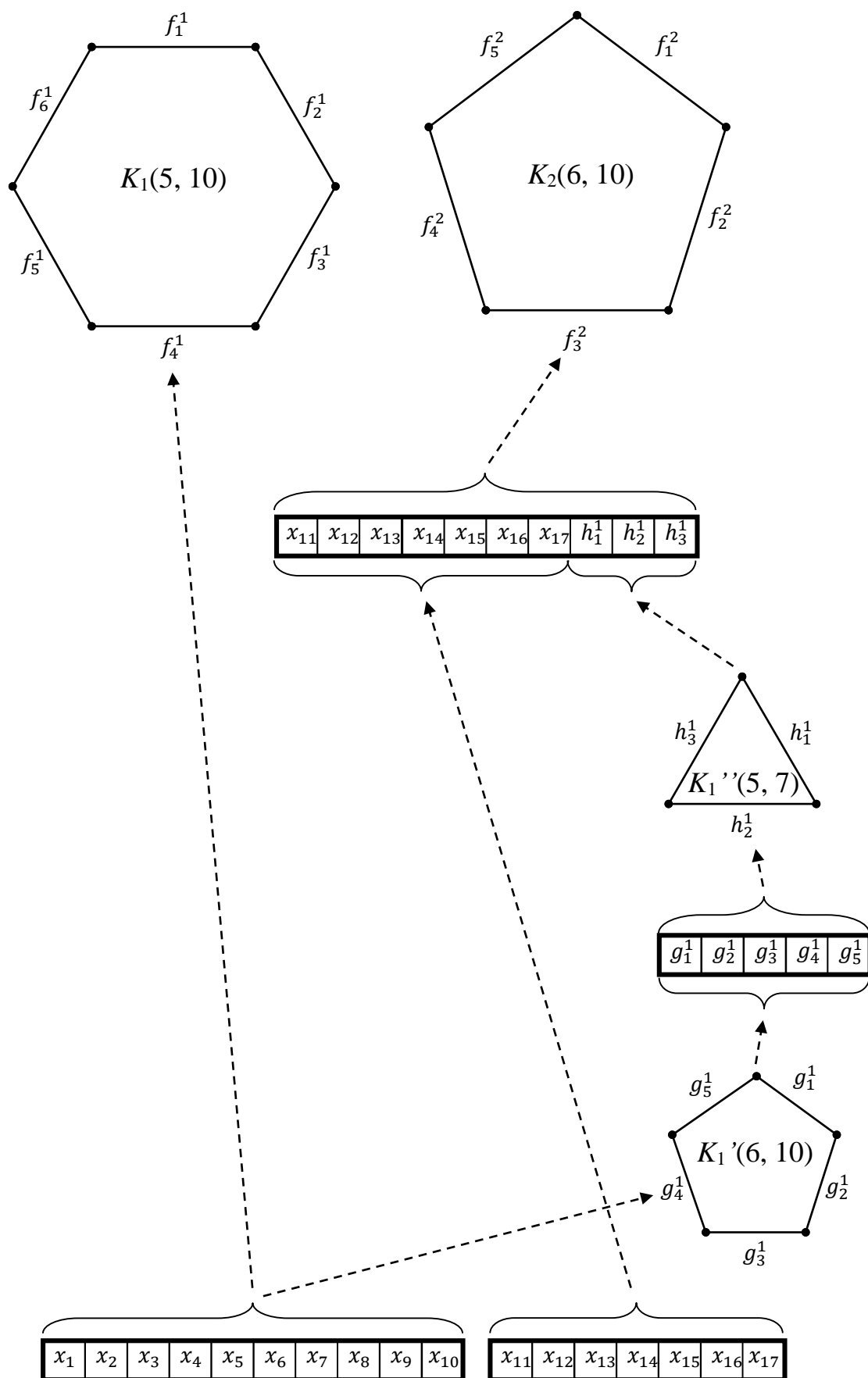


Рис. 2. Модель системи «донор»-«реципієнт» з обмеженнями

Додаток Л

Модель складної системи з ковзним резервом

Структура системи наведена на рис. 1. Вважатимемо, що станам першої підсистеми відповідають змінні x_1, x_2, \dots, x_7 , другої — x_8, x_9, \dots, x_{17} , третьої — $x_{18}, x_{19}, \dots, x_{23}$, четвертої — $x_{24}, x_{25}, \dots, x_{31}$, а п'ятій — $x_{32}, x_{33}, \dots, x_{36}$.

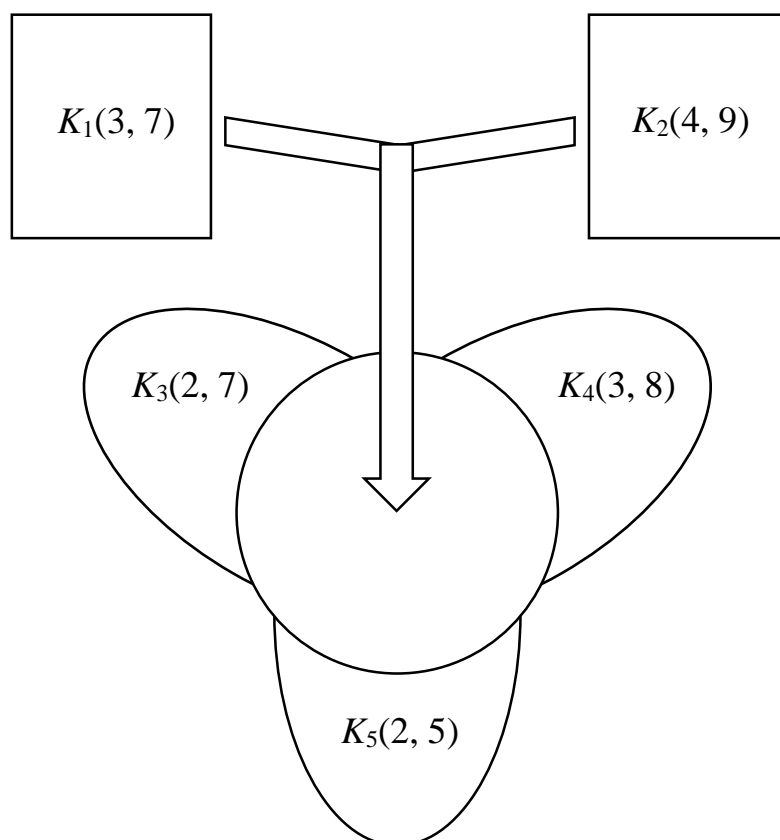


Рис. 1. Структура складної системи з ковзним резервом

Першій підсистемі відповідатиме модель $K_1(3, 7)$ з наступними реберними функціями:

$$f_1^1 = x_1 \vee x_2 \vee x_3 x_4$$

$$f_2^1 = x_1 x_2 \vee x_3 \vee x_4$$

$$f_3^1 = (x_1 \vee x_2)(x_1 x_2 \vee x_3 x_4)(x_3 \vee x_4) \vee x_5 x_6 x_7$$

$$f_4^1 = x_1 x_2 x_3 x_4 \vee (x_5 \vee x_6)(x_5 x_6 \vee x_7)$$

$$f_5^1 = x_5 \vee x_6 \vee x_7$$

Другій підсистемі відповідатиме модель $K_2(4, 9)$ із реберними функціями:

$$f_1^2 = x_8 \vee x_9 \vee x_{10} \vee x_{11}x_{12}$$

$$f_2^2 = (x_8 \vee x_9)(x_8x_9 \vee x_{10}) \vee x_{11} \vee x_{12}$$

$$f_3^2 = (x_8 \vee x_9 \vee x_{10})((x_8 \vee x_9)(x_8x_9 \vee x_{10}) \vee x_{11}x_{12})(x_8x_9x_{10} \vee x_{11} \vee x_{12}) \vee x_{13}x_{14}x_{15}x_{16}$$

$$f_4^2 = (x_8 \vee x_9)(x_8x_9 \vee x_{10})(x_8x_9x_{10} \vee x_{11}x_{12})(x_{11} \vee x_{12}) \vee (x_{13} \vee x_{14})(x_{13}x_{14} \vee x_{15}x_{16})(x_{15} \vee x_{16})$$

$$f_5^2 = x_8x_9x_{10}x_{11}x_{12} \vee (x_{13} \vee x_{14} \vee x_{15}x_{16})(x_{13}x_{14} \vee x_{15} \vee x_{16})$$

$$f_6^2 = x_{13} \vee x_{14} \vee x_{15} \vee x_{16}$$

Побудуємо допоміжну двоїсту модель $K_1'(5, 7)$ з реберними функціями:

$$g_1^1 = x_1x_2x_3x_4(x_5 \vee x_6 \vee x_7)$$

$$g_2^1 = (x_1x_2(x_3 \vee x_4) \vee (x_1 \vee x_2)x_3x_4)(x_5x_6 \vee (x_5 \vee x_6)x_7)$$

$$g_3^1 = (x_1x_2 \vee (x_1 \vee x_2)(x_3 \vee x_4) \vee x_3x_4)x_5x_6x_7$$

Побудуємо допоміжну двоїсту модель $K_2'(6, 9)$ із реберними функціями:

$$g_1^2 = x_8x_9x_{10}x_{11}x_{12}(x_{13} \vee x_{14} \vee x_{15} \vee x_{16})$$

$$g_2^2 = (x_8x_9x_{10}(x_{11} \vee x_{12}) \vee (x_8x_9 \vee (x_8 \vee x_9)x_{10})x_{11}x_{12}) \wedge (x_{13}x_{14} \vee (x_{13} \vee x_{14})(x_{15} \vee x_{16}) \vee x_{15}x_{16})$$

$$g_3^2 = (x_8x_9x_{10} \vee (x_8x_9 \vee (x_8 \vee x_9)x_{10})(x_{11} \vee x_{12}) \vee (x_8 \vee x_9 \vee x_{10})x_{11}x_{12}) \wedge (x_{13}x_{14}(x_{15} \vee x_{16}) \vee (x_{13} \vee x_{14})x_{15}x_{16})$$

$$g_4^2 = ((x_8x_9 \vee (x_8 \vee x_9)x_{10}) \vee (x_8 \vee x_9 \vee x_{10})(x_{11} \vee x_{12}) \vee x_{11}x_{12})x_{13}x_{14}x_{15}x_{16}$$

Вектор стану «надлишкових» процесорів r матиме вигляд:

$$r = \langle g_1^1, g_2^1, g_3^1, g_1^2, g_2^2, g_3^2, g_4^2 \rangle.$$

Допоміжна модель третьої підсистеми $K_3(3, 7)$ міститиме наступні реберні функції:

$$f_1^3 = x_{17} \vee x_{18} \vee x_{19}x_{20}$$

$$f_2^3 = x_{17}x_{18} \vee x_{19} \vee x_{20}$$

$$f_3^3 = (x_{17} \vee x_{18})(x_{17}x_{18} \vee x_{19}x_{20})(x_{19} \vee x_{20}) \vee x_{21}x_{22}x_{23}$$

$$f_4^3 = x_{17}x_{18}x_{19}x_{20} \vee (x_{21} \vee x_{22})(x_{21}x_{22} \vee x_{23})$$

$$f_5^3 = x_{21} \vee x_{22} \vee x_{23}$$

Допоміжна модель четвертої підсистеми $K_4(4, 8)$ міститиме наступні реберні функції:

$$f_1^4 = x_{24} \vee x_{25} \vee x_{26} \vee x_{27}$$

$$f_2^4 = (x_{24} \vee x_{25} \vee x_{26}x_{27})(x_{24}x_{25} \vee x_{26} \vee x_{27}) \vee x_{28}x_{29}x_{30}x_{31}$$

$$f_3^4 = (x_{24} \vee x_{25})(x_{24}x_{25} \vee x_{26}x_{27})(x_{26} \vee x_{27}) \vee$$

$$\vee (x_{28} \vee x_{29})(x_{28}x_{29} \vee x_{30}x_{31})(x_{30} \vee x_{31})$$

$$f_4^4 = x_{24}x_{25}x_{26}x_{27} \vee (x_{28} \vee x_{29} \vee x_{30}x_{31})(x_{28}x_{29} \vee x_{30} \vee x_{31})$$

$$f_5^4 = x_{28} \vee x_{29} \vee x_{30} \vee x_{31}$$

Допоміжна модель п'ятої підсистеми $K_5(3, 5)$ міститиме реберні функції:

$$f_1^5 = x_{32} \vee x_{33} \vee x_{34}$$

$$f_2^5 = (x_{32} \vee x_{33})(x_{32}x_{33} \vee x_{34}) \vee x_{35}x_{36}$$

$$f_3^5 = x_{32}x_{33}x_{34} \vee x_{35} \vee x_{36}$$

Вхідний вектор v матиме наступний вигляд:

$$v = \langle g_1^1, g_2^1, g_3^1, g_1^2, g_2^2, g_3^2, g_4^2, f_1^3, f_2^3, f_3^3, f_4^3, f_5^3, \\ f_1^4, f_2^4, f_3^4, f_4^4, f_5^4, f_1^5, f_2^5, f_3^5 \rangle.$$

