

УНИВЕРЗИТЕТ У БЕОГРАДУ
ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА

Драгана Д. Макајић-Николић

**НОВИ ПРИСТУП АНАЛИЗИ
ПОУЗДАНОСТИ СИСТЕМА ПРИМЕНОМ
ИНВЕРЗНИХ ПЕТРИЈЕВИХ МРЕЖА**

докторска дисертација

Београд, 2012

UNIVERSITY OF BELGRADE
FACULTY OF ORGANIZATIONAL SCIENCES

Dragana D. Makajić-Nikolić

**A NEW APPROACH TO THE SYSTEM
RELIABILITY ANALYSIS USING
REVERSE PETRI NETS**

Doctoral Dissertation

Belgrade, 2012

КОМИСИЈА

МЕНТОР

др Мирко Вујошевић, редовни професор, Универзитет у Београду, ФОН

ЧЛАНОВИ КОМИСИЈЕ:

др Мирјана Чангаловић, редовни професор, Универзитет у Београду, ФОН

др Милан Станојевић, ванредни професор, Универзитет у Београду, ФОН

др Драган Радојевић, научни саветник, Институт Михајло Пупин, Београд

др Владимир Поповић, доцент, Универзитет у Београду, Машински факултет

НОВИ ПРИСТУП АНАЛИЗИ ПОУЗДАНОСТИ СИСТЕМА ПРИМЕНОМ ИНВЕРЗНИХ ПЕТРИЈЕВИХ МРЕЖА

Анализа стабла неисправности (АСН) је техника за анализу поузданости која се користи за одређивање узрока и вероватноће отказа система. АСН је базирана на стаблу неисправности (СН), графичком моделу који користи логичка кола и отказне догађаје за представљање узрочно-последичних веза између догађаја који претходе отказу система. Квалитативни део АСН састоји се у одређивању минималних скупова пресека. Скуп пресека је скуп примарних догађаја који, када се догоде истовремено, доводе до отказа система. Минимални скуп пресека (минипресек) је скуп пресека који је редукован на минимални број елемената који изазивају отказ система.

У овој дисертацији је предложена нова метода за одређивање минипресека кохерентног СН, СН које садржи само И и ИЛИ логичка кола, са вишеструким догађајима. Метода је заснована на посебном типу Петријевих мрежа – инверзним Петријевим мрежама.

Прво је представљен нови алгоритам за редукцију скупова пресека кохерентног СН. Одређивање свих минипресека кохерентног СН је НП тежак проблем. У дисертацији се разматрају приступи којима се прво одређују сви скупови пресека датог СН а затим се врши елиминисање надскупова, односно скупова пресека који нису минимални. У тим приступима, СН се трансформише у еквивалентну булову једначину у којој се, затим, елиминишу сви редундантни скупови пресека. Већ је доказано да су скупови пресека, који не садрже вишеструке догађаје, минимални. Тиме се редукција ограничава само на скупове пресека са вишеструким догађајима. У овој дисертацији се посматра још једна врста скупова пресека: они који, ако садрже неки вишеструки догађај, садрже сва његова понављања. Овакви скупови пресека су означени са C^* . Показује се да је скуп пресека облика C^* , ако постоји, такође минималан. Тиме се додатно скраћује поступак редукције булове једначине. Затим се, даље, доказују услови за постојање скупова пресека облика C^* и одређује минимална број скупова пресека који се могу елиминисати као

надскуп од S^* . Предложен је нови алгоритам за редукцију булове једначине датог СН, који се базира на раздвајању скупова пресека у три групе: скупови пресека без вишеструких догађаја, скупови пресека облика S^* и остали скупови пресека. Ефикасност алгоритма је илустрована на групи тест.

У наставку је предложена нова метода за одређивање минипресека кохерентног СН са вишеструким догађајима. Метода је заснована на инверзним Петријевим мрежама (ИПМ). Петријеве мреже су оријентисани графови којима је придружено почетно стање, названо почетно маркирање. Структура Петријеве мреже је оријентисан, тежински, бипартитни граф са две врсте чворова: места, која представљају услов или последицу и прелаза, који представљају акцију у систему. Овде се користи посебна врста Петријевих мрежа – ИПМ. ИПМ има исту структуру као дата Петријева мрежа, односно, места прелазе и гране, али су јој гране супротно оријентисане у однову на дату Петријеву мрежу. Показано је да се дата СН може директно трансформисати у одговарајућу ИПМ.

Основна својства ИПМ су: ИПМ је Петријева мрежа у којој су примарни догађаји датог СН моделирани местима ушћа, а вршни догађај изворним местом; Сваки скуп пресека датог СН се може представити маркирањем ИПМ са жетонима у местима ушћа, која моделирају одговарајуће примарне догађаје. Такво маркирање је мртво; Свако маркирање $M = [1, M(p_2), \dots, M(p_m)]^T$, односно маркирање у коме је $M(p_1) = 1$, моделира одигравање вршног догађаја, што представља отказ система; Свако мртво маркирање M_0 са жетонима само у местима ушћа, које је достижно из маркирања $M = [1, M(p_2), \dots, M(p_m)]^T$, представља скуп пресека; Свако мртво маркирање M_0 са жетонима само у местима ушћа, које је достижно из маркирања $M = [1, 0, \dots, 0]^T$, представља минипресек.

Коришћењем својстава ИПМ и једначине стања, показано је да проблем одређивања минипресека датог СН, постаје следећи проблем достижности: одредити сва мртва маркирања достижна из почетног маркирања $M = [0, 0, \dots, 0, 1]^T$ ИПМ датог СН. Сви скупови пресека се тада добијају као листови графа достижности ИПМ. Формулисан је алгоритам за одређивање свих минипресека датог СН. Алгоритам захтева специјално нумерисање места и прелаза ИПМ, чиме

се омогућава да се не одређује цео граф достижности, већ само једно његово стабло. Алгоритам, такође, укључује раздвајање скупова пресека у три групе: скупови пресека без вишеструких догађаја, скупови пресека облика C^* и остали скупови пресека. За тестирање алгоритма коришћена је група тест примера.

На крају је разматран проблем комбинаторне експлозије. Како број минипресека у сложеним системима може бити веома велики, одређивање свих минипресека може бити веома сложен комбинаторни проблем. У овој дисертацији је предложен приступ за решавање овог проблема, заснован на одсецању стабла достижности и одређивању само минипресека вишег реда. Скуп пресека реда l садржи l примарних догађаја. Предложеним приступом се одређују само минипресеци вишег реда (минипресеци са l или мање примарних догађаја). Пошто постојање жетона у месту представља одигравање одговарајућег догађаја, укупан број жетона у ИПМ представља број догађаја који су се догодили истовремено. Одатле следи да број жетона у листу стабла достижности, представља ред одговарајућег скупа пресека. Због тога је број жетона у чворовима стабла достижности прихваћен као критеријум одсецања.

Ако СН садржи вишеструке догађаје, број жетона у скупу пресека може бити редукован применом закона апсорпције и идемпотенције. Због тога се, приликом одређивања параметра одсецања, у сваком чвору стабла достижности мора узети у обзир број жетона и будуће примене ова два закона. У овој дисертацији су предложени алгоритам за одређивање минипресека датог реда и алгоритам за одређивање параметра одсецања, који су илустровани на групи тест примера.

Кључне речи: стабло неисправности; минимални скупови пресека; инверзне Петријеве мреже; редукција скупова пресека

НАУЧНА ОБЛАСТ: ОПЕРАЦИОНА ИСТРАЖИВАЊА

УЖА НАУЧНА ОБЛАСТ: ПОУЗДАНОСТ

A NEW APPROACH TO THE SYSTEM RELIABILITY ANALYSIS USING REVERSE PETRI NETS

The Fault Tree Analysis (FTA) is a reliability analysis technique used to determine the root causes and probability of occurrence of a specified top event. FTA is based on a Fault Tree (FT), a graphical model using logic gates and fault events to model the cause-effect relationships involved in causing the top event. Determining minimal cut sets is a qualitative part in the FTA. The cut set is a set of basic events which, when simultaneous, cause the top event to occur. The minimal cut set (minicut) is a cut set which has been reduced to the minimum number of events that cause the top event to occur.

This Dissertation proposes a new method for minicuts generation of a coherent FT, constructed using AND and OR logic operator only, with repeated events. The approach is based on the special type of Petri Nets – Reverse Petri Net.

First, a new algorithm for reducing cut sets in coherent fault trees is presented. Determining all minicuts of a fault tree is NP-hard problem. Coherent fault trees and the top-down approaches for minicuts generation are considered. The FT can be translated into an equivalent Boolean expression. Obtained Boolean expression then should be reduced by eliminating all redundant cut sets. It is already proved that the cut sets not containing any repeated event are minicuts. This limits the reduction only to the cut sets containing repeated events. Cut sets containing all repetitions of its events are denoted by $\{C^*\}$. It is proved that C^* , if exists, is also minicut. This further limits the reduction of the Boolean expression. In addition, we proved conditions for existence of C^* and calculated the minimal number of cut sets that can be eliminated as subsets of C^* . Finally, a new algorithm for reduction of the Boolean expression which is based on the partition of the cut sets into three families: those not containing any repeated event, those of type C^* , and others, is proposed. The efficiency of the algorithm is shown by applying it to some benchmark fault trees.

Then, a new method for minicuts generation of a coherent FT with repeated events based on Reverse Petri Nets is presented. A Petri net is a particular kind of a directed

graph, together with an initial state called initial marking. The underlying graph of a Petri net is a directed, weighted, bipartite graph consisting of two kinds of nodes, called places and transitions, where arcs are either from a place to a transition or from a transition to a place. In modelling, using the concept of conditions and events, places represent conditions, and transitions represent events. Here, a special type of Petri net is used: Reverse Petri Net (RPN). RPN of a given Petri net has the same net structure, e.g. places, transitions and arcs as given Petri net but the arcs are reverse directed. It is shown that the reverse Petri Net (RPN) can be generated directly from the FT.

Main properties of RPN are: RPN is a Petri net with a sink place for each basic event and with one source place for the top event T of a FT; A cut set of a FT can be represented by marking of RPN with tokens only in sink places which models its basic events. Such marking is dead marking; Any marking $M = [1, M(p_2), \dots, M(p_m)]^T$, where $M(p_1) = 1$, represents the occurrence of the top event T, e.g. system failure; Each dead marking M_0 with tokens only in source places, which is reachable from the marking $M = [1, M(p_2), \dots, M(p_m)]^T$, represents a cut set. Each dead marking M_0 reachable from $M = [1, 0, \dots, 0]^T$, represents a minicut.

Using RPN properties and state equation show that the problem of finding all minimal cut sets becomes the following direct reachability problem: to find all dead markings reachable from marking $M = [0, 0, \dots, 0, 1]^T$ of RPN. All cut sets are then obtained as leaves of the reachability graph of RPN. An algorithm to determine all leaves of the reachability graph is presented in the paper. The algorithm requires special numeration of the places and transitions of RPN which enables the generation only one spanning trees of the reachability graph. The algorithm also includes the partition of the cut sets into three families: those not containing any repeated event, those of type C*, and others. The efficiency of the algorithm is illustrated on some benchmark fault trees.

Finally, the problem of state explosion is observed. Since the number of minicuts in complex systems can be very large and impossible to calculate, the enumeration of leafs of the reachability tree can be a complex combinatorial problem. An approach for solving the complexity problem, based on cutting the reachability tree of RPN and generating minicuts of a higher order only, is proposed. A cut set order is the number of

events in a cut set. The minicuts of an order l contain l basic events. Only the minicuts of a higher order using proposed approach (minicuts of an order l or less) will be found. Since the existence of a token in the place models the occurrence of the corresponding event, the total number of tokens in RPN represents the number of simultaneously occurred events. Consequently, the number of tokens in each node of the reachability tree of RPN represents the number of the occurred events. Finally, the number of tokens in each leaf of the reachability tree represents the order of the corresponding cut set. Hence, the numbers of tokens in nodes of a reachability tree may be accepted as cutting criteria.

In the FT contains repeated events, the number of events of cut set can be reduced by applying Boolean laws such as the absorption and idempotent laws. Therefore, in each node of the reachability tree, the number of token and the further applications of the absorption and idempotent laws must be taken into account during the determination of the cutting parameters. An algorithm for minicuts of a given order generation as well as the algorithm for the cutting parameters determination are given and tested on several benchmark FT.

Keywords: fault tree; minimal cut set; reverse Petri net; cut set reduction

SCIENTIFIC FIELD: OPERATIONS RESEARCH

PARTICULAR SCIENTIFIC FIELD: RELIABILITY

САДРЖАЈ

1. УВОД	1
2. АНАЛИЗА СТАБЛА НЕИСПРАВНОСТИ (АСН)	5
2.1. Историјски развој АСН	6
2.2. Методологија АСН	8
2.3. Дефинисање система и нежељеног догађаја	8
2.4. Конструкција стабла неисправности	10
2.5. Квалитативна АСН.....	14
2.5.1. Одређивање минималних скупова пресека	15
2.5.1.1. Методе одозго наниже	17
2.5.1.2. Методе одоздо навише	24
2.5.1.3. Бинарни дијаграми одлучивања	25
2.5.1.4. Одређивање минималних скупова путева	29
2.5.2. Рангирање минипресека и примарних догађаја	31
2.6. Квантитативна АСН.....	32
2.6.1. Одређивање вероватноће вршног догађаја	32
2.6.2. Мере значајности	36
3. АНАЛИЗА СН ВЕЛИКИХ ДИМЕНЗИЈА	39
3.1. Редукција пре одређивања скупова пресека	40
3.2. Редукција након одређивања скупова пресека	43
3.3. Директно издвајање минипресека	46
3.3.1. Теорема Лимниоса	47
3.3.2. Теорема о минималности скупова пресека са вишеструким догађајима	49
3.3.3. Структура СН које садржи C^*	57
3.3.4. Минимални број редукованих скупова пресека	61
3.3.5. Алгоритам редукције скупова пресека СН	63
3.3.6. Резултати експеримената	64
3.4. Одређивање минипресека датог реда	66

4. ПЕТРИЈЕВЕ МРЕЖЕ (ПМ)	69
4.1. Основни појмови, дефиниција и динамика ПМ	70
4.2. Врсте ПМ	76
4.2.1. Бинарне ПМ	76
4.2.2. Црно-беле ПМ	79
4.2.3. Петријеве мреже вишег нивоа	80
4.3. Проширења ПМ	84
4.4. Својства ПМ	86
4.5. Анализа ПМ	88
4.5.1. Редукција ПМ	88
4.5.2. Матрица инциденције и једначина стања	89
4.5.3. Инваријанте	91
4.5.4. Граф достижности	94
4.5.5. Симулација ПМ	95
4.6. Инверзне ПМ (ИПМ).....	96
4.7. Област примене ПМ	97
5. ПРИМЕНА ПМ У АСН	100
5.1. Графичка метода за одређивање минипресека	102
5.2. Матрична метода	104
5.3. Подела ПМ на нивое	106
5.4. Одређивање минипресека помоћу t -инваријанте	111
5.5. Одређивање минипресека на основу матрице инциденције ПМ	113
6. ОДРЕЂИВАЊЕ МИНИПРЕСЕКА СН БЕЗ ВИШЕСТРУКИХ ДОГАЂАЈА ПОМОЋУ ИПМ	117
6.1. Основни појмови	117
6.1.1. Презентација логичких кола помоћу ПМ	117
6.1.2. Проблем достижности и мртво маркирање	119
6.2. Еквивалента ПМ	119
6.3. Основна идеја методе	122
6.4. Одређивање минипресека помоћу стабла достижности	124
6.5. Алгоритам за одређивање свих минипресека	132

6.6. Одређивање минипресека задате дужине	139
6.7. Резултати експеримената	143
7. ОДРЕЂИВАЊЕ МИНИПРЕСЕКА СН СА ВИШЕСТРУКИМ ДОГАЂАЈИМА ПОМОЋУ ИПМ	146
7.1. Одређивање свих минипресека СН	146
7.2. Одређивање минипресека задате дужине	149
7.2.1. Одређивање параметра одсецања у СН	150
7.2.2. Одређивање параметра одсецања у ИПМ	158
7.2.3. Алгоритам за одређивање минипресека задате дужине	162
7.2.4. Алгоритам за одређивање $k_s^{j\max}$	169
7.3. Резултати експеримената	174
8. ЗАКЉУЧАК	178
ЛИТЕРАТУРА	182

1. УВОД

Неисправност и отказ су неизбежни феномени у свим производима и системима. Иако се под поузданошћу подразумева одсуство отказа, она се одређује управо истраживањем отказа и коришћењем података о отказу. Са научног и инжењерског аспекта, истраживање неисправности и начина отказа подразумева испитивање функционалних и физичких граница система, у покушају да се разуме како, зашто и када систем не функционише исправно. Једна од најстаријих метода, која на овај начин приступа одређивању поузданости, је анализа стабла неисправности (АСН).

АСН је заснована на теорији поузданости, Буловој алгебри, логици и теорији вероватноће. Врло једноставна правила и симболи обезбеђују механизам за анализу веома сложених система и комплексних веза унутар њих. Основни концепт АСН је превођење отказног понашања реалног система у графички и логички модел. Графички сегмент је стабло неисправности (СН) – граф који обезбеђује визуелни модел, који веома лако и јасно приказује везе отказа у систему, док логички сегмент омогућује квалитативну и квантитативну анализу.

Централни проблем квалитативне АСН је одређивање минималних скупова пресека. Минимални скупови пресека су минималне комбинације отказа компонентни система, које изазивају отказ целог система. Када се у СН примарни догађаји вишеструко појављују, издвајање минималних скупова пресека из фамилије свих скупова пресека, спада у класу НП тешких проблема. Због тога се највећи број публикација из области АСН бави решавањем овог проблема и то најчешће комбинујући АСН са другим техникама и приступима.

У овом раду се проблем одређивања минималних скупова пресека кохерентног СН решава помоћу Петријевих мрежа (ПМ). У ужем смислу, ПМ чине: структура – бипартитни граф, који моделира структуру система, и маркирање – расподела жетона у чворовима мреже, које моделира стање система. У ширем смислу, под ПМ се подразумева теорија, која омогућава графичко моделирање система и

разноврсне формалне аналитичке методе. Могућност примене аналитичких метода зависи од структуре ПМ и маркирања.

Приступ, који ће бити представљен у овом раду, заснива се на инверзним Петријевим мрежама (ИПМ) и анализи њиховог стабла достижности. Стабло достижности је граф чији корен представља почетно маркирање ПМ, листови – крајње маркирање, а гране – прелазак из једног маркирања у друго. СН се директно трансформише у структуру ИПМ, тако да корен ИПМ представља догађај када отказује систем, а листови ИПМ догађаје када отказују компоненте система. Присуство жетона у чвору ИПМ представља одигравање одговарајућег догађаја. Тиме се проблем одређивања минималних скупова пресека датог СН своди на проблем одређивања листова стабла достижности ИПМ из почетног маркирања, које представља отказ система.

Рад се састоји из осам целина. Након увода, у другом делу ће бити описани: основни концепти АСН, њен историјски развој, значајне примене и методологија спровођења. Највећа пажња ће бити посвећена квалитативној АСН, нарочито одређивању минималних скупова пресека. Приказаће се најпознатије и најстарије методе за налажење минималних скупова пресека, као и најновији приступ заснован на бинарним дијаграмима одлучивања. Након кратког осврта на минималне скупове путева и рангирање минималних скупова пресека, изложиће се основни елементи квантитативне АСН.

Трећи део посвећен је анализи СН великих димензија. Да би се превазишла комбинаторна експлозија, која се јавља код одређивања минималних скупова пресека СН великих димензија, потребно је редуковати проблем. При томе се може редуковати СН, пре одређивања скупова пресека, или се може редуковати добијена фамилија свих скупова пресека, да би се олакшало издвајање минималних. Представници оба приступа ће бити приказани у раду.

Крајем осамдесетих година прошлог века, показано је да је скуп пресека, који не садржи догађаје који се вишеструко појављују у СН, минималан. Овде ће бити представљена нова класа скупова пресека, C^* облика $x_1^{k_1} \cdot x_2^{k_2} \cdot \dots \cdot x_n^{k_n}$, где су

$k_i \geq 1, i = 1, \dots, n$ вишеструкости неких примарних догађаја x_1, x_2, \dots, x_n датог СН. Доказаће се теорема о минималности скупова пресека са вишеструким догађајима, којом се тврди да су скупови пресека, облика C^* , минимални, а минималност скупова пресека без вишеструких догађаја се своди на њен специјалан случај. Поред тога, биће уведени услови потребни за постојање скупа пресека облика C^* и одређен минималан број скупова пресека који се редукују као његов надскуп. Нови алгоритам редукције и допринос теореме о минималности скупова пресека са вишеструким догађајима, у смањивању броја операција редукције скупова пресека, биће илустрован на групи тест примера (*Benchmark Fault Trees*).

У четвртом делу ће бити описани основни појмови теорије ПМ: општа дефиниција ПМ, моделирање динамике система, класификација и проширења ПМ, и дефиниције оних врста ПМ које ће се користити касније у раду. Када се систем моделира помоћу ПМ, његова анализа се врши испитивањем својстава ПН. Овде ће бити наведена основна својства ПМ: достижност, живост, мртво маркирање и повратност, и методе анализе: редукција, матрица инциденције, инваријанте, граф достижности и симулација ПМ.

Пети део почиње основама моделирања СН помоћу ПМ, које су постављене осамдесетих година прошлог века и које нису промењене до данас. Након тога, биће описани различити приступи у одређивању минималних скупова пресека датог СН помоћу ПМ: графичка метода; матрична метода; подела ПМ на нивое; одређивање минималних скупова пресека помоћу t-инваријанте и на основу матрице инциденције ПМ

Основна идеја методе за одређивање минималних скупова пресека помоћу ИПМ биће описана у шестом делу. Идеја ће прво бити објашњена на кохерентном СН без вишеструких догађаја. На основу једначине стања, доказаће се својства ИПМ, која омогућавају да се проблем одређивања минималних скупова пресека сведе на проблем одређивања листова графа достижности ИПМ. Приказаће се алгоритам за одређивање минималних скупова пресека, који је имплементиран и тестиран на групи СН. На крају ће бити представљен приступ за добијање минималних

скупова пресека задатог реда (дужине), који се може користити онда када није могуће одредити све скупове пресека. Приступ је заснован на особини да укупан број жетона у ИПМ моделира број одиграних догађаја и да је тај број неоппадајући. Одређивање минималних скупова пресека састоји се у одсецању чворова графа достижности ИПМ, у којима је укупан број жетона већи од задатог реда. Резултати одсецања стабла достижности ће такође бити приказани.

У седмом делу, метода ће бити проширена на СН са вишеструким догађајима. Прво ће се, на основу резултата из трећег дела рада, алгоритам за одређивање свих минималних скупова пресека проширити тако да се одмах идентификују скупови пресека облика C^* . Након тога ће бити решаван проблем одређивања минималних скупова пресека датог реда за СН са вишеструким догађајима. У овим СН укупан број жетона у ИПМ није неоппадајући, што отвара проблем одређивања параметра одсецања графа достижности. Доказаће се да је параметар одсецања различит у сваком чвору стабла достижности и одредиће се горња граница вредности овог параметра која гарантује да се неће изоставити ниједан од тражених минималних скупова пресека. Ова граница је, такође, одређена на основу резултата из трећег дела рада. На крају ће бити приказан алгоритам за одређивање минималних скупова пресека СН са вишеструким догађајима, који је имплементиран и тестиран на групи тест примера (*Benchmark Fault Trees*).

У осмом делу ће бити изнета закључна разматрања и правци даљег истраживања.

2. АНАЛИЗА СТАБЛА НЕИСПРАВНОСТИ (АСН)

У управљању поузданошћу, ризиком и сигурношћу система до сада је, у различитим фазама, коришћена 701 метода и техника, специјализована или општа. Најинтензивнији развој одвијао се у периоду од 1991. до 2000. године (232 методе) а од 2000. до 2008. је развијено још 95 нових метода [30]. Оваквом развоју допринео је развој рачунара, нарочито у области прикупљања и анализе великог броја података.

Најзахтевнија фаза управљања поузданошћу и ризиком је фаза процене и анализе. Најпознатије методе које се могу користити у овој фази су [28]: прелиминарна анализа хазарда (*Preliminary Hazard Analysis, PHA*); анализа стабла догађаја (*Event Tree analysis, ETA*); анализа стабла неисправности (*Fault Tree Analysis, FTA*); анализа начина и ефеката отказа (*Failure Mode and Effect Analysis, FMEA*); модели Маркова (*Markov Analysis*); бинарни дијаграми одлучивања (*Binary Decision Diagram, BDD*); Петријеве мреже (*Petri Nets, PN*) итд.

Анализа стабла неисправности (АСН)* је, поред *FMEA*, најважнија и највише коришћена метода за анализу поузданости и ризика [123,39]. АСН је дедуктивна метода у којој се прво дефинише нежељени догађај, тзв. вршни догађај (*Top event*), а затим се анализира утицај понашања појединих компоненти система на његово појављивање. Анализа се ослања на дијаграм - стабло неисправности (СН) (*Fault tree*), који симболички описује логичке релације између догађаја.

У овој глави ће бити описан развој и значајне примене АСН, методологија њеног спровођења и преглед значајних метода за налажење минималних скупова пресека као најзахтевнијег корака у спровођењу АСН. Након кратког описа приступа у налажењу минималних скупова путева и метода за рангирање минималних скупова пресека, изложиће се основни елементи квантитативне АСН.

* Анализа стабла неисправности (АСН) је назив за *FTA* на српском језику по стандарду **SRPS EN 61025:2011** [103], мада се у домаћој литератури могу наћи и називи анализа стабла отказа [133] или анализа стабла грешака [137].

2.1. Историјски развој АСН

Аналізу стабла неисправности је развио *H.A. Watson* 1961. године у *Bell Telephone Laboratories* за процену безбедности у оквиру студије *Minuteman Launch Control System*, дела пројекта Поларис, за потребе војске САД. Инжењери у Беловим лабораторијама су уочили да метода која је коришћена да опише “исправну” логику у процесирању података, може бити коришћена за анализу “неисправне” логику која је последица отказа компоненти [38]. Касније је Дејв Хасл (*Dave Haasl*) у Компанији Боинг (*Boeing Company*) препознао вредности ове технике, први је назвао *Fault Tree Analysis* и предводио тим који је применио АСН у свим деловима *Minuteman Missile System* пројекта.

Након позитивних резултата примене, започело је коришћење АСН и у дизајну комерцијалних авиона у Компанији Боинг. 1965. године Компанија Боинг и Универзитет у Вашингтону организовали су прву конференцију о сигурности система (*System Safety Conference*) која представља почетак широког интересовања за АСН.

Следећи авионску индустрију, седамдесетих година двадесетог века почела је примена у дизајну и развоју нуклеарних постројења. Велики број алгоритама и софтвера развили су запослени у нуклеарној индустрији, за коју се сматра да је дала највећи индивидуални допринос развоју АСН [27]. Године 1975. Нуклеарна регулаторна комисија САД објавила је *WASH-1400* - Студију о безбедности нуклеарних реактора у којој је у додатку на 579 страна, детаљно описана примена АСН у анализи поузданости различитих подсистема нуклеарних електрана [109].

Осамдесетих година почиње интернационална примена АСН и примена у различитим областима: хемијска, аутомобилска, војна индустрија, свемирски програм итд. Први рад о АСН на српском језику објављен је 1983. године [122]. Од деведесетих година до данас, АСН добија велики број проширења и хибридни облици: Фази АСН [21,106,107,12,127]; Динамичка АСН [26,66]; *Posbist FTA* [42]; *Accident FT (AFT diagrams)* [48]; *Condition-based fault tree analysis (CBFTA)* [98]; *Software FTA (SFTA)* [74]; *Bouncing failure analysis (BFA)* која обједињује АСН и *FMEA* методологију [8] итд.

Могућност примене одређене методе значајно утиче на интересовање које за њу постоји и на њен развој. С обзиром да је АСН метода за анализу поузданости, сигурности и ризика, неколико несрећних догађаја је допринело да се она установи и потврди [27,50]:

– 27. јануара 1967. године - пожар на лансирној рампи Аполо 1. Након тога, НАСА и Компанија Боинг су увели нови сигурносни програм за цео Аполо пројекат који је обухватао и извођење АСН на читавом Аполо систему.

– 28. марта 1979. године – несрећа у нуклеарној електрани на острву Три Миље. Неколико студија испитивања несреће је извршено коришћењем АСН.

– 28. јануара 1986 године – Спејс шатл Челиндер је експлодирао 73 секунде након полетања. Независни тим за истраживање несреће је користио АСН за анализу главних мотора, како би се обезбедила адекватна сигурност већ у фази дизајна шатла.

– Фебруара 2003. године – експлозија спејс шатла Колумбија. Након овог догађаја, тежиште свемирске индустрије је поново стављено на безбедност и поузданост.

Сви описани догађаји односе са на системе који морају бити високопоуздани, било због окружења у коме функционишу, било због последица које би њихов квар могао да изазове. АСН је коришћена и приликом дизајна нуклеарног система Фукушима, који је доживео отказ са катастрофалним последицама у марту 2011. године. У извештају Јапанске владе на *IAEA Ministerial Conference on Nuclear Safety* у јуну 2011. године [91], наводи се да је АСН успешно спроведена у анализи неисправности система пумпи за морску воду, али да није разматрана могућност неисправности услед цунамија. Ово потврђује чињеницу да успешност сваке методе за анализу поузданости и ризика, па и АСН, зависи од познавања система на који се примењује и способности аналитичара да предвиди све релевантне факторе.

Данас, домен АСН постаје готово универзалан, са значајним резултатима у роботизици и поузданости софтвера и са све већим применама у анализи поузданости човека.

2.2. Методологија АСН

АСН идентификује, моделира и процењује везе између догађаја које воде до: отказа, нежељених догађаја или непланираних догађаја, односно стања система. Методологија АСН је дефинисана, структурирана и стриктна; лака је за савладавање, извођење и праћење. Међутим, АСН може бити скупа и временски захтевна, због чега треба увек упоредити трошкове спровођења анализе са трошковима које изазивју нежељени догађаји.

Најчешће фазе у спровођењу АСН су [86]:

- Дефинисање система, нежељеног догађаја (отказа) система и услова отказа.
- Конструкција стабла неисправности.
- Квалитативна анализа стабла неисправности.
- Квантитативна анализа стабла неисправности.

У наставку ће бити описана свака од фаза, а посебан нагласак ће бити на квалитативној АСН чији је корак налажења минималних скупова пресека главни предмет истраживања овог рада.

2.3. Дефинисање система и нежељеног догађаја

На почетку примене АСН мора се добро упознати систем који се анализира са аспекта поузданости и ризика. Због тога је често почетна тачка постојећа FMEA и блок дијаграм који обезбеђују његово боље разумевање. Након тога, потребно је одредити границе у оквиру којих се врши анализа:

- Физичке границе система: који делови система су укључени у анализу, а који се могу занемарити;
- Почетни услови: у ком се стању систем налази када се деси посматрани нежељени догађај;

- Спољашњи утицаји: који типови спољашњих утицаја треба да буду укључени у анализу (ратни услови, саботажа, земљотрес, нестанак електричне енергије итд.);
- Ниво детаљности: колико детаљна треба да буде анализа.

Као што је већ поменуто, АСН је дедуктивна анализа која се фокусира на одређени нежељени догађај система и утврђује његове узроке. Под догађајем се подразумева динамичка промена стања која се дешава у систему или елементу система. Са аспекта АСН, значајна су три нивоа догађаја:

- Вршни догађај – нежељени догађај система.
- Посредни догађај – догађај који се дешава као последица једног или више догађаја. То је догађај неисправности који се дешава зато што се догодио један или више претходних догађаја.
- Примарни догађај – догађај који се даље не разлаже и представља границу декомпозиције система.

Избор вршног догађаја је важан за успех анализе: уколико је сувише општи анализа постаје преобимна и тешко сагледива; ако је сувише специфичан анализа не обезбеђује довољно добар поглед на систем. Дефиниција вршног догађаја треба да да одговоре на питања: шта се десило, где и када [86]. Након тога, врши се декомпозиција система на примарне догађаје и приступом одозго наниже, утврђује се како одређени нежељени догађаји, везани за елементе система (подсистеми, делови, компоненте), утичу на вршни догађај.

Приликом одређивања догађаја могу се идентификовати две појаве [22]:

- Отказ (*failure*) – квар система или његових елемената изражена кроз одступање од планиране функције или понашања или кроз немогућност система, подсистема или компоненте да обаве потребну функцију. Степен отказа може бити различит: не ради у потребном тренутку, ради повремено, не престаје са радом у потребном тренутку, недостатак потребног излаза, промењен излаз или смањен оперативни капацитет.

– Неисправност (*fault*) – нежељено стање система или његових делова, узроковано присуством неодговарајућих команди или одсуством одговарајућих, или отказом. Сваки отказ изазива неисправност, али није свака неисправност изазвана отказом (систем који је искључен из безбедносних разлога није отказао). Неисправност је општи термин које се може рангирати од минорног дефекта до отказа.

Дефиниција 2.1. [58] Уколико су два догађаја комплементарна, називају се биформним (*biform*) догађајима у супротном су моноформни (*monoform*).

Упоредо са догађајима (неисправностима) у систему, утврђују се и логичке везе између њих и последице њиховог појављивања.






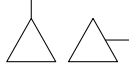
2.4. Конструкција стабла неисправности.

Стабло неисправности (СН) је модел који графички и логички представља комбинације примарних и посредних догађаја, које могу довести до вршног догађаја, односно до неисправности система. Конструкција СН се врши одозго наниже, од вршног догађаја до његових непосредних узрока и тако редом до примарних догађаја. У конструкцији стабла се полази од претпоставке да су догађаји бинарни, статистички независни и да везе између њих могу да се представе логичким операцијама (колима).







СН садржи три врсте симбола: симболе догађаја, симболе трансфера (табела 2.1) и симболе логичких кола (табела 2.2). Трансфери су чворови СН који омогућавају одовјени приказ делова СН. С обзиром да се у овом раду не разматрају, у табели је приказан само њихов општи изглед.

Поред наведених стандардних симбола, постоје различите врсте специјалних симбола, као што су симболи за: негације, кашњење, задату комбинацију улаза у логичко коло, догађај који мења статус у зависности од фазе анализе, функционалне зависности, итд.

Табела 2.1. Символи догађаја СН

Симбол	Значење симбола	
	Примарни догађај – примарна иницирајућа неисправност која не захтева даље развијање	Примарни догађаји
	Неразвијени догађај - догађај који није даље развијен или зато што то не би имало нарочит значај или зато што не постоји расположива информација.	
	Специфични услови или рестрикције које се могу применити на било које логичко коли (првенствено Приоритетно И и инхибиторско коло).	
	Спољашњи догађај (тригер) - догађај за који се очекује да ће се десити због природе пројекта и услова рада.	
	Посредни догађај	
	Символи за трансфер	

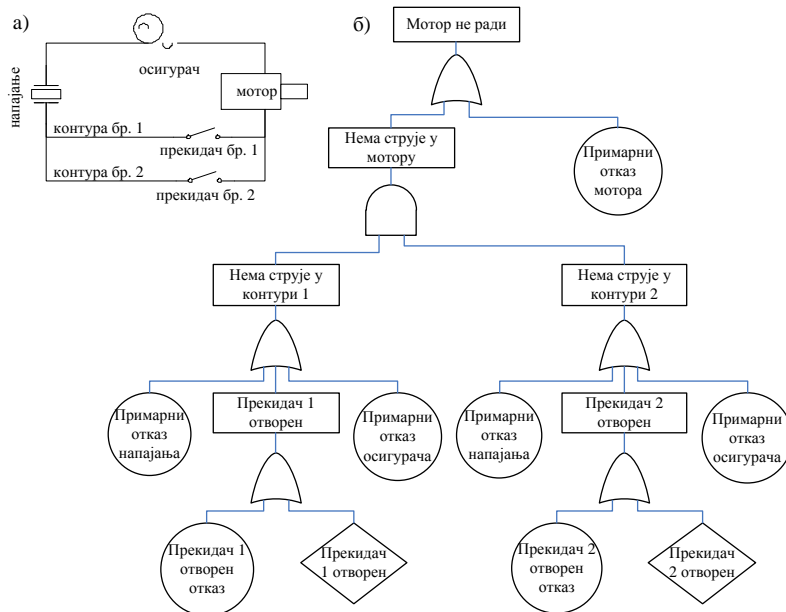
Табела 2.2. Символи логичких кола СН

Назив кола	Симбол кола	Значење симбола
И		Неисправност на излазу ће се десити ако се десе све улазне неисправности.
ИЛИ		Неисправност на излазу ће се десити ако се деси бар једна улазна неисправност.
Приоритетно И		Неисправност на излазу ће се десити ако се све улазне неисправности десе у специфицираној секвенци.
Ексклузивно ИЛИ		Неисправност на излазу ће се десити ако се деси тачно једна улазна неисправност.
m од n		Неисправност на излазу ће се десити ако се деси m од n улазних неисправности.
Инхибиторско коло		Неисправност на излазу ће се десити само ако се деси улазна неисправност и условни догађај.

Када је извршена декомпозиција нежељеног понашања система и када су утврђене логичке везе између догађаја у систему, догађаји и логичка кола се приказују графички, одозго наниже.

Пример

На слици 2.1. а) је приказано електрично коло у коме је потребно анализирати вршни догађај “мотор не ради” [122]. Овај догађај ће се десити ако откаже мотор или ако нема струје у мотору. Отказ мотора је догађај који се у овом случају даље не анализира а струје неће бити у мотору ако је истовремено не буде ни у једној од контура. Даљом декомпозицијом, добија се структура која је приказана одговарајућим стаблом неисправности на слици 2.1. б).



Слика 2.1. Стабло неисправности [122]

Формално, СН је укорењено бипартитно стабло* (U, V, A) , где је U скуп чворова догађаја, V скуп чворова логичких кола а A скуп грана. Сваки пар чворова (u, v) , $u \in U$, $v \in V$ може бити повезан највише једном граном.

Скуп U чине примарни, посредни и вршни догађаји. Вршни догађај је чвор на најнижем нивоу, односно корен СН, примарни догађаји су листови СН а посредни догађаји су сви остали чворови СН.

*С обзиром да је стабло, по дефиницији, бипартитни граф, термин бипартитно се овде користи у инжењерском смислу, да би се истакло постојање два дисконктна скупа чворова: скуп чворова који моделирају догађаје и скуп чворова који моделирају логичка кола.

С обзиром да је СН неоријентисано, овде ће се условно користити термини “улазне” и “излазне” гране. Под излазном граном чвора скупа U , подразумеваће се грана од чвора који моделира догађај до чвора који моделира логичко коло чији је посматрани догађај улаз. Под улазном граном чвора скупа U ће се подразумевати грана ка чвору који моделира догађај од чвора који моделира логичко коло чији је посматрани догађај излаз. Односно, посматраће се као да су гране оријентисане од чворова на вишем ка чворовима на нижем нивоу стабла. Аналогно је за улазне и излазне гране чворова који моделирају логичка кола. Уколико се другачије не нагласи, подразумеваће се да су оријентисане од чворова на вишем ка чворовима на нижем нивоу СН.

У складу са претходним образложењем, може се рећи да су примарни догађаји чворови без улазних грана, посредни догађаји су чворови са бар једном улазном и једном излазном граном, а вршни догађај је чвор без излазних грана.

Логичка кола се, поред скупа V , могу описати и индиректно преко елемената скупа догађаја U . Нека је ${}^{\#}D_u$ број улазних грана чвора u . Сваком чвору u који одговара вршном и посредном догађају, придружен је позитиван цели број b_u који представља логички индикатор. Ако је $b_u = 1$, онда чвору u одговара логичко коло И; ако је $b_u = {}^{\#}D_u$, онда чвору u одговара логичко коло ИЛИ [25].

Уколико СН садржи само И и ИЛИ логичка кола и моноформе примарне догађаје (примарне догађаје без њихових негација), назива се кохерентно СН. Кохерентним СН се моделирају кохерентни системи. То су системи у којима не постоје компоненте чије стање не утиче на стање система и у којима је структурна функција неоппадајућа [125]. У овом раду ће бити разматрана кохерентна СН. СН са вишеструким догађајима је оно у којем се поједини догађаји вишеструко појављују (*Multiple Occurring Event – MOE*). [28].

Пример

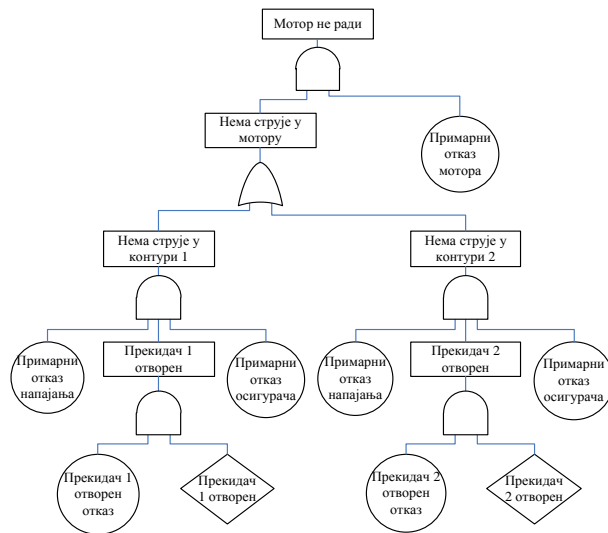
У СН са слике 2.1. догађаји “Примарни отказ напајања” и “Примарни отказ осигурача” су вишеструки и оба догађаја су вишеструкости 2.

Дефиниција 2.2. [25] Кохерентном СН (прималном СН – ПСН) се може придружити дуално СН (ДСН), које се разликује од ПСН само у вредности логичких индикатора. Ако је b_u логички индикатор чвора u у ПСН, онда је $\#D_u - b_u + 1$ индикатор чвора u у ДСН.

Дакле, дуално СН има исти скуп догађаја U као почетно СН, а у скупу V је свако логичко коло И замењено логичким колом ИЛИ и обрнуто [112].

Пример

На слици 2.2. приказано је ДСН за СН са Сликe 2.1.



Слика 2.2. Дуално СН стабла неисправности са слике 2.1.

2.5. Квалитативна АСН

Када је конструисано СН, оно може да се анализира са алгебарског аспекта. Ова анализа се још назива и квалитативна или логичка анализа и представља одређивање структурне функције СН у односу на вршни догађај.