Поведінці підсистем 3, 4 та 5 відповідатиме модель $K(7, 20)$ з наступними реберними функціями:

$$f_1 = g_1^1 \vee g_2^1 \vee g_3^1 \vee g_1^2 \vee g_2^2 \vee (g_3^2 \vee g_4^2)(g_3^2g_4^2 \vee f_1^3)(g_3^2g_4^2f_1^3 \vee f_2^3f_3^3)(f_2^3 \vee f_3^3)$$

$$f_2 = (g_1^1 \vee g_2^1 \vee g_3^1 \vee g_1^2g_2^2)((g_1^1 \vee g_2^1)(g_1^1g_2^1 \vee g_3^1) \vee g_1^2 \vee g_2^2) \vee$$

$$\vee (g_3^2 \vee g_4^2 \vee f_1^3)((g_3^2 \vee g_4^2)(g_3^2g_4^2 \vee f_1^3) \vee f_2^3f_3^3)(g_3^2g_4^2f_1^3 \vee f_2^3 \vee f_3^3)$$

$$f_3 = (g_1^1 \vee g_2^1 \vee g_3^1)((g_1^1 \vee g_2^1)(g_1^1g_2^1 \vee g_3^1) \vee g_1^2g_2^2)(g_1^1g_2^1g_3^1 \vee g_1^2 \vee g_2^2) \vee$$

$$\vee (g_3^2 \vee g_4^2 \vee f_1^3 \vee f_2^3f_3^3)((g_3^2 \vee g_4^2)(g_3^2g_4^2 \vee f_1^3) \vee f_2^3 \vee f_3^3)$$

$$f_4 = (g_1^1 \vee g_2^1)(g_1^1g_2^1 \vee g_3^1)(g_1^1g_2^1g_3^1 \vee g_1^2g_2^2)(g_1^2 \vee g_2^2) \vee g_3^2 \vee g_4^2 \vee f_1^3 \vee f_2^3 \vee f_3^3$$

$$f_5 = (g_1^1 \vee g_2^1 \vee g_3^1 \vee g_1^2 \vee g_2^2 \vee g_3^2g_4^2f_1^3f_2^3f_3^3) \wedge$$

$$\wedge \left((g_1^1 \vee g_2^1 \vee g_3^1 \vee g_1^2g_2^2)((g_1^1 \vee g_2^1)(g_1^1g_2^1 \vee g_3^1) \vee g_1^2 \vee g_2^2) \vee \right. \\ \left. \vee (g_3^2 \vee g_4^2)(g_3^2g_4^2 \vee f_1^3)(g_3^2g_4^2f_1^3 \vee f_2^3f_3^3)(f_2^3 \vee f_3^3) \right) \wedge$$

$$\wedge \left((g_1^1 \vee g_2^1 \vee g_3^1)((g_1^1 \vee g_2^1)(g_1^1g_2^1 \vee g_3^1) \vee g_1^2g_2^2)(g_1^1g_2^1g_3^1 \vee g_1^2 \vee g_2^2) \vee \right. \\ \left. \vee (g_3^2 \vee g_4^2 \vee f_1^3)((g_3^2 \vee g_4^2)(g_3^2g_4^2 \vee f_1^3) \vee f_2^3f_3^3)(g_3^2g_4^2f_1^3 \vee f_2^3 \vee f_3^3) \right) \wedge$$

$$\begin{aligned}
& \wedge \left(\begin{array}{c} (g_1^1 \vee g_2^1)(g_1^1 g_2^1 \vee g_3^1)(g_1^1 g_2^1 g_3^1 \vee g_1^2 g_2^2)(g_1^2 \vee g_2^2) \vee \\ \vee (g_3^2 \vee g_4^2 \vee f_1^3 \vee f_2^3 f_3^3)((g_3^2 \vee g_4^2)(g_3^2 g_4^2 \vee f_1^3) \vee f_2^3 \vee f_3^3) \end{array} \right) \wedge \\
& \wedge (g_1^1 g_2^1 g_3^1 g_1^2 g_2^2 \vee g_3^2 \vee g_4^2 \vee f_1^3 \vee f_2^3 \vee f_3^3) \vee f_4^3 f_5^3 f_1^4 f_2^4 f_3^4 f_4^4 f_5^4 f_1^5 f_2^5 f_3^5 \\
& f_6 = (g_1^1 \vee g_2^1 \vee g_3^1 \vee g_1^2 \vee g_2^2) \wedge \\
& \wedge \left((g_1^1 \vee g_2^1 \vee g_3^1 \vee g_1^2 g_2^2)((g_1^1 \vee g_2^1)(g_1^1 g_2^1 \vee g_3^1) \vee g_1^2 \vee g_2^2) \vee g_3^2 g_4^2 f_1^3 f_2^3 f_3^3 \right) \wedge \\
& \wedge \left(\begin{array}{c} (g_1^1 \vee g_2^1 \vee g_3^1)((g_1^1 \vee g_2^1)(g_1^1 g_2^1 \vee g_3^1) \vee g_1^2 g_2^2)(g_1^1 g_2^1 g_3^1 \vee g_1^2 \vee g_2^2) \vee \\ \vee (g_3^2 \vee g_4^2)(g_3^2 g_4^2 \vee f_1^3)(g_3^2 g_4^2 f_1^3 \vee f_2^3 f_3^3)(f_2^3 \vee f_3^3) \end{array} \right) \wedge \\
& \wedge \left(\begin{array}{c} (g_1^1 \vee g_2^1)(g_1^1 g_2^1 \vee g_3^1)(g_1^1 g_2^1 g_3^1 \vee g_1^2 g_2^2)(g_1^2 \vee g_2^2) \vee \\ \vee (g_3^2 \vee g_4^2 \vee f_1^3)((g_3^2 \vee g_4^2)(g_3^2 g_4^2 \vee f_1^3) \vee f_2^3 f_3^3)(g_3^2 g_4^2 f_1^3 \vee f_2^3 \vee f_3^3) \end{array} \right) \wedge \\
& \wedge \left(g_1^1 g_2^1 g_3^1 g_1^2 g_2^2 \vee (g_3^2 \vee g_4^2 \vee f_1^3 \vee f_2^3 f_3^3)((g_3^2 \vee g_4^2)(g_3^2 g_4^2 \vee f_1^3) \vee f_2^3 \vee f_3^3) \right) \wedge \\
& \wedge (g_3^2 \vee g_4^2 \vee f_1^3 \vee f_2^3 \vee f_3^3) \vee (f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4)(f_4^3 f_5^3 f_1^4 \vee f_2^4 f_3^4)(f_2^4 \vee f_3^4) \wedge \\
& \wedge (f_4^3 f_5^3 f_1^4 f_2^4 f_3^4 \vee f_4^4 f_5^4 f_1^5 f_2^5 f_3^5)(f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5)(f_4^4 f_5^4 f_1^5 \vee f_2^5 f_3^5) \wedge \\
& \wedge (f_2^5 \vee f_3^5) \\
& f_7 = (g_1^1 \vee g_2^1 \vee g_3^1 \vee g_1^2 g_2^2)((g_1^1 \vee g_2^1)(g_1^1 g_2^1 \vee g_3^1) \vee g_1^2 \vee g_2^2) \wedge \\
& \wedge \left(\begin{array}{c} (g_1^1 \vee g_2^1 \vee g_3^1)((g_1^1 \vee g_2^1)(g_1^1 g_2^1 \vee g_3^1) \vee g_1^2 g_2^2)(g_1^1 g_2^1 g_3^1 \vee g_1^2 \vee g_2^2) \vee \\ \vee g_3^2 g_4^2 f_1^3 f_2^3 f_3^3 \end{array} \right) \wedge \\
& \wedge \left(\begin{array}{c} (g_1^1 \vee g_2^1)(g_1^1 g_2^1 \vee g_3^1)(g_1^1 g_2^1 g_3^1 \vee g_1^2 g_2^2)(g_1^2 \vee g_2^2) \vee \\ \vee (g_3^2 \vee g_4^2)(g_3^2 g_4^2 \vee f_1^3)(g_3^2 g_4^2 f_1^3 \vee f_2^3 f_3^3)(f_2^3 \vee f_3^3) \end{array} \right) \wedge \\
& \wedge \left(\begin{array}{c} g_1^1 g_2^1 g_3^1 g_1^2 g_2^2 \vee \\ (g_3^2 \vee g_4^2 \vee f_1^3)((g_3^2 \vee g_4^2)(g_3^2 g_4^2 \vee f_1^3) \vee f_2^3 f_3^3)(g_3^2 g_4^2 f_1^3 \vee f_2^3 \vee f_3^3) \end{array} \right) \wedge \\
& \wedge (g_3^2 \vee g_4^2 \vee f_1^3 \vee f_2^3 f_3^3)((g_3^2 \vee g_4^2)(g_3^2 g_4^2 \vee f_1^3) \vee f_2^3 \vee f_3^3) \vee \\
& \vee (f_4^3 \vee f_5^3 \vee f_1^4) \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4) \vee f_2^4 f_3^4 \right) (f_4^3 f_5^3 f_1^4 \vee f_2^4 \vee f_3^4) \wedge \\
& \wedge \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4)(f_4^3 f_5^3 f_1^4 \vee f_2^4 f_3^4)(f_2^4 \vee f_3^4) \vee f_4^4 f_5^4 f_1^5 f_2^5 f_3^5 \right) \wedge \\
& \wedge \left(f_4^3 f_5^3 f_1^4 f_2^4 f_3^4 \vee (f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5)(f_4^4 f_5^4 f_1^5 \vee f_2^5 f_3^5)(f_2^5 \vee f_3^5) \right) \wedge \\
& \wedge (f_4^4 \vee f_5^4 \vee f_1^5) \left((f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5) \vee f_2^5 f_3^5 \right) (f_4^4 f_5^4 f_1^5 \vee f_2^5 \vee f_3^5) \\
& f_8 = (g_1^1 \vee g_2^1 \vee g_3^1)((g_1^1 \vee g_2^1)(g_1^1 g_2^1 \vee g_3^1) \vee g_1^2 g_2^2)(g_1^1 g_2^1 g_3^1 \vee g_1^2 \vee g_2^2) \wedge \\
& \wedge \left((g_1^1 \vee g_2^1)(g_1^1 g_2^1 \vee g_3^1)(g_1^1 g_2^1 g_3^1 \vee g_1^2 g_2^2)(g_1^2 \vee g_2^2) \vee g_3^2 g_4^2 f_1^3 f_2^3 f_3^3 \right) \wedge
\end{aligned}$$