Нека су $P_i, i=1, \dots, I$ примарни догађаји посматраног система и нека је сваком од њих придружена бинарна променљива x_i која представља индикатор i -тог примарног догађаја, односно [58]:

$$x_i = \begin{cases} 1, & \text{ако се деси } i\text{-ти догађај (отказ)} \\ 0, & \text{у супротном} \end{cases} \quad (2.1)$$

Структурна функција СН у односу на вршни догађај је:

$$\varphi(x) = \varphi(x_1, \dots, x_l) = \begin{cases} 1, & \text{ако се деси вршни догађај} \\ 0, & \text{у супротном} \end{cases}$$

За примарне догађаје се каже да су релевантни ако се њихова променљива јавља у минималној форми $\varphi(x)$, односно нерелевантни или небитни у супротном.

У ширем смислу, квалитативна анализа СН обухвата одређивање минималних скупова пресека и путева и различите начине њиховог рангирања и анализа.

2.5.1. Одређивање минималних скупова пресека

Скуп пресека (*Cut set*) је скуп примарних догађаја чије истовремено одигравање доводи до вршног догађаја. Минимални скуп пресека (*Minimum cut sets*), у наставку минипресек, се може дефинисати на различите начине:

Дефиниција 2.3: Минимални скуп пресека је скуп пресека који се не може редуковати без губљења статуса скупа пресека. Уколико се било који од догађаја, из минималног скупа пресека, не догоди, тада се неће десити вршни догађај.

Дефиниција 2.4: Минимални скуп пресека је скуп пресека који не садржи неки други скуп пресека.

Структурна функција СН се може добити директно из СН комбиновањем примарних догађаја и логичких кола помоћу Булових оператора. Налажење минипресека се тада своди на одређивање простих импликаната (*prime implicants*) Булове функције [24]. Проблем одређивања простих импликаната је у директној вези са САТ (*Satisfiability problem* –проблем задовољења), првим проблемом за који је доказано да је НП-комплетан. Једна од формулација САТ проблема је [72]:

САТ:

Улаз: Булова формула φ од n променљивих.

Питање: Да ли φ има прости импликант?

Дакле, проблем одређивања минипресека је NP-тежак проблем. Међутим, с обзиром да је сваки минипресек скуп независних примарних догађаја, вероватноћа њиховог истовременог појављивања једнака је производу њихових вероватноћа. Како примарни догађаји представљају отказе компоненти система, њихове вероватноће су релативно мале, односно веома мале код високопоузданих система. Због тога вероватноћа минипресека драстично опада са повећањем броја његових примарних догађаја па је довољно добро решење одредити само минипресеке са задатим бројем примарних.

Структурна функција СН се може изразити у коњунктивној нормалној форми (КНФ) и дисконјунктивној нормалној форми (ДНФ) [7]. Када је изражена у ДНФ, свака коњункција ДНФ структурне функције СН је један скуп пресека [75]. Уколико у СН не постоје вишеструки догађаји, сви скупови пресека су минимални. Тек у случају СН са вишеструким догађајима се појављује НП-тежак проблем одређивања минипресека, односно елиминисања скупова пресека који нису минимални.

За решавање овог проблема постоје бројне методе и приступи. Седамдесетих година је развијен велики број тзв. старих метода, које се по приступу решавања могу поделити на: методе одозго наниже (*Top-down* методе) и одоздо навише (*Bottom-up* методе). Приступом одозго наниже се одређују скупови пресека, а затим се елиминишу они који нису минимални. У приступу одоздо навише се покушава да избегне потенцијално велики број неминималних скупова пресека тако што се, идући од примарних ка вршном догађају, за сваки посредни догађај елиминишу надскупови [18].

Осамдесетих година почео је да се развија приступ решавања помоћу Петријевих мрежа, о чему ће више бити писано у осталим деловима овог рада. Почетком деведесетих година појавио се нови приступ у АСН, који се базира на бинарним дијаграмима одлучивања. С обзиром на број објављених радова, нарочито у последњих десет година, овај приступ се може сматрати доминантним у АСН.

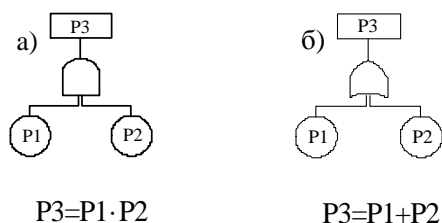
2.5.1.1. Методе одозго наниже

Овим методама се одређивање минипресека врши од чвора на најнижем нивоу, корена СН, ка чворовима на највишем нивоу, листовима СН. Најстарије и најчешће коришћене методе ове групе су: директна метода заснована на Буловој алгебри, *MOCUS* и *FATRAM* алгоритми.

Директна метода

Директна метода се може применити и за одређивање минипресека некохерентних СН, а ослања се на аксиоме и законе Булове алгебре [2]. Резултат методе је ДНФ структурне функције $\varphi(x)$ датог СН у којој, након редукције, сваки сабирак представља један минимални скуп пресека. У наставку ће бити описана сама метода најчешћом нотацијом у литератури, што значи да Булове променљиве неће бити означене као у (2.1).

Дакле, сваком логичком колу се може придружити једна Булова једначина. С обзиром да се у овом раду разматрају кохерентна СН, разматрају се само Булове једначине за И и ИЛИ коло (слика 2.3. а и б).



Слика 2.3. Логичка кола и одговарајуће логичке операције

У општем случају, када логичко коло И има n улазних грана из чворова P_1, P_2, \dots, P_n , посредни догађај P , који је излаз из тог логичког кола се може изразити као:

$$P = \prod_{i=1}^n P_i. \quad (2.2)$$

Ако логичко коло ИЛИ има n улазних грана из чворова P_1, P_2, \dots, P_n , посредни догађај P , који је излаз из тог логичког кола, се може изразити као:

$$P = \sum_{i=1}^n P_i . \quad (2.3)$$

Када се директна метода користи за одређивање минискупова кохерентног СН, довољне су следеће Булове аксиоме и закони:

- а) Комутативност: $P \cdot Q = Q \cdot P$
 $P + Q = Q + P$
- б) Асоцијативност: $P \cdot (Q \cdot R) = (P \cdot Q) \cdot R$
 $P + (Q + R) = (P + Q) + R$
- в) Дистрибутивност: $P \cdot (Q + R) = (P \cdot Q) + (P \cdot R)$
 $P + (Q \cdot R) = (P + Q) \cdot (P + R)$
- г) Закон идемпотенције: $P \cdot P = P$
 $P + P = P$
- д) Закон апсорпције: $P + (P \cdot Q) = P$
 $P \cdot (P + Q) = P$

Уводе се Булове променљиве:

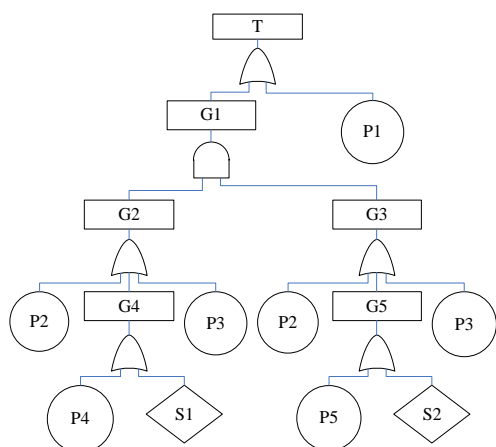
- Т - вршни догађај (*top event*),
- G_j- посредни догађаји (*gates*),
- P_n и S_m - примарни и неразвијени догађаји (*primary* и *secondary events*).

Алгоритам директне методе:

1. Написати Булове једначине за свако логичко коло. Сваком колу одговара једна једначина типа (2.2) или (2.3).
2. Вршити замену променљивих њиховим изразима док се не добије вршни догађај као функција само примарних догађаја. Притом се користе комутативни, асоцијативни и дистрибутивни закони Булове алгебре.
3. Извршити редукцију применом закона Булове алгебре. За кохерентна СН довољно је користити закон идемпотенције и закон апсорпције.

Пример

У наставку ће бити илустрована директна метода. На слици 2.4. је приказано СН са слике 2.1. са догађајима означеним Буловим променљивама.



Слика 2.4. СН са слике 2.1. означено Буловим променљивама

$$1. \quad T = P1 + G1$$

$$G1 = G2 \cdot G3$$

$$G2 = P2 + G4 + P3$$

$$G3 = P2 + G5 + P3$$

$$G4 = P4 + S1$$

$$G5 = P5 + S2$$

$$2. \quad T = P1 + G1 \Rightarrow \quad T = P1 + G2 \cdot G3$$

$$T = P1 + (P2 + G4 + P3) \cdot (P2 + G5 + P3) = P1 + P2 \cdot P2 + G4 \cdot P2 + P3 \cdot P2 + P2 \cdot G5 + G4 \cdot G5 + P3 \cdot G5 + P2 \cdot P3 + G4 \cdot P3 + P3 \cdot P3$$

$$T = P1 + P2 \cdot P2 + (P4 + S1) \cdot P2 + P3 \cdot P2 + P2 \cdot (P5 + S2) + (P4 + S1) \cdot (P5 + S2) + P3 \cdot (P5 + S2) + P2 \cdot P3 + (P4 + S1) \cdot P3 + P3 \cdot P3$$

$$T = P1 + P2 \cdot P2 + P2 \cdot P4 + P2 \cdot S1 + P2 \cdot P3 + P5 \cdot P2 + P5 \cdot P4 + P5 \cdot S1 + P5 \cdot P3 + S2 \cdot P2 + S2 \cdot P4 + S2 \cdot S1 + S2 \cdot P3 + P3 \cdot P2 + P3 \cdot P4 + P3 \cdot S1 + P3 \cdot P3$$

$$3. \quad T = P1 + P2 + P3 + P5 \cdot P4 + P5 \cdot S1 + S2 \cdot P4 + S2 \cdot S1$$

Сваки сабирак добијене минималне форме представља минипресек. Минипресеци су: $\{P1\}$, $\{P2\}$, $\{P3\}$, $\{P5, P4\}$, $\{P5, S1\}$, $\{S2, P4\}$, $\{S2, S1\}$.

Код кохерентних СН са вишеструким догађајима, резултат корака 2 су сви скупови пресека, а корака 3 сви минипресеци датог СН. Уколико у СН нема вишеструких догађаја, минипресеци се добијају већ након корака 2.

***MOCUS* алгоритам**

MOCUS (*Method of Obtaining CUt Sets*) је алгоритам за генерисање минипресека кохерентног СН. Први пут је објављен у [31]. Базиран је на чињеници да свако логичко коло типа ИЛИ повећава број минипресека док логичко коло типа И повећава величину (дужину) минипресека.

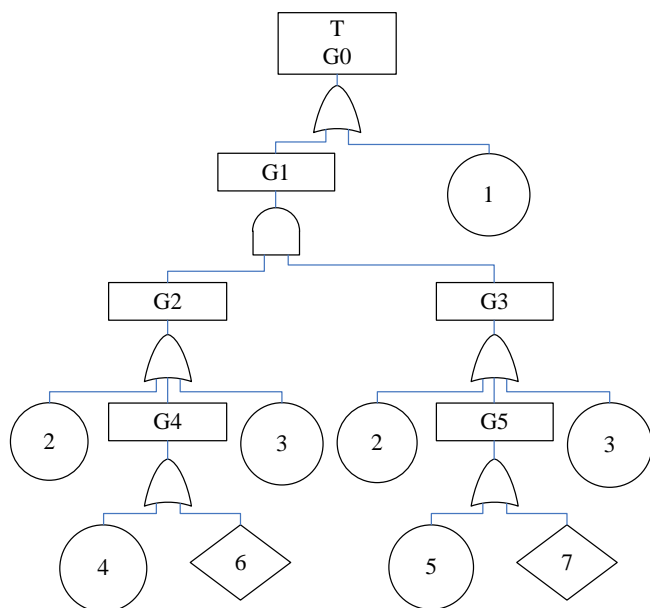
MOCUS алгоритам [49]:

1. Именовати сва логичка кола (посредне догађаје).
2. Нумерисати све примарне догађаје.
3. Уписати највиши чвор стабла неисправности у прву колону и ред матрице.
4. Понављати итеративно сваку од основних пермутација (а) или (б) одозго наниже:
 - 4.1. Заменити ИЛИ коло вертикално поређаним улазима тог кола и повећати број минипресека. Односно, када је елемент матрице ИЛИ коло, заменити га колоном чији су елементи улази тог кола.
 - 4.2. Заменити И коло хоризонтално поређаним улазима тог кола и повећати величину минипресека. Односно, када је елемент матрице И коло, заменити га редом чији су елементи улази тог кола.
5. Када су сва кола замењена примарним догађајима, уклонити пресеке који нису минимални, користећи правила Булове алгебре.
6. Коначна листа садржи минималне скупове пресека.

У литератури се може наћи и одоздо навише верзија *MOCUS* алгоритма у којој се корак 4 обавља од најнижих посредних догађаја ка вршном [28].

Пример

MOCUS алгоритам ће бити илустрован на примеру СН са слике 2.1.. За примену *MOCUS* алгоритма уобичајено је обележавање примарних догађаја целим бројевима, а посредних са G_i (слика 2.5). У табели 2.1. приказани су кораци у одређивању минипресека *MOCUS* алгоритмом.



Слика 2.5. СН са слике 2.1. нумерисано за примену *MOCUS* алгоритма

Табела 2.1. *MOCUS* алгоритам за СН са Слике 2.5.

Кораци	1 (к3)	2 (к4)	3 (к4)	4 (к4)	5 (к4)	6 (к4)	7 (к4)	8 (к5)
	G0	1	1	1	1	1	1	1
		G1	G2, G3	2, G3	2, 2	2, 2	2, 2	2
			G4, G3	2, G5	2, 5	2, 5	2, 5	4, 5
			3, G3	2, 3	2, 7	2, 7	2, 7	4, 7
				4, G3	2, 3	2, 3	2, 3	6, 5
				6, G3	4, 2	4, 2	4, 2	6, 7
				3, 2	4, G5	4, 5	4, 5	3
				3, G5	4, 3	4, 7	4, 7	
				3, 3	6, 2	4, 3	4, 3	
					6, G5	6, 2	6, 2	
					6, 3	6, 5	6, 5	
					3, 2	6, 7	6, 7	
					3, 5	6, 3	6, 3	
					3, 7	3, 2	3, 2	
					3, 3	3, 5	3, 5	
						3, 7	3, 7	
						3, 3	3, 3	

У последњем, 7. кораку, добијени су исти минипресеци стабла неисправности са слике 2.1, као и применом директне методе.

***FATRAM* алгоритам**

FATRAM (*FAult Tree Reduction AlgorithM*) алгоритам, је по приступу сличан алгоритму *MOCUS*. У методи се, приликом коришћења термина “улаз” и “излаз” подразумева да су гране СН оријентисане одоздо навише, од чворова на највишем ка чворовима на нижим нивоима. Кораци алгоритма су [85]:

1. Ако је вршни догађај излазни чвор И кола, сви улазни догађаји се додељују истом скупу, а ако је вршни догађај излазни чвор ИЛИ кола, сваки улазни догађај се додељује посебном скупу.
2. Понављати поступак док сва логичка кола не буду разложена. ИЛИ кола, са само примарним догађајима на улазу, се у овом кораку не разлажу.
3. Уклонити све редувантне елементе и надскупове.
4. За сваки вишеструки догађај применити следеће кораке:
 - а. Вишеструки догађај замењује сва неразложена логичка кола чији је улаз, чиме се формирају нови скупови.
 - б. Новоформиране скупове придружити постојећим.
 - в. Уклонити посматрани догађај из одговарајућих логичких кола.
 - г. Уклонити редувантне елементе и надскупове.
5. Разложити преостала ИЛИ кола. Сви добијени скупови су минипресеци.

Кључни кораци алгоритма, чија се идеја разликује од *MOCUS* алгоритма, доказани су следећим тврђењима.

Тврђење 2.1 [85]: Корак 4 *FATRAM* алгоритма одржава логички модел СН, односно чува све минипресеке датог СН.

Доказ: Нека је G изазни догађај ИЛИ логичког кола, чији су улази примарни догађај B и остали догађаји који су заједно означени са γ . Нека је S скуп који садржи G и друге догађаје који су излази ИЛИ кола и/или примарни догађаји и који су заједно означени са T . Односно, $G = B \cup \gamma$ и $S = \{G, T\}$. Тврђење следи из чињенице да је $S = [(G - B) \cap T] \cup [B \cap T]$. ♦

Тврђење 2.2 [85]: Сви резултујући скупови у кораку 5 FATRAM алгоритма су минипресеци.

Доказ: Након корака 2, као неразвијена кола остала су ИЛИ кола, чији су улази само примарни догађаји. По алгоритму, у кораку 4 се из неразвијених логичких кола уклањају сви вишеструки примарни догађаји, који су улазни догађаји тих кола и који се јављају више пута у тим колима или међу скуповима пресека који су већ идентификовани. Затим се у корацима 3. и 4.г. уклањају редувантни елементи и надскупови, чиме се задржавају само минипресеци. Последњи, 5. корак алгоритма, представља разлагање неразложених ИЛИ кола, која у том тренутку садрже само јединствене примарне догађаје, односно примарне догађаје који нису вишеструки. Тиме се већ постигнута минималност скупова пресека не нарушава јер ће сваки од новодобијених скупова пресека бити минималан. ♦

Пример

FATRAM алгоритам ће бити илустрован на истом примеру СН са слике 2.1.

Да би алгоритам, посебно корак 4, био детаљније илустрован, претпоставиће се да су P4 и P5 понављања истог вишеструког догађаја. Тиме се минипресек P4 P5 своди на минипресек P4, а минипресеци P4 S2 и P5 S1 се редукују, односно СН са слике 2.1. има следеће минипресеке: P1, P2, P3, P4 и S1S2.

1. Вршни догађај T одговара ИЛИ колу $\Rightarrow \{G1\}, \{P1\}$;

2. Догађај G1 одговара И колу $\Rightarrow \{G2,G3\}, \{P1\}$;

Догађај G2 одговара ИЛИ колу $\Rightarrow \{P2,G3\}, \{G4,G3\}, \{P3,G3\}, \{P1\}$;

Догађај G3 одговара ИЛИ колу $\Rightarrow \{P2,P2\}, \{P2,G5\}, \{P2,P3\}, \{G4,P2\}, \{G4,G5\}, \{G4,P3\}, \{P3,G2\}, \{P3,G5\}, \{P3,P3\}, \{P1\}$;

Догађаји G4 и G5 одговарају ИЛИ колу, чији су улаз само примарни догађаји, тако да се у овом кораку они не разлажу.

3. Уклањањем редувантних елемената добијају се скупови: $\{P2\}, \{P2,G5\}, \{P2,P3\}, \{G4,P2\}, \{G4,G5\}, \{G4,P3\}, \{P3,G2\}, \{P3,G5\}, \{P3\}, \{P1\}$;

Уклањањем надскупова добијају се скупови: $\{P2\}, \{G4,G5\}, \{P3\}, \{P1\}$;

4. У СН постоје три вишеструка догађаја: P2, P3 и P4. Догађаји P2 и P3 нису улазни ни за једно од неразложених логичких кола. Кораци a-d се примењују само за догађај P4:
 - a. Заменом догађаја P4 у неразложеним логичким колима добија се скуп {P4, P4};
 - б. Важећи скупови сада су: {P2}, {G4,G5}, {P3}, {P1}, {P4, P4};
 - в. Догађај P4 се уклања као улаз из кола G4 и G5, тако да је сада G4 ИЛИ коло, чији је једини улазни догађај S1, а G5 ИЛИ коло чији је једини улазни догађај S2.
 - г. Уклањањем редувантних елемената, важећи скупови остају: {P2}, {G4,G5}, {P3}, {P1}, {P4};
 5. Разлагањем кола G4, уместо скупа {G4,G5} добија се скуп {S1,G5}, а даљим разлагањем кола G5, добија се скуп {S1, S2}. Коначни скупови, односно минипресеци, сада су: {P2}, {S1, S2}, {P3}, {P1} и {P4}.
-

2.5.1.2. Методе одоздо навише

Основна идеја ових метода је да се што већи део редукције неминималних скупова пресека обави пре него што се добије структурна функција целог СН. Како су главни резултати овог рада везани за приступ одозго наниже, овде ће бити приказан само један, најпознатији представник приступа одоздо навише.

MICSUP алгоритам

MICSUP (*M*inimal *C*ut *S*ets *U*Pward) алгоритам је базиран на запажању да су скупови пресека неког подстабла СН подскупови скупова пресека датог СН. Уз то, уколико међу скуповима пресека подстабла СН, може по неком основу да се изврши редукција, она ће се по истом основу извршити и међу скуповима пресека целог СН. Алгоритам није формализован, али се укратко може описати на следећи начин: разматрати сукцесивно посредне догађаје СН одоздо навише и за свако од њих одредити минипресеке. Поступак понављати док се не достигне вршни

догађај [115]. У сваком кораку у коме се одређују минипресеци, примењује се нека од метода одозго наниже.

Пример

MICSUP алгоритам ће бити илустрован у комбинацији са директном методом за СН са слике 2.4.

$$G5 = P5 + S2$$

$$G4 = P4 + S1$$

$$G3 = P2 + G5 + P3 = P2 + P5 + S2 + P3$$

$$G2 = P2 + G4 + P3 = P2 + P4 + S1 + P3$$

$$G1 = G2 \cdot G3 = (P2 + P5 + S2 + P3) \cdot (P2 + P4 + S1 + P3)$$

$$= P2 \cdot P2 + P2 \cdot P4 + P2 \cdot S1 + P2 \cdot P3 + P5 \cdot P2 + P5 \cdot P4 + P5 \cdot S1 + P5 \cdot P3 + S2 \cdot P2 + S2 \cdot P4 + S2 \cdot S1 + S2 \cdot P3 + P3 \cdot P2 + P3 \cdot P4 + P3 \cdot S1 + P3 \cdot P3$$

$$= P2 + P3 + P5 \cdot P4 + P5 \cdot S1 + S2 \cdot P4 + S2 \cdot S1$$

$$T = P1 + G1 = P1 + P2 + P3 + P5 \cdot P4 + P5 \cdot S1 + S2 \cdot P4 + S2 \cdot S1$$

Добијено је свих 7 минипресека: {P1}, {P2}, {P3}, {P5, P4}, {P5, S1}, {S2, P4}, {S2, S1}.

На приказаном примеру може да се схвати основна идеја али не и ефикасност алгоритма, зато што се редукција врши тек у претпоследњем кораку.

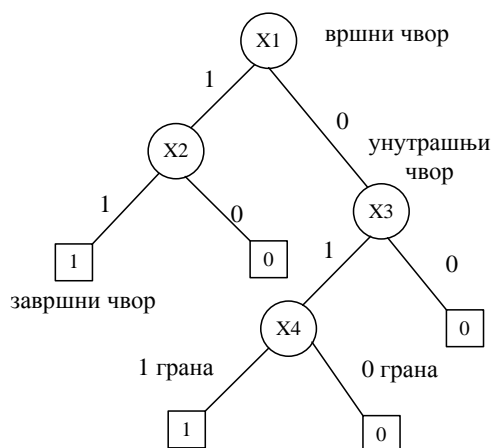
2.5.1.3. Бинарни дијаграми одлучивања

Бинарне дијаграме одлучивања (*Binary Decision Diagram*, БДО) је увео Lee [55] као начин за представљање и манипулацију са логичким функцијама који редукује рачунарско време и меморијске захтеве. Прва примена у АСН описана је у [88]. где је приказана трансформација СН у БДО, из кога се затим директно добијају минискупови.

Најсложенији корак у одређивању минипресека помоћу БДО је управо трансформација СН, јер димензије БДО експоненцијално расту са бројем примарних догађаја. Због тога је највећи број радова посвећен испитивању

ефикасности различитих поредака примарних догађаја [100] и различитих метода трансформације [114]. Показало се да сложеност трансформације СН у БДО може значајно да се смањи оптимизацијом поретка примарних догађаја. Међутим, тај оптимизациони проблем је НП-комплетан [23].

БДО којим се моделира СН, је укореењено бинарно стабло у коме се сви путеви завршавају у једном од два стања: 1, што одговара отказу система и 0, што одговара успеху система. БДО се састоји од завршних и унутрашњих чворова, који су повезани гранама. Унутрашњи чворови одговарају примарним догађајима СН, а завршни чворови имају вредност 1 или 0, која представља одигравање или неодигравање примарног догађаја. Пример БДО приказан је на слици 2.6 [100].



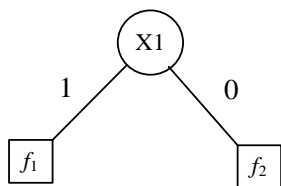
Слика 2.6. Бинарни дијаграм одлучивања

Сваки унутрашњи чвор има две гране: леву (1) која одговара одигравању примарног догађаја (отказу) и десну (0) која одговара неодигравању примарног догађаја. Сви чворови на путевима који се завршавају у стању 1 и улазни су чворови 1-гране представљају скупове пресека. У БДО се подразумева да су гране оријентисане од чворова на нижим ка чворовима на вишим нивоима. За БДО на слици 2.6. скупови пресека су:

- (1) {X1,X2}
- (2) {X3,X4}.

Сваком унутрашњем чвору БДО одговара једна *if-then-else* структура која се записује у облику $ite(X1, f_1, f_2)$. Ако се догађај X1 (отказ) деси *ite* има вредност f_1 ,

у супротном има вредност f_2 , односно f_1 је излазни чвор гране 1 а f_2 излазни чвор гране 0 (слика 2.7).



Слика 2.7. БДО за $ite(X1, f_1, f_2)$

Пре формирања БДО за дато СН, потребно је дефинисати релацију поретка међу примарним догађајима. Као што је већ поменуто, ефикасност трансформације СН у БДО зависи од изабраног поретка.

У наставку је приказана једна процедура трансформисања СН у БДО.

1. Придружити сваком примарном догађају X_i у СН ite структуру $ite(X_i, 1, 0)$.
2. Нека за два логичка кола важи $X > Y$ и $J = ite(X, F1, F2)$ и $H = ite(Y, G1, G2)$. Тада је $J \langle op \rangle H = ite(X, F1 \langle op \rangle H, F2 \langle op \rangle H)$.
3. Нека за два логичка кола важи $X \equiv Y$ и $J = ite(X, F1, F2)$ и $H = ite(X, G1, G2)$. Тада је $J \langle op \rangle H = ite(X, F1 \langle op \rangle G1, F2 \langle op \rangle G1)$.

$\langle op \rangle$ је оператор који одговара Буловим операцијама, односно логичким колима СН. При томе важи следеће:

$1 \langle op \rangle H = 1$ ако је $\langle op \rangle$ ИЛИ логичко коло

$1 \langle op \rangle H = H$ ако је $\langle op \rangle$ И логичко коло

$0 \langle op \rangle H = H$ ако је $\langle op \rangle$ ИЛИ логичко коло

$0 \langle op \rangle H = 0$ ако је $\langle op \rangle$ И логичко коло

Примењујући описану процедуру од дна СН ка врху, добија се сложена ite структура за вршни догађај СН, у којој фигуришу само примарни догађаји и на основу које се формира БДО.

Пример

Трансформација СН у БДО и одређивање минипресека ће бити илустрована на примеру СН са слике 2.1.

С обзиром да је у питању СН са вишеструким догађајима, поредак примарних догађаја ће бити утврђен техником специјалног распоређивања. Ова техника се састоји у томе да се прво разматрају вишеструки догађаји и да се већа вредност додељује догађајима са већим бројем понављања. Догађаји са истим бројем понављања се рангирају на основу положаја у самом СН. Утврђени поредак примарних догађаја за дато СН је: $S2 > X5 > S1 > X4 > X1 > X3 > X2$.

Применом процедуре 1-3, добијају се следеће *ite* структуре:

$$G4 = \text{ite}(X4, 1, 0) + \text{ite}(S1, 1, 0) = \text{ite}(X4, 1, \text{ite}(S1, 1, 0))$$

$$G5 = \text{ite}(X5, 1, 0) + \text{ite}(S2, 1, 0) = \text{ite}(X5, 1, \text{ite}(S2, 1, 0))$$

$$G2 = \text{ite}(X2, 1, 0) + G4 + \text{ite}(X3, 1, 0) = \text{ite}(X2, 1, 0) + \text{ite}(X4, 1, \text{ite}(S1, 1, 0)) + \text{ite}(X3, 1, 0) = \text{ite}(X2, 1, \text{ite}(X4, 1, \text{ite}(S1, 1, 0))) + \text{ite}(X3, 1, 0) = \text{ite}(X2, 1, \text{ite}(X4, 1, \text{ite}(S1, 1, 0))) + \text{ite}(X3, 1, 0) = \text{ite}(X2, 1, \text{ite}(X3, 1, \text{ite}(X4, 1, \text{ite}(S1, 1, 0))))$$

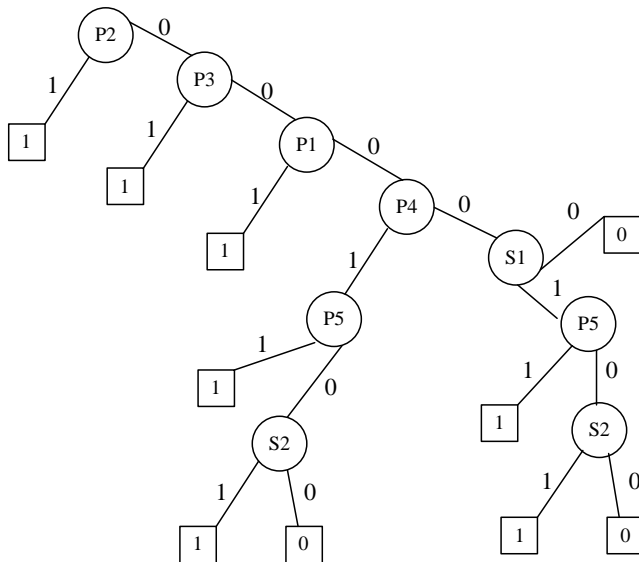
$$G3 = \text{ite}(X2, 1, 0) + \text{ite}(X5, 1, \text{ite}(S2, 1, 0)) + \text{ite}(X3, 1, 0) = \text{ite}(X2, 1, \text{ite}(X3, 1, \text{ite}(X5, 1, \text{ite}(S2, 1, 0))))$$

$$G1 = G2.G3 = \text{ite}(X2, 1, \text{ite}(X3, 1, \text{ite}(X4, 1, \text{ite}(S1, 1, 0)))) . \text{ite}(X2, 1, \text{ite}(X3, 1, \text{ite}(X5, 1, \text{ite}(S2, 1, 0)))) = \text{ite}(X2, 1, \text{ite}(X3, 1, \text{ite}(X4, 1, \text{ite}(S1, 1, 0)))) . \text{ite}(X3, 1, \text{ite}(X5, 1, \text{ite}(S2, 1, 0)))) = \text{ite}(X2, 1, \text{ite}(X3, 1, \text{ite}(X4, 1, \text{ite}(S1, 1, 0)))) . \text{ite}(X5, 1, \text{ite}(S2, 1, 0)))) = \text{ite}(X2, 1, \text{ite}(X3, 1, \text{ite}(X4, \text{ite}(X5, 1, \text{ite}(S2, 1, 0))), \text{ite}(S1, 1, 0)) . \text{ite}(X5, 1, \text{ite}(S2, 1, 0)))) = \text{ite}(X2, 1, \text{ite}(X3, 1, \text{ite}(X4, \text{ite}(X5, 1, \text{ite}(S2, 1, 0))), \text{ite}(S1, \text{ite}(X5, 1, \text{ite}(S2, 1, 0))), 0))))$$

$$T = \text{ite}(X1, 1, 0) + G1 = \text{ite}(X1, 1, 0) + \text{ite}(X2, 1, \text{ite}(X3, 1, \text{ite}(X4, \text{ite}(X5, 1, \text{ite}(S2, 1, 0))), \text{ite}(S1, \text{ite}(X5, 1, \text{ite}(S2, 1, 0))), 0)))) = \text{ite}(X2, 1, \text{ite}(X3, 1, \text{ite}(X4, \text{ite}(X5, 1, \text{ite}(S2, 1, 0))), \text{ite}(S1, \text{ite}(X5, 1, \text{ite}(S2, 1, 0))), 0)))) + \text{ite}(X1, 1, 0) = \text{ite}(X2, 1, \text{ite}(X3, 1, \text{ite}(X4, \text{ite}(X5, 1, \text{ite}(S2, 1, 0))), \text{ite}(S1, \text{ite}(X5, 1, \text{ite}(S2, 1, 0))), 0)))) + \text{ite}(X1, 1, 0)$$

$$T = \text{ite}(X2, 1, \text{ite}(X3, 1, \text{ite}(X1, 1, \text{ite}(X4, \text{ite}(X5, 1, \text{ite}(S2, 1, 0))), \text{ite}(S1, \text{ite}(X5, 1, \text{ite}(S2, 1, 0))), 0))))$$

БДО који се формира на основу последње ите структуре за вршни догађај приказан је на слици 2.8.



Слика 2.8. БДО за СН са слике 2.1.

Сви путеви у БДО који почињу из чвора који нема улазну грану или чвора чија је улазна грана 0, а завршавају се у чвору 1, одговарају минипресецима. На тим путевима, минипресецима припадају само они чворови чије су излазне гране 1. На пример пут $\{S1.P5.S2\}$ одговара минипресеку, јер почетни чвор $S1$ има улазну грану 0, а крајњи чвор $S2$ има излазну грану 1. Међутим, на том путу, излазна грана чвора $P5$ је 0, тако да је одговарајући минипресек $\{S1,S2\}$. На исти начин, добијено је свих седам минипресека СН са слике 2.1:

(1) $P2$; (2) $P3$; (3) $P1$; (4) $P4.P5$; (5) $P4.S2$; (6) $S1.P5$; (7) $S1.S2$

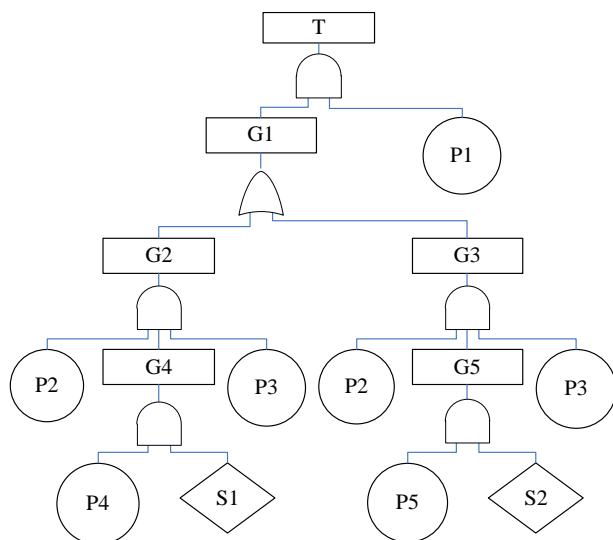
2.5.1.4. Одређивање минималних скупова путева

Минимални скупови путева (минипутеви) (*Minimum path sets*) су дуални (комплементни) скупови минипресека. Скупови путева представљају скуп примарних догађаја, чије неодигравање гарантује да се неће десити вршни догађај. Минипутеви су скупови путева који се не могу редуковати без губљења статуса скупа путева. Ако се ниједан од догађаја из минипута не деси, неће се десити ни вршни догађај.

Један од начина за одређивање минипутева је преко дуалног СН. Минипресеци дуалног СН су минипутеви полазног (прималног) СН [112].

Пример

На слици 2.9. је приказано дуално СН са слике 2.2, означено Буловим променљивама. Применом неке од метода за одређивање минипресека, добијају се минипутеви почетног СН (табела 2.2).



Слика 2.9. Дуално СН стабла неисправности са слике 2.4.

Табела 2.2. Минипутеви СН са слике 2.1.

Минипут	Опис
P1 P2 P3 P4 S1	Примарни отказ мотора, примарни отказ напајања, примарни отказ осигурача, отказ прекидача 1 и отворен прекидач 1
P1 P2 P3 P5 S2	Примарни отказ мотора, примарни отказ напајања, примарни отказ осигурача, отказ прекидача 2 и отворен прекидач 2

На основу табеле се може закључити да до нежељеног догађаја “мотор не ради” неће доћи ако се не деси ниједан од догађаја из прве врсте или ниједан од догађаја из друге врсте табеле.

2.5.2. Рангирање минипресека и примарних догађаја

Поред одређивања минипресека, квалитативна анализа обухвата и њихово рангирање по значају као и рангирање примарних догађаја. Минипресеци се рангирају на основу броја примарних догађаја од којих се састоје. Ако је број примарних догађаја у минипресеку l или мањи, за минипресека се каже да је реда l . С обзиром да су примарни догађаји независни, вероватноћа да ће се десити истовремено, једнака је производу њихових вероватноћа. Због тога су, ако су вероватноће примарних догађаја истог реда величине, минипресеци мањег реда већег значаја (вишег ранга).

Пример

У табели 2.3. је дата листа и опис свих минипресека СН са слике 2.1. За дато СН, сви минипресеци дужине 1 су првог, а сви остали минипресеци другог ранга

Табела 2.3. Минипресеци за СН са слике 2.1.

Ранг	Минипресека	Опис
1	P1	Примарни отказ мотора
1	P2	Примарни отказ напајања
1	P3	Примарни отказ осигурача
2	P5 P4	Отказ оба прекидача
2	P5 S1	Отворен прекидач 1 и отказ прекидача 2
2	S2 P4	Отворен прекидач 2 и отказ прекидача 1
2	S2S1	Отворена оба прекидача

У квалитативној АСН, значај примарних догађаја се утврђује на основу броја њиховог појављивања у различитим минипресецима. Међутим, већ из датог примера се лако уочава да подаци које даје квалитативна анализа, нису довољни за прецизно рангирање примарних догађаја. Наиме, сваки од догађаја P4, P5, S1, S2 се јавља у два минипресека и због тога би требало да буду вишег ранга. С друге стране, догађаји P1, P2 и P3 имају појављивање у само једном минипресеку,

али као једини догађај тог скупа, што би могло да буде разлог да се њима додели виши ранг. С обзиром да су у оба случаја минипресеци мале дужине, јасно је да би вероватноћа одигравања примарних догађаја омогућила далеко прецизније рангирање. Поред тога, вероватноћа одигравања примарних догађаја би утицала и на рангирање минипресека.

2.6. Квантитативна АСН

Квантитативна анализа СН се састоји у одређивању или процени вероватноће одигравања нежељеног догађаја и мерењу значајности примарних догађаја. Алгоритми за квантитативну анализу се могу сврстати у три групе [34]:

- Индиректне методе које користе минипресеке или БДО;
- Модуларне методе и
- Директне методе које користе одсецања, објектно оријентисани приступ, рекурзивност, препознавање патерна итд.

С обзиром да је акценат овог рада на одређивању минипресека, овде ће бити само укратко изложени основни појмови квантитативне анализе СН.

2.6.1. Одређивање вероватноће вршног догађаја

Вероватноћа вршног догађаја представља вероватноћу отказа система. Она се аналитички може одредити на два начина: директно из СН датог система и помоћу минипресека. У оба случаја се претпоставља да су познате вероватноће свих примарних догађаја.

Када СН не садржи вишеструке догађаје, вероватноћа вршног догађаја се може одредити директно кроз СН, без одређивања минипресека. На основу вероватноћа примарних догађаја и логичких кола се, идући уз СН, одређују вероватноће посредних догађаја, све док се не стигне до вршног догађаја. За кохерентна СН користе се само изрази за одређивање вероватноће у случају И и ИЛИ логичких кола између догађаја [125].

Рачунање вероватноће за свако логичко коло зависи од односа између улазних догађаја. У табели 2.4. су приказане формуле за рачунање вероватноћа логичких кола. Улазни догађаји су означени са A и B , а излазни са E .

Табела 2.4. Рачунање вероватноћа излаза из логичких кола

Зависност догађаја	Вероватноћа	
	логичко коло И	логичко коло ИЛИ
A и B су независни	$P(E) = P(A) \cdot P(B)$	$P(E) = P(A) + P(B) - P(A) \cdot P(B)$
A и B су зависни	$P(E) = P(A) \cdot P(B A)$ $= P(B) \cdot P(A B)$	$P(E) = P(A) + P(B) - P(A \cap B)$ $= P(A) + P(B) - P(A) \cdot P(B A)$
A и B зависни и $A \subset B$	$P(E) = P(A)$	$P(E) = P(B)$

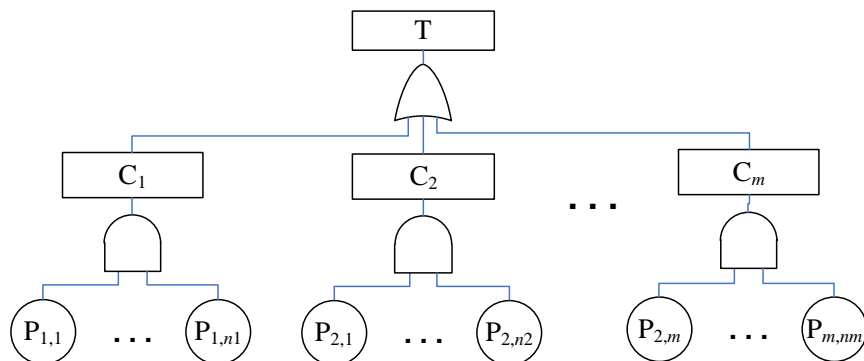
Уколико СН садржи вишеструке догађаје, вероватноћа вршног догађаја се, управо због вишеструких догађаја, не може одредити директно из СН. Један од начина за њено одређивање је на основу минипресека. Ако се посматра ДНФ структурне функције $\varphi(x)$, лако се закључује да је вероватноћа минипресека једнака производу вероватноћа његових примарних догађаја. Међутим, вероватноћа вршног догађаја је мања од суме вероватноћа минипресека, зато што се исти примарни догађај може јавити у више минипресека.

Један од начина одређивања вероватноће вршног догађаја се заснива на представљању вршног догађаја као уније минипресека [49]. Минипресеци се могу представити као пресек својих примарних догађаја, односно:

$$\begin{aligned}
 C_1 &= \{P_{1,1}, P_{1,2}, \dots, P_{1,n_1}\} = \left\{ \bigcap_{j=1}^{n_1} P_{1,j} \right\}, \\
 &\vdots \\
 C_m &= \{P_{m,1}, P_{m,2}, \dots, P_{m,n_m}\} = \left\{ \bigcap_{j=1}^{n_m} P_{m,j} \right\},
 \end{aligned} \tag{2.4}$$

где је P_{ij} i -ти примарни догађај у j -том минипресеку C_j .