$$\begin{aligned}
& \wedge (g_1^1 g_2^1 g_3^1 g_1^2 g_2^2 \vee (g_3^2 \vee g_4^2)(g_3^2 g_4^2 \vee f_1^3)(g_3^2 g_4^2 f_1^3 \vee f_2^3 f_3^3)(f_2^3 \vee f_3^3)) \wedge \\
& \wedge (g_3^2 \vee g_4^2 \vee f_1^3)((g_3^2 \vee g_4^2)(g_3^2 g_4^2 \vee f_1^3) \vee f_2^3 f_3^3)(g_3^2 g_4^2 f_1^3 \vee f_2^3 \vee f_3^3) \vee \\
& \vee (f_4^3 \vee f_5^3 \vee f_1^4 \vee f_2^4 f_3^4)((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4) \vee f_2^4 \vee f_3^4) \wedge \\
& \wedge \left((f_4^3 \vee f_5^3 \vee f_1^4) \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4) \vee f_2^4 f_3^4 \right) (f_4^3 f_5^3 f_1^4 \vee f_2^4 \vee f_3^4) \vee \right. \\
& \quad \left. \vee f_4^4 f_5^4 f_1^5 f_2^5 f_3^5 \right) \wedge \\
& \wedge \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4)(f_4^3 f_5^3 f_1^4 \vee f_2^4 f_3^4)(f_2^4 \vee f_3^4) \vee \right. \\
& \quad \left. \vee (f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5)(f_4^4 f_5^4 f_1^5 \vee f_2^5 f_3^5)(f_2^5 \vee f_3^5) \right) \wedge \\
& \wedge \left(\begin{array}{c} f_4^3 f_5^3 f_1^4 f_2^4 f_3^4 \vee \\ \vee (f_4^4 \vee f_5^4 \vee f_1^5) \left((f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5) \vee f_2^5 f_3^5 \right) (f_4^4 f_5^4 f_1^5 \vee f_2^5 \vee f_3^5) \end{array} \right) \wedge \\
& \wedge (f_4^4 \vee f_5^4 \vee f_1^5 \vee f_2^5 f_3^5) \left((f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5) \vee f_2^5 \vee f_3^5 \right) \\
f_9 = & (g_1^1 \vee g_2^1)(g_1^1 g_2^1 \vee g_3^1)(g_1^1 g_2^1 g_3^1 \vee g_1^2 g_2^2)(g_1^2 \vee g_2^2) \wedge \\
& \wedge (g_1^1 g_2^1 g_3^1 g_1^2 g_2^2 \vee g_3^2 g_4^2 f_1^3 f_2^3 f_3^3)(g_3^2 \vee g_4^2)(g_3^2 g_4^2 \vee f_1^3)(g_3^2 g_4^2 f_1^3 \vee f_2^3 f_3^3) \wedge \\
& \wedge (f_2^3 \vee f_3^3) \vee (f_4^3 \vee f_5^3 \vee f_1^4 \vee f_2^4 \vee f_3^4) \wedge \\
& \wedge \left((f_4^3 \vee f_5^3 \vee f_1^4 \vee f_2^4 f_3^4) \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4) \vee f_2^4 \vee f_3^4 \right) \vee f_4^4 f_5^4 f_1^5 f_2^5 f_3^5 \right) \wedge \\
& \wedge \left((f_4^3 \vee f_5^3 \vee f_1^4) \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4) \vee f_2^4 f_3^4 \right) (f_4^3 f_5^3 f_1^4 \vee f_2^4 \vee f_3^4) \vee \right. \\
& \quad \left. \vee (f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5)(f_4^4 f_5^4 f_1^5 \vee f_2^5 f_3^5)(f_2^5 \vee f_3^5) \right) \wedge \\
& \wedge \left(\begin{array}{c} (f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4)(f_4^3 f_5^3 f_1^4 \vee f_2^4 f_3^4)(f_2^4 \vee f_3^4) \vee \\ \vee (f_4^4 \vee f_5^4 \vee f_1^5) \left((f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5) \vee f_2^5 f_3^5 \right) (f_4^4 f_5^4 f_1^5 \vee f_2^5 \vee f_3^5) \end{array} \right) \wedge \\
& \wedge \left(f_4^3 f_5^3 f_1^4 f_2^4 f_3^4 \vee (f_4^4 \vee f_5^4 \vee f_1^5 \vee f_2^5 f_3^5) \left((f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5) \vee f_2^5 \vee f_3^5 \right) \right) \wedge \\
& \wedge (f_4^4 \vee f_5^4 \vee f_1^5 \vee f_2^5 \vee f_3^5) \\
f_{10} = & g_1^1 g_2^1 g_3^1 g_1^2 g_2^2 g_3^2 g_4^2 f_1^3 f_2^3 f_3^3 \vee (f_4^3 \vee f_5^3 \vee f_1^4 \vee f_2^4 \vee f_3^4 \vee f_4^4 f_5^4 f_1^5 f_2^5 f_3^5) \wedge \\
& \wedge \left((f_4^3 \vee f_5^3 \vee f_1^4 \vee f_2^4 f_3^4) \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4) \vee f_2^4 \vee f_3^4 \right) \vee \right. \\
& \quad \left. \vee (f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5)(f_4^4 f_5^4 f_1^5 \vee f_2^5 f_3^5)(f_2^5 \vee f_3^5) \right) \wedge \\
& \wedge \left((f_4^3 \vee f_5^3 \vee f_1^4) \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4) \vee f_2^4 f_3^4 \right) (f_4^3 f_5^3 f_1^4 \vee f_2^4 \vee f_3^4) \vee \right. \\
& \quad \left. \vee (f_4^4 \vee f_5^4 \vee f_1^5) \left((f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5) \vee f_2^5 f_3^5 \right) (f_4^4 f_5^4 f_1^5 \vee f_2^5 \vee f_3^5) \right) \wedge
\end{aligned}$$

$$\begin{aligned}
& \wedge \left(\begin{aligned} & (f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4)(f_4^3 f_5^3 f_1^4 \vee f_2^4 f_3^4)(f_2^4 \vee f_3^4) \vee \\ & \vee (f_4^4 \vee f_5^4 \vee f_1^5 \vee f_2^5 f_3^5) \left((f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5) \vee f_2^5 \vee f_3^5 \right) \end{aligned} \right) \wedge \\
& \wedge (f_4^3 f_5^3 f_1^4 f_2^4 f_3^4 \vee f_4^4 \vee f_5^4 \vee f_1^5 \vee f_2^5 \vee f_3^5) \\
f_{11} &= f_4^3 \vee f_5^3 \vee f_1^4 \vee f_2^4 \vee f_3^4 \vee \\
& \vee (f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5)(f_4^4 f_5^4 f_1^5 \vee f_2^5 f_3^5)(f_2^5 \vee f_3^5) \\
f_{12} &= (f_4^3 \vee f_5^3 \vee f_1^4 \vee f_2^4 f_3^4) \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4) \vee f_2^4 \vee f_3^4 \right) \vee \\
& \vee (f_4^4 \vee f_5^4 \vee f_1^5) \left((f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5) \vee f_2^5 f_3^5 \right) (f_4^4 f_5^4 f_1^5 \vee f_2^5 \vee f_3^5) \\
f_{13} &= (f_4^3 \vee f_5^3 \vee f_1^4) \left((f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4) \vee f_2^4 f_3^4 \right) (f_4^3 f_5^3 f_1^4 \vee f_2^4 \vee f_3^4) \vee \\
& \vee (f_4^4 \vee f_5^4 \vee f_1^5 \vee f_2^5 f_3^5) \left((f_4^4 \vee f_5^4)(f_4^4 f_5^4 \vee f_1^5) \vee f_2^5 \vee f_3^5 \right) \\
f_{14} &= (f_4^3 \vee f_5^3)(f_4^3 f_5^3 \vee f_1^4)(f_4^3 f_5^3 f_1^4 \vee f_2^4 f_3^4)(f_2^4 \vee f_3^4) \vee f_4^4 \vee f_5^4 \vee f_1^5 \vee f_2^5 \vee f_3^5
\end{aligned}$$

Структуру моделі системи наведено на рис. 2.

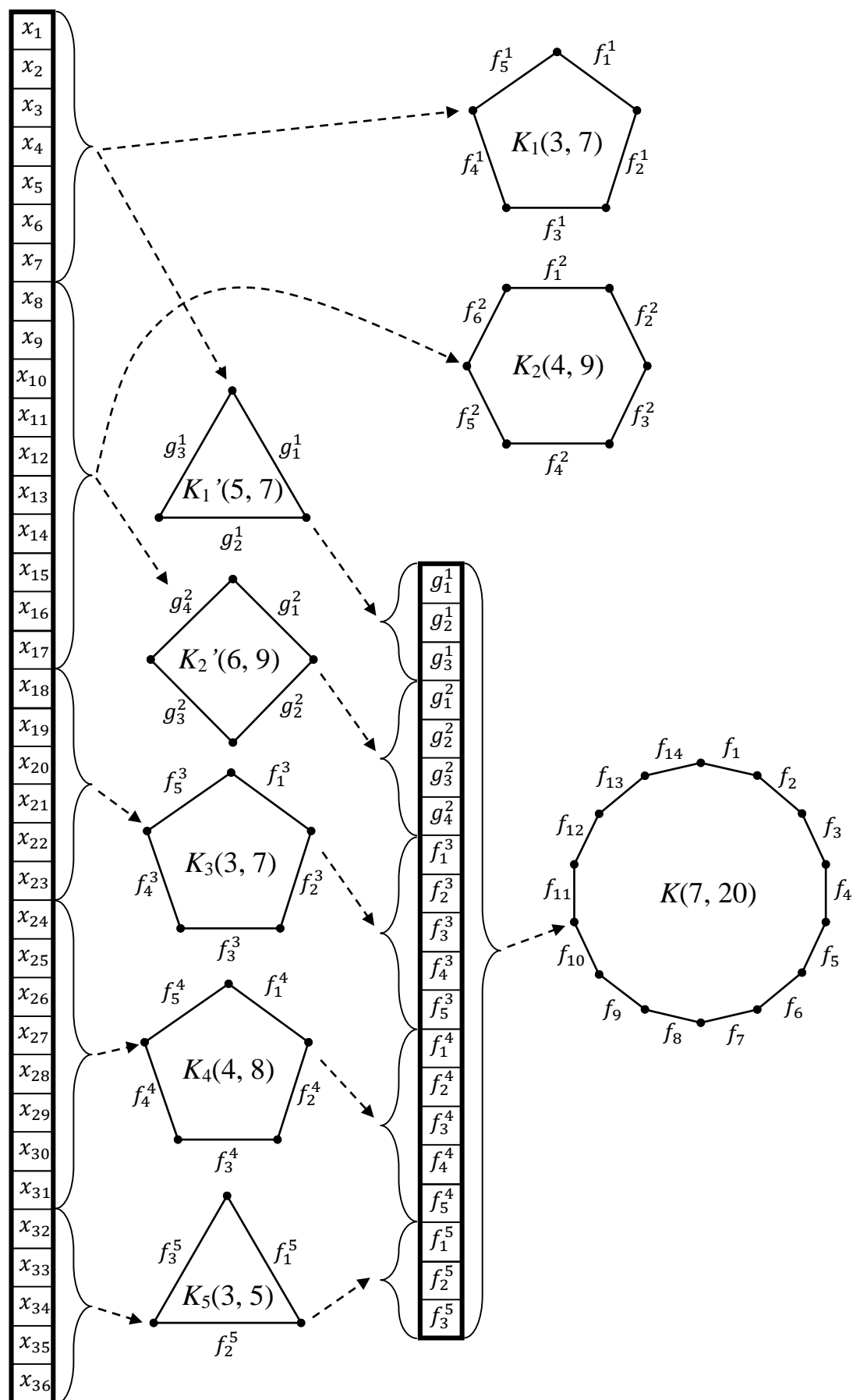


Рис. 2. Модель складної системи з ковзним резервом

Додаток М

Альтернативна модель системи типу «донор»-«реципієнт» з обмеженнями

Структура системи наведена на рис. 1. Вважатимемо, що станам процесорів першої підсистеми відповідають змінні x_1, x_2, \dots, x_{10} , а другої – змінні $x_{11}, x_{12}, \dots, x_{17}$.

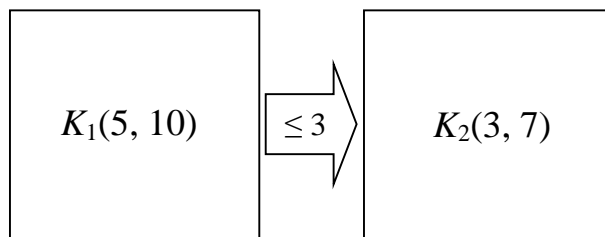


Рис. 1. Структура системи «донор»-«реципієнт» з обмеженнями

Для другої підсистеми побудуємо допоміжну модель $K_2'''(4, 7)$ з реберними функціями:

$$e_1^2 = x_{11} \vee x_{12} \vee x_{13} \vee x_{14}$$

$$e_2^2 = (x_{11} \vee x_{12} \vee x_{13}x_{14})(x_{11}x_{12} \vee x_{13} \vee x_{14}) \vee x_{15}x_{16}x_{17}$$

$$e_3^2 = (x_{11} \vee x_{12})(x_{11}x_{12} \vee x_{13}x_{14})(x_{13} \vee x_{14}) \vee (x_{15} \vee x_{16})(x_{15}x_{16} \vee x_{17})$$

$$e_4^2 = x_{11}x_{12}x_{13}x_{14} \vee x_{15} \vee x_{16} \vee x_{17}$$

Сформуємо вхідний вектор $\langle e_1^2, e_2^2, e_3^2, e_4^2, h_1^1, h_2^1, h_3^1 \rangle$ для моделі $K_2(3, 7)$, що міститиме наступні реберні функції:

$$f_1^2 = e_1^2 \vee e_2^2 \vee e_3^2 e_4^2$$

$$f_2^2 = e_1^2 e_2^2 \vee e_3^2 \vee e_4^2$$

$$f_3^2 = (e_1^2 \vee e_2^2)(e_1^2 e_2^2 \vee e_3^2 e_4^2)(e_3^2 \vee e_4^2) \vee h_1^1 h_2^1 h_3^1$$

$$f_4^2 = e_1^2 e_2^2 e_3^2 e_4^2 \vee (h_1^1 \vee h_2^1)(h_1^1 h_2^1 \vee h_3^1)$$

$$f_5^2 = h_1^1 \vee h_2^1 \vee h_3^1$$

Структуру моделі системи наведено на рис. 2.