СН се сада може приказати као на слици 2.10.



Слика 2.10. СН представљено помоћу минипресека

Другим речима, када су познати минипресеци, отказ система се може приказати серијском везом свих минипресека у којима су сви примарни догађаји повезани паралелно [39].

Вршни догађај Т ће се десити, ако се деси бар један од минипресека, тј.

$$\{T\} = \left\{ \bigcup_{i=1}^m C_i \right\}, \quad (2.5)$$

Заменом (2.4) у (2.5) се вероватноћа одигравања вршног догађаја може израчунати на следећи начин:

$$P\{T\} = P\left\{ \bigcup_{i=1}^m C_i \right\} = \sum_{i=1}^m P(C_i) - \sum_{i=1}^{m-1} \sum_{j=i+1}^m P(C_i \cap C_j) + \sum_{i=1}^{m-2} \sum_{j=i+1}^{m-1} \sum_{k=j+1}^m P(C_i \cap C_j \cap C_k) + \dots + (-1)^{m-1} P\left(\bigcap_{i=1}^m C_i \right) \quad (2.6)$$

Применом (2.6) се, за СН малих димензија, може одредити тачна вредност вероватноће вршног догађаја.

Пример

У табели 2.5. су приказане вредности вероватноћа примарних догађаја СН са слике 2.1, а у табели 2.6. вероватноће минипресека и вршног догађаја, као и ранг минипресека на основу њихове вероватноће.

Табела 2.5. Вероватноће примарних догађаја

Догађај	Опис	Вероватноћа
P ₁	Примарни отказ мотора	2·10 ⁻¹⁵
P ₂	Примарни отказ напајања	7·10 ⁻¹⁷
P ₃	Примарни отказ осигурача	4·10 ⁻¹⁴
P ₄	Отказ прекидача 1	1·10 ⁻¹³
P ₅	Отказ прекидача 2	3·10 ⁻¹⁴
S ₁	Отворен прекидач 1	2·10 ⁻¹²
S ₂	Отворен прекидач 2	5·10 ⁻¹⁰

Табела 2.6. Вероватноће минипресека и вршног догађаја

Ранг	Минипресека	Опис	Вероватноћа
1	P ₁	Примарни отказ мотора	2·10 ⁻¹⁵
1	P ₂	Примарни отказ напајања	7·10 ⁻¹⁷
1	P ₃	Примарни отказ осигурача	4·10 ⁻¹⁴
2	P ₅ P ₄	Отказ оба прекидача	3·10 ⁻²⁷
2	P ₅ S ₁	Отворен прекидач 1 и отказ прекидача 2	6·10 ⁻²⁶
2	S ₂ P ₄	Отворен прекидач 2 и отказ прекидача 1	5·10 ⁻²³
2	P ₁ P ₂	Отворена оба прекидача	1·10 ⁻²¹
		Вршни догађај	4,207·10 ⁻¹⁴

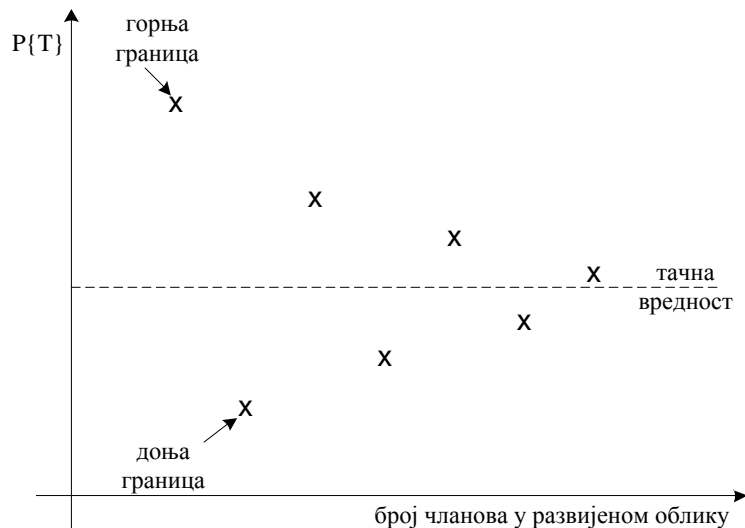
Код СН великих димензија, ако су вероватноће примарних догађаја мале, вероватноћа вршног догађаја се може апроксимирати сумом вероватноћа минипресека. Добијена вредност се најчешће користи као горња граница вероватноће вршног догађаја [104]:

$$P\{T\} \leq \sum_{i=1}^m P(C_i) \quad (2.7)$$

Доњу границу вероватноће вршног догађаја представљају прва два члана (2.6):

$$P\{T\} \geq \sum_{i=1}^m P(C_i) - \sum_{i=1}^{m-1} \sum_{j=i+1}^m P(C_i \cap C_j) \quad (2.8)$$

Што је већи број чланова у (2.6) одређен то је вредност процењене вероватноће ближа тачној вредности (слика 2.11)



Слика 2.11. Оцена вероватноће вршног догађаја на основу (2.6)

Вероватноћа вршног догађаја се може одредити и помоћу вероватноћа минипутева. Минипутеви се могу представити као унија својих примарних догађаја, а вршни догађај као пресек свих минипутева.

2.6.2. Мере значајности

Мере значајности, које могу бити релативне и апсолутне, су један од најважнијих резултата АСН. Њима се утврђује значај појединачних догађаја и њихов допринос вероватноћи вршног догађаја, као и осетљивост вероватноће вршног догађаја на повећање или смањење њихове вероватноће. У највећем броју система, свега 20% примарних догађаја СН је значајно са аспекта вероватноће. На основу анализе мера значајности, могу се доносити одлуке о расподели ресурса за тестирање, одржавање и контролу. Овде ће бити приказане најчешће коришћене мере значајности [104].

Ф-В значајност (*Fussell-Vesely (F-V) Importance*) (релативна и апсолутна) - утврђује утицај свих догађаја у СН на вероватноћу вршног догађаја, што даље

омогућује њихово рангирање. Ф-В значајност се рачуна сабирањем вероватноћа свих минипресека који садрже посматрани догађај.

Пример

У табели 2.7. је приказан ранг примарних догађаја и вероватноће свих минипресека који садрже посматрани догађај.

Табела 2.7. Ранг примарних догађаја на основу Ф-В мере значајности

Ранг	Догађај	ОПИС	Вероватноћа
1	P ₃	Примарни отказ осигурача	$4 \cdot 10^{-14}$
2	P ₁	Примарни отказ мотора	$2 \cdot 10^{-15}$
3	P ₂	Примарни отказ напајања	$7 \cdot 10^{-17}$
4	S ₁	Отворен прекидач 1	$1 \cdot 10^{-21}$
4	S ₂	Отворен прекидач 2	$1 \cdot 10^{-21}$
6	P ₄	Отказ прекидача 1	$5 \cdot 10^{-23}$
7	P ₅	Отказ прекидача 2	$6.3 \cdot 10^{-26}$

ВРР мера. Вредност редукције ризика (*Risk Reduction Worth, RRW*) утврђује колико се смањује вероватноћа вршног догађаја, ако се осигура неодиигравање посматраног примарног догађаја. Ова мера се обично одређује постављањем да је вероватноћа посматраног догађаја једнака 0 и поновним рачунањем вероватноће вршног догађаја.

Пример

У табели 2.8. је приказан ранг примарних догађаја на основу ове ВРР мере. Вероватноће вршног догађаја добијене су на основу (2.6) за вероватноће примарних догађаја из табеле 2.5. и вероватноћу 0 за посматрани примарни догађај.

Табела 2.8. Ранг примарних догађаја на основу BPP мере значајности

Догађај	ОПИС	Вероватноћа вршног догађаја
P ₃	Примарни отказ осигурача	$2,07 \cdot 10^{-15}$
P ₁	Примарни отказ мотора	$4,01 \cdot 10^{-14}$
P ₂	Примарни отказ напајања	$4,2 \cdot 10^{-14}$
S ₁	Отворен прекидач 1	$4,207 \cdot 10^{-14}$
S ₂	Отворен прекидач 2	$4,207 \cdot 10^{-14}$
P ₄	Отказ прекидача 1	$4,207 \cdot 10^{-14}$
P ₅	Отказ прекидача 2	$4,207 \cdot 10^{-14}$

ВОР мера. Вредност одигравања ризика (*Risk Achievement Worth, RAW*) утврђује колико се повећава вероватноћа вршног догађаја, ако посматрани догађаја на нижем нивоу СН сигурно одиграо. Ова мера указује на догађаје чија је превенција најважнија. Одређује се постављањем вероватноће посматраног догађаја на 1 и поновним рачунањем вероватноће вршног догађаја.

Пример

Табела 2.9. Ранг примарних догађаја на основу ВОР мере значајности

Догађај	ОПИС	Вероватноћа вршног догађаја
P ₁	Примарни отказ мотора	1
P ₂	Примарни отказ напајања	1
P ₃	Примарни отказ осигурача	1
S ₁	Отворен прекидач 1	$5 \cdot 10^{-10}$
P ₄	Отказ прекидача 1	$5 \cdot 10^{-10}$
S ₂	Отворен прекидач 2	$2,14 \cdot 10^{-12}$
P ₅	Отказ прекидача 2	$2,14 \cdot 10^{-12}$

БМ мера. Бирнбаумова мера значајности (*Birnbaum's Importance Measure, BM*) утврђује утицај промене вероватноће догађаја на нижем нивоу у СН, на промену вероватноће вршног догађаја. Ова мера обједињује претходне две мере а када су све три мере апсолутне, може се израчунати као њихов збир: $BM = BPP + WOR$.

3. АНАЛИЗА СН ВЕЛИКИХ ДИМЕНЗИЈА

Као што је поменуто у претходној глави, проблем одређивања минипресека СН је НП-тежак. Постоје различити приступи у решавању комбинаторне експлозије код СН великих димензија. Заједничко за све њих је да на неки начин редукују проблем.

Прва идеја за решавање овог проблема била је раздвајање СН на независне модуле, који се затим одвојено анализирају [122]. При томе се може разлагати дато СН или се може декомпоновати његова структурна функција [23]. Ефикасност овог приступа зависи од структуре СН, а са порастом вишеструкости примарних догађаја, број модула који се могу издвојити опада.

Другим приступом се полази од структуре СН и, уколико је могуће, редукују се (одсецају) његови делови, пре одређивања скупова пресека. При томе се одсецају чворови, гране или подстабла, чијим се уклањањем елиминишу само скупови пресека који нису минимали. У 3.1 ће бити приказане две овакве редукције СН.

Трећа група метода полази од ДНФ структурне функције и бави се елиминисањем скупова пресека који нису минимални. Овај корак је рачунарски веома захтеван, и просторно и временски, јер се упоређује сваки пар скупова пресека. Методама из ове групе углавном се предлажу различите технике записа примарних догађаја које убрзавају поступак упоређивања скупова пресека и елиминисања надскупова. У 3.2 ће бити описане две овакве технике.

Иако може да се сврста у трећи приступ, директно издвајање минипресека ће бити приказано у посебном делу (3.3). Наиме, код директног издвајања минипресека се, пре елиминисања надскупова, неки скупови пресека могу издвојити из ДНФ структурне функције СН и тиме смањити број скупова пресека који се међусобно пореде. На основу теореме Лимниоса (*Limnios*) [59] показано је да је скуп пресека, који не садржи вишеструке догађаје, минипресека и да не може бити подскуп неког од скупова пресека. У овом делу ће бити приказана нова теорема којом се доказује да је и скуп пресека, који садржи сва понављања својих вишеструких догађаја,

минималан. Тиме се број међусобних поређења, за једну класу СН, значајно смањује. На крају ће бити приказан нови алгоритам редукције СН.

Четврти приступ представљају методе скраћивања (*truncating*) скупа минипресека, којима се одређују само минипресеци значајни са структурног или са аспекта њихове вероватноће. У 3.4 ће бити приказан најстарији алгоритам за одређивање минипресека датог реда.

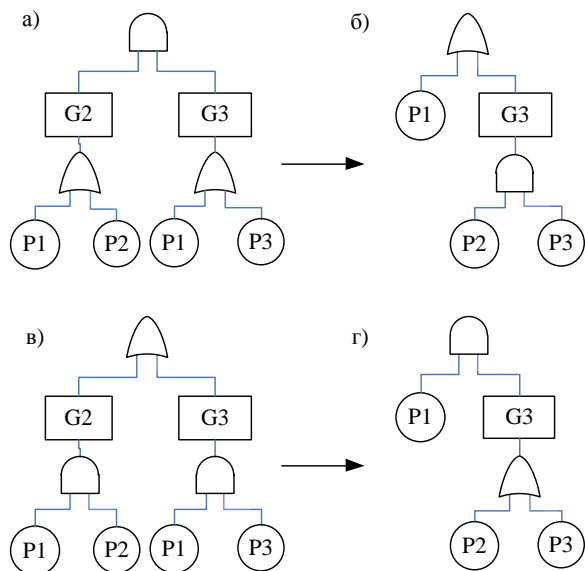
3.1. Редукција пре одређивања скупова пресека

Ниједна од метода из ове групе не гарантује елиминацију свих редувантних чворова СН, али смањују димензије СН и број понављања вишестуких догађаја. Најстарија и најпознатија метода редукције СН је Фонеова (*Faunet*) редукција, која је интегрисана у многе алгоритме, нарочито оне који се базирају на БДО. Поред тога, значајно је поменути и шеме редукције Ериксона (*Ericson*), којима се редукују појединачни догађаји или целе гране СН.

Фонеова редукција

Фонеова редукција је техника која се користи за свођење СН на минимални облик, да би се елиминисали сви “шумови” у систему уз задржавање његове логике. Често се користи у комбинацији са БДО. Ефективност технике је показана на примеру СН великих димензија, где је резултујући БДО био 50% мањи од БДО добијеног без претходне редукције СН. Метода се састоји из три фазе (примарни догађаји који се јављају само једанпут у СН су названи јединствени) [89]:

- Скраћивање - јединствене примарне догађаје који су улаз истог логичког кола, спојити у један чвор.
- Факторизација – парове примарних догађаја који се увек јављају заједно, заменити једним сложеним догађајем.
- Екстракција – идентификовати структуре са слика 3.1. а) и в) и заменити их структурама са слика 3.1. б) и г), респективно.

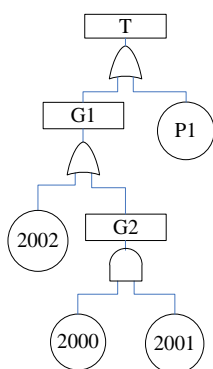


Слика 3.1. Екстракција СН

Ова три корака се понављају, све док више није могућа ниједна од промена у СН, а резултат је компактнији приказ система. При сваком скраћивању и факротизацији чувају се информације о структури сложених догађаја. Када се одреде скупови пресека редукованог СН, разлагањем сложених догађаја добијају се скупови пресека полазног СН.

Пример

Применом Фонеове редукције на СН са слике 2.1. добија се редуковано СН са информацијама о сложеним догађајима на слици 3.2.



Сложени догађај	Логичко коло	Догађај 1	Догађај 2
2000	ИЛИ	P4	S1
2001	ИЛИ	P5	S2
2002	ИЛИ	P2	P3

Слика 3.2. Редуковано СН са слике 2.4. применом Фонеове редукције

Коришћењем неке од метода из претходног поглавља, на пример Директне методе, добијају се скупови пресека: {2002}, {2000-2001} и {P1}. Разлагањем сложених догађаја и применом одговарајућих операција Булове алгебре, добија се свих седам минипресека датог СН.

Шеме редукције Ериксона

Ериксон је у [29] предложио редукцију СН засновану на препознавању шаблона (*patterns*) међусобног положаја вишеструкости примарних догађаја. Одсецање (*pruning*) чворова или подстабала се врши тако да се сачувају сви минипресеци. Редукција користи само два закона Булове алгебре: комутативност и апсорпцију. У табели 3.1. приказано је шест Ериксоних шаблона и правила редукције за сваки од њих.

Табела 3.1. Шаблони редукције СН

Случај 1	Случај 2	Случај 3
<p>Сва логичка кола су типа ИЛИ</p> $CS = \{A, B, C, D, E, A, F\}$ $= \{A, B, C, D, E, F\}$	<p>Сва логичка кола су типа ИЛИ осим И кола непосредно изнад једног од понављања</p> $CS = \{A, B, C, D, E, AF\}$ $= \{A, B, C, D, E\}$	<p>Сва логичка кола су типа ИЛИ осим И кола изнад једног од понављања</p> $CS = \{A, B, C, D, EA, EF\}$ $= \{A, B, C, D, EF\}$
$A + A = A$	$A + A \cdot F = A$	$A + A \cdot E = A$
<p>Правило: Одсећи једно од понављања</p>	<p>Правило: Одсећи подстабло испод И кола</p>	<p>Правило: Одсећи понављање испод И кола</p>

Случај 4	Случај 5	Случај 6
<p>Сва логичка кола су типа ИЛИ, вршно коло је типа И</p> $CS = \{AD, AE, AA, AF, BD, BE, BA, BF, CD, CE, CA, CF\}$ $= \{A, BD, BE, BF, CD, CE, CF\}$	<p>Вршно коло је типа И, са једне стране су сва кола типа ИЛИ, а са друге типа И</p> $CS = \{ADEAF, BDEAF, CDEAF\}$ $= \{ADEF\}$	<p>Вршно коло је типа ИЛИ, са једне стране су сва кола типа ИЛИ, а са друге типа И</p> $CS = \{A, B, C, DEAF\}$ $= \{A, B, C\}$
$A \cdot A = A$ $A + A \cdot D = A$	$A \cdot D \cdot E \cdot A \cdot F = A \cdot D \cdot E \cdot F$ $A \cdot D \cdot E \cdot F + B \cdot D \cdot E \cdot A \cdot F = A \cdot D \cdot E \cdot F$	$A + D \cdot E + A \cdot F = A$
<p>Правило: Не може се извршити одсецање</p>	<p>Правило: Може се одсећи цела страна са ИЛИ колима</p>	<p>Правило: Може се одсећи цела страна са И колима</p>

Ова правила редукције се могу лако интегрисати у програм за одређивање минипресека или се могу вршити ручно на самом СН.

Пример

На СН са слике 2.1. се не може применити ни један од приказаних случајева.

3.2. Редукција након одређивања скупова пресека

Као што је већ поменуто, ова група метода полази од ДНФ структурне функције и елиминисање скупова пресека који нису минимални. Елиминисање надскупова се врши директним упоређивањем сваког пара скупова пресека и одређивање да ли

је један садржан у другом. Овде ће бити приказани *ELRAFT* и Техника манипулације битовима које, специјалним записом скупова пресека, убрзавају њихово међусобно поређење.

ELRAFT

ELRAFT (Efficient Logic Reduction Analysis of Fault Tree) је редукциони алгоритам који је предложио Семандер (*Semanderes*) у [97]. Сваки примарни догађај се обележава различитим простим бројем, а вредност скупа пресека представља производ бројева његових примарних догађаја. Алгоритмом се из скупова пресека издвајају минипресеци директним упоређивањем вредности скупова, а не њихових појединачних догађаја.

Редукција скупова пресека се састоји у следећем: Нека су дата два скупа пресека S_1 , чија је вредност x и S_2 , чија је вредност y . Ако је x дељиво са y онда је S_2 подскуп скупа S_1 и скуп пресека S_1 се може редуковати. Да би се побољшала ефикасност редукције, пре њене примене је потребно применити законе апсорпције, да би се елиминисали редундантни догађаји у скуповима пресека.

Пример

ELRAFT редукција ће бити илустрована на примеру СН са слике 2.4. Нека су примарним догађајима, редом, додељени следећи прости бројеви: $P_1=2$, $P_2=3$, $P_3=5$, $P_4=7$, $P_5=11$, $S_1=13$ и $S_2=17$. Након добијања скупова пресека и примене закона апсорпције, добијају се скупови са следећим вредностима:

$P_1=2$, $P_2=3$, $P_2 \cdot P_4=21$, $P_2 \cdot S_1=39$, $P_2 \cdot P_3=15$, $P_5 \cdot P_2=33$, $P_5 \cdot P_4=77$, $P_5 \cdot S_1=143$, $P_5 \cdot P_3=55$, $S_2 \cdot P_2=51$, $S_2 \cdot P_4=119$, $S_2 \cdot S_1=221$, $S_2 \cdot P_3=85$, $P_3 \cdot P_2=15$, $P_3 \cdot P_4=35$, $P_3 \cdot S_1=65$, $P_3=5$.

Упоређивањем прво дужих скупова са краћим, редукују се скупови $P_2 \cdot P_4$, $P_2 \cdot S_1$, $P_2 \cdot P_3$, $P_5 \cdot P_2$, $S_2 \cdot P_2$, $P_3 \cdot P_2$, дељиви са 3 (скуп P_2), а затим скупови $P_5 \cdot P_3$, $S_2 \cdot P_3$, $P_3 \cdot P_4$, $P_3 \cdot S_1$, дељиви са 5 (скуп P_3). Скупови $P_5 \cdot P_4$, $P_5 \cdot S_1$, $S_2 \cdot P_4$ и $S_2 \cdot S_1$ нису

дељиви са вредношћу ни једног скупа, тако да, поред скупова са једним догађајем (P1, P2 и P3) представљају свих седам минипресека СН са слике 2.4.

Техника манипулације битовима

Техника манипулације битовима (*Bit manipulation*) се први пут појавила 1977. године у [116], а касније је унапређена у [19]. Развијена је као техника која омогућава компјутерску АСН, а заснована је на бинарном кодирању догађаја кохерентног СН и манипулисању битовима за редукцију СН.

Дужина бинарног записа, којим ће примарни догађаји бити представљени, једнака је броју примарних догађаја. Онда се, редом, примарним догађајима додељују бинарни записи са јединицом на различитом месту. Скупови пресека се представљају као збир бинарних записа својих примарних догађаја. Редукција се врши тако што се над сваким паром скупова пресека прво примени операција ИЛИ, а затим се над резултатом и сваким појединачним скупом пресека, из посматраног пара, примени операција Ексклузивно ИЛИ. Ако је коначни резултат 0, скуп пресека је редундантан и елиминише се.

Пример

Редукција манипулацијом битовима ће бити илустрована на примеру СН са слике 2.4. Пошто постоји 7 примарних догађаја, њихов бинарни запис ће бити дужине 7. Нека су примарним догађајима додељени следећи бинарни записи: P1=1000000, P2=0100000, P3=0010000, P4=0001000, P5= 0000100, S1=0000010 и S2=0000001.

Након добијања скупова пресека и примене закона апсорције, добијају се скупови са следећим бинарним записима:

P1=1000000, P2=0100000, P2·P4=0101000, P2·S1=0100010, P2·P3=0110000,
P5·P2=0100100, P5·P4=0001100, P5·S1=0000110, P5·P3=0010100, S2·P2=0100001,
S2·P4=0001001, S2·S1=0000011, S2·P3=0010001, P3·P2=0110000, P3·P4=0011000,
P3·S1=0010010, P3=0010000.

Редукција ће бити приказана на примеру два пара скупова пресека, једном у коме има редукције: P2 и P5·P2 и другом у коме нема редукције: P5·P4 и S2·P4.

Применом оператора ИЛИ над овим паровима добија се:

P2	0100000	P5·P4	0001100
P5·P2	<u>0100100</u>	S2·P4	<u>0001001</u>
ИЛИ	0100100	ИЛИ	0001101

Применом оператора ексклузивно ИЛИ (ЕИЛИ), добија се:

P2	0100000	P5·P4	0001100
ИЛИ	<u>0100100</u>	ИЛИ	<u>0001101</u>
ЕИЛИ	0000100	ЕИЛИ	0000001

P2 није редувантно.

P5·P4 није редувантно

P5·P2	0100100
ИЛИ	<u>0100100</u>
ЕИЛИ	0000000

S2·P4	0001001
ИЛИ	<u>0001101</u>
ЕИЛИ	0000100

P5·P2 је редувантно.

S2·P4 није редувантно

3.3. Директно издвајање минипресека

Када се редукција врши након што су одређени сви скупови пресека, потребно је извршити велики број поређења у којима се проверава да ли је један скуп пресека подскуп неког другог. Нека је s број скупова пресека. С обзиром да се поређење два скупа пресека врши у оба смера, максимални број операција које треба извршити је $s \cdot (s - 1)$.

Уколико би за неке од скупова пресека могло директно, на основу њихових својстава, да се утврди да су минипресеци, онда би се број операција смањило. Нека је међу s скупова пресека m оних који су минипресеци. То значи да они не могу бити надскуп неког другог скупа пресека и да је број скупова пресека које би

требало испитати $s-m$. Другим речима, потребно је $s-m$ скупова пресека поредити међусобно и за сваки од њих проверити да ли је неки од m минипресека његов подскуп. Тиме број максималних операција које треба извршити постаје:

$$(s-m) \cdot (s-m-1) + (s-m) \cdot m = (s-m) \cdot (s-1). \quad (3.1)$$

Однос максималног броја операција које треба извршити и укупног броја операција је линеаран:

$$\frac{(s-m) \cdot (s-1)}{s \cdot (s-1)} = 1 - \frac{m}{s}. \quad (3.2)$$

3.3.1. Теорема Лимниоса

У [59] је доказано да су скупови пресека кохерентног СН, који не садрже вишеструке догађаје, минипресеци. Овде ће бити наведени: оригинална теорема са доказом, последица теореме и алгоритам редуковања СН. Нотација која је у оригиналном раду коришћена је следећа:

- X, Y, Z - скупови пресека;
- K - фамилија свих скупова пресека;
- $K1$ - фамилија свих скупова пресека који садрже вишеструке догађаје;
- $K2$ - фамилија свих скупова пресека који не садрже вишеструке догађаје;
- $P[G]$ - домен логичког кола G (или Буловог израза G). Садржи ознаке свих примарних догађаја који се јављају у подстаблу кореном у колу G (ознаке свих примарних догађаја који се јављају у Буловом изразу G).

Посматрајмо фамилију K скупова пресека СН, добијену развијањем вршног догађаја или MOCUS алгоритмом.

Теорема 3.1: Посматрајмо скупове пресека $X, Z \in K$. Ако

- а) X захтева редукцију $x \cdot x = x$, или
- б) X имплицира Z

тада X садржи бар један вишеструки догађај.

Доказ:

- а) Израз $x \cdot x$ се може добити само у случају И кола. Дакле, домени два различита улаза тог И кола садрже догађај x . Следи, x је вишеструки догађај.
- б) Ако X имплицира Z тада је $Z = X + XY$. Израз $Z = X + XY$ се може добити или у случају И кола, $X(R+Y) = X + XY$, где R имплицира X ; или у случају ИЛИ кола, $(X)+(XY)$. У оба случаја $P[X] \cap P[F] = P[S]$, где је $F = R+S$ или XY ; и $S = R$ или X , респективно. Следи да X садржи бар један вишеструки догађај.

◆

Последица 3.1:

- а) Елементи K_2 су минимални.
- б) Редуковањем елемената K_1 , добија се фамилија K_1^* а фамилија K^* свих минималних скупова је једнака $K_1^* \cup K_2$.

Алгоритам редукције АРСН1:

1. Одредити фамилију K помоћу МОСUS алгоритма. Ако СН не садржи вишеструке догађаје, $K^* = K$. Ићи на корак 5.
2. Поделити фамилију K на K_1 , фамилију скупова пресека који садрже бар један вишеструки догађај и K_2 , фамилију скупова пресека који не садрже вишеструке догађаје.
3. Редуковати фамилију K_1 да би се добила K_1^* .
4. Нека је $K^* = K_1^* \cup K_2$.
5. Сваки елемент K^* је минипресек.

Крај

На основу теореме 3.1 и последице 3.1, максимални број операција (3.1) се може смањити на

$$(s-m) \cdot (s-m-1). \quad (3.3)$$

Напоменимо да је доказ претходне теореме тешко читљив али да је, са спекта редукције, значајна његова последица. У наставку ће бити представљена теорема чија је једна од последица минималност скупова пресека без вишеструких догађаја.

3.3.2. Теорема о минималности скупова пресека са вишеструким догађајима

У наставку ће бити уведена и доказана теорема на основу које ће се показати да је скуп пресека минималан ако садржи све вишеструкости својих примарних догађаја. Тиме се уједно доказује и теорема 3.1 и последица 3.1, јер скупови пресека без вишеструких догађаја постају специјалан случај посматраних скупова. Након тога ће бити уведени услови под којима такав скуп пресека постоји и утврђено колики је минимални број скупова пресека који су његов надскуп.

Ради лакшег записа, примарни догађаји ће бити обележени бинарним променљивама $x_i, i = 1, \dots, J, J \leq I$, које су дефинисане на почетку поглавља 2.5, а које представљају њихов индикатор. Ако СН садржи вишеструке догађаје, сва понављања неког вишеструког догађаја ће бити означена истом бинарном променљивом. Параметар $k_i, i = 1, \dots, J$ је број понављања, односно вишеструкост примарног догађаја x_i . Уколико примарни догађај нема понављања, сматраће се да је вишеструкости 1. Како је $\sum_{i=1}^n k_i = I, J < I$ за СН са вишеструким догађајима, односно $J = I$ ако СН садржи само догађаје вишеструкости 1, односно само јединствене догађаје. Поред неведене, биће коришћена следећа нотација:

- $G_v, v \in V$ - посредни догађаји. Њихов број је једнак броју логичких кола у СН и они се овде разматрају само у смислу логичког кола;
- $P[G_v]$ - домен посредног догађаја G_v . То је мултискуп свих примарних догађаја који су који припадају подстаблу СН са кореном G_v . Подстабло СН са кореном G_v представљају сва деца чвора G_v . Како је T вршни догађај, односно корен СН, $P[T]$ је мултискуп свих примарних догађаја СН и $|P[T]| = I$;
- $D[G_v]$ - скуп свих скупова пресека у подстаблу са кореном G_v . Скуп свих скупова пресека СН обележаваће се са $D[T]$.
- C - било који скуп пресека.

Сваки скуп пресека $C \in D[T]$ је један елемент ДНФ структурне функције СН, односно представља конјункцију неких примарних догађаја: x_1, x_2, \dots, x_n . У складу са Буловом алгебром, у наставку ће скуп пресека C , односно конјункција: $x_1 \wedge x_2 \wedge \dots \wedge x_n$ бити записани у облику: $C = x_1 \cdot x_2 \cdot \dots \cdot x_n$, односно у облику $C = x_1^{k_1} \cdot x_2^{k_2} \cdot \dots \cdot x_n^{k_n}$ ако скуп пресека садржи више понављања посматраних примарних догађаја. На исти начин ће ДНФ структурне функције СН $\bigvee_{C \in D[T]}$, на

основу правила Булове алгебре, бити записана у облику: $\sum_{C \in D[T]} C$.

У складу са уведеном нотацијом, дефинисаћемо релацију \subset за скупове пресека: скуп пресека $C^1 = x_1^{k_1} \cdot x_2^{k_2} \cdot \dots \cdot x_n^{k_n}$ је “подскуп” скупа пресека $C^2 = x_1^{k_1^2} \cdot x_2^{k_2^2} \cdot \dots \cdot x_n^{k_n^2}$, тј. $C^1 \subset C^2$ ако $k_i^1 \leq k_i^2$ за свако $i = 1, \dots, n$.

У наредној теорему ће се посматрати скупови пресека C^* који садрже сва понављања својих примарних догађаја. Односно, ако неки примарни догађаји x_1, x_2, \dots, x_n припадају скупу пресека C^* , онда свих $k_i, i = 1, \dots, n$ понављања ових примарних догађаја, припада скупу пресека C^* . За скуп пресека C^* ће се рећи да је облика $x_1^{k_1} \cdot x_2^{k_2} \cdot \dots \cdot x_n^{k_n}$.

Напомена 3.1: У теорему 3.2, посматраће се скупови пресека пре примене закона идемпотенције. Према томе, иако по закону идемпотенције важи да је $x \cdot x = x$, овде су $x \cdot x = x^2$ и x различити скупови пресека, односно $x^2 \not\subset x$.

Теорема 3.2: Нека је T вршни догађај кохерентног СН и нека су $k_i \geq 1, i = 1, \dots, n$ максималне вишеструкости примарних догађаја x_1, x_2, \dots, x_n . Ако постоји скуп пресека C^* облика $x_1^{k_1} \cdot x_2^{k_2} \cdot \dots \cdot x_n^{k_n}$ из $D[T]$, тада међу осталим скуповима пресека из $D[T]$, не постоји скуп пресека \tilde{C} такав да је $\tilde{C} \subset C^*$ или $C^* \subset \tilde{C}$.

Доказ: Теорема ће бити доказана трансфинитном индукцијом по броју примарних догађаја I .

1. За $I=1$ једини скуп пресека је T , па тврђење очигледно важи. С обзиром да неке дефиниције СН подразумевају бар једно логичко коло са најмање два улазна догађаја, за почетну вредност биће узето $I=2$.

За $I=2$, примарни догађаји могу бити догађаји x_1 и x_2 или само догађај x_1 који је вишеструкости 2. СН може имати само једно логичко коло, у односу на које постоје два случаја:

а) Логичко коло је типа ИЛИ. Тада, ако постоје два догађаја x_1 и x_2 , скупови пресека су x_1 и x_2 , и ниједан није подскуп другог. Ако постоји један примарни догађај x_1 , скупови пресека су x_1 и x_1 и ниједан није облика C^* .

б) Логичко коло је типа И. Тада је једини скуп пресека $x_1 \cdot x_2$ ако постоје два догађаја x_1 и x_2 , односно x_1^2 ако постоји један примарни догађај. Оба скупа пресека су облика C^* и у оба случаја не постоји скуп пресека \tilde{C} такав да је $\tilde{C} \subset C^*$ или $C^* \subset \tilde{C}$.

2. Нека је $I \geq 2$ (односно $I \geq 3$ ако је почетна вредност $I=2$). Претпоставимо да тврђење важи за све СН која имају мање од I примарних догађаја. Докажимо да тврђење важи и за СН која имају I примарних догађаја.

Без умањења општости, претпоставимо да вршни догађај СН има два чвора детета на следећем нивоу СН: G_1 и G_2 . Нека је CH_1 подстабло са кореном G_1 и структурном функцијом $\sum_{C \in D[G_1]} C$ и CH_2 подстабло са кореном G_2 и структурном

функцијом $\sum_{C \in D[G_2]} C$. У односу на тип логичког кола у корену СН, могућа су два

случаја:

а) Логичко коло у корену СН је типа ИЛИ. Тада је структурна функција СН облика

$$\sum_{C \in D[G_1]} C + \sum_{C \in D[G_2]} C. \quad (3.4)$$

Ако скуп пресека C^* постоји, он је садржан у бар једној од сума из (3.4). На пример, нека је C^* садржан у $\sum_{C \in D[G_1]} C$. С обзиром да су k_1, k_2, \dots, k_n вишеструкости примарних догађаја x_1, x_2, \dots, x_n , следи да се сва понављања примарних догађаја x_1, x_2, \dots, x_n налазе у $P[G_1]$. Значи, ниједно понављање примарних догађаја x_1, x_2, \dots, x_n се не налази у $P[G_2]$, па у CH_2 не постоји скуп пресека \tilde{C} за кога је $\tilde{C} \subset C^*$ или $C^* \subset \tilde{C}$.

Како у CH_2 постоји бар један примарни догађај, следи да CH_1 има мање од I примарних догађаја. На основу индукцијске хипотезе следи да ни у CH_1 не постоји скуп пресека \tilde{C} такав да је $\tilde{C} \subset C^*$ или $C^* \subset \tilde{C}$.

Из (3.4) следи да у CH не постоји скуп пресека \tilde{C} такав да је $\tilde{C} \subset C^*$ или $C^* \subset \tilde{C}$.

б) Логичко коло у корену CH је типа И. Тада је структурна функција CH облика

$$\sum_{C \in D[G_1]} C \cdot \sum_{C \in D[G_2]} C. \quad (3.5)$$

Скуп пресека $C^* = x_1^{k_1} \cdot x_2^{k_2} \cdot \dots \cdot x_n^{k_n}$ постоји ако и само ако је,

$$\begin{aligned} \sum_{C \in D[G_1]} C &= C_1^* + \sum_{X \in D[G_1] \setminus C_1^*} X, \\ \sum_{C \in D[G_2]} C &= C_2^* + \sum_{Y \in D[G_2] \setminus C_2^*} Y, \end{aligned} \quad (3.6)$$

$$C_1^* = x_1^{k_1'} \cdot \dots \cdot x_n^{k_n'}, C_2^* = x_1^{k_1''} \cdot \dots \cdot x_n^{k_n''}, k_i' + k_i'' = k_i, k_i', k_i'' \geq 0, i = 1, \dots, n.$$

Како су k_1, k_2, \dots, k_n вишеструкости примарних догађаја x_1, x_2, \dots, x_n у CH , следи да је k_i' (k_i'') вишеструкост догађаја x_i у CH_1 (CH_2) ако је $k_i' \geq 1$ ($k_i'' \geq 1$). У случају $k_i' = 0$ ($k_i'' = 0$), x_i се не појављује у CH_1 (CH_2). Одатле, следи да је C_1^* (C_2^*) из (3.6) скуп пресека облика C^* за подстабло CH_1 (CH_2). Из $k_i' + k_i'' = k_i, i = 1, \dots, n$ следи да је $C_1^* \cdot C_2^* = C^*$.

Заменом (3.6) у (3.5) добија се структурна функција СН:

$$C^* + C_1^* \cdot \sum_{Y \in D[G_2] \setminus C_2^*} Y + C_2^* \cdot \sum_{X \in D[G_1] \setminus C_1^*} X + \sum_{X \in D[G_1] \setminus C_1^*} X \cdot \sum_{Y \in D[G_2] \setminus C_2^*} Y \quad (3.7)$$

Како подстабла СН₁ и СН₂ садрже мање од I примарних догађаја, на основу индукцијске хипотезе, за СН₁ не постоји скуп пресека $\tilde{C}_1 \in D[G_1] \setminus C_1^*$ такав да је $C_1^* \subset \tilde{C}_1$ или $\tilde{C}_1 \subset C_1^*$, односно за СН₂ не постоји скуп пресека $\tilde{C}_2 \in D[G_2] \setminus C_2^*$ такав да је $C_2^* \subset \tilde{C}_2$ или $\tilde{C}_2 \subset C_2^*$.

Претпоставимо, супротно тврђењу, да за СН са I примарних догађаја постоји скуп пресека $\tilde{C} \in D[T] \setminus C^*$ такав да је $\tilde{C} \subset C^*$ или $C^* \subset \tilde{C}$.

Нека је $C^* \subset \tilde{C}$. Из (3.7) следи је сваки скуп пресека из $D[T] \setminus C^*$ облика: $C_1^* \cdot Y$, $C_2^* \cdot X$ или $X \cdot Y$, где $X \in D[G_1] \setminus C_1^*$ и $Y \in D[G_2] \setminus C_2^*$. Како је $C^* = C_1^* \cdot C_2^*$, тада је $C_1^* \subset C_r$ или $C_2^* \subset C_s$, што је супротно индукцијској претпоставци.

Нека је сада $\tilde{C} \subset C^*$. Као и у претходном случају, \tilde{C} је облика: $C_1^* \cdot Y$, $C_2^* \cdot X$ или $X \cdot Y$, где $X \in D[G_1] \setminus C_1^*$ и $Y \in D[G_2] \setminus C_2^*$. Како је $C^* = C_1^* \cdot C_2^*$, тада је $X \subset C_1^*$ или $Y \subset C_2^*$, што је такође супротно индукцијској претпоставци.

Доказ је аналоган за случај када вршни догађај има $u > 2$ чвора детета на следећем нивоу СН.

У делу 2 а) доказа, структурна функција (3.4) је:

$$\sum_{C \in D[G_1]} C + \sum_{C \in D[G_2]} C + \dots + \sum_{C \in D[G_u]} C \quad (3.8)$$

где се сва понављања примарних догађаја x_1, x_2, \dots, x_n и скуп пресека C^* јављају у

$\sum_{C \in D[G_1]} C$. У том случају може се увести смена:

$$\sum_{C \in D[G_2]} C + \dots + \sum_{C \in D[G_u]} C = \sum_{C \in D[\bigcup_{j=2}^u G_j]} C$$

чиме се израз (3.8) своди на (3.4).

У делу 2 б) доказа, структурна функција (3.5) је:

$$\sum_{C \in D[G_1]} C \cdot \sum_{C \in D[G_2]} C \cdot \dots \cdot \sum_{C \in D[G_u]} C \quad (3.9)$$

где свако $\sum_{C \in D[G_v]} C$ представља структурну функцију подстабла CH_v , $v = 1, \dots, u$.

Скуп пресека C^* постоји ако и само ако је,

$$\begin{aligned} \sum_{C \in D[G_v]} C &= C_v^* + \sum_{C \in D[G_v] \setminus C_v^*} C, \quad v = 1, \dots, u \\ \sum_{j=1}^u k_j^v &= k_i, \quad k_i^v \geq 0, \quad i = 1, \dots, n, \quad v = 1, \dots, u \end{aligned} \quad (3.10)$$

Заменом (3.10) у (3.9) добија се структурна функција СН:

$$C^* + C_1^* \cdot \prod_{v=2}^u \sum_{C \in D[G_v]} C + C_1^* \cdot C_2^* \cdot \prod_{v=3}^u \sum_{C \in D[G_v]} C + \dots + \prod_{v=1}^u \sum_{C \in D[G_v]} C$$

Наставак доказа је аналоган доказу у 2 б). Тиме је теорема доказана. \blacklozenge

По теорему 3.2, не постоји скуп пресека \tilde{C} такав да је $C^* \subset \tilde{C}$ или $\tilde{C} \subset C^*$. Када је $k_i = 1$, $i = 1, \dots, n$, догађаји x_1, x_2, \dots, x_n су јединствени, а $C^* = x_1 \cdot x_2 \cdot \dots \cdot x_n$ је скуп пресека у коме су сви догађаји вишеструкости 1. Овим је последица теореме 3.1 сведена на специјалан случај теореме 3.2. Односно, на основу теореме 3.2 Дефиниције 2.4. из поглавља 2.5.1, важе следеће последице:

Последица 3.2. Ниједан скуп пресека се не може редуковати као надскуп скупа пресека у коме су сви догађаји вишеструкости 1.