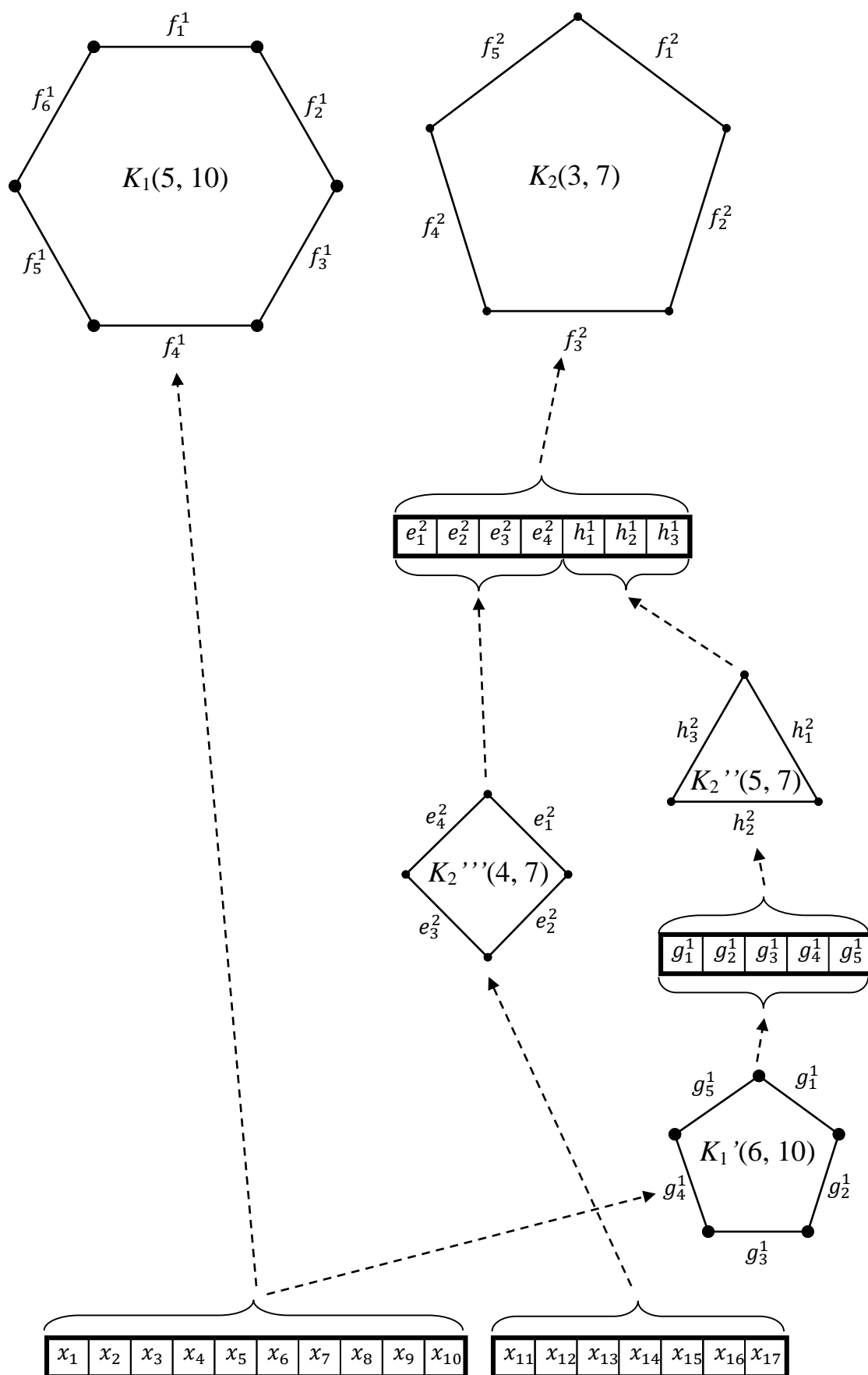


Рис. 2. Альтернатива модель системи «донор»-«реципієнт» з обмеженнями

Додаток Н

ЗАТВЕРДЖУЮ
Перший проректор Національного
технічного університету України
«Київський політехнічний інститут
ім. Ігоря Сікорського»

Юрій ЯКИМЕНКО

« 27 » січня 2021 р.

А К Т

впровадження результатів дисертаційного дослідження асистента кафедри Системного програмування і спеціалізованих комп'ютерних систем факультету прикладної математики Національного технічного університету України «КПІ ім. Ігоря Сікорського» Морозова Костянтина В'ячеславовича на тему «Методи і засоби побудови моделей поведінки небазових відмовостійких багатопроцесорних систем» на здобуття наукового ступеня кандидата технічних наук.

Комісія у складі: голова – декан факультету прикладної математики Національного технічного університету України «КПІ ім. Ігоря Сікорського», д.т.н., проф. Дичка І.А.; члени комісії – професор кафедри СПіСКС НТУУ «КПІ ім. І. Сікорського», д.т.н., проф. Романкевич О.М., доцент кафедри СПіСКС НТУУ «КПІ ім. І. Сікорського», к.т.н., доц. Орлова М.М. цим Актом засвідчує, що результати дисертаційного дослідження Морозова Костянтина використані співробітниками кафедри СПіСКС КПІ ім. Ігоря Сікорського при підготовці та викладанні курсу лекцій «Тестування, надійність, контроль та діагностика комп'ютерних систем» спеціальності «Спеціалізовані комп'ютерні системи». Зокрема впроваджено метод оцінки заданої множини взаємоперевірок та визначення станів процесорів відмовостійкої багатопроцесорної системи при виконанні її самотестування.

Голова комісії

д.т.н., проф.



Іван ДИЧКА

Члени комісії

д.т.н., проф.



Олексій РОМАНКЕВИЧ

к.т.н., доц.



Марія ОРЛОВА

Акт

впровадження результатів дисертаційної роботи Морозова К.В.

Даний акт підтверджує, що при виконанні наукового завдання «Підвищення ефективності роботи ТЕЦ шляхом утилізації теплових втрат із використанням теплонасосної технології» в 2019 - 2020 рр. в рамках імовірнісного аналізу безпеки ТЕЦ та виконання Цільової програми наукових досліджень НАН України «Інтелектуальна екологічно безпечна енергетика з традиційними та відновлюваними джерелами енергії» було використано наступні результати дисертаційної роботи Морозова Костянтина В'ячеславовича *«Методи і засоби побудови GL-моделей для небазових відмовостійких багатопроцесорних систем»*:

- методи побудови моделей поведінки в потоці відмов ієрархічних систем;
- методи побудови моделей поведінки систем із ковзним резервом;
- оцінка параметрів надійності розроблених моделей шляхом проведення статистичних експериментів.

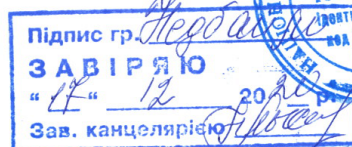
Результати дисертаційної роботи Морозова К.В. також можуть бути поширені та успішно використані на об'єктах енергетики, в тому числі, ТЕС та АЕС.

Д.Т.Н., С.Н.С.,

старший науковий співробітник

відділу ТОЕТ ІТТФ НАН України

Недбайло О.М.



Додаток II

1. Romankevich A.M. Hierarchical Graph-Logical Models of Multiprocessor Systems Based on Grouping of Their Components / A.M. Romankevich, K.V. Morozov, V.A. Romankevich // IJCSNS International Journal of Computer Science and Network Security. – 2019.– Vol. 19, No.10.– P. 138-143.
2. Romankevich A.M. Graph-Logic Models of Hierarchical Fault-Tolerant Multiprocessor Systems / A.M. Romankevich, K.V. Morozov, V.A. Romankevich // IJCSNS International Journal of Computer Science and Network Security.– 2019.– Vol. 19, No.7.– P. 151-156.
3. Морозов К.В. О характере влияния модификации рёберных функций GL-модели на её поведение в потоке отказов / К.В. Морозов, А.М. Романкевич, В.А. Романкевич // Радіоелектронні і комп'ютерні системи. – №6, 2016.– С.108-112.
4. Морозов К.В. Про модифікацію графо-логічної моделі / К.В. Морозов, В.О. Романкевич, К.Р. Потапова // Інформаційні технології та комп'ютерна інженерія.– Том 3, № 34.– 2015.– С.42-49..
5. Романкевич В.А. Об одном методе модификации рёберных функций GL-моделей / В.А. Романкевич, К.В. Морозов, А.П. Фесенюк // Радіоелектронні і комп'ютерні системи. – №6, 2014. – С.95-99.
6. Романкевич В.А. Определение достаточного уровня отказоустойчивости для обеспечения заданной вероятности безотказной работы многопроцессорной системы / В.А. Романкевич, А.П. Фесенюк, К.В. Морозов // Journal of Qafqaz university – Mathematics and computer science. – Volume 1, Number 2. – 2013. – pp.118-126.
7. Романкевич А.М. Об одной GL модели системы со скользящим резервом / А.М. Романкевич, В.А. Романкевич, К.В. Морозов // Радіоелектронні і комп'ютерні системи. – №5, 2013.– С.333-336.
8. Романкевич А.М. Об одной задаче реконфигурирования в многопроцессорных системах / А.М. Романкевич, Мораведж Сейед Милад,

- В.А. Романкевич, К.В. Морозов // Радіоелектронні і комп'ютерні системи. – №5, 2012.– С.91-94.
9. Майданюк І.В. Об одном свойстве GL-модели с минимальным количеством теряемых рёбер / И.В. Майданюк, К.В. Морозов, Е.Р. Потапова, А.В. Шурига // Науковий вісник Чернівецького університету. Сер.: Комп'ютерні системи та компоненти.– 2010. – Т.1, вип. 2.– С.31-34.
 10. Морозов К.В. О модификации графо-логической модели для систем и их компонентов с множественными состояниями / К.В. Морозов, В.А. Романкевич, Т.Г. Сапсай // Системный анализ и информационные технологии: материалы 19 й Международной научно-практической конференции SAIT 2017. – К.: УНК «ИПСА» КПИ им. Игоря Сикорского, 2017. – С.94.
 11. Морозов К.В. GL-модель иерархической системы с процессорами на всех уровнях иерархии / К.В. Морозов, Е.Р. Потапова, Н.К. Кичигин // Комп'ютерні системи і проектування технологічних процесів та обладнання: Матеріали всеукраїнської науково-практичної конференції (17-19 лютого 2016р). – Чернівці: ЧФ НТУ «ХП», 2016.- С.35-37.
 12. Морозов К.В. О самотестировании многопроцессорных систем / К.В. Морозов, Е.Р. Потапова // Системный анализ и информационные технологии: материалы 18-й Международной научно-практической конференции SAIT 2016. – К.: УНК «ИПСА» НТУУ «КПИ», 2016. – С.396.
 13. Морозов К.В. О графо-логических моделях взвешенных систем / К.В. Морозов, В.А. Романкевич, В.В. Олейник // Інформаційні технології в освіті, техніці та промисловості. Матеріали II всеукраїнської науково-практичної конференції молодих учених і студентів. – Івано-Франківськ – 2015. – С. 206-207..
 14. Потапова К.Р. Формування GL-моделей для систем з ієрархічною структурою / К.Р. Потапова, К.В. Морозов, М.І. Трошина // Прикладна математика та комп'ютинг (ПМК-2015). Збірник тез доповідей. – Київ, НТУУ "КПІ". – 2015. – С. 156 160.

15. Романкевич А.М. Об одном условии модификации рёберных функций GL-модели / А.М. Романкевич, В.А. Романкевич, К.В. Морозов // Системный анализ и информационные технологии: материалы 17-й Международной научно-практической конференции SAIT 2015. - К.: УНК «ИПСА» НТУУ «КПИ», 2015. – С.106.
16. Романкевич А.М. О модификации реберных функций GL-моделей / А.М. Романкевич, Т.Г. Сапсай, К.В. Морозов // Інформаційно-керуючі системи на залізничному транспорті. Тези доповідей. – 2013. – №4 (додаток). – С. 36.
17. Гроль В.В. Упрощенная 3 отказоустойчивая GL-модель для отказоустойчивых многопроцессорных систем / В.В. Гроль, К.В. Морозов, В.А. Романкевич // Інформаційно-керуючі системи на залізничному транспорті. Тези доповідей. – 2012. – №5. – С. 90.
18. Романкевич В.О. Базова $K(3,n)$ GL-модель, яка будується методом зсуву змінних / В.О. Романкевич, К.В. Морозов // Прикладна математика та комп'ютинг (ПМК 2012). Збірник тез доповідей. – Київ, НТУУ "КПІ". – 2012. – С.230-235.
19. Романкевич В.А. Построение GL моделей поведения в потоке отказов для систем со скользящим резервом / В.А. Романкевич, Мораведж Милад, К.В. Морозов // Тези доповідей Третьої міжнародної науково-практичної конференції «Методи та засоби кодування, захисту й ущільнення інформації». – Вінниця, ВНТУ. – 2011. – С.36-37.
20. Романкевич В.А. О расчёте надёжности отказоустойчивых многопроцессорных систем, подсистемы которых имеют общие процессоры / В.А. Романкевич, А.П. Фесенюк, К.В. Морозов // Інформаційно-керуючі системи на залізничному транспорті. Тези доповідей. – 2010. – №4 (додаток). – С. 6.
21. Романкевич В.А. Об оптимизации вычисления рёберных функций в GL-моделях / В.А. Романкевич, К.В. Морозов // Інформаційні технології та комп'ютерна інженерія. Тези доповідей Міжнародної науково-практичної конференції. – Вінниця, ВНТУ. – 2010. – С.379-380.

- 22.Потапова К.Р. Комп'ютерна програма «Програма підтримки експериментів з графо-логічними моделями відмово стійких багатопроцесорних систем» / К.Р. Потапова, М.І. Трошина, К.В. Морозов, Б.А. Корнейчик // Свідоцтво про реєстрацію авторського права на твір №59822 від 27.05.2015р.
- 23.Шурига О.В. Комп'ютерна програма «Програма формування та оцінки складності реберних функцій графо-логічної моделі поведінки відмовостійких багатопроцесорних систем у потоці відмов методом поділу множини змінних на дві рівні частини і методом нерівного поділу» / О.В. Шурига, В.О. Романкевич, К.В. Морозов // Свідоцтво про реєстрацію авторського права на твір №39445 від 02.08.2011р.