Последица 3.3: Скуп пресека у коме су сви догађаји вишеструкости 1 је минималан.

Из последица 3.2. и 3.3. следи да, приликом редукције скупова пресека, ове скупове не треба упоређивати са осталим скуповима пресека.

Напомена 3.2: Над скупом пресека C^* се може $\sum_{i=1}^n (k_i - 1)$ пут применити закон идемпотенције.

Нека је C скуп пресека облика $x_1 \cdot x_2 \cdot \dots \cdot x_n$ добијен из C^* применом закона идемпотенције. На основу теореме 3.2, за свако \tilde{C} из $D[T] \setminus C^*$, важи $\tilde{C} \not\subset C^*$. Како је $C \subset C^*$ онда је $\tilde{C} \not\subset C$. На основу Дефиниције 2.4 из 2.5.1, важи:

Последица 3.4. Скуп пресека C облика $x_1 \cdot x_2 \cdot \dots \cdot x_n$, добијен из скупа пресека C^* применом закона идемпотенције, је минималан.

Ова последица је нови резултат и из ње следи да, приликом редукције скупова пресека, за ове скупове не треба проверавати да ли су надскуп неког скупа пресека. Међутим, ови скупови могу бити подскуп неког скупа пресека, јер из $C^* \not\subset \tilde{C}$ и $C \subset C^*$ не следи $C \not\subset \tilde{C}$.

На основу свега наведеног, фамилија скупова пресека од n елемената се може поделити у три групе:

1. $K1$, фамилија од m_1 скупова пресека са свим догађајима вишеструкости 1 у СН. Ови скупови пресека су минипресеци који не могу бити садржани ни у једном од осталих скупова пресека.
2. $K2$, фамилија од m_2 скупова пресека са свим вишеструкостима својих примарних догађаја, односно скупова пресека C^* . Ови скупови пресека су минипресеци који могу бити садржани у осталим скуповима пресека.
3. $K3$, фамилија од $s - m_1 - m_2$ скупова пресека међу којима треба уклонити надскупове.

Сада се максимални број операција поређења (3.3) може смањити на:

$$(s - m_1 - m_2) \cdot m_2 + (s - m_1 - m_2) \cdot (s - m_1 - m_2 - 1). \quad (3.11)$$

Дакле, потребно је $s - m_1 - m_2$ скупова пресека поредити међусобно и за сваки од њих проверити да ли је неки од m_2 минипресека његов подскуп. При томе је поређење ефикасније ако се прво изврши провера да ли међу елементима $K1$ има

подскупова елемената из $K3$. Сада се Алгоритам редукције АРСН1 може модификовати увођењем додатних корака.

Алгоритам редукције АРСН2.

1. Одредити фамилију K свих скупова пресека кохерентног СН. Ако СН садржи само догађаје вишеструкости 1, $K^* = K$. Ићи на корак 6.
2. Поделити фамилију K на фамилију $K1$ скупова пресека који садрже само догађаје вишеструкости 1, фамилију $K2$ скупова пресека који садрже сва понављања вишеструких догађаја и фамилију $K3 = K \setminus K1 \setminus K2$.
3. Применити закон идемпотенције на $K2$. Редуковати $K3$ уклањањем сваког елемента који је надскуп неког елемента из $K2$, да би се добила $K3^*$.
4. Редуковати фамилију $K3^*$ елиминисањем надскупова да би се добила $K3^{**}$.
5. Нека је $K^* = K1 \cup K2 \cup K3^{**}$.
6. Сваки елемент K^* је минипресек.

Крај

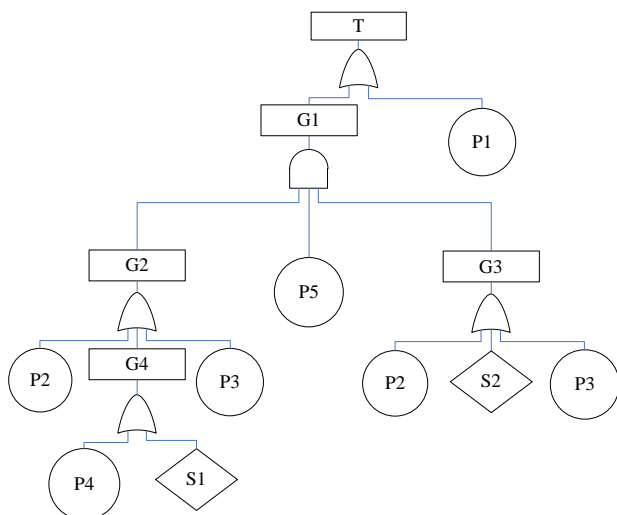
Након корака 3, добија се фамилија $K3^*$, чији је број елемената $m_3 \leq s - m_1 - m_2$, тако да се максимални број операција поређења (3.11) додатно може смањити на:

$$(s - m_1 - m_2) \cdot m_2 + m_3 \cdot (m_3 - 1). \quad (3.12)$$

Први члан израза (3.12) представља број операција поређења у кораку 3, док је други члан број операција поређења приликом елиминације надскупова у кораку 4 алгоритма редукције СН2.

Пример

Да би се боље илустровале претходне тврдње и алгоритам, СН са слике 2.4 је модификовано премештањем примарног догађаја P5 (слика 3.3).



Слика 3.3. Модификовано СН са слике 2.4

Скупови пресека овог СН су: $P1$, $P2^2 \cdot P5$, $P2 \cdot P4 \cdot P5$, $P2 \cdot S1 \cdot P5$, $P2 \cdot P3 \cdot P5$, $S2 \cdot P2 \cdot P5$, $S2 \cdot P4 \cdot P5$, $S2 \cdot S1 \cdot P5$, $S2 \cdot P3 \cdot P5$, $P3 \cdot P2 \cdot P5$, $P3 \cdot P4 \cdot P5$, $P3 \cdot S1 \cdot P5$, $P3^2 \cdot P5$. Сада је:

$$K1 = \{P1, S2 \cdot P4 \cdot P5, S2 \cdot S1 \cdot P5\},$$

$$K2 = \{P2^2 \cdot P5, P3^2 \cdot P5\}, \text{ односно } K2 = \{P2 \cdot P5, P3 \cdot P5\} \text{ и}$$

$$K3 = \{P2 \cdot P4 \cdot P5, P2 \cdot S1 \cdot P5, P2 \cdot P3 \cdot P5, S2 \cdot P2 \cdot P5, S2 \cdot P3 \cdot P5, P3 \cdot P2 \cdot P5, P3 \cdot P4 \cdot P5, P3 \cdot S1 \cdot P5\}.$$

Применом корака 3 алгоритма, добија се $K3^* = \emptyset$.

3.3.3. Структура СН које садржи C^*

Скуп пресека C^* не постоји у сваком, већ само у СН одређене структуре. У овом поглављу ће бити описани услови које СН мора да задовољава да би садржало скупове пресека C^* . Поред раније описане, користиће се следећа нотација:

– $G_{ij}, i, j = 1, \dots, J$ - минимални посредни догађај (минимално логичко коло) примарних догађаја x_i и x_j , односно корен минималног подстабла које садржи дате примарне догађаје.

Логична претпоставка теореме 3.3 је да у СН не постоји логичко коло чији су чворови деца догађаји који представљају понављања истог примарног догађаја.

Теорема 3.3: Нека су $k_i \geq 1, i = 1, \dots, n$ вишеструкости догађаја x_1, x_2, \dots, x_n кохерентог СН. Ако скуп пресека C^* постоји, тада је за сваки пар примарних догађаја из C^* минимално логичко коло G_{ij} типа И.

Доказ: Теорема може бити доказана трансфинитном индукцијом по броју примарних догађаја I . У случају $I=1$ или $I=2$ тврђење је очигледно (видети теорему 3.2).

У случају $I \geq 2$ ($I \geq 3$ ако је почетна вредност $I=2$), нека тврђење важи за сва СН која имају мање од I примарних догађаја. Докажимо да тврђење важи и за СН која имају I примарних догађаја.

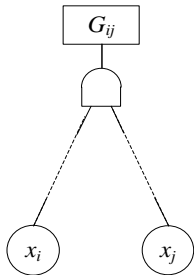
На основу доказа Теореме 3.2, ако постоји скуп пресека $C^* = x_1^{k_1} \cdot x_2^{k_2} \cdot \dots \cdot x_n^{k_n}$, структурна функција СН је облика (3.4) или облика (3.5). СН је облика (3.4) ако је логичко коло у корену СН типа ИЛИ и сва понављања примарних догађаја x_1, x_2, \dots, x_n се налазе у СН₁ које има мање од I примарних догађаја. На основу индукцијске хипотезе, за пар понављања x_i и x_j из C^* , минимално логичко коло G_{ij} је типа И у СН₁, а самим тим и у СН.

СН је облика (3.5) ако је логичко коло у корену СН типа И. Нека се x_i и x_j налазе у истом подстаблу, на пример у СН₁. Тада је $C_1^* = x_1^{k_1} \cdot \dots \cdot x_n^{k_n}$ скуп пресека облика C^* за подстабло СН₁. На основу индукцијске хипотезе, минимално логичко коло G_{ij} за пар x_i и x_j , је типа И у СН₁, а самим тим и у СН. Ако се x_i и x_j налазе у различитим подстаблима СН₁ и СН₂, онда је минимално логичко коло за пар x_i и x_j корен СН типа И.

Доказ је аналоган за случај када логичко коло које одговара вршном догађају има $I > 2$ улазних догађаја (као у случају доказа Теореме 3.2.). ♦

Нека је дато СН са кореном T у коме постоји C^* . Посматрајмо једно понављање догађаја $x_i, i \in \{1, \dots, n\}$. Догађај x_i је лист датог СН. Идући од листа x_i ка корену,

нађимо прво минимално логичко коло G_{ij} за дато понављање x_i и неко понављање x_j , $j \in \{1, \dots, n\}$ (j може бити једнако i). На основу теореме 2.3, G_{ij} је типа И (слика 3.4.).



Слика 3.4. Минимално логичко коло G_{ij}

Ако је неко логичко коло G на путу од x_i до G_{ij} типа И, онда у $P[G]$ постоје само примарни догађаји $x \notin C^*$, јер је G_{ij} прво минимално логичко коло за x_i и неко x_j . Одатле следи да у структурној функцији СН постоји скуп пресека $\tilde{C} = \tilde{x} \cdot x_1^{k_1} \cdot x_2^{k_2} \cdot \dots \cdot x_n^{k_n}$, што је супротно тврђењу теореме 3.2. Дакле, сва логичка кола на путу од x_i до G_{ij} су типа ИЛИ, па је структурна функција подстабла са кореном G_{ij} облика:

$$(x_i + \sum_i \bullet) \cdot (x_j + \sum_j \bullet) \quad (3.13)$$

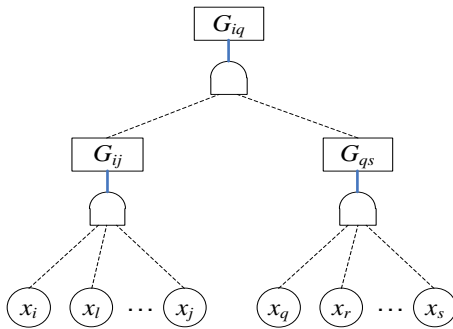
где је $\sum_i \bullet$ сума која не садрже ни једно понављање догађаја x_1, x_2, \dots, x_n (јер је G_{ij} минимално логичко коло за x_i и неко од x_j , $j \in \{1, \dots, n\}$) и која може бити сложени израз. Уз то, $x_i + \sum_i \bullet$ је структурна функција подстабла чији је корен G посредни догађај, који је непосредни улаз минималног кола G_{ij} . Другим речима, G је последњи посредни догађај на путу од од x_i до G_{ij} .

Минимално логичко коло G_{ij} је у (3.13) приказано као бинарно. Није тешко приметити да ако G_{ij} није бинарно, онда је оно минимално логичко за више понављања догађаја x_1, x_2, \dots, x_n и да у сваком његовом подстаблу мора постојати

бар једно понављање догађаја x_1, x_2, \dots, x_n . Тада је структурна функција подстабла са кореном G_{ij} облика:

$$(x_i + \sum_i \bullet) \cdot (x_j + \sum_j \bullet) \cdot \dots \cdot (x_l + \sum_l \bullet) \quad (3.14)$$

Идући даље од G_{ij} ка корену СН, нађимо следеће минимално логичко коло G_{iq} за x_i и неко $x_q \notin P[G_{ij}]$, $q \in C^*$. G_{iq} је такође типа И (слика 3.5).



Слика 3.5. Минимално логичко коло G_{iq}

Као и у случају G_{ij} , ако G_{iq} није бинарно, у сваком његовом подстаблу мора постојати бар једано понављање догађаја x_1, x_2, \dots, x_n , па је структурна функција подстабла са кореном G_{iq} :

$$[(x_i + \sum_i \bullet) \cdot \dots \cdot (x_j + \sum_j \bullet) + \sum_{ij} \bullet] \cdot \dots \cdot [(x_q + \sum_q \bullet) \cdot \dots \cdot (x_s + \sum_s \bullet) + \sum_{qs} \bullet] \quad (3.15)$$

Ако се израз (3.15) трансформише тако да се издвоје сви чланови, који множењем дају чинилац у коме су само понављања догађаја x_1, x_2, \dots, x_n , добија се:

$$(x_i + \sum_i \bullet) \cdot \dots \cdot (x_j + \sum_j \bullet) \cdot \dots \cdot (x_q + \sum_q \bullet) \cdot \dots \cdot (x_s + \sum_s \bullet) + \sum_{iq} \bullet \quad (3.16)$$

где је $\sum_{iq} \bullet$ члан који може садржати и понављања (али не сва) догађаја x_1, x_2, \dots, x_n и који може бити сложени израз.

Настављајући процес, од листова ка корену СН, добиће се структурна функција СН:

$$(x_1 + \sum_{11} \bullet) \cdot \dots \cdot (x_1 + \sum_{1k_1} \bullet) \cdot \dots \cdot (x_n + \sum_{n1} \bullet) \cdot \dots \cdot (x_n + \sum_{nk_n} \bullet) + \sum_{C^o \in D[T]} C^o \quad (3.17)$$

где је $\sum_{C^o \in D[T]} C^o$ остатак трансформисаног израза у коме се могу појављивати и

x_1, x_2, \dots, x_n . ДНФ функције (3.17) је:

$$x_1^{k_1} \cdot x_2^{k_2} \cdot \dots \cdot x_n^{k_n} + \sum_{C \in D[T] \setminus C^*} C$$

На основу претходног, важи и супротно тврђење Теореме 3.3:

Тврђење 3.1: Ако је за сваки пар догађаја из $\{x_1, x_2, \dots, x_n\}$ минимално логичко коло G_{ij} типа И, скуп пресека C^* постоји.

Пример

За СН са слике 3.3, $P2^2 \cdot P5$ и $P3^2 \cdot P5$ су скупови пресека облика C^* . За сваки пар из скупа пресека $P2^2 \cdot P5$: $(P2, P5)$, $(P2, P2)$ и $(P5, P2)$ минимално логичко коло је типа И чији је излазни догађај $G1$. Исто тако, за сваки пар из скупа пресека $P3^2 \cdot P6$: $(P3, P5)$, $(P3, P3)$ и $(P5, P3)$ минимално логичко коло је типа И (у овом случају са истим излазним догађајем $G1$).

3.3.4. Минимални број редукованих скупова пресека

Издвојимо из структурне функције (3.17) СН, у коме постоји C^* , све чланове који множењем дају C^* :

$$(x_1 + \sum_{i1} \bullet) \cdot \dots \cdot (x_1 + \sum_{1k_1} \bullet) \cdot \dots \cdot (x_n + \sum_{n1} \bullet) \cdot \dots \cdot (x_n + \sum_{nk_n} \bullet) \quad (3.18)$$

Где је $x_i + \sum_{ij} \bullet$ структурна функција подстабла СН које садржи само j -то понављање примарног догађаја x_i . Посматрајмо k_i чинилаца у (3.18) који садрже сва понављања догађаја x_i :

$$(x_i + \sum_{i1} \bullet) \cdot \dots \cdot (x_i + \sum_{ik_i} \bullet) \quad (3.19)$$

Подсетимо да $\sum_{il} \bullet$, као у (3.13), не садржи ни једно понављање догађаја x_1, x_2, \dots, x_n и да може бити сложени израз.

Нека је $\sigma_{il}, l=1, \dots, k_i$ број сабирака у суми \sum_{il} . У развијеном облику, (3.19) има укупно $(1 + \sigma_{i1}) \cdot \dots \cdot (1 + \sigma_{ik_i})$ сабирака, од којих $\sigma_{i1} \cdot \dots \cdot \sigma_{ik_i}$ сабирака не садржи ни једно x_1, x_2, \dots, x_n . Број сабирака који садрже бар једно x_i је:

$$(1 + \sigma_{i1}) \cdot \dots \cdot (1 + \sigma_{ik_i}) - \sigma_{i1} \cdot \dots \cdot \sigma_{ik_i}.$$

Одатле следи да је у (3.18) укупан број сабирака који садрже бар по једно понављање свих x_1, x_2, \dots, x_n једнак:

$$\prod_{i=1}^n [(1 + \sigma_{i1}) \cdot \dots \cdot (1 + \sigma_{ik_i}) - \sigma_{i1} \cdot \dots \cdot \sigma_{ik_i}]. \quad (3.20)$$

Сабирци из (3.18) су скупови пресека СН чија је структурна функција (3.17). Како (3.17) садржи и остатак $\sum_{C^o \in D[T]} C^o$ у коме се могу појављивати x_1, x_2, \dots, x_n , (3.20) је

минималан број скупова пресека који садрже бар по једно појављивање догађаја x_1, x_2, \dots, x_n . Један од тих скупова пресека је C^* .

Када се примени закон идемпотенције над добијеним скуповима пресека, C^* постаје $C = x_1 \cdot x_2 \cdot \dots \cdot x_n$, а минимални број скупова пресека који садрже C , односно који ће се редуковати као његов надскуп, је:

$$\prod_{i=1}^n [(1 + \sigma_{i1}) \cdot \dots \cdot (1 + \sigma_{ik_i}) - \sigma_{i1} \cdot \dots \cdot \sigma_{ik_i}] - 1. \quad (3.21)$$

Уколико су у C^* све вишеструкости једнаке 1, тада (3.20) постаје:

$$\prod_{i=1}^n [(1 + \sigma_{i1}) - \sigma_{i1}] = 1.$$

На основу (3.21), број скупова пресека који се може редуковати као његов надскуп је једнак 0, што је наведено и у последици 3.2.

Пример

Посматрајмо скуп пресека $P2^2 \cdot P5$ за СН са слике 3.3. За прво понављање $P2$, $\sum_{21} = P3 + P4 + S1$, односно $\sigma_{21} = 3$, за једино понављање $P5$, $\sum_{51} = 0$, $\sigma_{51} = 0$ и за друго понављање $P2$, $\sum_{22} = P3 + S2$, $\sigma_{22} = 2$. Након примене закона идемпотенције,

скуп пресека $P2^2 \cdot P5$ постаје минипресек $P2 \cdot P5$, а минимални број скупова пресека који ће се редуковати као његов надскуп, на основу (21), је:

$$[(1 + \sigma_{21}) \cdot (1 + \sigma_{22}) - \sigma_{21} \cdot \sigma_{22}] \cdot [(1 + \sigma_{61}) - \sigma_{61}] - 1 = (4 \cdot 3 - 3 \cdot 2) \cdot (1 - 0) - 1 = 5.$$

Заиста, шест скупова пресека из $K3$ који се редукују као надскуп минипресека $P2 \cdot P6$ су: $P2 \cdot P4 \cdot P5$, $P2 \cdot S1 \cdot P5$, $P2 \cdot P3 \cdot P5$, $S2 \cdot P2 \cdot P5$, $P3 \cdot P2 \cdot P5$.

На исти начин се могу одредити надскупови скупа пресека $P3^2 \cdot P5$, односно минипресека $P3 \cdot P5$: $P2 \cdot P3 \cdot P5$, $S2 \cdot P3 \cdot P5$, $P3 \cdot P2 \cdot P5$, $P3 \cdot P4 \cdot P5$, $P3 \cdot S1 \cdot P5$.

Лако је приметити да, ако у СН постоји више скупова пресека облика C^* , укупан број редукованих скупова пресека неће увек бити једнак збиру скупова пресека редукованих по сваком од C^* . Тај број може бити мањи од збира, јер исти скуп пресека може истовремено бити надскуп више скупова пресека облика C^* .

3.3.5. Алгоритам редукције скупова пресека СН

Наравно, број скупова пресека који ће се редуковати зависи од структуре СН и може бити и нула. Међутим, у општем случају, ако минипресек $C = x_1 \cdot x_2 \cdot \dots \cdot x_n$, добијен идемпотенцијом из $C^* = x_1^{k_1} \cdot x_2^{k_2} \cdot \dots \cdot x_n^{k_n}$, постоји, може се очекивати велики број његових надскупова који расте са порастом вишеструкости догађаја. Због тога је оправдано очекивати да ће редукција скупова пресека бити ефикаснија ако се врши редоследом датим у Алгоритму редукције АРСН2.

Поред тога, што је вишеструкост неког догађаја већа, то је очекивани број скупова пресека у којима ће се појавити већи. Корак 3 Алгоритма АРСН2 ће бити ефикаснији ако се елементи скупова $K2$ и $K3$ сортирају на основу броја понављања, од највећег ка најмањем. Алгоритам редукције СН се коначно може формулисати на следећи начин:

Алгоритам редукције АРСН3.

1. Одредити фамилију K свих скупова пресека кохерентног СН. Ако СН садржи само догађаје вишеструкости 1, $K^* = K$. Ићи на корак 6.

2. Поделити фамилију K на фамилију $K1$ скупова пресека који садрже само догађаје вишструкости 1, фамилију $K2$ скупова пресека који садрже сва понављања вишструких догађаја и фамилију $K3 = K \setminus K1 \setminus K2$.
3. Сортирати скунове пресека из $K2$ и $K3$ на основу вишструкости примарних догађаја, од највеће ка најмањој. Применити закон идемпотенције на $K2$. Редуковати $K3$ уклањањем сваког елемента који је надскуп неког елемента из $K2$, да би се добила $K3^*$.
4. Применити закон идемпотенције на $K3^*$. Редуковати фамилију $K3^*$ да би се добила $K3^{**}$.
5. Нека је $K^* = K1 \cup K2 \cup K3^{**}$.
6. Сваки елемент K^* је минипресек.

Крај

3.3.6. Резултати експеримената

Алгоритам редукције АРСНЗ је примењен на групи СН (*Benchmark Fault Tree*) из [87]. Сви скунови пресека добијени су помоћу инверзне Петријеве мреже, алгоритмом који ће бити описан у главама 6 и 7. У табели 3.2. је приказан број операција поређења скупова пресека применом АРСН1 одређен на основу (3.3), и АРСНЗ на основу (3.12). За свако СН су приказане карактеристике примарних догађаја: укупан број примарних догађаја (БПД), број догађаја чија је вишструкост већа од 1 (БВД) и распон вишструкости (РВ). Поред тога, дате су и карактеристике фамилије скупова пресека: укупан број скупова пресека (s), број скупова пресека са свим догађајима вишструкости 1 (m_1), број скупова пресека облика C^* (m_2) и број скупова пресека у $K3^*$ (m_3). У последњој колони је приказан проценат броја операција у АРСН1, који се не извршава (уштеда) ако се примени АРСНЗ.

Знак * значи да је поређење вршено за првих 40000 добијених скупова пресека. Знак “ф” значи да је на датом СН прво извршена факторизација на основу *Faunet-*

ове редукције. У случају “фм” извршена је факторизација и просто раздвајање СН на два модула. Факторизација и модулација су извршене да би се скратило рачунарско време и простор. Међутим, оне не утичу на однос бројева s , m_1 , m_2 и m_3 . За оригинална СН ове вредности би биле сразмерно веће тако да би вредности у последњој колони остале исте.

Табела 3.2. Број операција поређења Алгоритмима редукције СН1 и СН3

рб	СН	БПД	БВД	РВ	s	m_1	m_2	m_3	АРСН1 (3.3)	АРСН3 (3.12)	%
1	das9205	51	5	2	20160	17280	0	2880	$8,29 \cdot 10^6$	$8,29 \cdot 10^6$	0
2	fr10	175	63	2-9	690	72	4	598	$3,81 \cdot 10^5$	$3,59 \cdot 10^5$	5,73
3	isp9606	89	14	2	2292	867	12	1197	$2,03 \cdot 10^6$	$1,45 \cdot 10^6$	28,61
4	chinese*	35	24	2-4	40000	0	12	23649	$1,60 \cdot 10^9$	$5,6 \cdot 10^8$	65,02
5	chinese(ф)	25	9	2-4	272	0	3	22	$7,37 \cdot 10^4$	$1,27 \cdot 10^3$	98,28
6	das9202(фм)	37	12	3-11	18750	2	20	0	$3,51 \cdot 10^8$	$3,75 \cdot 10^5$	99,89
7	das9203(ф)	51	15	2-3	7000	945	195	248	$3,67 \cdot 10^7$	$1,2 \cdot 10^6$	96,72

Из табеле 3.2. се може уочити да, очекивано, АРСН3 не даје резултате када не постоје скупови пресека облика C^* (рб. 1) или да смањење броја операција поређења може бити веома мало (рб. 2). Међутим, у неким случајевима се може постићи значајна (рб. 3-4) и чак веома значајна (рб. 5-7) уштеда рачунарског времена ако се примени АРСН3.

Такође се може закључити да ефикасност алгоритма зависи од броја догађаја чија је вишеструкост већа од 1 и њихових вишеструкости. У СН под редним бројем 3, где је око 16% примарних догађаја вишеструкости 2, постигнута је уштеда од 28,61% операција поређења. С друге стране, код СН под редним бројевима од 5 до 7, у којима је проценат догађаја, чија је вишеструкост већа од 1, већи (45%, 55% и 29%, респективно) и већи распон вишеструкости, постиже се уштеда од преко 95% операција поређења.

Међутим, може се закључити и да је ефикасност алгоритма веома зависна од структуре датог СН. Наиме, у СН под редним бројем 2, проценат вишеструкости

догађаја је велики (36 %) као и распон вишеструкости (од 2 до 9), а проценат уштеде операција поређења је мали (5,73 %). Овде је у питању СН које у вршном догађају има логичко коло типа ИЛИ са 35 догађаја на следећем нивоу.

Зависност ефикасности АРСНЗ од структуре СН се може уочити и анализом броја скупова пресека, који су редуковани као надскупови скупова пресека облика C^* . Број скупова пресека редукованих на овај начин је једнак: $s - m_1 - m_2 - m_3$. Посматрајмо, на пример, СН под редним бројевима 3 и 4, која имају по 12 скупова пресека облика C^* . Број надскупова од C^* је у 3. СН 216 (9,4% од укупног броја скупова пресека) а у 4. СН 16339 (40,8%). Велика разлика у броју редукованих скупова пресека у ова два СН је последица великог процента догађаја чија је вишеструкост већа од 1 и великог броја свих скупова пресека у СН под редним бројем 4. Посматрајмо сада СН под редним бројевима 2 и 3, у којима је проценат скупова пресека облика C^* у укупном броју скупова пресека, исти - око 5%. У 3. СН је, као што је већ поменуто, 9,4% од укупног броја скупова пресека који су надскупови C^* , док је у 2. СН таквих скупова пресека само 1,7% од укупног броја скупова пресека. Разлог за мали број надскупова у СН под редним бројем 2 је, већ помињана структура, са логичким колом типа ИЛИ у вршном догађају овог СН.

3.4. Одређивање минипресека датог реда

Одређивање свих минипресека за СН великих димензија је често немогуће због ограничења рачунарског времена и/или простора. Због тога је још седамдесетих година започео развој метода скраћивања (*truncating*) скупа минипресека. Основна идеја свих тих метода је да се елиминишу минипресеци чији је допринос вероватноћи нежељеног догађаја испод задатог прага или минипресеци чија дужина прелази задату вредност. Оправдање за овакав приступ лежи и у чињеници да велики број минипресека не обезбеђује квалитетну анализу, па да их није ни потребно све одредити.

Број примарних догађаја у минипресеку се назива ред минипресека, тако да одређивање минипресека датог реда представља одређивање минипресека чији

број примарних догађаја не прелази задату вредност. Овде ће бити приказан један од првих алгоритама скраћивања.

Расмусон (*Rasmuson*) и Маршал (*Marshal*), уз алгоритам *FATRAM*, који је описан у поглављу 2.5.1.1 предлажу и алгоритам за одређивање минискупова задате дужине. Током примене *FATRAM* алгоритма сваки догађај СН се сврстава у једну од шест категорија [85]:

- I Примарни догађаји.
- II Излазни догађаји II кола.
- III Излазни догађаји ИЛИ кола, чији су улаз посредни и примарни догађаји.
- IV Излазни догађаји ИЛИ кола, чији су улаз само примарни догађаји који су већ категорије I.
- V Излазни догађаји ИЛИ кола, чији су улаз само примарни догађаји који још увек нису категорије I, али су понављања истог вишеструког догађаја.
- VI Излазни догађаји ИЛИ кола, чији су улаз само примарни догађаји који још увек нису категорије I и нису понављања истог вишеструког догађаја.

Алгоритам за одређивање минискупова задате дужине се састоји у следећем:

0. Нека је M задата максимална дужина минипресека.
1. У сваком кораку *FATRAM* алгоритма у коме се формирају нови скупови, за сваки нови скуп одредити:
 α = број примарних догађаја категорије I,
 β = број догађаја категорије IV,
$$\delta = \begin{cases} 1 & \text{ако категорија } V \text{ није празна} \\ 0 & \text{ако је категорија } V \text{ празна} \end{cases}$$
2. Израчунати $V = \alpha + \beta + \delta$;
3. Ако је $V > M$ елиминисати скуп, а ако је $V \leq M$ задржати га.

Пример

Нека је за СН са слике 2.4 које је до сада разматрано, задата максимална дужина минипресека $M=1$.

1. У првом кораку добијени су скупови $\{G1\}$ и $\{P1\}$. $V_{G1}=0$ јер је $G1$ II категорије, а $V_{P1}=0$ јер је $P1$ I категорије. Оба скупа се задржавају.
2. Нови скуп је $\{G2,G3\}$. $V_{G2G3}=0$ јер су $G2$ и $G3$ III категорије. Скуп се задржава.

Нови скупови су $\{P2,G3\}$, $\{G4,G3\}$ и $\{P3,G3\}$. $V_{P2G3}=1$ јер је $G3$ III, а $P2$ I категорије. $V_{G4G3}=0$ јер је $G4$ VI, а $G3$ III категорије. $V_{P3G3}=1$ јер је $G3$ III, а $P3$ I категорије. Сви скупови се задржавају.

Нови скупови су: $\{P2,P2\}$, $\{P2,G5\}$, $\{P2,P3\}$, $\{G4,P2\}$, $\{G4,G5\}$, $\{G4,P3\}$, $\{P3,G2\}$, $\{P3,G5\}$, $\{P3,P3\}$.

3. Уклањањем редувантних елемената и надскупова нови скупови су: $\{P2\}$, $\{G4,G5\}$, $\{P3\}$. $V_{P2}=1$ јер је $P2$ I категорије. $V_{G4G5}=0$ јер су $G4$ и $G5$ VI категорије. $V_{P3}=1$ јер је $P3$ I категорије. Сви скупови се задржавају.
 4. Нема нових скупова.
 5. Нови скуп је $\{S1, S2\}$. $V_{S1S2}=2$ јер су $S1$ и $S2$ I категорије. Скуп се одбацује.
-

4. ПЕТРИЈЕВЕ МРЕЖЕ (ПМ)

Петријеве мреже (ПМ, енг. *Petri Nets*^{*}) су графичка и математичка моделујућа техника за опсивање система и њихове динамике. Концепт ПМ је увео Карл Адам Петри (*Carl Adam Petri*), у докторској дисертацији под називом “*Kommunikation mit Automaten*”, 1962. године на Факултету за математику и физику Универзитета *Darmstadt* у Немачкој [13]. Резултати дисертације први пут су објављени у [79], а сама дисертација је објављена на енглеском језику 1966. године у [78]. Петри је дошао до концепта ПМ решавајући проблем израчунавања рекурзивне функције, покушавајући да превазиђе тадашњи проблем резервисања потребног меморијског простора унапред. Развио је модел састављен из секвенцијалних модула који садрже по један елемент и повезани су (комуницирају) само са суседним модулима. Том приликом, Петри је увео и формализам, који подржава нову архитектуру, у коме се први пут појављују нови термини “место” и “прелаз” за означавање локалних стања и акција.

Петријеве мреже, у облику у коме су данас познате, први пут се појављују у излагању “*Fundamentals on the description of discrete processes*” на *The 3rd Colloquium on Automata Theory* у Хановеру 1966. године. Мрежа, која је том приликом представљена, је названа *Condition/Event Net* (CE). Временом је концепт Петријевих мрежа толико развијен да је CE постала једна од најједноставнијих класа ПМ.

До сада је објављено преко 250 књига посвећених ПМ и више од 2000 књига у којима су обрађене у оквиру поглавља. Публиковано је преко 8500 радова посвећених ПМ. Од 1980. године се сваке године одржава “*International Conferences on Application and Theory of Petri Nets and Other Models of Concurrency*” [80]. Међу конференције посвећене ПМ, које се одржавају редовно спадају и: *Workshop and Tutorial on Practical Use of Coloured Petri Nets and the*

* Рус: Сети Петри [124], франц: *Réseaux de Petri* [84], нем: *Petrinetze* [82] .

CPN Tools (од 1998. године), и *QEST* у оквиру кога се од 2004. године одржава *The International Workshop on Petri Nets and Performance Models (PNPM)*. Поред наведених, постоји велики број конференција на којима су Петријеве мреже стална програмска област.

Анализа Петријевих мрежа се ослања на теорију графова, а у њеној практичној примени је неопходна употреба рачунара. Велики број истраживачких група има свој софтверски пакет за цртање ПМ, њихову анализу и симулацију, од којих је више од седамдесет, углавном слободних, доступно на глобалној мрежи [80].

4.1. Основни појмови, дефиниција и динамика ПМ

Два основна елемента ПМ су структура и маркирање. У случају ПМ вишег нивоа, трећи елемент је декларација ПМ, која може обухватати: типове података, временске параметре и функције, функције расподеле, додатне услове паљења прелаза, чак и делове програмског кода који се придружују чворовима на мрежи.

Структура ПМ је оријентисан, тежински, бипартитни граф, који се састоји из две групе чворова:

- Прелаза (*transitions* – *t*) – који представљају догађаје који треба да се одиграју или операције које треба да се изврше. Графички се приказују правоугаонцима или линијама.
- Места (*places* – *p*) – која представљају узрок и/или последицу догађаја или бафер, у коме су смештене информације или други ресурси потребни за извршење операције. У литератури се може наћи и ознака *s* (од немачког *stellen* [41]). Места се графички приказују круговима или елипсама.

У табели 4.1. су приказане неке уобичајене интерпретације прелаза и места у моделима ПМ.

Табела 4.1. Типичне интерпретације прелаза и места [71]

Улазна места	Прелази	Излазна места
Услов	Догађај (акција, логички оператор)	Последица
Улазни подаци	Рачунска операција	Излазни подаци
Улазни сигнал	Процесор	Излазни сигнал
Потребни ресурси	Задатак или посао	Реализовани ресурси
Бафер	Операција	Бафер

Како је ПМ бипартитни граф, оријентисане гране повезују места са прелазима и прелазе са местима, никад два прелаза или два места. Свакој грани може бити придружена тежина којом се директно утиче на динамику ПМ. Уколико недостаје, подразумева се да је тежина гране једнака 1.

Формално се структура ПМ описује на следећи начин:

Дефиниција 4.1 [83]: Структура Петријеве мреже је четворка $N = (P, T, A, W)$, где је:

$$P = \{p_1, p_2, \dots, p_m\} \text{ коначан скуп места,}$$

$$T = \{t_1, t_2, \dots, t_n\} \text{ коначан скуп прелаза,}$$

$$A \subseteq (P \times T) \cup (T \times P) \text{ коначан скуп грана,} \tag{4.1}$$

$$W : A \rightarrow \{1, 2, \dots\} \text{ функција тежина придружена гранама,}$$

где је $P \cap T = \emptyset$ и $P \cup T \neq \emptyset$.

Структура ПМ може бити циклични или ациклични граф, што зависи од структуре моделираног система. Пошто су предмет овог рада ацикличне ПМ, прецизније ПМ које имају структуру стабла, овде ће бити уведени још неки појмови који ће се користити у даљем тексту.

Нека су:

$${}^0t - \text{скуп свих улазних места прелаза } t, \text{ тј. } {}^0t = \{p \mid p \in P, (p, t) \in A\},$$

t^0 – скуп свих излазних места прелаза t , тј. $t^0 = \{p \mid p \in P, (t, p) \in A\}$,

0p – скуп свих улазних прелаза места p , тј. ${}^0p = \{t \mid t \in T, (t, p) \in A\}$,

p^0 – скуп свих излазних прелаза места p , тј. $p^0 = \{t \mid t \in T, (p, t) \in A\}$.

Дефиниција 4.2 [71,83]: p је изворно место (*source place*) ако је ${}^0p = \emptyset$; а место ушћа (*sink place*) ако је $p^0 = \emptyset$.

Дефиниција 4.3 [71,83]: t је изворни прелаз (*source transition*) ако је $t^0 = \emptyset$; а прелаз ушћа (*sink transition*) ако је $t^0 = \emptyset$.

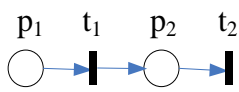
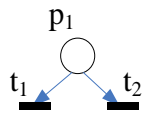
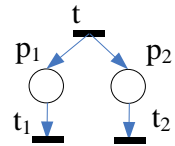
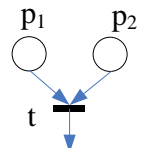
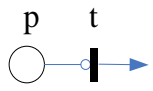
У табели 4.2. су приказани извори и ушћа ПМ и услови за њихово постојање.

Табела 4.2. Места и прелази извора и ушћа [71]

Назив	Услов	Приказ
изворно место	${}^0p = \emptyset$	
место ушћа	$p^0 = \emptyset$	
изворни прелаз	$t^0 = \emptyset$	
прелаз ушћа	$t^0 = \emptyset$	

Да би се могао конструисати ПМ модел, потребно је овладати са релативно мало примитива. У табели 4.3. су приказани основни примитиви [15,108].

Табела 4.3. Примитиви Петријевих мрежа

Примитив	Услов	Приказ
Секвенца	$t_1 \in p_1^0, t_1 \in {}^0 p_2, t_2 \in p_2^0$ $M(p_1) \neq \emptyset$	
Конфликт	$t_1, t_2 \in p^0$ $M(p) \neq \emptyset$	
Истовременост (concurrency)	$p_1, p_2 \in t^0, t_1 \in p_1^0, t_2 \in p_2^0$	
Синхронизација	$p_1, p_2 \in {}^0 t$ $M(p_1) \neq \emptyset, M(p_2) \neq \emptyset$	
Инхибитор	$p \in {}^0 t, M(p) = \emptyset$	

Други елемент ПМ је маркирање. Свако место у ПМ може да садржи један или више жетона (токена, знакова - *token*), који омогућавају моделирање динамике система. Број жетона у месту p се назива маркирање места (*place marking*) и означава са $M(p)$.

Расподела жетона у местима Петријеве мреже се назива маркирање ПМ и означава са M . Једно маркирање ПМ $M = [M(p_1), M(p_2), \dots, M(p_m)]^T$ представља једно стање моделираног система. Почетно маркирање на мрежи означава се са M_0 , $M_0 = [M_0(p_1), M_0(p_2), \dots, M_0(p_m)]^T$, тако да се Петријева мрежа најопштије може приказати као $PN = (N, M_0)$.

Променом маркирања ПМ се моделира промена стања система. Маркирање се мења када се неки од прелаза запали* (*fire, occur*). Да би се одређени прелаз

* У литератури на српском језику се користи и термин активирање [2].

запалио, односно догодио, потребно је да буду испуњени одређени услови, чиме је тај прелаз омогућен (*enabled*).

Прелаз t је омогућен (*enabled*) ако

$$\forall p \in {}^0t, M(p) \geq W(p,t) \quad (4.2)$$

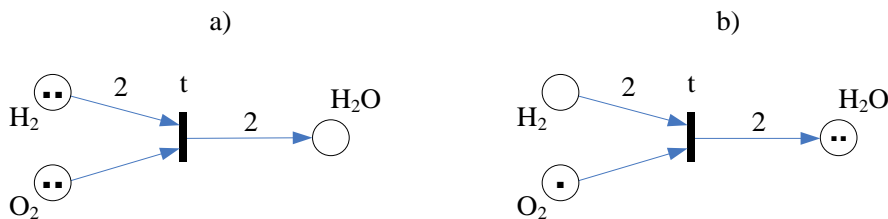
тј. када улазно место садржи најмање онолико жетона колика је тежина гране која води од места p до прелаза t , где је $W(p,t)$ тежина гране (p,t) .

Формално, паљење прелаза је трансформација неког маркирања M у маркирање M' , дефинисана са:

$$M'(p) = \begin{cases} M(p) - W(p,t) & \text{ако } p \in {}^0t \\ M(p) + W(t,p) & \text{ако } p \in t^0 \\ M(p) & \text{иначе} \end{cases} \quad (4.3)$$

Паљењем прелаза t се из свих места $p \in {}^0t$ ослобађају жетони, а у свим местима $p \in t^0$ се појављују жетони. Тиме се добија ново маркирање, односно ново стање моделираног система. Колико ће се жетона ослободити, односно појавити зависи од тежина $W(p,t)$ и $W(t,p)$.

На слици 4.1. је улустрована динамика на примеру ПМ која моделира познату хемијску реакцију: $2\text{H}_2 + \text{O}_2 \rightarrow 2\text{H}_2\text{O}$.



Слика 4.1. Илустрација динамике ПМ [71]

ПМ са слике 4.1. се може написати у облику $PN = (P, T, A, W, M_0)$, где је:

$$P = \{\text{H}_2, \text{O}_2, \text{H}_2\text{O}\},$$

$$T = \{t\},$$

$$A = \{(H_2, t), (O_2, t), (t, H_2O)\},$$

$$W = [2, 1, 2],$$

$$M_0 = [2, 2, 0].$$

По два жетона у улазним местима на слици 4.1. а) показују да су на располагању по две јединице H_2 и O_2 и да је прелаз t омогућен. Односно:

$${}^0t = \{H_2, O_2\}$$

$$\left. \begin{array}{l} M(H_2)=2, M(O_2)=2, M(H_2) \geq W(H_2, t) \\ W(H_2, t)=2, W(O_2, t)=1, M(O_2) \geq W(O_2, t) \end{array} \right\} \Rightarrow \text{ прелаз } t \text{ је омогућен .}$$

Након паљења прелаза t , маркирање се мења на следећи начин:

$$M'(H_2) = M(H_2) - W(H_2, t) = 2 - 2 = 0$$

$$M'(O_2) = M(O_2) - W(O_2, t) = 2 - 1 = 1$$

$$M'(H_2O) = M(H_2O) + W(t, H_2O) = 0 + 2 = 2$$

и постаје као на слици 4.1. б), где прелаз t више није омогућен.

Низ прелаза, којим се из почетног маркирања M_0 достиже неко маркирање M , назива се секвенца паљења σ , и означава се:

$$M_0 \xrightarrow{\sigma} M.$$

Скуп свих маркирања која се могу достићи из неког почетног маркирања M_0 , различитим секвенцама паљења означава се са $R(M)$.

Дефиниција 4.4 [83]: Маркирање M које се може достићи из почетног маркирања M_0 је мртво (*deadlock*) ако у M ниједан од прелаза ПМ није омогућен.

Ако само места ушћа садже жетон, тада нема прелаза чија улазна места садже жетон, тј. да нема омогућених прелаза. Према дефиницији 4.4, маркирање M је мртво. Сада се може формулисати следеће својство:

Својство 4.1: Нека је M маркирање ПМ у коме само места ушћа садрже жетоне. Маркирање M је мртво (*deadlock*).

Петља (*self-loop*) је пар (p, t) такав да $p \in {}^0t$ и $p \in t^0$, тј. p је улазно и излазно место прелаза t . За ПМ се каже да је **чиста** ако нема петљи. Уколико ПМ има петљи, увек се може трансформисати у чисту ПМ, увођењем фиктивних парова прелаза и места [71].

4.2. Врсте ПМ

Петријеве мреже се најчешће класификују у односу на број и структуру жетона које могу да садрже њихова места. По том критеријуму се могу класификовати у три врсте [5]. Прву врсту чине бинарне ПМ, чија маркирања могу да се изразе Буловим вредностима. Друга врста су “црно-беле” (обичне) ПМ, чија маркирања могу да се изразе целобројним вредностима. Трећу групу чине више ПМ (*high-level* ПМ), чија маркирања могу да се изразе мултискуповима структурираних жетона.

Бинарне и елементарне мреже се још називају и Петријеве мреже нижег нивоа (*low-level* ПМ). Црно-беле ПМ имају значајно већу моћ у моделирању од бинарних ПМ, а при томе задржавају могућност за детаљну анализу моделираног процеса. За ове мреже се може рећи да су, по могућности примене, између бинарних и виших Петријевих мрежа које имају изузетну моћ моделирања, али далеко сиромашнији апарат за анализу у односу на ниже ПМ.

4.2.1. Бинарне ПМ

Уколико место у ПМ представља услов, онда присуство, односно одсуство жетона указује да ли је услов испуњен (*true*) или не (*false*). Ови жетони се још називају буловски жетони, зато што њихов број у месту може бити 0 или 1. Буловски жетони се графички представљају (црном) тачком у месту. Мрежа која садржи само места са буловским жетонима, се назива ПМ нивоа 1 или бинарна ПМ.

Услов/Последица мреже

Услов/Последица мрежа (*Condition/Event*, C/E) је мрежа коју је представио Петри у својој докторској дисертацији, односно то је прва ПМ. C/E је мрежа нивоа која захтева да ПМ структура буде чиста и може се формално представити на следећи начин [5]:

Дефиниција 4.5: C/E мрежа је четворка (P, T, A, C) , где је

$P = \{p_1, p_2, \dots, p_n\}$ коначан скуп места,

$T = \{t_1, t_2, \dots, t_n\}$ коначан скуп прелаза,

$A \subseteq (P \times T) \cup (T \times P)$ коначан скуп грана,

$C = \{M_0, M_1, M_2, \dots\}$ скуп свих маркирања која нису мртва.

Правило паљења се у C/E мрежама може описати на следећи начин: сваки прелаз t C/E мреже може да се упали у неком маркирању M ако:

$$(\forall t \in T) (\exists M \in C) (M(t) = 1 \wedge M(t') = 0)$$

односно, ако у свим улазним местима прелаза t постоји по један жетон и ако су сва излазна места прелаза t празна. Када се прелаз t запали, из његових улазних места се ослобађа по један жетон, а у његовим излазним местима се производи по један жетон. Важно својство C/E мрежа је да свако маркирање може да се достигне из било ког маркирања у коначном броју корака, унапред или уназад.

Елементарне мреже

Елементарне мреже су C/E мреже са задатим почетним маркирањем.

Дефиниција 4.6: EN мрежа је четворка (P, T, A, M_0) , где је

$P = \{p_1, p_2, \dots, p_n\}$ коначан скуп места,

$T = \{t_1, t_2, \dots, t_n\}$ коначан скуп прелаза,

$A \subseteq (P \times T) \cup (T \times P)$ коначан скуп грана,

M_0 почетно маркирање.

1-Safe мреже

Ове мреже припадају нивоу 1, али се често посматрају и као елементарне (црно-беле) ПМ са буловским маркирањем. За маркирање ПМ се каже да је *1-safe* ако за свако место p ПМ важи да је $M(p) \leq 1$. ПМ је *1-safe* ако су сва њена маркирања *1-safe* [20]

Дефиниција 4.7: *1-Safe* мрежа је петорка (P, T, A, W, M_0) , где је

$$\begin{aligned} P &= \{p_1, p_2, \dots, p_n\} \text{ коначан скуп места,} \\ T &= \{t_1, t_2, \dots, t_m\} \text{ коначан скуп прелаза,} \\ A &\subseteq (P \times T) \cup (T \times P) \text{ коначан скуп грана,} \\ W &: A \rightarrow \{1\} \text{ функција тежина придружена гранама,} \\ M_0 &: P \rightarrow \{0,1\} \text{ почетно маркирање.} \end{aligned} \tag{4.4}$$

State machine

State machine је подкласа *1-Safe* мрежа, у којој сваки прелаз t има тачно једно улазно и тачно једно излазно место. *State machine* се може формално описати као петорка (P, T, A, W, M_0) , где уз (4.4) важи [71]:

$$|{}^0t| = |t^0| = 1, \forall t \in T$$

где је $|{}^0t|$ ($|t^0|$) број улазних (излазних) места прелаза t .

Marked graph (MG)

MG је чиста проста ПМ, таква да свако место p има тачно један улазни и тачно један излазни прелаз. *Marked graph* се може формално описати као петорка (P, T, A, W, M_0) , где уз (4.4) важи [71]:

$$|{}^0p| = |p^0| = 1, \forall p \in P$$

где је $|{}^0p|$ ($|p^0|$) број улазних (излазних) прелаза места p .

4.2.2. Црно-беле ПМ

Ако је местом ПМ моделиран бафер, онда број жетона у месту указује на број, тј. количину информација или другог ресурса у баферу. ПМ чија места могу да садрже више од једног жетона се назива мрежа 2. нивоа, елементарна, обична или “црно-бела” ПМ. Друго важно својство ове мреже је да су жетони у њој неструктурирани и да су сви графички представљени (црним) тачкама.

Place/Transition (P/T) мреже

Place/Transition мрежа је елементарна (црно-бела) ПМ дефинисана са (4.1). У развоју Петријевих мрежа, ове мреже су биле прве које су дозвољавале да место садржи више од једног жетона.

Проста (Ordinary) ПМ

Проста ПМ је елементарна ПМ у којој је тежина свих грана једнака 1.

Дефиниција 4.8: Проста ПМ је петорка (P, T, A, W, M_0) , где је

$P = \{p_1, p_2, \dots, p_n\}$ коначан скуп места,

$T = \{t_1, t_2, \dots, t_q\}$ коначан скуп прелаза,

$A \subseteq (P \times T) \cup (T \times P)$ коначан скуп грана,

$W : A \rightarrow \{1\}$ функција тежина придружена гранама,

$M_0 : P \rightarrow \{0, 1, 2, \dots\}$ почетно маркирање.

Free Choice (FC) мреже

FC мрежа је проста ПМ таква да је свака грана из неког места p или јединствена излазна грана места p или јединствена улазна грана неког прелаза t [5]:

$$\forall p \in P, |p^0| > 1 \Rightarrow {}^0(p^0) = \{p\}$$

или, еквивалентно $\forall p_1, p_2 \in P, p_1^0 \cap p_2^0 \neq \emptyset \Rightarrow |p_1^0| = |p_2^0| = 1$

или, еквивалентно $\forall p \in P, \forall t \in T, (p, t) \in A \Rightarrow p^0 = \{t\} \vee \{p\} = {}^0t$

где је $|{}^0p|$ ($|p^0|$) број улазних (излазних) прелаза места p .

Дефиниција 4.9 [10]: ПМ $PM=(P,T,A,M0)$ је *free-choice* ако и само ако за свако место p важи или $|p^0| < 2$ или ${}^0(p^0) = \{p\}$.

***S*-мреже**

S-мрежа је подкласа FC мреже у којој сваки прелаз t има највише једно улазно и једно излазно место, тј.:

$$\forall t \in T, |{}^0t| \leq 1 \wedge |t^0| \leq 1$$

***T*-мреже**

T-мрежа је подкласа FC мреже у којој свако место p има највише један улазни и један излазни прелаз, тј.:

$$\forall p \in P, |{}^0p| \leq 1 \wedge |p^0| \leq 1$$

4.2.3. Петријеве мреже вишег нивоа

У раду са нижим Петријевим мрежама, основни проблем су велике димензије модела реаланих система. Да би се овај проблем превазишао, уведене су више Петријеве мреже у којима се користе структурирани жетони и алгебарски изрази. За разлику од ПМ, које су дефинисане на различите начине, за више ПМ постоје интернационални стандарди за синтаксу, семантику и графичку презентацију [44]. Међутим, још увек нису обухваћени и разрешени сви елементи, као што су својства понашања и структурна својства мрежа, чак ни ознаке места (p од енглеског *place* и италијанског *posti* или s од немачког *stellen*).

У ВПМ спадају: ПМ са апстрактним типовима података - алгебарске ПМ где су жетони и правила преласка из једног маркирања у друго дефинисани помоћу алгебарских израза; *Environment Relationship* (ER) мреже, у којима жетони представљају функције које пресликавају скуп променљивих у скуп вредности; *Product* (Prod) мреже, које садрже две специјалне врсте грана: инхибитор гране и бришуће гране; традиционалне ВПМ, које чине *Predicate/Transition* мреже (PrTN) и Обојене ПМ итд. Последње две врсте мрежа су веома сличне. PrTN је прва виша ПМ која је прво теоретски уведена, а њена (широка) примена је објављивана у каснијим радовима. Овде ће детаљније бити описане само Обојене ПМ.

Обојене Петријеве мреже

Прву верзију обојених ПМ (*Coloured Petri Nets*, ОПМ), названу CP⁸¹-мрежа, дефинисао је 1981. Године Курт Јенсен (*Kurt Jensen*) у раду “*Coloured Petri Nets and the Invariant Method*”, а прва свеобухватна дефиниција је објављена 1986. године у раду “*Coloured Petri Nets*” истог аутора [46,47]. Основна одлика обојених ПМ је да је сваком чвору типа место придружен тип податка, који одређује врсту податка које дато место може да садржи. Сваки жетон у месту носи вредност податка, која припада типу тог места. Ова вредност податка се назива и боја жетона, а тип податка – скуп боја.

Како место p у ОПМ може да садржи два или више жетона са истом вредношћу, садржај места се не може представити као скуп, већ као мултикуп (*multi-set*) жетона [138].

Дефиниција 4.10: Мултикуп $M = (S, m)$, над непразним скупом S , где је функција $m: S \rightarrow \mathbb{N}$ се представља као формална сума $\sum_{s \in S} m(s) \cdot s$.

Са S_{MS} означава се скуп свих мултикупова над S , где је MS скраћеница за мултикуп (*multi-set*). Коефицијенти мултикупа су ненегативне целобројне вредности $\{m(s) | s \in S\}$. $s \in m$ акко $m(s) \neq 0$. Другим речима, $m(s)$ представља број појављивања елемента $s \in S$ у мултикупу M . Ако се у неком мултикупу над скупом S сваки елемент из S појављује само по једном ($1 \cdot s, s \in S$), онда се

мултискуп може представити и самим скупом S . Мултискупови поседују велики број алгебарских својстава и над њима се могу извршити бројне стандардне операције: сабирање, одузимање, множење скаларом итд.

Маркирање у обојеним ПМ може да се представи као функција која сваком месту p придружује мултискуп жетона одговарајућег типа. Овај тип податка обухвата све жељене атрибуте моделираног објекта, а може, ако је потребно да буде и неструктуриран.

Ознаке које се односе на прелазе и гране обојене ПМ су:

- Тип променљиве v , који се означава са $Type(v)$;
- Тип описа $expr$, који се означава са $Type(expr)$;
- Скуп променљивих у опису гране $expr$, који се означава са $Var(expr)$;
- Повезивање (*binding*) скупа променљивих V , који свакој променљивој $v \in V$ придружује елемент $b(v)$ типа $Type(v)$. Овим се формира повезујући елемент за који посматрани прелаз t може да се догоди;
- Вредност добијена евалуацијом описа гране ($expr$), која се означава са $expr \langle b \rangle$. За $Var(expr)$ се захтева да буде истог типа као b , а евалуација се врши замењивањем сваке променљиве $v \in V(expr)$ вредношћу $b(v)$ типа $Type(v)$, која је одређена повезивањем. Евалуација се врши за све улазне гране посматраног прелаз t .

Прелаз t могу садржати израз, тј. функцију чији је резултат буловска променљива. Ова функција се назива чување (*guard*). Увођењем ове функције чувања, обезбеђује се да одређени жетон буде омогућен само ако је испуњен услов паљења прелаз t , односно ако је вредност функције чувања *true*. Буловски тип променљиве ($\{true, false\}$) означава се са **B**.

Сада је могуће формално дефинисати ОПМ.

Дефиниција 4.11 [46]: Обојена ПМ је деветорка $CPN = (\Sigma, P, T, A, N, C, G, E, M_0)$

где је

- (i) Σ коначан скуп типова (боја),

- (ii) $P = \{p_1, p_2, \dots, p_n\}$ коначан скуп места,
- (iii) $T = \{t_1, t_2, \dots, t_n\}$ коначан скуп прелаза,
- (iv) $A \subseteq (P \times T) \cup (T \times P)$ коначан скуп грана
- (v) $N: A \rightarrow (P \times T) \cup (T \times P)$ функција чворова,
- (vi) $C: P \rightarrow \Sigma$ функција типова (боја),
- (vii) G функција чувања, која пресликава T у израз такав да:

$$\forall t \in T: [\text{Тур}\{\mathbf{G}(t)\} = \mathbf{B} \wedge \text{Тур}\{\text{Var}(\mathbf{G}(t))\} \subseteq \Sigma],$$
- (viii) E опис гране, који пресликава A у израз такав да:

$$\forall a \in A: [\text{Тур}\{\mathbf{E}(a)\} = C(p)_{MS} \wedge \text{Тур}\{\text{Var}(\mathbf{E}(a))\} \subseteq \Sigma],$$
 где је p место у $N(a)$,
- (ix) M_0 почетна функција, која пресликава P у затворени израз такав да:

$$\forall p \in P: [\text{Тур}\{\mathbf{M}_0(p)\} = C(p)_{MS}].$$

Израз (i) дефинише типове, тј. боје садржане у мрежи. Изразима (ii -v) је описана структура мреже. Изразом (vi) је описано да се сваком месту придружује тип жетона који могу да се налазе у том месту. Са (vii) је дефинисана буловска функција чувања којом се задају додатни услови паљења прелаза. Изразом (viii) је дефинисано да у опису грана морају да буду садржане променљиве истог типа који је додељен улазном, односно излазном месту гране. На крају, као и у свим врстама ПМ, (ix) представља почетно маркирање на мрежи.

Петријева мрежа којом се моделира конкретан процес се углавном не описује на формалан начин, већ се директно приказује графом и описима на графу. Када се ОПМ приказује графички, она се састоји из 3 дела:

- Структура мреже, односно оријентисан, бипартитни граф дефинисан са (ii -v) из дефиниције 4.11;
- Опис мреже или чвор описа мреже (*decsription node*), који се графички приказује као правоугаоник у коме се налази опис променљивих, константи и функција који се користе у мрежи;
- Ознаке на мрежи: типови места, описи грана, услови паљења прелаза и почетно маркирање на мрежи.

4.3. Проширења ПМ

Проширења ПМ су последица потреба моделирања конкретних зависности или особина процеса. Тако су нпр. за потребе моделирања слободних ресурса уведене ПМ са инхибиторским гранама [83]. Инхибиторске гране моделирају услов да њихов улазни прелаз може да се запали само када је излазно место празно. Због неодређености многих информација о моделираним процесима, уведене су фази ПМ [17,113]. Ово проширење ПМ увођењем елемената фази логике се може реализовати на више начина: фази маркирања која месту придружују фази број као број жетона или функцију припадности неком месту; фази правила паљења прелаза, итд. Да би се моделирале методе одлучивања уведене су објектне ПМ [40]. У овим мрежама се жетонима моделирају методе одлучивања које представљају жетоне веће системске мреже, у којој су места доносиоци одлуке или конкретни проблеми којима се придружују методе решавања проблема. Листа могућих проширења ПМ је велика. Овде су детаљније описане хијерархијске, временске и стохастичке ПМ.

Хијерархијске ПМ

Сврха увођења хијерархијских мрежа је разлагање ПМ модела на већи број мањих, који се могу независно анализирати. Овакво хијерархијско моделирање омогућава декомпозицију процеса на различитим нивоима детаљности. Начин увођења хијерархијских мрежа је такав да се хијерархијска ПМ увек може превести у одговарајућу нехијерархијску, и обрнуто. Хијерархијска и њој еквивалентна нехијерархијска мрежа имају иста својства и исто понашање.

Хијерархијска ОПМ се формира заменом прелаза подмрежом. Основна идеја замене је да се прелазу и гранама које га окружују додели мрежа која детаљније описује активност коју прелаз представља. Посматрани прелаз се назива надчвор. Мрежа која садржи надчвор се назива надмрежа, а мрежа којом се прелаз описује – подмрежа. Сваки пут када је прелаз омогућен, покреће се једно извршавање (инстанца) придружене подмреже. Свако извршавање креће од другачијег почетног маркирања, зато што је оно увек промењено претходним извршавањем.

Временске ПМ

У временским ПМ је местима и/или прелазима придружено одређено трајање. Ако је трајање придружено месту p , оно се односи на време које жетон, од момента пристизања у место p , треба да проведе у том месту [83]. У литератури је далеко више разматрано трајање које се придружује прелазима. Разлог за то је што места углавном представљају услове или стања, а прелази представљају операције или акције које имају одређено трајање, односно кашњење.

У временској ПМ прелаз се може запалити када је омогућен и спреман (*ready*). Прелаз је омогућен, као и у случају невременске ПМ, када се у његовим улазним местима налази бар онолико жетона колико је задато описом одговарајуће гране. Прелаз је спреман када је временска вредност жетона у улазним местима већа или једнака тренутном времену глобалног сата [47]. Нека је паљењу прелаза t придружено трајање τ , и нека паљење t почиње у тренутку T_0 . Онда се паљење прелаза t састоји у: уклањању жетона из свих места $p \in {}^0t$ у тренутку T_0 , и додавању жетона у сва места $p \in t^0$ у тренутку $T_0 + \tau$.

Број и врста жетона који се уклањају, односно додају зависи од описа грана које повезују прелаз са местима. Претпоставља се да између тренутака T_0 и $T_0 + \tau$ жетони остају у прелазу t . Ако у ПМ постоји бар један прелаз са трајањем паљења или бар једно место у коме се жетон задржава одређено време, ПМ је временска. Уобичајено је, ако у ПМ постоје оба типа прелаза (временски и тренутни), да се прелази са трајањем графички приказују правоугаоником, а тренутни линијом.

Стохастичке ПМ

Трајање придружено паљењу прелаза може бити детерминистичко и стохастичко. Уколико у ПМ постоји бар један прелаз, чије је трајање паљења стохастичко, мрежа се назива стохастичком ПМ (СПМ). Иако је СПМ временска мрежа, постоји велики број аутора који је посматрају као посебну класу ПМ. У СПМ је трајање паљења прелаза случајна величина која подлеже некој расподели, а време потребно за паљење датог прелаза је различито за свако ново паљење прелаза. СПМ имају велику примену у моделирању променљивог трајања операција, што

је веома често у системима у којима операције обавља човек или када на трајање операција утичу неки спољни догађаји. Ако у мрежи постоје и тренутни и прелази са стохастичким трајањем, мрежа се назива генерализована стохастичка ПМ (ГСПМ). У [69] је показано како је k -ограничена ПМ, са експоненцијалном расподелом паљења прелаза, изоморфна са коначним Марковљевим ланцем. У [63] је дата веза између ГСПМ и Марковљевих ланаца и алгоритам за реализацију Марковљевих процеса, заснован на ПМ.

4.4. Својства ПМ

Основна снага ПМ је у могућности анализирања различитих својстава моделираног система или процеса на основу својстава саме мреже. Својства ПМ се могу поделити у две групе: структурна својства, која зависе само од структуре ПМ и независна су од почетног маркирања и својства понашања (*behavioural properties, marking-dependent*), која зависе од структуре ПМ и од почетног маркирања. Овде ће бити описана најважнија својства понашања зато што су она значајна за тему овог рада.

Достижност

Проблем достижности (*reachability*) се састоји у испитивању да ли систем из тренутног стања може достићи неко жељено или нежељено стање. Када се систем моделира помоћу ПМ, ово испитивање се своди на анализу да ли се жељено или нежељено маркирање може достићи из почетног маркирања M_0 .

Дефиниција 4.12 [81]: Дата је $PN = (N, M_0, M_f)$. Проблем достижности састоји се у испитивању да ли $M_f \in R(M_0)$.

Сложеност проблема достижности зависи од врсте ПМ. За *1-safe* ПМ, проблем достижности спада у PSPACE-комплетне проблеме, за ацикличне и *1-safe* ацикличне ПМ у NP-комплетне, за *Free Choice* у EXPSpace-тешке а за *1-safe Free Choice* ПМ у PSPACE-комплетне проблеме [20]. PSPACE проблеми су

проблеми код којих рачунарски простор, при решавању Тјуринговом машином, полиномијално расте са димензијама. За EXPSPACE-тешке проблеме, решавање заузима $O(2^{p(n)})$ простора, односно потребан рачунарски простор експоненцијално расте са димензијама проблема.

У овом раду ће се посматрати још један облик проблема достижности: одредити $R(M_0)$, скуп свих маркирања достижних из почетног маркирања M_0 . У решавању неких проблема (као што ће овде бити случај) је довољно одредити неки подскуп скупа $R(M_0)$, а не читав скуп, што је лакши проблем.

Живост

У теорији ПМ се, за испитивање да ли систем може да достигне стање у којем је блокиран, користи концепт живости (*liveness*). Прелаз t у $PN=(N, M_0)$ је жив ако важи $\forall M \in R(M_0) \exists M' \in R(M)$ такво да је t омогућен, тј. t може бити запаљен из сваког маркирања $M \in R(M_0)$. ПМ је жива ако су јој сви прелази живи [83].

Дефиниција 4.12 [20]: Проблем живости састоји се у испитивању да ли је $PN=(N, M_0, M_f)$ жива.

Сложеност проблема живости такође зависи од врсте ПМ. За 1-safe ПМ он спада у PSPACE-комплетне проблеме, за ацикличне ПМ је време његовог решавања линеарно а за 1-safe ацикличне константно, за *Free Choice* спада у NP-комплетне проблеме а за 1-safe *Free Choice* ПМ време решавања је полиномијално [20].

Мртво маркирање

Мртво маркирање (*deadlock*) је већ дефинисано (Дефиниција 4.4) као маркирање у коме ни један од прелаза ПМ није омогућен. Проблем мртвог маркирања се може дефинисати на следећи начин:

Дефиниција 4.13: Дата је $PN=(N, M_0)$. Проблем мртвог маркирања је одређивање да ли је неко од маркирања $M \in R(M_0)$ мртво.

Сложеност овог проблема је иста као проблема живости за све поменуте врсте ПМ, осим за *1-safe Free Choice* ПМ, за које проблем проверавања мртвог маркирања спада у NP-комплетне проблеме [20]. Проблем који ће се разматрати у овом раду је одређивање свих мртвих маркирања достижних из датог M_0 .

Повратност и почетно стање

У функционисању система потребно је, када дође до грешке у раду или из неког другог разлога, вратити систем у почетно стање или у неко друго изабрано стање, из кога он даље наставља са радом. Могућност да се системом управља на овај начин се у ПМ испитује помоћу својства повратности и изворног стања (*Reversibility* и *Home State*).

ПМ је повратна ако је могуће вратити се у почетно маркирање из неког маркирања достижног из почетног, тј. $M_0 \in R(M) \forall M \in R(M_0)$. Маркирање M_a у ПМ са почетним маркирањем M_0 , је изворно стање ако је достижно из било ког маркирања достижног из M_0 , тј. $M_a \in R(M) \forall M \in R(M_0)$ [83].

4.5. Анализа ПМ

Петријеве мреже, нарочито ПМ нижег нивоа, представљају моћан алат за анализу многих својстава система. Код виших ПМ се многе од метода за анализу не могу уопште или могу применити веома ограничено. У овом поглављу ће прво бити описана редукција ПМ и једначина стања, а затим технике за анализу ПМ: инваријанте, стабло достижности и симулација.

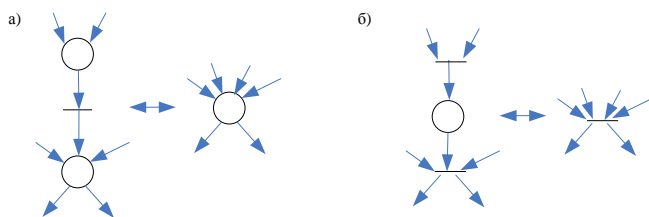
4.5.1. Редукција ПМ

Један од проблема који се јавља у раду са ПМ је величина модела. У неким случајевима је могуће цео ПМ модел декомпоновати на модуле који се могу одвојено анализирати. Редукционе методе се, међутим, користе локално тј. редукују се поједини делови ПМ модела. Редукција је процедура која

трансформише ПМ у редуковане мреже или подмреже које чувају жељена својства оригиналне мреже.

У литератури се могу наћи различите технике и класификације техника редукције невременских [56,110,71] и временских ПМ [101]. С обзиром на структуру ПМ које ће се проучавати у наставку, овде ће бити приказана само два правила редукције која се на њима могу применити: фузија прелаза и фузија места.

Нека је $X = P \cup T$ скуп свих места и прелаза у ПМ, која се сада може представити као четворка: $PN = (X, A, W, M_0)$. Нека је $D(x)$ степен чвора x , $x \in X$. Улазни степен чвора једнак је броју грана које улазе у тај чвор, а излазни степен је једнак броју грана које из чвора излазе [138]. Чвор x се може редуковати ако је и улазни и излазни степен чвора једнак јединици. На слици 4.2. су приказане фузија два места у јединствено место а) и фузија два прелаза у јединствени прелаз б).



Слика 4.2. Фузија места а) и прелаза б)

Фузија прелаза и фузија места чувају својства: живости, мртвог маркирања и повратности.

4.5.2. Матрица инциденције и једначина стања

Структура ПМ се може описати помоћу матрице инциденције. Дефиниција матрице инциденције је спој дефиниције матрице инциденције бипартитног графа и дефиниције тежинске матрице (матрице растојања) [140].

Дефиниција 4.14 [83]: Матрица инциденције $A = [a_{ij}]$, $i = 1, \dots, n$, $j = 1, \dots, m$ чисте (без петљи) $PN = (N, M_0)$ је дефинисана на следећи начин:

$$a_{ij} = \begin{cases} W(t_j, p_i) & \text{ако } t_j \in {}^0 p_i \\ -W(t_j, p_i) & \text{ако } t_j \in p_i^0 \\ 0 & \text{иначе} \end{cases} \quad (4.5)$$

где је n број места, m број прелаза у ПМ, а $W(t_j, p_i)$ тежина гране (t_j, p_i) .

За чисту бинарну $PN = (N, M_0)$ (4.5) постаје:

$$a_{ij} = \begin{cases} 1 & \text{ако } t_j \in {}^0 p_i \\ -1 & \text{ако } t_j \in p_i^0 \\ 0 & \text{иначе} \end{cases}$$

Захтев да ПМ буде чиста је зато што елемент који одговара месту p и прелазу t петље (p, t) може бити 1 и -1 истовремено, а сви елементи матрице инциденције треба да буду јединствени.

Матрица инциденције се може дефинисати и помоћу матрица: $A^+ = [a_{ij}^+]$ и $A^- = [-a_{ij}^-]$, где је

$$a_{ij}^+ = \max(0, a_{ij}) \quad \text{и} \quad a_{ij}^- = \min(0, a_{ij})$$

при чему важи $A = A^+ - A^-$.

Лако се уочава да је a_{ij}^+ број жетона који се додају месту p_i , односно a_{ij}^- број жетона који се уклања из места p_i , после једног паљења прелаза t_j .

Прелаз t_j је омогућен у маркирању M ако [71]:

$$-a_{ij} \leq M(p_i), \quad i = 1, \dots, n$$

Матрица инциденције се може користити за формирање једначине стања посматраног процеса. Ова једначина омогућава да се аналитички одреди маркирање, које се достиже када се из неког почетног маркирања M_0 запали неки дати скуп прелаза. Пре дефинисања једначине стања је потребно увести неке појмове.

Нека је дато M_0 почетно маркирање и σ допуштена секвенца паљења прелаза у ПМ. Вектор паљења (*firing count*) је вектор $V_\sigma = [v_1, \dots, v_m]$, где је v_j број појављивања (догађања) прелаза t_j у секвенци σ , односно вредност која показује колико се пута прелаз t_j запалио. Вектор V_σ се у литератури назива и карактеристични вектор секвенце паљења σ [54].

Дефиниција 4.15 [83]: Нека је M маркирање које се добија из почетног маркирања M_0 , паљењем секвенце σ . Једначина стања којом се описује трансформација из M_0 у M има следећи облик:

$$M^T = M_0^T + A \cdot V_\sigma^T \quad (4.6)$$

или
$$A \cdot V_\sigma^T = M^T - M_0^T = \Delta M \quad (4.7)$$

Потребно је нагласити да маркирање, које се одреди помоћу једначине стања (4.6), не може да се достигне ако секвенца паљења σ није допуштена.

Напомена 4.1: Вектор паљења означава само који су се прелази запалили приликом трансформације из M_0 у M , а не и којим редоследом. То значи да ће све пермутације секвенце паљења σ имати исти вектор паљења. Ова важна особина ће се користити у даљем раду.

Ако је потребно одредити маркирање које се добија из почетног маркирања M_0 паљењем само једног прелаза t_j , вектор паљења V_σ се састоји из $m-1$ нула и 1 на позицији j .

4.5.3. Инваријанте

У теорији ПМ, инваријанта је својство које остаје непромењено за сва достижна маркирања ПМ [94]. Анализа инваријанти је структурни приступ у испитивању својстава ПМ. Могу се анализирати p -инваријанта која се односи на ограниченост ПМ и t -инваријанта којом се испитује живост ПМ. Између њих постоји дуалност, односно свака t -инваријанта ПМ је p -инваријанта дуалне ПМ која се добија транспоновањем матрице инциденције дате ПМ [120].

p-инваријанта

Вредност која у сваком стању система остаје инваријантна након паљења прелаза у ПМ представља p -инваријанту [96]. Вектор Z , n ненегативних целих бројева је p -инваријанта ако

$$Z \cdot A = 0 \quad (4.8)$$

где је A матрица инциденције ПМ, а n број места у ПМ.

Својство 4.1: Нека је M_0 почетно маркирање мреже $PN=(N, M_0)$ и $M \in R(M_0)$. Онда је

$$Z \cdot M_0^T = Z \cdot M^T \quad (4.9)$$

где је Z p -инваријанта.

Доказ: Множење једначине стања (4.6) са леве стране са A , резултира у:

$$Z \cdot M^T = Z \cdot M_0^T + Z \cdot A \cdot V_\sigma^T \quad (4.10)$$

Заменом (4.8) у (4.10) добија се (4.9). ◆

Обрнути исказ својства 4.1 не важи, тј. може постојати M такво да важи (4.9), али да $M \notin R(M_0)$.

Ако су све компоненте p -инваријанте Z строго позитивне, онда је укупан број жетона у ПМ ограничен. Ако су, уз то, све компоненте једнаке 1, онда укупан број жетона у ПМ остаје константан након било које допустиве секвенце паљења.

Скуп свих места која одговарају строго позитивним компонентама p -инваријанте Z се назива носач (*support*) p -инваријанте и означава са $|Z|$. Носач p -инваријанте је минималан ако не садржи носаче других p -инваријанти. За p -инваријанту се каже да је минимална ако не постоји друга p -инваријанта Z_1 таква да $Z_1(p_i) \leq Z(p_i), \forall i = 1, \dots, n$. Скуп свих минималних p -инваријанти је база за генерисање p -инваријанти, тј. свака p -инваријанта је линеарна комбинација минималних p -инваријанти.

t-инваријанта

У контексту теорије ПМ, *t*-инваријанта се обично односи на вектор паљења који враћа систем у почетно стање [96]. Вектор W , m ненегативних целих бројева је *t*-инваријанта ако

$$A \cdot W^T = 0 \quad (4.11)$$

где је A матрица инциденције ПМ, а m број прелаза у ПМ.

Својство 4.2: Нека је σ допустива секвенца паљења, а V_σ одговарајући вектор паљења. Ако је V_σ *t*-инваријанта, M_0 почетно маркирање мреже $PN=(N, M_0)$, а M маркирање добијено стартовањем σ из M_0 , тада је $M=M_0$.

Доказ следи из једначине стања (4.6). Како је V_σ *t*-инваријанта и важи (4.11), следи да је $A \cdot V_\sigma^T = 0$, односно из једначине стања, $M^T = M_0^T$. ♦

Ово својство значи да, ако постоје допустиве секвенце паљења $\sigma_1, \sigma_2, \dots, \sigma_k$, и ако су $V_{\sigma_1}, V_{\sigma_2}, \dots, V_{\sigma_k}$ *t*-инваријанте, стартовањем било које од ових секвенци, једном или више пута, ПМ се враћа у почетно маркирање, односно $M_0 \xrightarrow{\sigma_l} M_0, l=1, \dots, k$. Обрнути исказ својства 4.2 не важи, тј. може постојати σ такво да је V_σ *t*-инваријанта, али да секвенца паљења σ није допустива.

Ако је у свим *t*-инваријантама иста компонента која одговара прелазу t_j једнака нули, онда је немогуће вратити се у почетно маркирање након стартовања прелаза t_j . У реалном систему, то значи да је немогуће вратити се у почетно стање након извођења операције која одговара нула компоненти *t*-инваријанте. Скуп свих прелаза који одговарају строго позитивним компонентама *t*-инваријанте W се назива носач (*support*) *t*-инваријанте и означава са $|W|$. Носач *t*-инваријанте је минималан ако не садржи носаче других *t*-инваријанти. За *t*-инваријанту се каже да је минимална ако не постоји друга *t*-инваријанта W_1 таква да $W_1(t_j) \leq W(t_j), \forall j=1, \dots, q$. Скуп свих минималних *t*-инваријанти је база за генерисање *t*-инваријанти, тј. свака *t*-инваријанта је линеарна комбинација минималних *t*-инваријанти.

4.5.4. Граф достижности

Испитивање својстава ПМ преко графа достижности (*Reachability graph*) подразумева пребројавање свих достижних маркирања ПМ. Анализа помоћу графа достижности може да се примени на све класе ПМ. У случају цикличних ПМ, иста маркирања и исте секвенце паљења се могу појавити више пута, па се за краћи приказ свих достижних маркирања користити и граф покривања (*Coverability graph*). У вишим ПМ се чешће може наћи термин потпуни граф догађања, О-граф (*Full Occurrence Graph, O-graph*) који садржи сва достижна маркирања у ПМ, граф догађања са класама еквиваленције, ОЕ-граф (*Occurrence Graph with Equivalence Classes, OE-graph*) који садржи класе еквивалентних маркирања и граф догађања са симетријом, ОС-граф (*Occurrence Graph with Symetries, OS-graph*) који се може формирати само за више ПМ у којима постоје симетрична маркирања.

Пошто су овде предмет проучавања ацикличне ПМ, у наставку ће бити описан граф достижности.

Нека је дата $PN=(N,M_0)$ и маркирање M . Као што је већ поменуто, проблем достижности састоји се у проверавању да ли постоји секвенца паљења σ која води у M почевши од M_0 , тј.

$$M_0 \xrightarrow{\sigma} M$$

односно $M \in R(M_0)$.

За дату $PN=(N,M_0)$, из почетног маркирања M_0 се може добити онолико нових маркирања колико је омогућених прелаза. Из новодобијених маркирања се, паљењем омогућених прелаза добијају нова маркирања, итд. Сва достижна маркирања представљају се графом достижности. Граф достижности почиње са почетним маркирањем (корен стабла). Свако маркирање на мрежи се представља једним чвором стабла достижности. Грана стабла достижности, која повезује два чвора, представља прелаз чијим се паљењем прелази из улазног чвора (маркирања ПМ) у излазни чвор графа достижности (маркирање ПМ).

Формално се граф достижности може описати на следећи начин [33]:

Дефиниција 4.16: Нека је дата $PN = (N, M_0)$. Граф достижности дате ПМ је усмерени граф $RG(PN) = (V, E)$, где су је $V = R(M_0)$ скуп чворова и $E = \{ \langle M, t, M' \rangle \mid M, M' \in R(M_0) \wedge M \xrightarrow{t} M' \}$ скуп грана.

Конструкција графа достижности је веома тежак проблем, са рачунарске стране, јер простор стања (достижних маркирања) може експоненцијално да расте са димензијама ПМ. Због тога је развијен велики број алгоритама којим се покушава да се превазиђе комбинаторна експлозија. У наставку је приказан општи алгоритам конструкције графа достижности [33].

Алгоритам 4.1 Конструкција графа достижности

Улаз: $PN = (N, M_0)$

Излаз: $RG(PN) = (V, E)$

1. Иницијализација: $RG(PN) = (\{M_0\}, \emptyset)$, M_0 је необележен;
2. Док има необележених чворова у V ради:
 - 2.1. изабрати необележени чвор $M \in V$ и обележити га
 - 2.2 за сваки омогућени прелаз t у M израчунати све M' такве да $M \xrightarrow{t} M'$;
 - 2.3 $V = V \cup \{M'\}$, M' је неозначени чвор;
 - 2.4 $E = E \cup \{ \langle M, t, M' \rangle \}$
3. Крај алгоритма. $RG(PN) = (V, E)$ је граф достижности.

4.5.5. Симулација ПМ

Већина ПМ је намењена испитивању логичке исправности система и процеса, односно њихових динамичких својстава и функционалности. Међутим, ПМ, нарочито оне вишег реда, се могу користити и за испитивање перформанси као што су: време потребно за извршење циклуса, максимално време извршења неке активности, просечно време чекања на испуњење неког услова, просечан број

објекта који чекају да се над њима изврши нека активност итд. За овакву анализу користи се симулација. Како трајања активности углавном нису константна већ подлежу некој расподели, најбоље је симулирати ГСПМ.

На основу резултата симулације, мере се вредности статистичких променљивих које представљају перформансе које је потребно испитати. Ове статистичке променљиве могу бити невременске (*untimed*) и временске (*timed*) [60]. Максимална, минимална, прва, последња, просечна вредност, број појављивања, сума, просечна вредност и варијанса су перформансе које се могу мерити помоћу оба типа променљивих. Тренутак првог или последњег појављивања, време извршавања активности и други временски интервали, се могу мерити само временским променљивама.

Да би се симулирала ПМ, неопходна је примена софтверског пакета који омогућава симулацију и мерење њених резултата.

4.6. Инверзне ПМ

Метода за генерисање минискупова, која ће бити представљена у овом раду, заснива се на коришћењу инверзних Петријевих мрежа (ИПМ). У литератури на енглеском језику се, за ову врсту ПМ, могу наћи термини *inversed*, *reverse* и *backward*, тако да се могу назвати и Обрнуте ПМ. Појам инверзне ПМ преузет је из теорије графова у којима постоји дефиниција обрнутих графова.

Дефиниција 4.17 [11]: Обрнути граф усмереног графа графа $G = (V, E)$ је усмерени граф $G^R = (V, E^R)$ са истим скупом чворова као G али са супротно оријентисаним гранама: $E^R = \{(u, v) | (v, u) \in E\}$.

ИПМ су у [82] дефинисане на следећи начин:

Дефиниција 4.18: Инверзна ПМ дате ПМ, чија је структура $N = (P, T, F)$, је Петријева мрежа са структуром $RN = (P, T, F^R)$, где је F^R скуп супротно оријентисаних грана из F , тј. $F^R = \{(u, v) | (v, u) \in F\}$.

Напомена 4.2: Пошто су све гране обрнуто оријентисане, матрица инциденције ИПМ може да се изрази као $-A = [-a_{ij}], i = 1, \dots, m; j = 1, \dots, n$, где је $A = [a_{ij}], i = 1, \dots, m; j = 1, \dots, n$ матрица инциденције дате ПМ.

Својство ИПМ, које је веома значајно за развој методе описане у овом раду, је:

Својство 4.3: Нека су M_0 и M маркирања ПМ и њене ИПМ, и нека је M достижно из M_0 у ПМ. Тада је маркирање M_0 достижно из маркирања M у ИПМ.

Доказ: Трансформација из M_0 у M у ПМ је дата једначином стања (4.6) која је еквивалентна следећим једначинама:

$$\begin{aligned}M_0 &= M - A \cdot V_\sigma \\M_0 &= M + (-A) \cdot V_\sigma.\end{aligned}\tag{4.12}$$

Једначина (4.12) је једначина стања која описује прелазак из M у M_0 у ПМ са матрицом инциденције $-A$, тј. у ИПМ дате ПМ. ♦

Напомена 4.3: Секвенце паљења којима се врше трансформације $M_0 \rightarrow M$ и $M \rightarrow M_0$ су обрнуте. Међутим, вектор паљења V_σ је исти у обе једначине, зато што он означава само број паљења одређеног прелаза у секвенци али не и саму секвенцу паљења.

4.7. Област примене ПМ

Петријеве мреже имају веома широку област примене, а користе их углавном истраживачки центри великих компанија у сарадњи са универзитетским институтима и департманима. Прва практична примена везана је за *Bell Telephone Laboratories* шездесетих година XX века, где су коришћене, у сарадњи са К.А. Петријем, за потребе тима за симулацију пројекта *ESS-1 (Electronic switching system)* [37,14]. Већина врста и проширења ПМ су развијене из практичних примена, а број врста које су прво теоријски конципиране је незнатан. Област

примене различитих класа ПМ је широка али са још увек великим простором за истраживање. Овде је издвојено неколико карактеристичних области, као и донети коришћења ПМ код нас. Поред примене у анализи позданости, која ће бити детаљније описана у следећој глави, неке од најчешћих области примене су:

– Протоколи и мреже: Моделирање и валидација IPv6 протокола у *Ericsson Telebit*. Обојени ПМ модели и симулација су коришћени у анализи и избору сценарија за комуникацију [52]. Телекомуникационе мреже у Комапнији *Deutsche Telekom*. Објектно оријентисане ПМ су коришћене у раздвајању процеса на улоге чија је валидација извршена помоћу ПМ, прво појединачно, а затим процеса као целине [16]. Моделирање и анализа различитих варијанти *TCP* протокола, као део већег пројекта моделирања у сарадњи са *Hewlett-Packard* [6].

– Војне примене: Програм управљања нуклеарним отпадом у САД [70]. Анализа перформанси командних компоненти *NASA* система осматрања земље [68]. Симулатор рада пројектила у Аустралији (управа Јужноаустралијског Универзитета и *DSTO - Defence Science and Tehnology Organisation*) [35]. Систем поморске команде и контроле у Канади. ПМ су коришћене за оцену перформанси у случају различитих одлука о рапореду наоружања [4].

– Софтвер и хардвер: Моделирање и анализа дизајна и понашања *VLSI* чипова (*Meta Software, Cambridge MA*) [99]. Моделирање и симулација система за складиштење докумената [95]. Моделирање интеракције композитног веб сервиса са осталим веб сервисима [121].

– Примене у саобраћају: Систем контроле Европских возова у Немачкој (*Institute for Control and Automation Engineering at the Technical University of Braunschweig*) [45]. Саобраћајна сигнализација у Бразилу [76]. Планирање секвенце слетања авиона на фреквентним аеродромима (*German Aerospace Center*) [73]. Симулација сигнализације Т-раскрснице [126].

– *Workflows* и пословни процеси: Моделирање механизма за расподелу посла са аспекта ресурса [77]. Конфигурација и тестирање пословних процеса софтверским пакетом *Protos2CPN*, базираним на обојеним ПМ [36]. Моделирање пословних

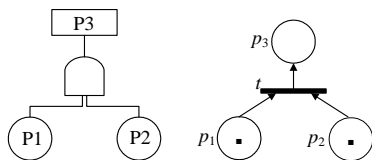
процеса у финансијама [53]. Симулација и анализа ефекта бича у ланцима снабдевања [64]. Моделирање и симулација процеса у производњи [57,1,62,130]. Моделирање и анализа процеса контроле залиха [3]. Моделирање и симулација процеса пријема робе код велепродаваца воћа [136].

– Остале примене: Симулација биохемијских процеса у организму [32,93]. СПМ модел система за праћење пацијената [102]. Моделирање процеса издавања часописа YUJOR [132]. Моделирање и симулација система одржавања аутобуса ГСП [128]. Закључивање о исправности аутобуса помоћу фази ПМ [113]. Симулација система са редовима чекања [131].

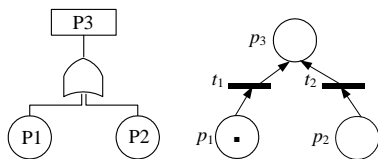
5. ПРИМЕНА ПМ У АСН

Прве примене ПМ у анализи поузданости почеле су већ седамдесетих година прошлог века [71]. Квалитативна анализа СН помоћу ПМ је први пут објављена касних осамдесетих година у [43]. Већ 1989. године, овај приступ је практично примењен у војној индустрији [67]. Интересовање за АСН помоћу ПМ постоји континуирано до данас и, поред БДО, највећи број публикација везаних за АСН је из области ПМ. Пошто је предмет овог рада квалитативна анализа СН, овде ће бити приказан преглед алгоритама за одређивање минипресека СН заснованих на ПМ. Пре тога ће бити објашњене основне поставке моделирања СН помоћу ПМ приказане у [43], на којима се заснивају касније методе других аутора.

Хура (*Hura*) и Атвуд (*Atwood*) су први пут применили ПМ у анализи кохерентног СН. Полазна тачка моделирања СН помоћу ПМ је да се логичка кола типа И и ИЛИ могу директно моделирати Петријевим мрежама (слике 5.1. и 5.2).



Слика 5.1. Логичко коло типа И и његова еквивалентна ПМ



Слика 5.2. Логичко коло типа ИЛИ и његова еквивалентна ПМ

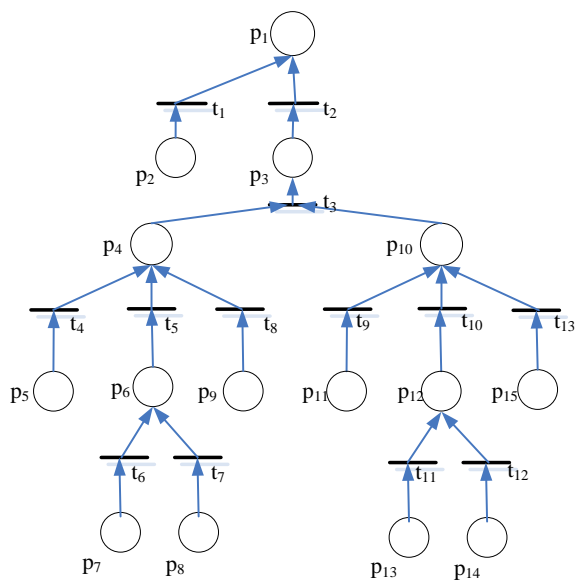
Местима ПМ моделирани су догађаји СН (примарни и посредни). Присуство жетона у месту означава да се одговарајући отказ одиграо. У случају оба логичка кола са слика 5.1. и 5.2, почетно маркирање $M = [0, 0, 1]^T$ одговара стању када се десио нежељени (вршни) догађај P3.

Прелазима су моделирана логичка кола. У случају логичког кола типа И, до отказа Р3 ће доћи ако се истовремено десе откази Р1 и Р2. На еквивалентној ПМ (ЕПМ) са слике 5.1, жетон ће се појавити у месту p_3 паљењем прелаза t само ако у оба улазна места p_1 и p_2 тог прелаза постоје жетони. Ако је логичко коло типа ИЛИ, до отказа Р3 ће доћи ако се деси бар један од отказа Р1 и Р2. На ЕПМ са слике 5.2, жетон ће се појавити у месту p_3 паљењем прелаза t_1 или t_2 , односно ако бар једном од њихових улазних места p_1 и p_2 постоји жетон.

Кохерентно СН је скуп логичких кола типа И и ИЛИ, тако да се сваком кохерентном СН може придружити ЕПМ.

Пример

На слици 5.3. је приказана ЕПМ СН са слике 2.4. ЕПМ је добијена разлагањем СН на појединачна логичка кола, њиховим моделирањем одговарајућим ПМ структурама и поновним спајањем у јединствену ПМ.



Слика 5.3. ЕПМ за СН са слике 2.4.

Динамика ЕПМ се може анализирати једначином стања (4.6):

$$M^T = M_0^T + A \cdot V_\sigma^T$$

У случају логичког кола типа И, једначина стања којом се приказује прелазак из почетног маркирања $M_0 = [1, 1, 0]^T$ у нежељено стање $M = [0, 0, 1]^T$, има облик:

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \cdot [1].$$

У случају логичког кола типа ИЛИ, једначина стања којом се приказује прелазак из почетног маркирања $M_0 = [1, 0, 0]^T$ у нежељено стање $M = [0, 0, 1]^T$, има облик:

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Метода за одређивање отказа, која је приказана у [43], заснована је на једначини стања. Основна идеја је преузета из [71] где је други Кирхофов закон примењен на маркирани граф. Нека су M_0 и M почетно и коначно маркирање и $\Delta M = M - M_0$ њихова разлика. Једначина стања (4.6) се сада може написати у облику:

$$\Delta M = A \cdot V_\sigma^T \quad (5.1)$$

Вектори ΔM и $A \cdot V_\sigma^T$ треба да буду једнаки, односно њихова разлика $E = A \cdot V_\sigma^T - \Delta M$ треба да буде нула-вектор. Ако није добијен нула-вектор, онда елемент i , различит од нуле у колони вектору E , указује да i -том месту у ПМ треба променити маркирање. Ова метода се примењује за суседна маркирања (стања) и омогућава одређивање само по једног отказа у сваком кораку. То значи да ће у сваком кораку вектор паљења V_σ садржати само један елемент различит од нуле.

5.1. Графичка метода за одређивање минипресека

Графичка метода је приказана у [61], а основна идеја је преузета из *MOCUS* алгоритма. Ради једноставнијег описа алгоритма биће уведени следећи појмови:

- Примарна места – места Петријеве мреже која моделирају примарне догађаје,

- Вршно место – место Петријевој мреже које моделира вршни догађај,
- Рута – низ грана, места и прелаза којима су повезана два места у Петријевој мрежи.

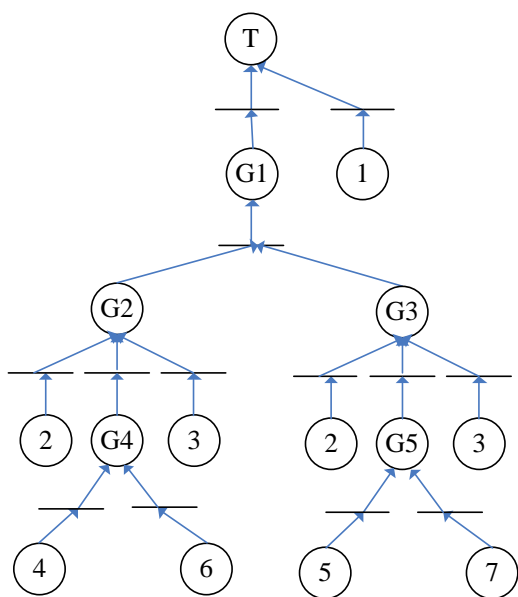
Уколико у СН постоје догађаји чија је вишеструкост већа од 1, свако њихово понављање се моделира местима са истом ознаком.

Алгоритам методе се састоји од следећих пет корака:

1. Ако је примарно место повезано са вршним местом тако да на њиховој рути нема ни једног прелаза са више улазних места, онда одговарајући примарни догађај представља минипресек.
2. Ако на рути од примарног до вршног места постоји прелаз са више улазних места, поређати хоризонтално (у исти ред) сва улазна места тог прелаза. (Није неопходно разматрати руте од хоризонтално поређаних места до вршног места.)
3. Заменити хоризонтално поређана места њиховим примарним местима, односно хоризонтално поређана места посматрати као суб-вршна места, и понављати кораке 1-3 док се не дође до примарних места.
4. Уклонити надскупове, односно скупове пресека који садрже друге скупове пресека. Резултат су минипресеци.

Пример

Метода ће бити илустрована на СН са слике 2.1. На основу СН нумерисаног за примену *MOCUS* алгоритма, са слике 2.5, формирана је ПМ која је приказана на слици 5.4, заједно са поступком добијања минипресека помоћу графичке методе. У табели са слике, у трећем кораку су подвучени скупови пресека који се редукују, а у четвртом је добијено свих седам минипресека датог СН



Кораци			
1	2	3	4
1	1	1	1
2, G3	2,2	2,2	2
	2,G5	<u>2,5</u>	
		<u>2,7</u>	
	2,3	<u>2,3</u>	
4,G3	4,2	<u>4,2</u>	
	4,G5	4,5	4,5
		4,7	4,7
	4,3	<u>4,3</u>	
6,G3	6,2	<u>6,2</u>	
	6,G5	6,5	6,5
		6,7	6,7
	6,3	<u>6,3</u>	
3,G3	3,2	<u>3,2</u>	
	3,G5	<u>3,5</u>	
		<u>3,7</u>	
	3,3	<u>3,3</u>	
2, G2	2,2	<u>2,2</u>	
	2,G4	<u>2,4</u>	
		<u>2,6</u>	
	2,3	<u>2,3</u>	
5,G2	5,2	<u>5,2</u>	
	5,G4	<u>5,4</u>	
		<u>5,6</u>	
	5,3	<u>5,3</u>	
7,G2	7,2	<u>7,2</u>	
	7,G4	<u>7,4</u>	
		<u>7,6</u>	
	7,3	<u>7,3</u>	
3, G2	3,2	<u>3,2</u>	
	3,G4	<u>3,4</u>	
		<u>3,6</u>	
	3,3	3,3	3

Слика 5.4. ПМ за СН са слике 2.5. и кораци графичке методе

5.2. Матрична метода

У [119] је приказана одозго наниже матрична методе за одређивање минималних скупова пресека помоћу ПМ. Метода полази од већ формиране и нумерисане ПМ за дато СН. Препорука аутора је да се нумерисање врши одоздо на горе и то прво места којима се моделирају примарни, а затим места којима се моделирају

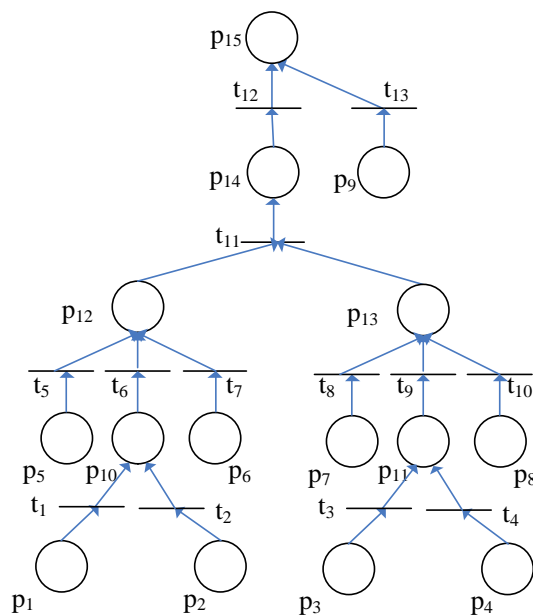
посредни догађаји. Уколико у СН постоје догађаји чија је вишеструкост већа од 1, свако понављање догађаја се моделира посебним местом ПМ.

Правила за добијање минипресека матричном методом су:

1. За ИЛИ везу: уписати ознаке излазних места у истом реду.
2. За И везу: уписати ознаке излазних места у истој колони.
3. Приликом прослеђивања ИЛИ везе неког улазног места, елементи из његове колоне постају заједнички за сва његова излазна места.
4. Полазећи од вршног догађаја примењивати кораке 1-3, док сва места не буду замењена примарним догађајима. На тај начин је формирана матрица примарних догађаја. Колоне вектори ове матрице представљају скупове пресека.
5. Применити редукцију уклањањем надскупова. Колоне вектори које преостају су минипресеци.

Пример

ПМ са СН са слике 2.4, нумерисана одоздо на горе, приказана је на слици 5.5.



Слика 5.5. ПМ за СН са слике 2.4. нумерисана за примену матричне методе

Поступак примене матричне методе је:

$$\left[\begin{array}{ccc} & & 14 \end{array} \quad \begin{array}{ccc} & & 9 \end{array} \right]$$

$$\left[\begin{array}{ccc} & & 12 \\ & & 13 \\ & & 9 \end{array} \right]$$

$$\left[\begin{array}{ccc} 5 & 5 & 5 \\ 7 & 11 & 8 \end{array} \quad \begin{array}{ccc} 10 & 10 & 10 \\ 7 & 11 & 8 \end{array} \quad \begin{array}{ccc} 6 & 6 & 6 \\ 7 & 11 & 8 \\ 9 \end{array} \right]$$

$$\left[\begin{array}{cccc} 5 & 5 & 5 & 5 \\ 7 & 3 & 4 & 8 \end{array} \quad \begin{array}{cccc} 1 & 1 & 1 & 1 \\ 7 & 3 & 4 & 8 \end{array} \quad \begin{array}{cccc} 2 & 2 & 2 & 2 \\ 7 & 3 & 4 & 8 \end{array} \quad \begin{array}{cccc} 6 & 6 & 6 & 6 \\ 7 & 3 & 4 & 8 \\ 9 \end{array} \right]$$

$$\left[\begin{array}{cccc} 5 & 5 & 5 & 5 \\ 5 & 3 & 4 & 6 \end{array} \quad \begin{array}{cccc} 1 & 1 & 1 & 1 \\ 5 & 3 & 4 & 6 \end{array} \quad \begin{array}{cccc} 2 & 2 & 2 & 2 \\ 5 & 3 & 4 & 6 \end{array} \quad \begin{array}{cccc} 6 & 6 & 6 & 6 \\ 5 & 3 & 4 & 6 \\ 9 \end{array} \right]$$

$$\left[\begin{array}{ccc} & & 1 & 1 \\ & & 2 & 2 \end{array} \quad \begin{array}{ccc} & & 3 & 4 \\ & & 3 & 4 \end{array} \quad \begin{array}{ccc} & & 6 & 9 \end{array} \right]$$

Колоне четврте матрице представљају свих 17 скупова пресека СН са слике 2.4. У петој матрици су сва понављања вишеструких догађаја означена истим бројевима, а у последњој матрици је, након примене закона апсорпције и уклањања надскупова, добијено свих 7 минипресека датог СН.

5.3. Подела ПМ на нивое

У [92] је приказан алгоритам за налажење минипресека кохерентног СН, којим се ПМ дели и траже путеви од листова до корена преко веза између прелаза различитих нивоа. Алгоритам се састоји од следећих осам корака:

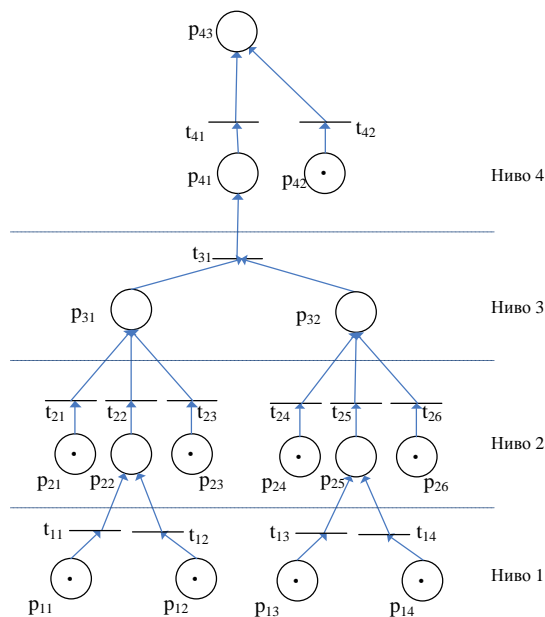
1. Анализирати структуру СН и поделити га у нивое.
2. Поставити почетно маркирање тако да свако место p за које важи ${}^0p = \emptyset$ садржи жетон.
3. Наћи све омогућене прелазе (прелазе који се могу запалити).
4. Направити све могуће комбинације омогућених прелаза различитих нивоа

и у свакој комбинацији запалити све прелазе t_{ij} , идући од нижег нивоа ка вишем. Индекс i се односи на ниво СН, а индекс j на прелаз у нивоу СН.

5. Издвојити све секвенце паљења прелаза $\sigma = \{\sigma_k, k \text{ коначно}\}$ које трансформишу M_0 у M_n . M_n су сва маркирања у којима место које моделира вршни догађај саджи жетон.
6. Заменити свако $t_{ij} \in \sigma_k$ скупом места ${}^0t_{ij}$. Нађени скуп места је скуп пресека.
7. Одбацити места p за која важи ${}^0p \neq \emptyset$ и $p^0 \neq \emptyset$.
8. Упоредити свака два скупа пресека и издвојити минипресеке.

Пример

На слици 5.6. је приказана ПМ за СН са слике 2.4, подељена по нивоима, нумерисана за примену алгоритма и са почетним маркирањем одређеним на основу корака 2 алгоритма.



Слика 5.6. ПМ за СН са слике 2.4, подељена на нивое

С обзиром да се предложени алгоритам односи на СН без вишеструких догађаја, резултат алгоритма у случају ПМ са слике 5.6. ће бити скупови пресека датог СН.

1. ПМ је подељена у четири нивоа.

2. Почетно маркирање је $M_0 = [111110110100010]$.
3. На нивоу 1 омогућени су прелази: $t_{11}, t_{12}, t_{13}, t_{14}$, на нивоу 2 прелази: $t_{21}, t_{23}, t_{24}, t_{26}$, на нивоу 3 нема омогућених прелаза и на нивоу 4 омогућен је прелаз t_{42} .
4. Комбинације прелаза су:
1. 0 прелаза нивоа 1 и 2 и 1 прелаз нивоа 4:
 - $\sigma_1 = t_{42} \Rightarrow$ крајње маркирање $p_{41} = 1$
 2. 0 прелаза нивоа 1, 1 прелаз нивоа 2 и 0 прелаза нивоа 4:
 - $\sigma_2 = t_{21} \Rightarrow t_{31}$ није омогућен, крајње маркирање $p_{41} = 0$
 - $\sigma_3 = t_{23} \Rightarrow t_{31}$ није омогућен, крајње маркирање $p_{41} = 0$
 - $\sigma_4 = t_{24} \Rightarrow t_{31}$ није омогућен, крајње маркирање $p_{41} = 0$
 - $\sigma_5 = t_{26} \Rightarrow t_{31}$ није омогућен, крајње маркирање $p_{41} = 0$
 3. 0 прелаза нивоа 1, 2 прелаза нивоа 2 и 0 прелаза нивоа 4:
 - $\sigma_6 = t_{21}, t_{23} \Rightarrow t_{31}$ није омогућен, крајње маркирање $p_{41} = 0$
 - $\sigma_7 = t_{21}, t_{24} \Rightarrow t_{31}$ и t_{41} су омогућени, крајње маркирање $p_{41} = 1$
 - $\sigma_8 = t_{21}, t_{26} \Rightarrow t_{31}$ и t_{41} су омогућени, крајње маркирање $p_{41} = 1$
 - $\sigma_9 = t_{23}, t_{24} \Rightarrow t_{31}$ и t_{41} су омогућени, крајње маркирање $p_{41} = 1$
 - $\sigma_{10} = t_{23}, t_{26} \Rightarrow t_{31}$ и t_{41} су омогућени, крајње маркирање $p_{41} = 1$
 - $\sigma_{11} = t_{24}, t_{26} \Rightarrow t_{31}$ није омогућен, крајње маркирање $p_{41} = 0$
 4. 0 прелаза нивоа 1, 3 прелаза нивоа 2 и 0 прелаза нивоа 4:
 - $\sigma_{12} = t_{21}, t_{23}, t_{24} \Rightarrow t_{31}$ и t_{41} су омогућени, крајње маркирање $p_{41} = 1$
 - $\sigma_{13} = t_{21}, t_{23}, t_{26} \Rightarrow t_{31}$ и t_{41} су омогућени, крајње маркирање $p_{41} = 1$
 - $\sigma_{14} = t_{23}, t_{24}, t_{26} \Rightarrow t_{31}$ и t_{41} су омогућени, крајње маркирање $p_{41} = 1$
 5. 0 прелаза нивоа 1, 4 прелаза нивоа 2 и 0 прелаза нивоа 4:
 - $\sigma_{15} = t_{21}, t_{23}, t_{24}, t_{26} \Rightarrow t_{31}$ и t_{41} су омогућени, крајње маркирање $p_{41} = 1$

На исти начин су испитане све остале комбинације омогућених прелаза. У табели 5.1. је дата листа комбинација и броја секвенци за сваку од комбинација. Укупно

постоји 49 комбинација. Међутим, из табеле су изостављене: 4 комбинације са прелазима другог и четвртог нивоа, 4 комбинације са прелазима првог и четвртог нивоа и 16 комбинација са прелазима сва три нивоа. Ове комбинације су изостављене, јер је очигледно да ће све бити редуковане у кораку 8, с обзиром да постоји само један омогућени прелаз нивоа 4, који је већ укључен као једини прелаз прве комбинације. Поред тога, у табели су за сваку комбинацију приказане секвенце паљења које трансформишу M_0 у M_n , односно резултат корака 5. Због великог броја секвенци, након 18. комбинације приказане су само ознаке секвенци које се добијају у овом кораку. У следећој колони табеле приказани су одговарајући скупови места ПМ који су резултат корака 6. У последњој колони су приказани резултати корака 7 и 8.

Табела 5.1. Кораци 2-4 алгоритма *Rochdi*-а

р.б.	број t по нивоима			секвенце	5	кораци	
	1	2	4			6	7 и 8
1	0	0	1	σ_1	$\sigma_1 = t_{42}$	P_{42}	P_{42}
2	0	1	0	$\sigma_2 - \sigma_5$	/	/	
3	0	2	0	$\sigma_6 - \sigma_{11}$	$\sigma_7 = t_{21}, t_{24}, t_{31}, t_{41}$ $\sigma_8 = t_{21}, t_{26}, t_{31}, t_{41}$ $\sigma_9 = t_{23}, t_{24}, t_{31}, t_{41}$ $\sigma_{10} = t_{23}, t_{26}, t_{31}, t_{41}$	$P_{21}, P_{24}, P_{31}, P_{32}, P_{41}$ $P_{21}, P_{26}, P_{31}, P_{32}, P_{41}$ $P_{23}, P_{24}, P_{31}, P_{32}, P_{41}$ $P_{23}, P_{26}, P_{31}, P_{32}, P_{41}$	P_{21}, P_{24} P_{21}, P_{26} P_{23}, P_{24} P_{23}, P_{26}
4	0	3	0	$\sigma_{12} - \sigma_{14}$	$\sigma_{12} = t_{21}, t_{23}, t_{24}, t_{31}, t_{41}$ $\sigma_{13} = t_{21}, t_{23}, t_{26}, t_{31}, t_{41}$ $\sigma_{14} = t_{23}, t_{24}, t_{26}, t_{31}, t_{41}$	$P_{21}, P_{23}, P_{24}, P_{31}, P_{32}, P_{41}$ $P_{21}, P_{23}, P_{26}, P_{31}, P_{32}, P_{41}$ $P_{23}, P_{24}, P_{26}, P_{31}, P_{32}, P_{41}$	/
5	0	4	0	σ_{15}	$\sigma_{15} = t_{21}, t_{23}, t_{24}, t_{26}, t_{31}, t_{41}$	$P_{21}, P_{23}, P_{24}, P_{26}, P_{31}, P_{32}, P_{41}$	/
6	1	0	0	$\sigma_{16} - \sigma_{19}$	/	/	
7	2	0	0	$\sigma_{20} - \sigma_{25}$	$\sigma_{21} = t_{11}, t_{13}, t_{31}, t_{41}$ $\sigma_{22} = t_{11}, t_{14}, t_{31}, t_{41}$ $\sigma_{23} = t_{12}, t_{13}, t_{31}, t_{41}$ $\sigma_{24} = t_{12}, t_{14}, t_{31}, t_{41}$	$P_{11}, P_{13}, P_{31}, P_{32}, P_{41}$ $P_{11}, P_{14}, P_{31}, P_{32}, P_{41}$ $P_{12}, P_{13}, P_{31}, P_{32}, P_{41}$ $P_{12}, P_{14}, P_{31}, P_{32}, P_{41}$	P_{11}, P_{13} P_{11}, P_{14} P_{12}, P_{13} P_{12}, P_{14}
8	3	0	0	$\sigma_{26} - \sigma_{28}$	$\sigma_{26} = t_{11}, t_{12}, t_{13}, t_{31}, t_{41}$ $\sigma_{27} = t_{11}, t_{12}, t_{14}, t_{31}, t_{41}$ $\sigma_{28} = t_{12}, t_{13}, t_{14}, t_{31}, t_{41}$	$P_{11}, P_{12}, P_{13}, P_{31}, P_{32}, P_{41}$ $P_{11}, P_{12}, P_{14}, P_{31}, P_{32}, P_{41}$ $P_{12}, P_{13}, P_{14}, P_{31}, P_{32}, P_{41}$	/
9	4	0	0	σ_{29}	$\sigma_{29} = t_{11}, t_{12}, t_{13}, t_{14}, t_{31}, t_{41}$	$P_{11}, P_{12}, P_{13}, P_{14}, P_{31}, P_{32}, P_{41}$	/

р.б.	број t по нивоима			секвенце	5	кораци	6	7 и 8
	1	2	4					
18	1	1	0	$\sigma_{30} - \sigma_{45}$	$\sigma_{32}, \sigma_{33}, \sigma_{36} - \sigma_{39},$ σ_{42}, σ_{43}	8		P_{11}, P_{24} P_{11}, P_{26} P_{12}, P_{24} P_{12}, P_{26} P_{13}, P_{21} P_{13}, P_{22} P_{14}, P_{21} P_{14}, P_{22}
19	1	2	0	$\sigma_{46} - \sigma_{69}$	$\sigma_{47} - \sigma_{51},$ $\sigma_{53} - \sigma_{62},$ $\sigma_{64} - \sigma_{68}$	20		/
20	1	3	0	$\sigma_{70} - \sigma_{81}$	$\sigma_{70} - \sigma_{81}$	22		/
21	1	4	0	$\sigma_{82} - \sigma_{85}$	$\sigma_{82} - \sigma_{85}$	4		/
22	2	1	0	$\sigma_{86} - \sigma_{109}$	$\sigma_{88} - \sigma_{107}$	10		/
23	2	2	0	$\sigma_{110} - \sigma_{145}$	$\sigma_{111} - \sigma_{144}$	34		/
24	2	3	0	$\sigma_{146} - \sigma_{163}$	$\sigma_{146} - \sigma_{163}$	18		/
25	2	4	0	$\sigma_{164} - \sigma_{169}$	$\sigma_{164} - \sigma_{169}$	6		/
26	3	1	0	$\sigma_{170} - \sigma_{181}$	$\sigma_{170} - \sigma_{181}$	12		/
27	3	2	0	$\sigma_{182} - \sigma_{199}$	$\sigma_{182} - \sigma_{199}$	18		/
28	3	3	0	$\sigma_{200} - \sigma_{208}$	$\sigma_{200} - \sigma_{208}$	9		/
29	3	4	0	$\sigma_{209} - \sigma_{211}$	$\sigma_{209} - \sigma_{211}$	3		/
30	4	1	0	$\sigma_{212} - \sigma_{215}$	$\sigma_{212} - \sigma_{215}$	4		/
31	4	2	0	$\sigma_{216} - \sigma_{221}$	$\sigma_{216} - \sigma_{221}$	6		/
32	4	3	0	$\sigma_{222} - \sigma_{224}$	$\sigma_{222} - \sigma_{224}$	3		/
33	4	4	0	σ_{225}	σ_{225}	1		/

Резултат корака 8 је свих 17 скупова пресека СН са слике 2.4.

Сами аутори у [92] наглашавају непогодност алгоритма за одређивање минипресека за СН великих димензија и предлажу редукцију ПМ помоћу фузије прелаза за превазилажење проблема комбинаторне експлозије.

5.4. Одређивање минипресека помоћу t -инваријанте

Одређивање минипресека помоћу t -инваријанте ПМ приказано је у [9]. Овај приступ се разликује од претходних зато што се не пролази кроз СН да би се одредили минипресеци, већ се директно, на основу структуре целог СН израженог преко матрице инциденције, одређују минипресеци. Алгоритам је првенствено намењен налажењу минипресека ГСПМ датог СН. Међутим, првим кораком алгоритма, ГСПМ се брисањем инхибиторских грана и уклањањем петљи, своди на елементарну ПМ.

Уколико у СН постоје примарни догађаји чија је вишеструкост већа од 1, сва понављања истог вишестуког догађаја се моделирају истим местом ПМ. Након формирања ПМ датог СН, примењују се следећа четири корака:

1. Додати по један прелаз извор сваком месту које моделира примарни догађај СН и прелаз ушће месту које моделира вршни догађај.
2. Одредити t -инваријанте. Из t -инваријанти елиминисати све прелазе који нису извори.
3. Одредити носаче t -инваријанти. Добијени скупови одговарају скуповима пресека датог СН.
4. Елиминисати надскупове. Преостали скупови одговарају минипресецима датог СН.

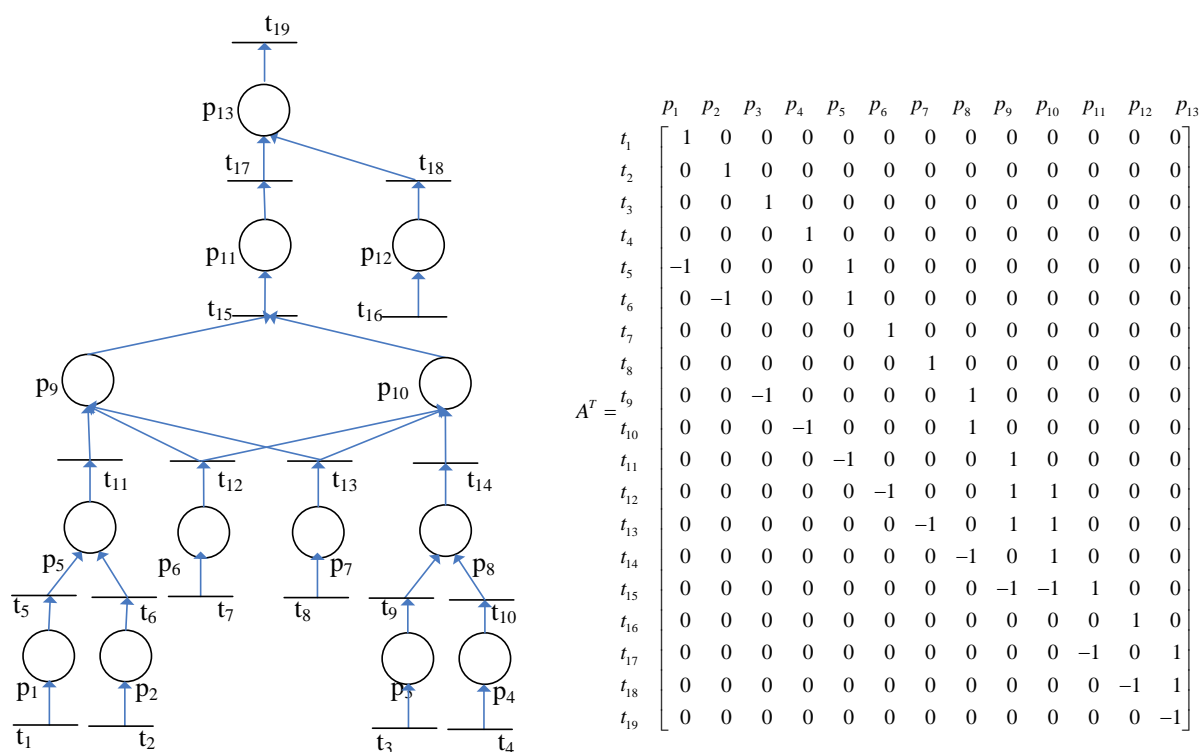
Овде су извршене две модификације алгоритма: Прво, изостављен је први корак оригиналног алгоритма, јер се овде разматрају само елементарне ПМ. Друга модификација убрзава одређивање минипресека. Наиме, оригиналним алгоритмом се прво одређују t -инваријанте и скупови места која одговарају добијеним инваријантама. Након тога се, из добијених скупова места, искључују места која не моделирају примарне догађаје. На крају се елиминишу надскупови. Пошто у ПМ датог СН прелази извори постоје само као улазни извори места која одговарају примарним догађајима, ефикасније је прво из t -варијанти елиминисати прелазе који су улазни за места која моделирају посредне догађаје, па тек након тога одредити одговарајуће скупове места инваријанти.

Идеја приступа из [9] је оригинална и веома интересантна. Наиме, када се Петријевом мрежом моделира СН, t -иваријанте се могу интерпретирати као секвенце паљења прелаза које, стартујући из празног маркирања, воде у празно маркирање. Пошто је посматрана ПМ ациклична, празно маркирање је могуће постићи само ако се на почетку запали одговарајућа комбинација прелаза извора, а као последњи запали прелаз ушће. Прелаз ушће ће се запалити само ако се у његовом улазном месту, које моделира вршни догађај односно отказ система, појави жетон.

Пример

Алгоритам ће бити илустрован на истом примеру као и остали у овом поглављу.

1. На слици 5.7. је приказана ПМ за СН са слике 2.4, модификована увођењем седам прелаза извора и једног прелаза ушћа и њена транспонована матрица инциденције.



Слика 5.7. ПМ са додатним прелазима и матрица инциденције

2. t -инваријанте су n -вектори W који се одређују на основу (4.11):

$$A \cdot W^T = 0$$

где је A матрица инциденције ПМ, а n број прелаза у ПМ. У овом случају постоји додатни услов да су све вредности вектора W бинарне.

Решавањем система (4.11) за матрицу инциденције са слике 5.7, добијају се следећа решења:

$$W_1 = [1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1]$$

$$W_2 = [1\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1]$$

$$W_3 = [0\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1]$$

$$W_4 = [0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1]$$

$$W_5 = [0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1]$$

$$W_6 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$$

$$W_7 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 1]$$

Одговарајуће t -инваријанте су: $\{t_1, t_3, t_5, t_9, t_{11}, t_{14}, t_{15}, t_{17}, t_{19}\}$,

$\{t_1, t_4, t_5, t_{10}, t_{11}, t_{14}, t_{15}, t_{17}, t_{19}\}$, $\{t_2, t_3, t_6, t_9, t_{11}, t_{14}, t_{15}, t_{17}, t_{19}\}$,

$\{t_2, t_4, t_6, t_{10}, t_{11}, t_{14}, t_{15}, t_{17}, t_{19}\}$, $\{t_7, t_{12}, t_{15}, t_{17}, t_{19}\}$, $\{t_8, t_{13}, t_{15}, t_{17}, t_{19}\}$ и $\{t_{16}, t_{18}, t_{19}\}$.

Елиминисањем свих прелаза који нису извор, добијају се следеће редуковане t -инваријанте: $\{t_1, t_3\}$, $\{t_1, t_4\}$, $\{t_2, t_3\}$, $\{t_2, t_4\}$, $\{t_7\}$, $\{t_8\}$ и $\{t_{16}\}$.

3. Носачи добијених t -инваријанти су: $\{p_1, p_3\}$, $\{p_1, p_4\}$, $\{p_2, p_3\}$, $\{p_2, p_4\}$, $\{p_6\}$, $\{p_7\}$ и $\{p_{12}\}$ и одговарају минипресецима датог СН.

5.5. Одређивање минипресека на основу матрице инциденције ПМ

Одређивање минипресека на основу веза између улаза и излаза у матрици инциденције ПМ датог СН приказано је у [117] и [118]. Због лакшег разумевања алгоритма, уведени су појмови: вршно, посредно и примарно место за описивање места ПМ, којима се моделирају вршни, посредни и примарни догађаји СН, респективно. Поред тога, уведен је однос улаза и излаза између места, тако да се

за место p_i каже да је улазно место места p_j , ако постоји прелаз t такав да $t \in p_i^0$ и $t \in {}^0_j p$, односно прелаз чије је p_i улазно а p_j излазно место.

Иако се одређивање минипресека врши помоћу матрице инциденције, у овом алгоритму се то не врши директно, као у алгоритму из 5.1.4. Наиме, овде се пролази кроз матрицу инциденције, чиме се, посредно пролази кроз дату ПМ.

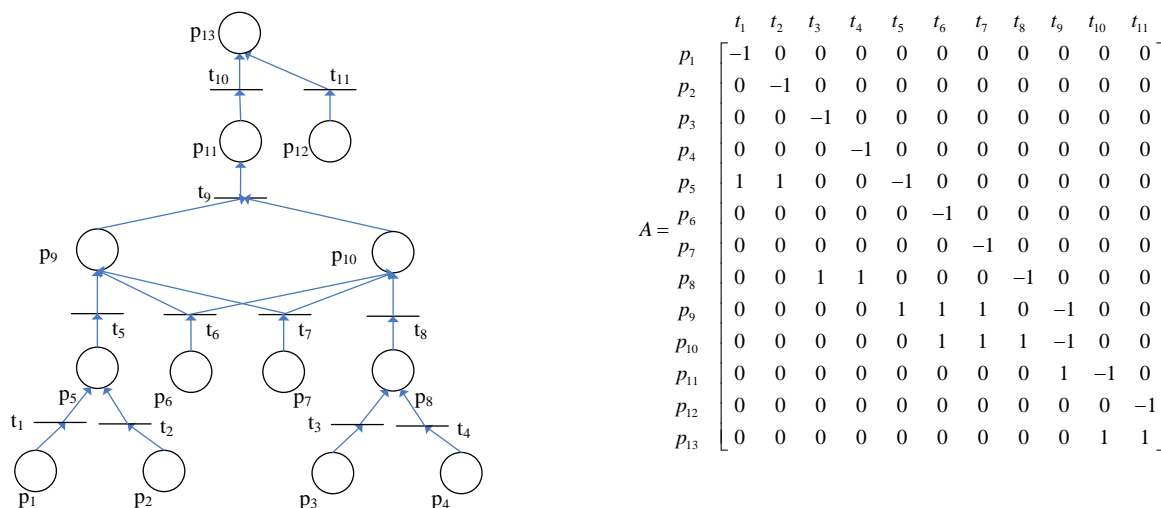
Алгоритам за одређивање минипресека помоћу матрице инциденције ПМ (АИ) се састоји од следећих 6 корака:

1. Наћи ред који садржи само вредности 1 и 0 и не садржи -1. Тај ред се назива ред вршног места, односно то је ред који садржи само излазна места. Затим почети претраживање од вршног реда.
2. Наћи вредност -1 у колони у којој вршни ред има вредност 1. Место које одговара реду са вредношћу -1 у тој колони је улазно место вршног места. Ако у колони постоји више вредности -1, то значи да одговарајући прелаз има више излазних места, што даље значи да су сва та места у И вези.
3. Наћи вредност 1 у редовима места која су пронађена у кораку 2. Ако вредност 1 постоји у реду, значи да је у питању посредно место ПМ и тада понављати корак 2, све док се не дође до реда који садржи само вредност -1. Ако вредност 1 не постоји у реду, нађено је примарно место. Ако у реду постоји више вредности 1, то значи да одговарајуће место има више улазних места која су у ИЛИ вези.
4. Понављати кораке 2 и 3 док се не нађу сва примарна места.
5. Заменити сва примарна места на основу И и ИЛИ веза да би се одредили скупови пресека.
6. Користити законе апсорпције и идемпотенције за одређивање минипресека.

Уколико у СН постоји догађај чија је вишеструкост већа од 1, сва његова понављања се моделирају једним местом ПМ и повезују са свим својим излазним местима. Тиме се постиже графичка јасноћа и аритметичка једноставност ПМ.

Пример

На слици 5.8. је приказана ПМ за СН са слике 2.4. која је формирана на описани начин и њена матрица инциденције.



$$A = \begin{matrix} & \begin{matrix} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 & t_{10} & t_{11} \end{matrix} \\ \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \\ p_9 \\ p_{10} \\ p_{11} \\ p_{12} \\ p_{13} \end{matrix} & \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

Слика 5.8. ПМ за СН са слике 2.4, нумерисана за примену алгоритма АИ

1. Тринаести ред је вршни ред.
2. Вредност 1 је у колонама 10 и 11. Улазна места прелаза t_{10} и t_{11} су у ИЛИ вези. У 10. колони вредност -1 је у 11. реду а у 11. колони у 12. реду. $\Rightarrow p_{13} = p_{11} + p_{12}$
3. У 11. реду вредност 1 је у 9. колони.

У 9. колони вредност -1 је у 9. и 10. реду. Места p_9 и p_{10} су у И вези, односно

$$p_{11} = p_9 p_{10}.$$

У 9. реду вредност 1 је у колонама 5, 6 и 7. Улазна места прелаза t_5 , t_6 и t_7 су у ИЛИ вези. У 5. колони вредност -1 је у 5. реду, у 6. колони у 6. реду а у 7. колони у 7. реду. $\Rightarrow p_9 = p_5 + p_6 + p_7.$

У 5. реду вредност 1 је у колонама 1 и 2. Улазна места прелаза t_1 и t_2 су у ИЛИ вези. У 1. колони вредност -1 је у 1. реду, у 2. колони у 2. реду. $\Rightarrow p_5 = p_1 + p_2.$

У редовима 1 и 2 нема вредности 1, што значи да су p_1 и p_2 примарна места.

У редовима 6 и 7 нема вредности 1, што значи да су p_6 и p_7 примарна места.

$$\Rightarrow p_9 = p_1 + p_2 + p_6 + p_7$$

4. У 10. реду вредност 1 је у колонама 6, 7 и 8. Улазна места прелаза t_6 , t_7 и t_8 су у ИЛИ вези. У 6. колони вредност -1 је у 6. реду, у 7. колони у 7. реду а у 8. колони у 8. реду. $\Rightarrow p_{10} = p_6 + p_7 + p_8$.

У редовима 6 и 7 нема вредности 1, што значи да су p_6 и p_7 примарна места.

У 8. реду вредност 1 је у колонама 3 и 4. Улазна места прелаза t_3 и t_4 су у ИЛИ вези. У 3. колони вредност -1 је у 3. реду, у 4. колони у 4. реду. $\Rightarrow p_8 = p_3 + p_4$.

У редовима 3 и 4 нема вредности 1, што значи да су p_3 и p_4 примарна места.

$$\Rightarrow p_{10} = p_6 + p_7 + p_3 + p_4$$

5. Свих 17 скупова пресека се одређују заменом свих претходно добијених релација.

$$p_{13} = p_{11} + p_{12}$$

$$p_{13} = p_9 p_{10} + p_{12}$$

$$p_{13} = (p_1 + p_2 + p_6 + p_7)(p_6 + p_7 + p_3 + p_4) + p_{12}$$

$$p_{13} = p_1 p_6 + p_1 p_7 + p_1 p_3 + p_1 p_4 + p_2 p_6 + p_2 p_7 + p_2 p_3 + p_2 p_4 + p_6 p_6 + p_6 p_7 + p_6 p_3 + p_6 p_4 + p_7 p_6 + p_7 p_7 + p_7 p_3 + p_7 p_4 + p_{12}$$

6. Применом закона апсорпције и идемпотенције, добија се свих седам минипресека:

$$p_{13} = p_1 p_3 + p_1 p_4 + p_2 p_3 + p_2 p_4 + p_6 + p_7 + p_{12}$$

6. ОДРЕЂИВАЊЕ МИНИПРЕСЕКА СН БЕЗ ВИШЕСТРУКИХ ДОГАЂАЈА ПОМОЋУ ИПМ

У овој глави ће бити описана метода за одређивање минипресека помоћу инверзне ПМ. Основна идеја методе је објављена у [129] и [65]. Овде ће бити приказана на кохерентном СН без вишеструких догађаја, односно чији су сви догађаји вишеструкости 1, а у наредној глави ће бити проширена на кохерентна СН са вишеструким догађајима.

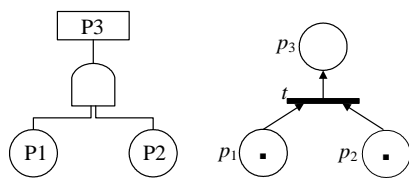
6.1. Основни појмови

Неки појмови и објашњења из глава 2 и 4 овог рада, овде ће биће поновљени, да би се олакшало читање овог дела као посебне целине.

6.1.1. Презентација логичких кола помоћу ПМ

Логичка кола СН се могу директно моделирати помоћу елементарне ПМ. Како су предмет овог рада кохерентна СН, у наставку ће бити приказана само трансформација И и ИЛИ кола у одговарајуће ПМ.

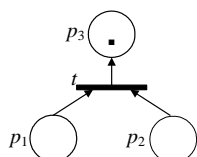
И коло: На слици 6.1. је приказано логичко коло типа И и његова еквивалентна бинарна ПМ [43].



Слика 6.1. Логичко коло И и еквивалентна ПМ

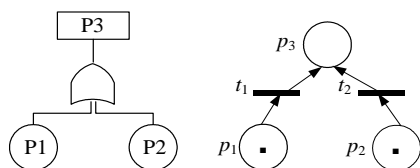
Примарни догађаји P1 и P2 су представљени изворним местима p_1 и p_2 , а вршни догађај P3 местом ушћа p_3 . Нека је $M = [M(p_1), M(p_2), M(p_3)]$ маркирање Петријеве мреже. Постојање жетона у месту моделира одигравање одговарајућег

догађаја. Логичко коло И је моделирано прелазом t , који је омогућен само ако оба његова улазна места p_1 и p_2 садрже жетон, тј. ако је почетно маркирање $M_0 = [1, 1, 0]^T$. Након паљења прелаза t , резултујуће маркирање ће бити $M_1 = [0, 0, 1]^T$ (слика 6.2). Маркирање M_1 је мртво маркирање које представља одигравање вршног догађаја P3.



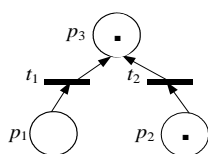
Слика 6.2. ПМ са слике 6.1. након паљења прелаза t .

ИЛИ коло: На слици 6.3. је приказано логичко коло типа ИЛИ и његова еквивалентна бинарна ПМ [43].



Слика 6.3. Логичко коло ИЛИ и еквивалентна ПМ

Логичко коло ИЛИ је моделирано прелазом t_1 , који је омогућен само ако његово улазно место p_1 садржи жетон и прелазом t_2 , који је омогућен само ако његово улазно место p_2 садржи жетон, тј. ако је почетно маркирање $M_{01} = [1, 0, 0]^T$, $M_{02} = [0, 1, 0]^T$ или $M_{03} = [1, 1, 0]^T$. На слици 6.3. је приказано почетно маркирање $M_{03} = [1, 1, 0]^T$. Из овог маркирања је могуће запалити оба прелаза. Након паљења нпр. прелаза t_1 добиће се маркирање $M_{13} = [0, 1, 1]^T$ (слика 6.4). Маркирање M није мртво маркирање јер је прелаз t_2 и даље омогућен.



Слика 6.4. ПМ са слике 6.3. након паљења прелаза t_1

6.1.2. Проблем достижности и мртво маркирање

Нека је M_0 почетно маркирање ПМ и $R(M_0)$ скуп свих маркирања која се могу достићи из M_0 . Проблем достижности овде ће бити посматран шире од проблема достижности из дефиниције 4.2. У ПМ којима се моделира СН, проблем достижности ће бити посматран као проблем налажења свих $M \in R(M_0)$.

Прелазак из једног маркирања у друго може се изразити једначином стања (4.6):

$$M = M_0 + A \cdot V_\sigma$$

где је A матрица инциденције, а V_σ вектор паљења.

У 4.2 је на основу једначине стања доказано да је маркирање M , са жетонима само у местима ушћа, мртво маркирање.

Сва маркирања, достижна из почетног маркирања M_0 се могу приказати графом достижности. Граф достижности је граф у коме сваки чвор одговара једном маркирању ПМ, а гране одговарају прелазима чијим паљењем се из једног маркирања прелази у друго. Сваки чвор графа достижности има онолико излазних грана колико је омогућених прелаза у одговарајућем маркирању. Почетно маркирање M_0 је корен, а мртва маркирања су листови графа достижности.

6.2. Еквивалента ПМ

Кохерентно СН без вишеструких догађаја се може разложити на И и ИЛИ логичка кола и на тај начин директно трансформисати у еквивалентну ПМ (ЕПМ). Пошто се у овој глави разматрају СН без вишеструких догађаја, свако место, примарно или посредно, је улазно место тачно једног логичког кола. Одатле следи да ће свако место у ЕПМ имати тачно један излазни прелаз. То ЕПМ чини *Free Choice* (FC) мрежом (видети 4.2.2).

Нека је $u = |U|$ број свих догађаја СН (примарних, посредних и вршног), а $v = |V|$ број логичких кола. Као што је показано у глави 2, пошто сваком посредном догађају и вршном догађају СН одговара по једно логичко коло, број примарних

догађаја у СН једнак је $u - v$. Како сваком догађају одговара тачно по једно место ЕПМ, укупан број места у ЕПМ једнак је u . Број прелаза ЕПМ зависи од логичких кола у СН. Уколико су сва логичка кола у СН типа И, број прелаза ће бити једнак u , а ако су сва типа ИЛИ, број прелаза ће бити једнак $u - 1$, јер тада сваки прелаз има тачно једно улазно место.

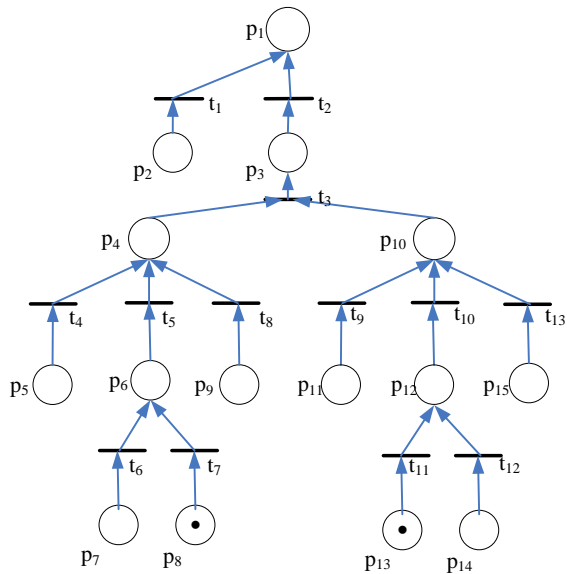
Својство 6.1: Нека је $P = \{p_1, p_2, \dots, p_m\}$ скуп места ЕПМ, где место p_1 моделира вршни догађај. Основне особине ЕПМ су:

1. ЕПМ је *FC* Петријева мрежа у којој су примарни догађаји датог СН моделирани изворним местима, а вршни догађај местом ушћа.
2. Сваки скуп пресека датог СН се може представити маркирањем ЕПМ са жетонима, само у изворним местима која моделирају одговарајуће примарне догађаје.
3. Свако маркирање $M = [1, M(p_2), \dots, M(p_m)]^T$, односно маркирање у коме је $M(p_1) = 1$, моделира одигравање вршног догађаја, што представља отказ система.
4. Свако почетно маркирање M_0 са жетонима само у изворним местима, из кога се достиже мртво маркирање $M = [1, M(p_2), \dots, M(p_m)]^T$, представља један скуп пресека.
5. Ако почетно маркирање M_0 моделира минипресек онда ће, према дефиницији минипресека, мртво маркирање ЕПМ бити $M = [1, 0, \dots, 0]^T$ [43].

Пример

На слици 6.5. је приказана ЕПМ за СН са слике 2.4. ЕПМ је нумерисана одозго на ниже. Разлог за такво нумерисање ће бити објашњен у поглављу 6.4. Вршни догађај је моделиран местом p_1 , посредни догађаји G1, G2, G3, G4 и G5 местима p_3, p_4, p_{10}, p_6 и p_{12} , респективно, а примарни догађаји P1, P4, P5, S1 и S2 местима p_2, p_7, p_{13}, p_8 и p_{14} , респективно. С обзиром да се у овој глави разматрају само СН без вишеструких догађаја, оба понављања примарног догађаја P2 су посматрана као различити догађаји који су моделирани местима p_5 и p_{11} док су оба

повнављања примарног догађај P3 моделирана местима p_9 и p_{15} . На слици је приказано почетно маркирање $M_0 = [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0]^T$ које одговара минипресеку P5·S1.



Слика 6.5. ЕПМ за СН са слике 2.4. са почетним маркирањем M_0

Мртво маркирање које се достиже из почетног маркирања M_0 је $M = [1, 0, \dots, 0]^T$. Одговарајући вектор паљења прелаза којим се M достиже из M_0 је $V_\sigma = [0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0]^T$.

Уколико би се поставило почетно маркирање које не одговара ни једном минипресеку, мртво маркирање које се из њега достиже не би било облика $M = [1, 0, \dots, 0]^T$. Тако се, нпр. из $M_0 = [0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0]^T$ достиже мртво маркирање $M_1 = [1, 0, 0, 2, 0, \dots, 0]^T$ за вектор паљења $V_\sigma = [0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0]^T$. Маркирање M_1 одговара одигравању вршног догађаја, јер је $M(p_1) = 1$, што значи да M_0 одговара неком скупу пресека. Међутим, пошто је $M(p_4) = 2$, тај скуп пресека није минималан. Нека је почетно маркирање $M_0 = [0, 0, 0, 0, 1, 0, 0, 1, 0, \dots, 0]^T$. Мртво маркирање које се достиже из M_0 је $M_2 = [0, 0, 0, 2, 0, \dots, 0]^T$ са вектором паљења $V_\sigma = [0, 0, 0, 1, 1, 0, 1, 0, \dots, 0]^T$. Маркирање M_2 одговара неодигравању вршног догађаја јер је $M(p_1) = 0$.

Тврђење 6.1: Проблем налажења свих минипресека датог СН своди се на следећи обрнути (инверзни) проблем достижности: наћи сва почетна маркирања M_0 ЕПМ из којих је маркирање $M = [1, 0, \dots, 0]^T$ доступно, односно таква да важи $M = [1, 0, \dots, 0]^T \in R(M_0)$, где је маркирање у коме је $M(p_1) = 1$ и p_1 моделира одигравање вршног догађаја.

Овај проблем достижности је тежи од стандардног проблема достижности који се састоји у испитивању да ли $M_f \in R(M_0)$ и за који је показано да у случају *Free Choice* мрежа спада у *EXPSAPCE*-тешке (видети 4.4).

6.3. Основна идеја методе

Основна идеја методе за одређивање минипресека је у томе да се они не добијају на основу ЕПМ, већ на основу Инверзне ПМ (ИПМ). У поглављу 4.6 је детаљно описана ИПМ. Подсетимо само да је ИПМ дате ПМ са структуром $N = (P, T, F)$, Петријева мрежа са структуром $RN = (P, T, F^R)$, где је F^R скуп супротно оријентисаних грана из F дате ПМ. Подсетимо се и својства 4.3, на основу кога важи да је маркирање и M_0 достижно из M у ИПМ, ако је у ПМ маркирање M достижно из M_0 .

На слици 6.6. су приказане ИПМ за ПМ са слика 6.1. и 6.3, са почетним маркирањем $M = [0, 0, 1]$.



Слика 6.6. ИПМ за ПМ са слика 6.1. и 6.3.

Мреже на слици 6.6. представљају ИПМ логичких кола типа И и ИЛИ, тако да се ИПМ за дато СН може формирати директно, трансформацијом његових логичких кола у одговарајуће ИПМ структуре. ИПМ за неко СН ће бити инверзна мрежа

ЕПМ датог СН. Како сваком логичком колу СН одговара један посредни догађај, сваки прелаз у ИПМ ће имати тачно једно улазно место, односно излазну грану. Значи да је ИПМ такође ФС мрежа.

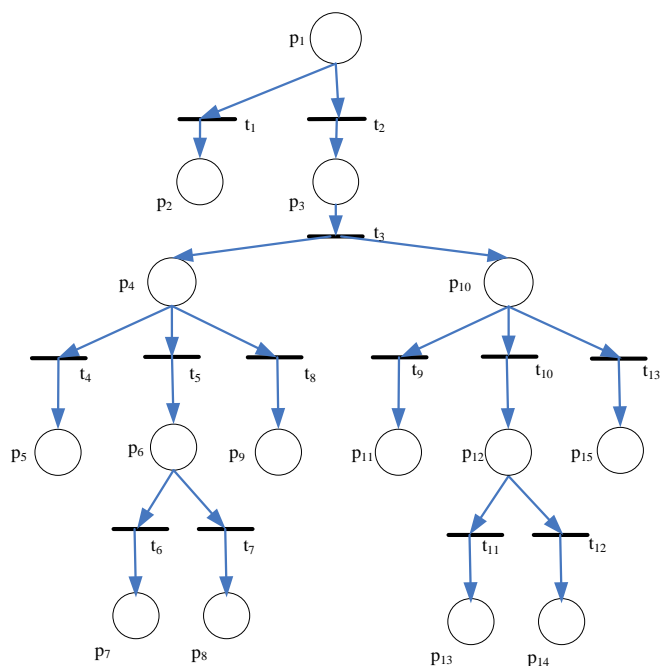
На основу особина ЕПМ и својства ИПМ, могу се дефинисати значајне особине ИПМ датог СН.

Својство 6.2: Нека је ИПМ инверзна ЕПМ датог СН. Основне особине ИПМ су:

1. ИПМ је Петријева мрежа у којој су примарни догађаји датог СН моделирани местима ушћа, а вршни догађај изворним местом.
2. Сваки скуп пресека датог СН се може представити маркирањем ИПМ са жетонима у местима ушћа, која моделирају одговарајуће примарне догађаје. Такво маркирање је мртво.
3. Свако маркирање $M = [1, M(p_2), \dots, M(p_m)]^T$, односно маркирање у коме је $M(p_1) = 1$, моделира одигравање вршног догађаја, што представља отказ система.
4. Свако мртво маркирање M_0 са жетонима само у местима ушћа, које је достижно из маркирања $M = [1, M(p_2), \dots, M(p_m)]^T$, представља скуп пресека.
5. Свако мртво маркирање M_0 са жетонима само у местима ушћа, које је достижно из маркирања $M = [1, 0, \dots, 0]^T$, представља минипресек.

Пример

На слици 6.7. је представљена ИПМ за СН на слике 2.4. То је ЕПМ са слике 6.5, са супротно оријентисаним гранама. С обзиром да се у овом поглављу разматрају само СН без вишеструких догађаја, сва понављања неког вишеструког догађаја овде су третирана као посебни догађаји. Због тога ће се као минипресеци добити свих 17 скупова пресека датог СН.



Слика 6.7. ИПМ за СН са слике 2.4.

С обзиром да ће се ИПМ увек посматрати са почетним маркирањем $M = [1, 0, \dots, 0]^T$, и с обзиром на то да је ИПМ стабло, максималан број жетона у сваком месту је 1, односно за свако место p је $M(p) \leq 1$. То ИПМ чини и *1-safe* мрежом (видети 4.2.1).

Централна тачка предложене методе је у следећем тврђењу.

Тврђење 6.2: Проблем налажења свих минипресека датог СН своди се на следећи директни проблем достижности: наћи сва крајња маркирања M_0 ИПМ која су достижна из маркирања $M = [1, 0, \dots, 0]^T$ као почетног, односно таква да важи $M_0 \in R(M)$, $M = [1, 0, \dots, 0]^T$, где је маркирање у коме је $M(p_1) = 1$ и p_1 моделира одигравање вршног догађаја.

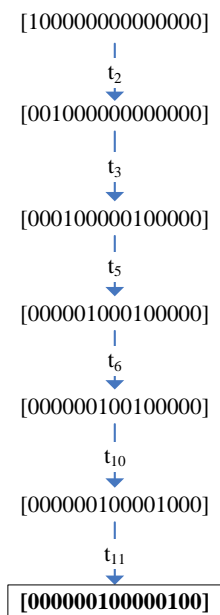
6.4. Одређивање минипресека помоћу стабла достижности

Као што је већ поменуто, проблем одређивања свих маркирања достижних из почетног маркирања може се решити помоћу графа достижности. У случају ИПМ,

маркирање $M = [1, 0, \dots, 0]^T$ је корен, а мртва маркирања која моделирају минипресеке датог СН су листови графа достижности.

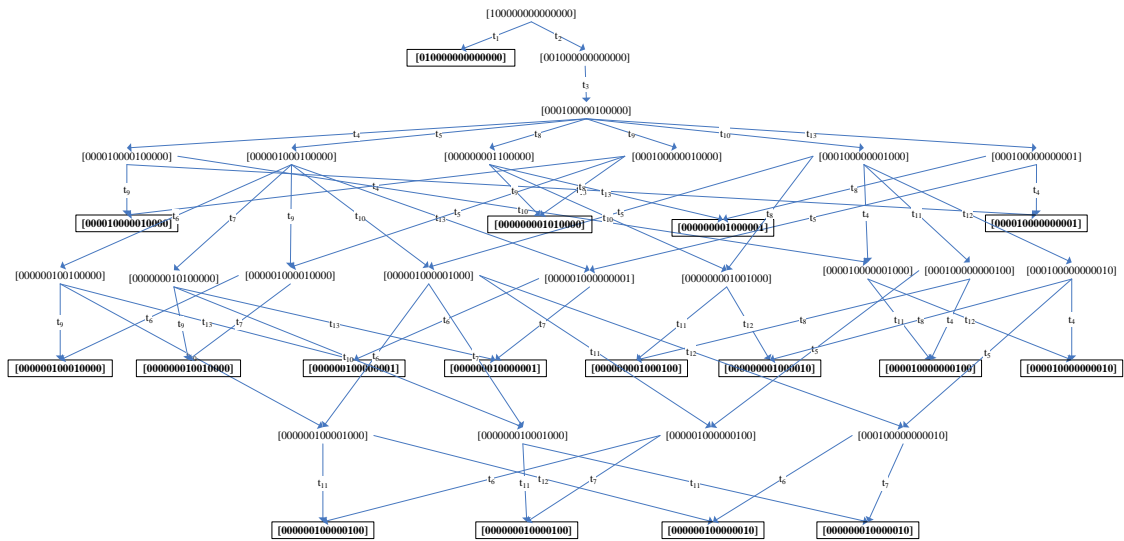
Пример

Посматрајмо ИПМ са слике 6.7. С обзиром да је вршни догађај моделиран местом p_1 , почетно маркирање ће бити $M_0 = [1, 0, \dots, 0]^T$. Посматрајмо и један од минипресека, на пример Р4-Р7, СН са слике 2.4. Крајње маркирање које представља овај минипресек је $M = [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0]^T$. То је маркирање у коме се жетони налазе у местима p_7 и p_{13} , која моделирају примарне догађаје Р4 и Р7. Једна од секвенци паљења, којом се из почетног маркирања M_0 достиже маркирање M је $\sigma = \langle t_2, t_3, t_5, t_6, t_{10}, t_{11} \rangle$. Граф достижности којим се приказује трансформација из M_0 у M секвенцом паљења σ , приказан је на слици 6.8.



Слика 6.8. Граф достижности за секвенцу паљења $\sigma = \langle t_2, t_3, t_5, t_6, t_{10}, t_{11} \rangle$

Одређивањем свих могућих стања ИПМ са слике 6.7. из почетног маркирања $M_0 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$, добија се цео граф достижности (слика 6.9).



Слика 6.9. Граф достижности ИПМ са слике 6.7.

Приказани граф достижности садржи 39 чворова: корен, 21 унутрашњи чвор и 17 листова. Листови су маркирања која одговарају минипресецима датог СН. У табели 6.1. је приказано свих 17 маркирања и одговарајући скупови пресека.

Табела 6.1. Минипресеци СН са слике 2.4. добијених помоћу ИПМ са слике 6.7.

Листови графа достижности	Места ИПМ која садрже жетон	Минипресеци
1. [01000000000000]	p_2	P1
2. [000010000010000]	p_5, p_{11}	P2, P5
3. [000000001010000]	p_9, p_{11}	P3, P5
4. [000000001000001]	p_9, p_{15}	P3, P6
5. [000010000000001]	p_5, p_{15}	P2, P6
6. [000000100010000]	p_7, p_{11}	P4, P7
7. [000000010010000]	p_8, p_{11}	S1, P7
8. [000000100000001]	p_7, p_{15}	P4, P6
9. [000000010000001]	p_8, p_{15}	S1, P6
10. [000000001000100]	p_9, p_{13}	P3, P5
11. [000000001000010]	p_9, p_{14}	P3, S2
12. [000010000000100]	p_5, p_{13}	P2, P5
13. [000010000000010]	p_5, p_{14}	P2, S2
14. [000000100000100]	p_7, p_{13}	P4, P7
15. [000000010000100]	p_8, p_{13}	S1, P7
16. [000000100000010]	p_7, p_{14}	P4, S2
17. [000000010000010]	p_8, p_{14}	S1, S2

С обзиром да се понављања вишеструких догађаја овде посматрају као посебни догађаји, добијено је свих 17 скупова пресека датог СН.

Листови графа достижности ИПМ неког СН су коњункције примарних догађаја. Збир тих листова предствља ДНФ која је енквивалнтна структурној функцији датог СН. Граф достижности може бити великих димензија и за релативно мале ПМ. Такође, ако су потребна само крајња маркирања, у графу постоји велики број грана и међустања, који су непотребни, јер више путева води до истог маркирања. У наставку ће бити показано како се не мора одредити цео граф достижности да би се добила сва мртва маркирања, већ само једно његово разапињуће стабло. Да би се то омогућило, потребно је увести посебан начин нумерисања ИПМ.

ИПМ ће бити нумерисана у дубину и с леве на десну страну, тј. “при-ордер” (КЛД корен – лево подстабло – десно подстабло [134], енг. *pre-order* [51]), као што је нумерисана ИПМ са слике 6.7. С обзиром да је ИПМ бипартитни граф, чворови типа место и чворови типа прелаз ће бити различито нумерисани. Како у бипартитном графу, чворови истог типа нису међусобно повезани, између њих мора прво да се успостави поредак.

- Приликом нумерисања места, за два места p_i и p_j сматраће се да су суседна уколико постоји прелаз t_q такав да је:

$$p_i = {}^{\circ}t_q \text{ и } t_q = {}^{\circ}p_j \quad (6.1)$$

односно ако је p_i улазно место прелаза t_q , а t_q улазни прелаз места p_j . Дакле, ако у ИПМ постоје гране $p_i - t_q - p_j$, место p_i ће представљати претходника места p_j .

- Аналогно, приликом нумерисања прелаза, за два прелаза t_q и t_r сматраће се да су суседни уколико постоји место p_j такво да је:

$$t_q = {}^{\circ}p_j \text{ и } p_j = {}^{\circ}t_r \quad (6.2)$$

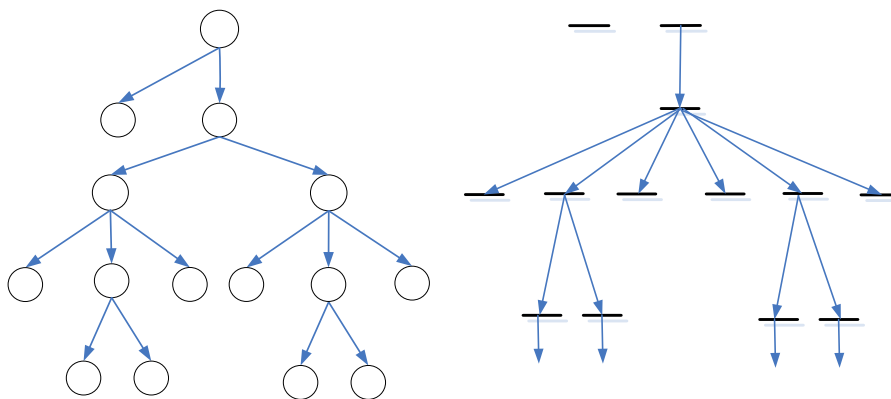
и тада ће прелаз t_q представљати претходника прелаза t_r .

Напомена 6.1: У изразима (6.1) и (6.2) свако место (осим корена) има тачно један улазни прелаз и сваки прелаз има тачно једно улазно место. Услов (6.1) се може краће записати $p_i \in {}^\circ({}^\circ p_j)$, а услов (6.2) $t_q \in {}^\circ({}^\circ t_r)$.

На овај начин су формирана два подграфа ИПМ. Ови подграфови су два стабла или стабло и шума. Подграф чији су чворови типа место је увек стабло чији је корен место које моделира вршни догађај. Подграф чији су чворови типа прелаз је стабло, уколико вршном догађају СН одговара логичко коло И, односно шума, уколико вршном догађају СН одговара логичко коло ИЛИ.

Пример

На слици 6.10. су приказани подграфови ИПМ са слике 6.7.



Слика 6.10. Подграфови ИПМ са слике 6.7.

Са слике се може уочити да је подграф прелаза шума, зато што вршном догађају са слике 6.7. одговара логичко коло ИЛИ.

Нумерисање у дубину се изводи аналогно претрази у дубину [139]. Нумерисање у дубину има три последице које су важне за развој методе за одређивање минипресека помоћу ИПМ.

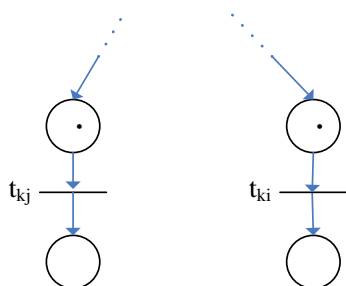
Последица 6.1: С обзиром да су све гране ИПМ оријентисане од корена ка листовима, на свакој грани ће за свака два суседна места важити $num(p_i) < num(p_j)$, ако важи $p_i \in {}^\circ({}^\circ p_j)$ (место p_i је претходник места p_j),

односно за свака два суседна прелаза важиће да је $num(t_q) < num(t_r)$, ако важи $t_q = \circ(\circ t_r)$ (прелаз t_q је претходник прелаза t_r).

Последица 6.2: У свакој секвенци паљења $\sigma = \langle t_{k_1}, t_{k_2}, \dots, t_{k_r} \rangle$ која води од корена графа достижности до неког његовог листа, односно која одговара једном путу ИПМ, важи да је $k_1 < k_2 < \dots < k_r$.

У складу са уведеном нотацијом, за било која два места на истом путу од листа до корена ИПМ може се записати да важи: $p_i = \circ \dots \circ(\circ p_j)$ (место p_i је индиректни претходник места p_j), односно за било која два прелаза на истом путу важи $t_q = \circ \dots \circ(\circ t_r)$ (прелаз t_q је индиректни претходник прелаза t_r).

Последица 6.3: Ако у некој секвенци паљења $\sigma = \langle t_{k_1}, \dots, t_{k_i}, t_{k_j}, \dots, t_{k_r} \rangle$ ИПМ за два суседна прелаза важи да је $k_i > k_j$, онда, на основу последице 6.1, прелази t_{k_i} и t_{k_j} нису на истом путу, односно $t_{k_i} \neq \circ \dots \circ(\circ t_{k_j})$, већ се ова два прелаза налазе на различитим путевима ИПМ (слика 6.11).



Слика 6.11. Прелази на различитим путевима ИПМ

Како је $t_{k_i} \in \sigma$, $t_{k_j} \in \sigma$, $t_{k_i} \neq \circ \dots \circ(\circ t_{k_j})$, онда важи $M(\circ t_{k_i}) = 1$ и $M(\circ t_{k_j}) = 1$ (улазна места оба прелаза садрже жетон). То значи да ће, након паљења прелаза t_{k_i} и даље важити $M(\circ t_{k_j}) = 1$, односно прелаз t_{k_j} је и даље омогућен. Ако се прво запали прелаз t_{k_j} и даље ће важити $M(\circ t_{k_i}) = 1$, односно прелаз t_{k_i} ће и даље бити омогућен. То, даље, значи да ова два прелаза могу да се запале у било ком редоследу, а да то не утиче на наредна маркирања ИПМ. Транспозицијом ова два

суседна прелаза у секвенци паљења, добија се секвенца $\sigma = \langle t_{k_1}, \dots, t_{k_j}, t_{k_i}, \dots, t_{k_r} \rangle$, таква да је $k_1 < \dots < k_j < k_i < \dots < k_r$.

Познато је да свака пермутација може да се добије као низ транспозиција суседних елемената [105].

На основу претходног доказано је следеће тврђење: Свака допустива секвенца паљења ИПМ има пермутацију $\sigma = \langle t_{k_1}, t_{k_2}, \dots, t_{k_r} \rangle$ такву да је $k_1 < k_2 < \dots < k_r$.

Пример

На графу достижности са слике 6.9. се може уочити да се до појединих листова (мртвих маркирања ИПМ) долази помоћу више путева. Поред тога, на слици 6.8. је на графу достижности показано како се мртво маркирање $M = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0]^T$, које представља минипресек P4-P7, може достићи из почетног маркирања $M_0 = [1, 0, \dots, 0]^T$, паљењем секвенце $\sigma_1 = \langle t_2, t_3, t_5, t_6, t_{10}, t_{11} \rangle$. Међутим, M се може достићи из M_0 и помоћу неких пермутација секвенце σ_1 : $\sigma_2 = \langle t_2, t_3, t_5, t_{10}, t_6, t_{11} \rangle$ и $\sigma_3 = \langle t_2, t_3, t_5, t_{10}, t_{11}, t_6 \rangle$.

Свакој секвенци паљења $\sigma = \langle t_{k_1}, t_{k_2}, \dots, t_{k_r} \rangle$ одговара вектор паљења V_σ . Како је V_σ вектор од n елемената, чији су елементи k_1, k_2, \dots, k_r једнаки 1, а остали елементи једнаки 0, он је исти за све пермутације секвенце паљења $\sigma = \langle t_{k_1}, t_{k_2}, \dots, t_{k_r} \rangle$.

Пример

Секвенци паљења $\sigma_1 = \langle t_2, t_3, t_5, t_6, t_{10}, t_{11} \rangle$ са слике 6.8. и свим њеним пермутацијама одговара исти вектор паљења $V_{\sigma_1} = [0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0]^T$.

Прелаз из почетног маркирања M_0 у маркирање M секвенцом паљења σ се може описати једначином стања (4.6):

$$M = M_0 + A \cdot V_\sigma.$$

Како V_σ одговара свим пермутацијама секвенце паљења $\sigma = \langle t_{k_1}, t_{k_2}, \dots, t_{k_r} \rangle$, за сваку допустиву секвенцу која је пермутација секвенце паљења σ важиће једначина стања (4.6), односно сваком допустивом секвенцом паљења која представља пермутацију секвенце паљења σ се такође из почетног маркирања M_0 достиже маркирање M . Одавде следи да је, ако је потребно из почетног маркирања M_0 достићи маркирање M , довољно одредити само једну од допустивих пермутација које одговарају вектору паљења V_σ .

На основу претходног важи:

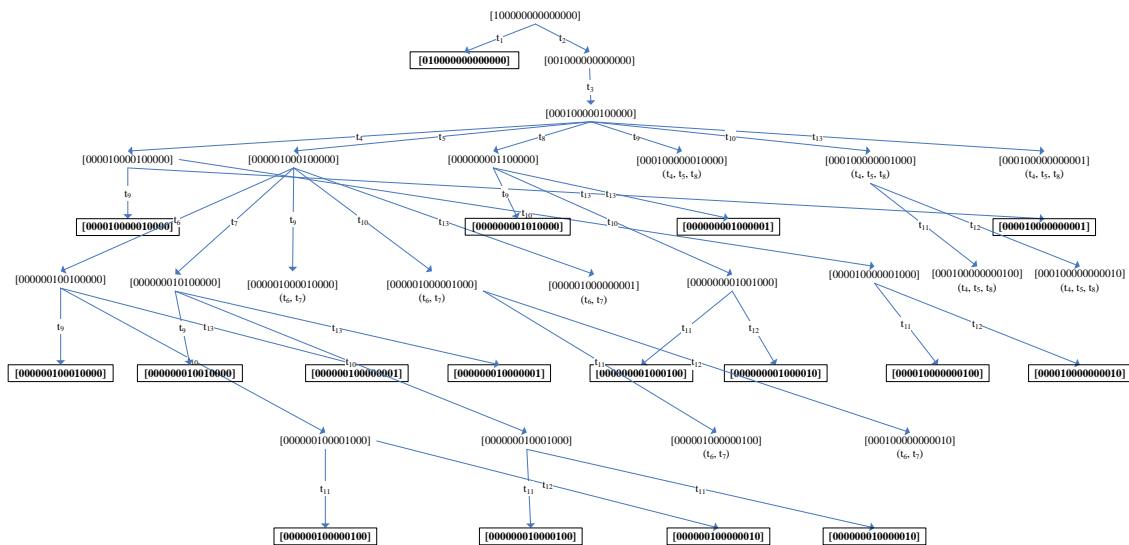
Тврђење 6.3: Нека су у ИПМ чворови означени по дубини и нека је дато почетно маркирање $M_0 = [1, 0, \dots, 0]$. Добијањем свих допустивих секвенци паљења $\sigma = \langle t_{k_1}, t_{k_2}, \dots, t_{k_r} \rangle$ за које важи $k_1 < k_2 < \dots < k_r$, добијају се сва допустива маркирања ИПМ.

Последица 6.4: Граф достижности ИПМ, који се добија одређивањем само допустивих секвенци паљења $\sigma = \langle t_{k_1}, t_{k_2}, \dots, t_{k_r} \rangle$ за које важи $k_1 < k_2 < \dots < k_r$, је стабло.

Пошто ће у наставку увек бити одређивано стабло достижности, у наставку ће се увек користити тај термин.

Пример

На слици 6.12. је приказано стабло достижности за ИПМ са слике 6.7, које се добија одређивањем само секвенци паљења $\sigma = \langle t_{k_1}, t_{k_2}, \dots, t_{k_r} \rangle$ за које важи $k_1 < k_2 < \dots < k_r$.



Слика 6.12. Стабло достижности ИПМ са слике 6.7.

Приказано стабло достижности садржи исте чворове као граф достижности на слици 6.9: Прелази у заградама испод чворова представљају гране графа достижности које су исечене, јер одговарајући прелаз нема мањи индекс од индекса прелаза на свим улазним гранама посматраног чвора. На тај начин је добијено нових осам листова графа достижности. Међутим, ови листови одговарају маркирањима која нису мртва, јер у њима има омогућених прелаза, односно у њима постоје жетони у местима која нису листови одговарајуће ИПМ.

Напомена 6.2: На овај начин се и даље одређује цео простор стања али је време његовог одређивања скраћено.

Напомена 6.3: Само листови графа достижности који одговарају мртвим маркирањима ИПМ представљају минипресеке датог СН. То су маркирања у којима само листови ИПМ, односно места која моделирају примарне догађаје, имају жетон.

6.5. Алгоритам за одређивање свих минипресека

Алгоритмом за одређивање свих минипресека кохерентног СН без вишеструких догађаја, се одређује разпињуће стабло графа достижности ИПМ за које важи да

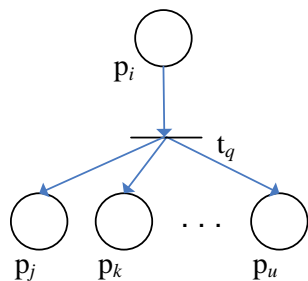
је на сваком путу од корена стабла до листа $k_1 < k_2 < \dots < k_r$, где су k_1, k_2, \dots, k_r индекси прелаза секвенце паљења $\sigma = \langle t_{k_1}, t_{k_2}, \dots, t_{k_r} \rangle$. Стабло достижности се формира приликом претраживања ИПМ датог СН.

Претраживање ИПМ врши се обилажењем чворова типа место. Свако место p_j ИПМ описано је помоћу свог прелаза претходника и места претходника, односно представљено је тројком (i, q, j) где су:

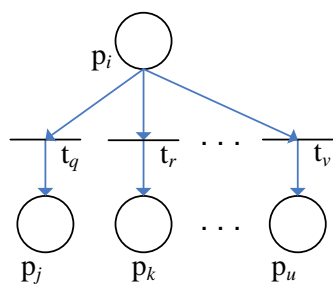
- q индекс улазног прелаза места p_j , тј, $t_q = {}^{\circ}p_j$.
- i индекс улазног места прелаза t_q , тј, $p_i = {}^{\circ}t_q$ или $p_i = {}^{\circ}({}^{\circ}p_j)$.
- j индекс места p_j

На овај начин је потпуно описана структура ИПМ. Корен ИПМ се представља тројком $(0, 0, 1)$. Уколико је место p_j лист, неће постојати ни једна тројка у којој је први елемент једнак j .

Ако места p_j, p_k, \dots, p_u ИПМ моделирају догађаје СН који су у И вези и која следе место p_i , биће представљени тројкама: (i, q, j) , (i, q, k) , \dots , (i, q, u) , односно прва два елемента тројки које их представљају су иста (слика 6.13). Ако места p_j, p_k, \dots, p_u ИПМ моделирају догађаје СН, који су у ИЛИ вези и која следе место p_i , биће представљени тројкама: (i, q, j) , (i, r, k) , \dots , (i, v, u) , односно само први елемент тројки које их представљају је исти (слика 6.14).



Слика 6.13. ИПМ И везе



Слика 6.14. ИПМ ИЛИ везе

Поред структуре, сваку ПМ карактерише и маркирање, односно дистрибуција жетона у мрежи. Маркирање ИПМ је представљено, на уобичајени начин, вектором $M = [M(p_1), M(p_2), \dots, M(p_m)]^T$.

Улазни подаци који се користе у алгоритму су:

- тројке (i, q, j) , $j = 1, \dots, m$, којима је описана структура ИПМ и
- почетно маркирање $M_0 = [1, 0, \dots, 0]$, које одговара одигравању вршног догађаја.

Параметри и скупови који се користе у алгоритму су:

- k, r - индекс активног места и активног прелаза;
- R - скуп места за која важи: $p_j = (p_i \circ)^{\circ}$. Овај скуп представља скуп оних места следбеника активног места (тројке) k , чији је индекс улазног прелаза већи или једнак индексу активног прелаза, односно $R = \{(k, q, j) \mid q \geq r\}$. На тај начин се одређују само секвенце паљења $\sigma = \langle t_{k_1}, t_{k_2}, \dots, t_{k_r} \rangle$, за које важи $k_1 < k_2 < \dots < k_r$. Ако је $P \neq \emptyset$, активно место представља посредни догађај, а ако је $P = \emptyset$, активно место представља примарни догађај датог СН, односно активно место је лист ИПМ;
- L - скуп индекса места који су листови ИПМ у тренутној итерацији. Овај скуп се може директно превести у одговарајуће маркирање јер је $M(p_j) = 1, \forall j \in L$, односно $M(p_j) = 0, \forall j \notin L$;

- R_{\wedge} - скуп места из R која су у И вези, $R_{\wedge} = \{(i, q, j) \mid q = q^*\}$, где је q^* индекс улазног прелаза активног места. Након паљења активног прелаза, сва места из R_{\wedge} добијају жетон $M(p_j) = 1$, $(\forall (i, q, j) \in R_{\wedge})$.
- $R \setminus R_{\wedge}$ - скуп места из R која су у ИЛИ вези. Пошто су у кохерентном СН логичка кола типа И или типа ИЛИ, у свакој итерацији ће један од скупова R_{\wedge} и $R \setminus R_{\wedge}$ бити празан. Међутим, ако се изврши редукција ИПМ, фузијом места, може се десити да ни један од поменутих скупова не буде празан;
- Q_{\wedge} - скуп парова: места у ИПМ која у тренутној итерацији садрже жетон и која нису у R_{\wedge} , односно $Q_{\wedge} = \{(i, q, j) \mid j = 1, j \notin L\}$. То је скуп примарних и/или посредних догађаја чији су излазни прелази омогућени. Када се дође до листа ИПМ, који се смешта у L , потребно је изабрати ново активно место. То место се бира из Q_{\wedge} тако да је $j^* = \min_{(i, q, j) \in Q_{\wedge}} j$. Ако је $Q_{\wedge} = \emptyset$, дошло се до листа стабла достижности, односно добијен је један скуп пресека, а индекси места која представљају тај скуп пресека су садржани у L ;
- Q_{\vee} - скуп места која су у ИЛИ вези са активним местом и важећег маркирања када је место постало активно, $Q_{\vee} = Q_{\vee} \cup \{[(i, q, j), M_j] \mid (i, q, j) \in R, [(i, q, j)]\}$. Када се дође до листа стабла достижности, потребно је изабрати ново активно место. Оно се бира из Q_{\vee} тако да је $j^* = \min_{\text{lex } M_j} \min_{(i, q, j) \in Q_{\vee}} j$. Исто место (тројка) може бити додељено скупу Q_{\vee} у различитим итерацијама са различитим маркирањима. Због тога је уведено $\text{lex } M_j$, односно у том случају активно постаје место са лексикографски мањим маркирањем. Ако је $Q_{\vee} = \emptyset$ добијени су сви скупови пресека и алгоритам стаје.
- SP - скуп скупова пресека (излаз из алгоритма);

Алгоритам за одређивање минипресека кохерентног СН без вишеструких догађаја се може сажети у следеће три фазе:

- Резултат *else* гране Корака 1 је један лист ИПМ;

- Резултат *else* гране Корака 3 је један лист стабла достижности који представља један минипресек;
- Резултат *else* гране Корака 4 су сви листови стабла достижности, односно сви минипресеци датог СН.

Алгоритам ИПМ_1

Почетак:

$(0,0,1)$, $k := 1$, $r := 0$, $M := (1,0,\dots,0)$, $R := \emptyset$, $R_\wedge := \emptyset$, $Q_\wedge := \emptyset$, $Q_\vee := \emptyset$, $L := \emptyset$,
 $SP := \emptyset$

Корак 1:

$R := \{(k,q,j) \mid q \geq r\}$;

If $R \neq \emptyset$ then Корак 2 else $L := L \cup \{j\}$; Go to Корак 3.

Корак 2:

$q^* := \min_{(i,q,j) \in R} q$;

$R_\wedge := \{(i,q,j) \mid q = q^*\}$;

$R := R \setminus R_\wedge$;

$M(p_k) := 0$;

$(\forall (i,q,j) \in R) (M(p_j) := 1, M_j := M; M(p_j) := 0)$

$Q_\vee := Q_\vee \cup \{[(i,q,j), M_j] \mid (i,q,j) \in R\}$;

$R := \emptyset$;

$(\forall (i,q,j) \in R_\wedge) M(p_j) := 1$;

$j^* := \min_{(i,q,j) \in P_\wedge} j$;

$k := j^*$;

$r := q$;

$R_\wedge = \emptyset$; Go to Корак 1.

Корак 3:

$$Q_{\wedge} := \{(i, q, j) \mid M(j) = 1, j \notin L\};$$

$$\text{If } Q_{\wedge} \neq \emptyset \text{ then } j^* := \min_{(i, q, j) \in Q_{\wedge}} j;$$

$$k = j^*; r = q; Q_{\wedge} = \emptyset; \text{ Go to Корак 1.}$$

$$\text{else } SP = SP \cup \{L\};$$

Корак 4:

$$\text{If } Q_{\vee} \neq \emptyset \text{ then } j^* = \min_{\text{lex } M_j} \min_{(i, q, j) \in Q_{\vee}} j;$$

$$k = i; r = q; M = M_j;$$

$$Q_{\vee} = Q_{\vee} \setminus \{(i, q, j), M_j\}, L = \emptyset; \text{ Go to Корак 1;}$$

else Крај.

Резултат алгоритма је скуп SP - скуп свих минипресека. Због услова $q \geq r$, при формирању скупа R у Кораку 1, листови графа достижности који представљају одсечена маркирања се неће генерисати.

Пример

У наставку ће бити илустровано како се алгоритмом ИПМ_1 добијају минипресеци СН са слике 2.4. на основу стабла достижности ИПМ са слике 6.7.

Улазни подаци који се користе у алгоритму су:

- тројке $(0,0,1), (1,1,2), (1,2,3), (3,3,4), (4,4,5), (4,5,6), (6,6,7), (6,7,8), (4,8,9), (3,3,10), (10,9,11), (10,10,12), (12,11,13), (12,12,14)$ и $(10,13,15)$;
- почетно маркирање $M_0 = [1, 0, \dots, 0]^T$.

Напомена: Маркирања ће у алгоритму бити означена као m -торке.

Почетак: $(0,0,1), k = 1, r = 0, M = (1000000000000000), R = \emptyset, R_{\wedge} = \emptyset, Q_{\wedge} = \emptyset,$
 $Q_{\vee} = \emptyset, L = \emptyset, SP = \emptyset.$

Корак 1: $R = \{(1,1,2), (1,2,3)\}$;

Корак 2: $q^* = \min_{(i,q,j) \in R} \{1, 2\} = 1$; $R_{\wedge} = \{(1,1,2)\}$; $M(p_1) = 0$; $R = \{(1,2,3)\}$; $M(p_3) = 1$, $M_3 =$

0010000000000000); $M(p_3) = 0$, $Q_{\vee} = \{[(1,2,3), M_3]\}$; $R = \emptyset$;

$M(p_2) = 1$, $M = (0100000000000000)$; $j^* = 2$; $k = 2$; $r = 1$, $R_{\wedge} = \emptyset$,

Корак 1: $R = \emptyset$ $L = \{2\}$;

пронађен је лист ИПМ

Корак 3: $Q_{\wedge} = \emptyset$, $SP = \{\{2\}\}$;

пронађен је један минипресек

Корак 4: $j^* = 3$; $k = 1$; $r = 2$; $M = (0010000000000000)$; $Q_{\vee} = \emptyset$, $L = \emptyset$;

Корак 1: $R = \{(3,3,4), (3,3,10)\}$;

Корак 2: $q^* = 3$; $R_{\wedge} = \{(3,3,4), (3,3,10)\}$; $R = \emptyset$; $M(p_3) = 0$; $Q_{\vee} = \emptyset$; $M(p_4) = 1$; $M(p_{10}) = 1$,

$M = (0001000001000000)$; $j^* = k = 4$; $r = 3$; $R_{\wedge} = \emptyset$;

Корак 1: $R = \{(4,4,5), (4,5,6), (4,8,9)\}$;

Корак 2: $q^* = 4$; $R_{\wedge} = \{(4,4,5)\}$; $R = \{(4,5,6), (4,8,9)\}$;

$M(p_4) = 1$; $M_4 = (0000010001000000)$, $M(p_4) = 0$;

$M(p_9) = 1$; $M_9 = (0000000011000000)$; $M(p_9) = 0$;

$Q_{\vee} = \{[(4,5,6), M_5], [(4,8,9), M_9]\}$; $R = \emptyset$;

$M(p_5) = 1$; $M = (0000100001000000)$; $j^* = k = 5$; $r = 4$; $R_{\wedge} = \emptyset$;

Корак 1: $R = \emptyset$ $L = \{5\}$;

пронађен је лист ИПМ

Корак 3: $Q_{\wedge} = \{(3,3,10)\}$; $j^* = k = 10$; $r = 3$; $Q_{\wedge} = \emptyset$;

Корак 1: $R = \{(10,9,11), (10,10,12), (10,13,15)\}$;

Корак 2: $q^* = 9$; $R_{\wedge} = \{(10,9,11)\}$; $R = \{(10,10,12), (10,13,15)\}$; $M(p_{10}) = 0$;

$M(p_{12}) = 1$; $M_{12} = (0000100000010000)$; $M(p_{12}) = 0$;

$$M(p_{15}) = 1; M_{15} = (000010000000001); M(p_{15}) = 0;$$

$$Q_{\vee} = \{[(4,5,6), M_5], [(4,8,9), M_9], [(10,10,12), M_{12}], [(10,13,15), M_{15}]\}; R = \emptyset;$$

$$M(p_{11}) = 1; M = (000010000010000); j^* = k = 11; r = 9; R_{\wedge} = \emptyset;$$

Корак 1: $R = \emptyset$ $L = \{5,11\}$; # пронађен је лист ИПМ

Корак 3: $Q_{\wedge} = \emptyset$, $SP = \{\{2\}, \{5,11\}\}$; # пронађен је још један минипресек

Корак 4: $j^* = 6$; $k = 6$; $r = 5$; $M = (000001000100000)$;

$$Q_{\vee} = \{[(4,8,9), M_9], [(10,10,12), M_{12}], [(10,13,15), M_{15}]\}; R = \emptyset;$$

Корак 1: $R = \{(6,6,7), (6,7,8)\}$;

Даљом применом алгоритма добијају се, редом, елементи скупа SP : $\{7,11\}$, $\{8,11\}$, $\{9,11\}$, $\{5,13\}$, $\{7,13\}$, $\{8,13\}$, $\{9,13\}$, $\{5,14\}$, $\{7,14\}$, $\{8,14\}$, $\{9,14\}$, $\{5,15\}$, $\{7,15\}$, $\{8,15\}$ и $\{9,15\}$.

6.6. Одређивање минипресека задате дужине

С обзиром да број минипресека у сложеним системима може бити веома велики, одређивање свих листова стабла достижности може бити веома сложен, временски и просторно захтеван, комбинаторни проблем. У неким системима, чак и када се могу одредити сви минипресеци, они са малом вероватноћом се могу занемарити. Због тога је у квалитативној анализи СН често довољно или чак потребно одредити само најзначајније минипресеке. Најзначајнији минипресеци су они чија је вероватноћа одигравања највећа. Како је вероватноћа минипресека једнака производу вероватноћа његових примарних догађаја, најзначајнијим минипресецима се могу сматрати они са најмањим бројем примарних догађаја. Ово важи под претпоставком да су вероватноће примарних догађаја истог или приближног реда величине.

Ред минипресека (*order*) представља број његових примарних догађаја [28]. Минипресеци реда l садрже l или мање примарних догађаја. У наставку ће бити

описан приступ у коме се, помоћу стабла достижности ИПМ датог СН, одређују само минипресеци датог реда, а остали се одсецају.

Како број жетона у месту моделира одигравање одговарајућег догађаја датог СН, укупан број жетона у једном маркирању ИПМ представља укупан број догађаја који су се истовремено одиграли. Пошто је свако маркирање ИПМ један чвор стабла достижности, следи да број жетона у сваком чвору стабла достижности ИПМ представља укупан број догађаја датог СН који су се одиграли. Коначно, број жетона у сваком листу стабла достижности, представља број примарних догађаја одговарајућег минипресека, односно његов ред.

Својство 6.2: Број жетона у чворовима стабла достижности ИПМ датог СН је неоппадајући идући од корена $M = [1, 0, \dots, 0]^T$ ка листовима.

Доказ: ИПМ датог СН је стабло: сваки прелаз t има тачно једно улазно место, односно $|{}^0t| = 1$. Број излазних места зависи од типа логичког кола. Ако се прелазом t моделира ИЛИ логичко коло, онда је $|t^0| = 1$, односно прелаз t има једно излазно место. Ако се прелазом моделира И логичко коло, онда је $|t^0| > 1$, односно прелаз t има више излазних места. Како су тежине свих грана у ИПМ једнаке 1, промена маркирања приликом паљења неког прелаза изражена са (4.3), постаје:

$$M'(p) = \begin{cases} M(p) - 1 & \text{ако } p \in {}^0t \\ M(p) + 1 & \text{ако } p \in t^0 \\ M(p) & \text{иначе} \end{cases} \quad (6.3)$$

Дакле, паљењем неког прелаза t , из његовог улазног места се троши, а у његовом излазном месту се производи један, ако је $|t^0| = 1$, или више жетона, ако је $|t^0| > 1$. Другим речима, разлика у броју жетона након и пре паљења прелаза t је увек ≥ 0 .

Нека је дата ИПМ са почетним маркирањем $M = [1, 0, \dots, 0]^T$ и нека је l број жетона у неком маркирању $M_1 \in R(M)$ у коме је омогућен прелаз t . На основу претходног, број жетона у ИПМ након паљења прелаза t је већи или једнак l .

Како сваком маркирању ИПМ одговара један чвор стабла достижности, следи да је, идући од корена $M = [1, 0, \dots, 0]^T$ ка листовима, број жетона у чворовима стабла достижности неопадајући. ♦

Да би се издвојили само минипресеци датог реда, потребно је увести критеријум на основу кога ће се одсецати маркирања која одговарају минипресецима дужим од задатог. На основу својства 6.3, као критеријум одсецања се може усвојити број жетона у чворовима графа достижности.

Алгоритам ИПМ_1 за одређивање минипресека се сада може проширити опцијом одсецања делова графа достижности. Прво је потребно увести нови улазни параметар и додатни излазни скуп:

- l - параметар одсецања
- \overline{SP} - скуп одсечених чворова.

Корак 2 алгоритма се сада модификује на следећи начин:

Корак 2'

$$q^* = \min_{(i,q,j) \in P} q;$$

$$R_{\wedge} = \{(i,q,j) \mid q = q^*\};$$

$$R = R - R_{\wedge};$$

$$M(p_k) = 0;$$

$$(\forall (i,q,j) \in R) (M(p_j) = 1, M_j = M; M(p_i) = 0)$$

$$Q_{\vee} = Q_{\vee} \cup \{(i,q,j), M_j \mid (i,q,j) \in R, [(i,q,j), M_j] \neq Q_{\vee}\};$$

$$R = \emptyset;$$

$$(\forall (i,q,j) \in R_{\wedge}) M(p_j) = 1;$$

$$\text{if } \sum_{j=1}^m M(p_j) \leq l \text{ then } j^* = \min_{(i,q,j) \in P_{\wedge}} j;$$

$k = j^*$; $r = q$; $R_\wedge = \emptyset$; Go to *Корак 1*.

else ($\forall (i, q, j) \in R_\wedge$) $M(p_j) = 0$; $M(p_k) = 1$

$L = L + \{j\}$; $\overline{SP} = \overline{SP} + \{L\}$; $L = L \setminus \{j\}$ $Q_\wedge = \emptyset$; Go to *Корак 4*.

Елементи скупа одсечених чворова \overline{SP} нису минипресеци већ представљају жива маркирања. Ако постоји потреба да се одреде минипресеци неког одсеченог дела графа достижности, довољно је одредити нови граф достижности са одговарајућим живим маркирањем као кореном.

Пример

Нека је за СН са слике 2.4. потребно одредити минипресеке дужине 1. Поред улазних података који су приказани у претходној илустрацији алгоритма, уводи се и параметар одсецања $l = 1$.

Почетак: $(0,0,1)$, $k = 1$, $r = 0$, $M = (1000000000000000)$, $R = \emptyset$, $R_\wedge = \emptyset$, $Q_\wedge = \emptyset$,

$Q_\vee = \emptyset$, $L = \emptyset$, $SP = \emptyset$, $\overline{SP} = \emptyset$, $l = 1$.

Корак 1: $R = \{(1,1,2), (1,2,3)\}$;

Корак 2: $q^* = \min_{(i,q,j) \in R} \{1,2\} = 1$; $R_\wedge = \{(1,1,2)\}$; $M(p_1) = 0$; $R = \{(1,2,3)\}$; $M(p_3) = 1$,

$M_3 = (0010000000000000)$; $Q_\vee = \{[(1,2,3), M_3]\}$; $R = \emptyset$; $M(p_2) = 1$,

$\sum_{j=1}^{15} M(p_j) = 1 \leq l$, $M = (0100000000000000)$; $j^* = 2$; $k = 2$; $r = 1$, $R_\wedge = \emptyset$,

Корак 1: $R = \emptyset$ $L = \{2\}$;

пронађен је лист ИПМ

Корак 3: $Q_\wedge = \emptyset$, $SP = \{\{2\}\}$;

пронађен је један минипресек

Корак 4: $j^* = 3$; $k = 1$; $r = 2$; $M = (0010000000000000)$; $Q_\vee = \emptyset$, $L = \emptyset$;

Корак 1: $R = \{(3,3,4), (3,3,10)\}$;

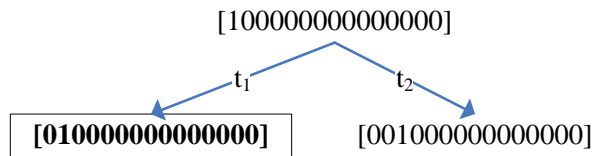
Корак 2: $q^* = 3; R_\wedge = \{(3,3,4), (3,3,10)\}; R = \emptyset; M(p_3) = 0; Q_v = \emptyset; M(p_4) = 1; M(p_{10}) = 1,$

$$\sum_{j=1}^{15} M(p_j) = 2 > l, M(p_4) = 0; M(p_{10}) = 0, M(p_3) = 1, M = (001000000000000),$$

$$L = \{3\}, \overline{SP} = \{\{3\}\}; L = \emptyset \quad \# \text{ одсечен је један чвор}$$

Корак 4: $Q_v = \emptyset$, Крај

Резултат алгоритма су скуп $SP = \{\{2\}\}$ и скуп $\overline{SP} = \{\{3\}\}$ са одсеченим маркирањем $M = (001000000000000)$. Стабло достижности, које је одређено на овај начин, приказано је на слици 6.15.



Слика 6.15. Стабло достижности минипресека дужине 1

С обзиром да се овде посматра СН без вишеструких догађаја, модификовани Корак 2 ће дати тачан резултат, односно све минипресеке датог реда. Међутим, у СН са вишеструким догађајима се, након примене закона апсорпције, број примарних догађаја неког минипресека смањује. У том случају, Својство 6.3 не може директно да се користи, већ морају да се уведу додатни услови, што ће детаљно бити описано у глави 7.

6.7. Резултати експеримената

Да би се тестирао алгоритам и његово проширење, одабрано је 10 СН тако што су су, у изабраним *Benchmark* СН, сва понављања вишеструких примарних догађаја посматрана као различити догађаји. У табели 6.2. су описане карактеристике одабраних СН у смислу: броја логичких кола, броја примарних догађаја, као и

димензије одговарајуће ИПМ датог СН. У последње три колоне приказан је: број минипресека и распон дужина минипресека.

Минипресеци су добијени применом алгоритма ИПМ_1, који је имплементиран у програмском језику *Visual Basic (VB)* [111]. За експерименте је коришћен *Intel Core 2 Duo CPU E8400 (3 GHz), RAM 3,25 GB*.

Табела 6.2. Карактеристике СН, ИПМ и резултат Алгоритма ИПМ_1

СН	СН		ИПМ				
	Број логичких кола	Број примарних догађаја	Број места	Број прелаза	Број минипресека	l_{\min}	l_{\max}
1	40	104	144	124	2292	1	5
2	92	303	394	365	690	1	4
3	34	61	95	81	18541	1	12
4	90	299	389	332	654	1	12
5	31	78	109	95	4582	2	7
6	46	117	163	138	11074	1	8
7	49	118	167	136	1948	2	8
8	88	292	380	347	3603	1	5
9	31	78	109	95	7474	1	8
10	21	59	80	68	1400	3	8

Након одређивања свих минипресека, применом модификације Корака 2 алгоритма, извршено одсецање да би се добили минипресеци реда 1,2,3,4,5,6 и 7. Резултати одсецања, који представљају број минипресека датог реда, су приказани у табели 6.3.

Време потребно за одређивање минипресека је било испод 10 секунди за резултате са ознаком ** и испод 1 секунде за резултате са ознаком *. Сви остали резултати су добијени готово тренутно (< 0.1 секунди).

Табела 6.3. Број минипресека датог реда

CH	Вредност параметра одсецања l						
	1	2	3	4	5	6	7
1	4	155	1319	2291	2292		
2	53	433	438	690			
3	3	13	73	157	469	1069	2269*
4	34	390	395	648	648	649	651
5	0	182	196	1033	1674*	2470*	4582**
6	5	6	18	274	1426	3730**	6178**
7	0	12	480	1320	1840*	1936*	1946*
8	29	376	381	633	3603**		
9	5	6	18	274	274	2578*	2578*
10	0	0	735	840	1085	1330	1330

Поред минипресека датог реда, резултат алгоритма је и скуп одсечених чворова графа достижности (\overline{SP}) и активно маркирање у тренутку одсецања.

7. ОДРЕЂИВАЊЕ МИНИПРЕСЕКА СН СА ВИШЕСТРУКИМ ДОГАЂАЈИМА ПОМОЋУ ИПМ

Метода и алгоритам који су описани у претходној глави, овде ће бити проширени на СН са догађајима чија је вишеструкост већа од или једнака 1. Посебно ће бити разматрана два проблема: како од свих скупова пресека издвојити минималне и како одредити минипресеке задатог реда. У првом случају ће алгоритам ИМП1 бити проширен раздвајањем скупова пресека на скупове K_1 , K_2 и K_3 описане у поглављу 3.3.5, након чега се примењује алгоритам редукције СНЗ. У другом случају је главни задатак одређивање параметра одсецања l , тако да се не изостави ниједан од минипресека задате дужине.

7.1. Одређивање свих минипресека СН

Уколико би се сва понављања истог примарног догађаја датог СН моделирала једним местом, као у 5.1.4 и 5.1.5, ИПМ не би била стабло, чиме би се изгубила многа од својстава коришћених у развоју алгоритма у глави 6. Због тога ће примарни догађаји, чија је вишеструкост већа од 1, бити представљени различитим местима ИПМ. На тај начин ће укупан број места бити једнак

$u' = v + \sum_{i=1}^{u-v} k_i$, где је u број свих догађаја СН, v број логичких кола, $u - v$ број

примарних догађаја у СН а k_i вишеструкост примарног догађаја x_i . Број прелаза ИПМ зависи од логичких кола у СН и може бити најмање v , уколико су сва логичка кола у СН типа И, а највише u' , ако су сва логичка кола типа ИЛИ.

Да би се сачували подаци о вишеструкости, уводи се улазни параметар, који повезује места која моделирају понављања истог примарног догађаја. Нека су местима $P_{s_1}, P_{s_2}, \dots, P_{s_k}$ моделирана понављања датог примарног догађаја вишеструкости k . За такав догађај формира се $(k+1)$ -торка $(k, s_1, s_2, \dots, s_k)$, чији први елемент представља вишеструкост примарног догађаја, а осталих k елемената индексе места којима су моделирана сва његова понављања.

Када се у кораку 3 алгоритма ИПМ_1 добије један скуп пресека, маркирање MC које га моделира, има једно од следећа три својства:

7.1. $\sum_{i=1}^k MC(p_{s_i}) = 0$, односно ниједно од k места која моделирају понављања посматраног догађаја не садржи жетон. За овакав скуп пресека је у поглављу 3.3 доказано да је минипресек.

7.2. $\sum_{i=1}^k MC(p_{s_i}) = k$, односно $M(p_{s_1}) = M(p_{s_2}) = \dots = M(p_{s_k}) = 1$ - сва места која моделирају понављања посматраног догађаја садрже жетон. Последицом 3.4 је и за овакав скуп пресека доказано да, након $k-1$ примена закона идемпотенције: $P \cdot P = P$, постаје минипресек.

7.3. $0 < \sum_{i=1}^k MC(p_{s_i}) < k$, односно нека, али не сва, места која моделирају понављање посматраног догађаја садрже жетон. За овакав скуп пресека треба испитати да ли је минипресек или се елиминише.

Нека у СН постоји n' примарних догађаја чија је вишеструкост већа од 1. Означимо са S скуп свих $(k+1)$ -торки $(k, s_1, s_2, \dots, s_k)$, формираних за такве догађаје. Подсетимо да сваком мртвом маркирању $M = [M(p_1), M(p_2), \dots, M(p_m)]^T$ ИПМ одговара један скуп пресека, тако да примарни догађај припада скупу пресека ако је маркирање места које га представља једнако 1. Сада се скуп свих скупова пресека SP , који се добијају алгоритмом ИМП1, може поделити на три дисјунктна подскупа:

- $MSP0 = \{C \in SP \mid (\forall s \in S) \sum_{i=1}^k MC(p_{s_i}) = 0\}$, који представља скуп свих скупова пресека који имају Својство 7.1. $MSP0$ је скуп $K1$ из алгоритма редукције СН3 из поглавља 3.3.5;

- $MSP = \{C \in SP \setminus MSP0 \mid (\forall s \in S) (\sum_{i=1}^k MC(p_{s_i}) = k \vee \sum_{i=1}^k MC(p_{s_i}) = 0)\}$, скуп свих скупова пресека који имају Својство 7.2, односно таквих да, ако садрже неки примарни догађај чија је вишеструкост већа од 1, онда садрже сва његова

понављања. MSP је скуп К2 из алгоритма редукције СНЗ, тј. скупови пресека из MSP су облика C^* ; и

- $SPR = SP \setminus (MSP0 \cup MSP)$, који представља скуп свих скупова пресека који имају Својство 7.3. SPR је скуп К3 алгоритма АРСНЗ;.

Сада се алгоритам ИПМ_1 може модификовати на следећи начин:

- Улазним подацима алгоритма ИПМ_1 придружује се скуп S .
- У почетном кораку се изоставља скуп $SP = \emptyset$, а додају дефинисани подскупови, тј. $MSP0 = \emptyset$, $MSP = \emptyset$ и $SPR = \emptyset$.
- Корак 3 алгоритма постаје:

Корак 3':

$$Q_{\wedge} = \{(i, q, j) \mid j = 1, j \notin L\};$$

If $Q_{\wedge} \neq \emptyset$ then $j^* = \min_{(i, q, j) \in Q_{\wedge}} j$, $k = j^*$, $r = q$, $Q_{\wedge} = \emptyset$, Go to *Корак1*;

else if $(\forall s \in S) \sum_{i=1}^k M(p_{s_i}) = 0$ then $MSP0 = MSP0 \cup \{L\}$;

else if $(\forall s \in S) (\sum_{i=1}^k M(p_{s_i}) = k \vee \sum_{i=1}^k M(p_{s_i}) = 0)$ then $MSP = MSP \cup \{L\}$;

else $SPR = SPR \cup \{L\}$;

Пример

Модификовани алгоритам ИПМ_1 за СН са слике 2.4, на основу стабла достижности ИПМ са слике 6.7, има исти број корака као у примеру у поглављу 6.5. Овде ће бити приказан само корак 3 за дати пример.

Поред постојећих, нови улазни подаци су елементи скупа $S = \{(2,5,11), (2,9,15)\}$. Прва тројка представља опис примарног догађаја Р2 СН са слике 2.4, које је вишеструкости 2 и чија су понављања моделирана местима p_5 и p_{11} . Другом тројком је моделиран примарни догађај Р3, чија су два понављања моделирана местима p_9 и p_{15} .

Почетак: $(0,0,1)$, $k = 1$, $r = 0$, $M = (1000000000000000)$, $R = \emptyset$, $R_\wedge = \emptyset$, $Q_\wedge = \emptyset$,
 $Q_\vee = \emptyset$, $L = \emptyset$, $MSP0 = \emptyset$, $MSP = \emptyset$ и $SPR = \emptyset$.

...

Корак 3: $Q_\wedge = \emptyset$,

$$M(p_5) + M(p_{11}) = 0 \text{ и } M(p_9) + M(p_{15}) = 0,$$

$$MSP0 = \{\{2\}\}; \quad \# \text{ пронађен је скуп пресека са својством 1.}$$

...

Корак 3: $Q_\wedge = \{(3,3,10)\}$; $j^* = k = 10$; $r = 3$; $Q_\wedge = \emptyset$;

...

Корак 3: $Q_\wedge = \emptyset$,

$$M(p_5) + M(p_{11}) = 2 \text{ и } M(p_9) + M(p_{15}) = 0,$$

$$MSP = \{\{5,11\}\}; \quad \# \text{ пронађен је скуп пресека са својством 2.}$$

...

Даљом применом алгоритма добијају се, редом, елементи скупа $MSP0$: $\{7,13\}$, $\{8,13\}$, $\{7,14\}$ и $\{8,14\}$, скупа MSP : $\{9,15\}$ и скупа SPR : $\{7,11\}$, $\{8,11\}$, $\{9,11\}$, $\{5,13\}$, $\{9,13\}$, $\{5,14\}$, $\{9,14\}$, $\{5,15\}$, $\{7,15\}$ и $\{8,15\}$.

Сада се над добијеним скуповима може применити алгоритам АРСНЗ, на исти начин како је на примеру истог СН примењено у поглављу 3.3.2.

7.2. Одређивање минипресека задате дужине

За СН са свим примарним догађајима вишеструкости 1, број жетона у чворовима графа достижности ИПМ је веома природан критеријум одсецања скупова пресека дужих од задате вредности l , односно одређивања само скупова пресека реда l . Сваки чвор стабла достижности представља једно маркирање ИПМ, односно скуп посредних и/или примарних догађаја. Подсетимо да је предложена метода одређивања минипресека помоћу стабла достижности ИПМ типа одозго на ниже и да је доказано да је број жетона у чворовима графа достижности неоппадајући. То

значи да, када се достигне маркирање M' у коме је $\sum_{i=1}^m M'(p_i) > l$, даљом

применом алгоритма се могу добити само маркирања M у којима је такође $\sum_{i=1}^m M(p_i) > l$, па се одговарајућа грана стабла достижности може “одсећи”.

Међутим, у СН са примарним догађајима чија је вишеструкост већа од 1, над скуповима пресека се примењује закон идемпотенције, након чега се њихова дужина смањује. Нека је, на пример, дат скуп пресека*:

$$x_1 \cdot \dots \cdot x_{j-1} \cdot x_j^{k_j} \cdot x_{j+1} \cdot \dots \cdot x_u$$

Овај скуп пресека садржи $u + k_j - 1$ примарних догађаја, међу којима је k_j понављања догађаја x_j . Нека је параметар одсецања $u \leq l < u + k_j - 1$. Директном применом корака 2' из поглавља 6.6, овај скуп пресека не би био одређен. Међутим, након $k_j - 1$ примене закона идемпотенције, дати скуп пресека постаје:

$$x_1 \cdot \dots \cdot x_{j-1} \cdot x_j \cdot x_{j+1} \cdot \dots \cdot x_u$$

чиме се његова дужина скраћује на u . Другим речима, дати скуп пресека јесте реда l и треба да буде одређен.



Како су скупови пресека комбинације примарних догађаја који изазивају отказ система, ниједна од њих не сме бити изостављена. Због тога у одређивању параметра l треба узети у обзир и будућу примену закона идемпотенције.

7.2.1. Одређивање параметра одсецања у СН

Нека је $G_v, v \in V$, посредни догађај датог СН и $P[G_v]$ (као у 3.3.2) његов домен, односно мултискуп свих примарних догађаја, који су непосредни или посредни улаз логичког кола чији је излазни догађај G_v . Нека је U' скуп примарних догађаја, чија је вишеструкост већа од 1. Претпоставимо прво да у $P[G_v]$ постоји само један догађај из U' , нпр. x_i вишеструкости k_i и да је k_i^v број понављања примарног догађаја x_i у $P[G_v]$.

* У првом делу наредног излагања биће коришћена нотација СН из главе 3, а на крају ће главни закључци бити у поглављима 7.2.2 и 7.2.3 преведени у нотацију Петријевих мрежа.

На основу тврђења 3.1, ако је за сваки пар понављања догађаја x_i минимално логичко коло G_{ij} типа И, у структурној функцији подстабла са кореном G_v , постоји скуп пресека облика C^* , односно скуп пресека са свих k_i^v понављања догађаја x_i . Нека посматрани скуп пресека садржи укупно k примарних догађаја. Након $k_i^v - 1$ примене закона идемпотенције, посматрани скуп пресека ће имати $k - (k_i^v - 1)$ елемената. Другим речима, скуп пресека облика C^* ће бити скраћен за $k_i^v - 1$. Међу осталим скуповима пресека постоји још скупова пресека који садрже више понављања догађаја x_i и који ће такође бити скраћени применом закона идемпотенције, али за мање од $k_i^v - 1$. За скуп пресека C^* ћемо рећи да је максимално скраћени скуп пресека, а $k_i^v - 1$ максимално скраћење скупова пресека у подстаблу са кореном G_v .

Уколико у подстаблу са кореном G_v не важи услов из Тврђења 3.1, број елемената у максимално скраћеном скупу пресека применом идемпотенције, је мањи од $k_i^v - 1$. У том случају је могуће одредити максималан број понављања догађаја x_i у $P[G_v]$, таквих да је за сваки њихов пар минимално логичко коло  типа И. Означимо га са . Тада, у структурној функцији подстабла са кореном G_v , постоји скуп пресека са k_i^{vmax} понављања догађаја x_i , у коме ће се применити $k_i^{vmax} - 1$ закон идемпотенције. Дакле, $k_i^{vmax} - 1$ је у том случају максимално скраћење скупова пресека у подстаблу са кореном G_v . Ако су сва посматрана минимална логичка кола типа ИЛИ, $k_i^{vmax} = 1$ и тада је $k_i^{vmax} - 1 = 0$, односно неће постојати ниједан скуп пресека у коме се може применити закон идемпотенције.

Нека су G_v и G_w , $v, w \in V$ два посредна догађаја, таква да $G_v \notin P[G_w]$ и $G_w \notin P[G_v]$ и нека је k_i^{vmax} и k_i^{wmax} максималан број понављања догађаја x_i у $P[G_v]$ и $P[G_w]$, таквих да је за сваки њихов пар минимално логичко коло G_{ij} типа И. Нека је (слично као у 3.3.3) G_{vw} минимално логичко коло догађаја G_v и G_w ,

односно корен минималног подстабла које садржи дате посредне догађаје. Ако посредном догађају G_{vw} одговара логичко коло типа И, у структурној функцији подстабла са кореном G_{vw} , постоји скуп пресека са k примарних догађаја у коме је $k_i^{v\max} + k_i^{w\max}$ понављања примарног догађаја x_i . Након примене закона идемпотенице, број елемената датог скупа пресека постаће $k - (k_i^{v\max} + k_i^{w\max} - 1)$, односно максимално скраћење је $k_i^{v\max} + k_i^{w\max} - 1$. Ако посредном догађају G_{vw} одговара логичко коло типа ИЛИ, максималан број понављања догађаја x_i у $P[G_{vw}]$ једнак је $\max\{k_i^{v\max}, k_i^{w\max}\}$.

У општем случају, нека је V' скуп индекса посредних догађаја таквих да за сваки пар $G_v, G_w, v, w \in V'$ важи $G_v \notin P[G_w]$ и $G_w \notin P[G_v]$ и нека је $k_i^{v\max}, v \in V'$ максималан број понављања догађаја x_i у $P[G_v], v \in V'$, таквих да је за сваки њихов пар минимално логичко коло G_{ij} типа И. Нека је G минимално логичко коло посредних догађаја $G_v, v \in V'$ типа И. Тада, у структурној функцији подстабла са кореном G , постоји скуп пресека са k примарних догађаја, у коме је $\sum_{v \in V'} k_i^{v\max}$ понављања примарног догађаја x_i . Након примене закона идемпотенице, број елемената датог скупа пресека постаје $k - (\sum_{v \in V'} k_i^{v\max} - 1)$. Дакле, максимално скраћење које се може постићи у подстаблу са кореном G је:

$$\sum_{v \in V'} k_i^{v\max} - 1 \quad (7.1).$$

Ако је минимално логичко коло посредних догађаја $G_v, v \in V'$, G типа ИЛИ, максималан број понављања догађаја x_i у $P[G]$ једнак је

$$\max_{v \in V'} \{k_i^{v\max}\}. \quad (7.2)$$

Напомена 7.1: Примарни догађаји, чија је вишеструкост већа од 1, овде ће такође бити посматрани као посредни догађаји, који су излаз логичког кола типа И и који сами чине свој домен, тј. $P[x_j] = x_j, j \in U$. За ове догађаје важи да је $k_j^{j\max} = 1$. Зато ће се у наставку под скупом V подразумевати $V \cup U$.

Претходно разматрање односило се на случај када се у домену посматраних логичких кола налази само један примарни догађај са вишеструкошћу већом од 1. Нека је сада $U'' \subset U'$ скуп свих примарних догађаја x_i , чија је вишеструкост k_i већа од 1 и који припадају домену датог G_v , тј. $x_i \in U''$ и $x_i \in P[G_v]$. Нека је k_i^v број појављивања догађаја x_i у $P[G_v]$. Ако за сва понављања догађаја $x_i \in U''$ у $P[G_v]$ важи услов из Тврђења 3.1, у структурној функцији подстабла са кореном G_v , постоји скуп пресека облика C^* . Тада је максимално скраћење скупова пресека у подстаблу СН са кореном G_v , које се добија применом закона идемпотенције, једнако $\sum_{i \in U''} (k_i^v - 1)$.

Ако у подстаблу са кореном G_v не важи услов из Тврђења 3.1, тада је за сваки од догађаја $x_i \in U''$ у $P[G_v]$, могуће одредити k_i^{vmax} - максималан број понављања догађаја таквих да је за сваки њихов пар минимално логичко коло G_{ij} типа И. Тада је горња граница максималног скраћења у подстаблу са кореном G_v једнака

$$\sum_{i \in U''} (k_i^{vmax} - 1) \quad (7.3)$$

Максимално скраћење ће бити једнако (7.3) само у случају када за свака два примарна догађаја $x_i, x_j \in U''$, њихова понављања која чине k_i^{vmax}, k_j^{vmax} , имају минимално логичко коло типа И. Тада, у структурној функцији подстабла са кореном G_v , постоји скуп пресека облика $\prod_{i \in U''} x_i^{k_i^{vmax}} \cdot \prod_{j \in U} y_j$, где су y_j неки примарни догађаји вишеструкости 1, односно $U \subset U \setminus U''$, који ће бити скраћен за (7.3) догађаја.

Дакле, (7.1) је максимално скраћење за случај једног примарног догађаја у више домена посредних догађаја, чије је минимално логичко коло G типа И. Израз (7.3) је горња граница максималног скраћења за случај више примарних догађаја у домену једног посредног догађаја. Потребно је одредити максимално скраћење (или његову границу) за најопштији случај: више примарних догађаја у више домена посредних догађаја, чије је минимално логичко коло G типа И.

Посматрајмо сада најопштији случај. Нека постоје примарни догађаји $x_i \in U'' \subset U'$ чија је вишеструкост k_i већа од 1 и који припадају доменима датих посредних догађаја $G_v, v \in V'$. Посредни догађаји $G_v, v \in V'$ су такви да је њихово минимално логичко коло G типа И и да за сваки пар $G_v, G_w, v, w \in V'$ важи $G_v \notin P[G_w]$ и $G_w \notin P[G_v]$. Нека је за свако x_i познат $k_i^{v\max}, v \in V'$, максималан број понављања догађаја x_i у $P[G_v], v \in V'$, таквих да је за сваки њихов пар (x_i, x_i) минимално логичко коло G_{ii} типа И.

Спајањем (7.1) и (7.3) добија се горња граница максималног скраћења за општи случај: $\sum_{i \in U''} (\sum_{v \in V'} k_i^{v\max} - 1)$. Претходном дискусијом доказано је следеће тврђење:

Тврђење 7.1: Нека су $G_v, v \in V'$ посредни догађаји датог СН, такви да је њихово минимално логичко коло G типа И и да за сваки пар $G_v, G_w, v, w \in V'$ важи $G_v \notin P[G_w]$ и $G_w \notin P[G_v]$. Скупови пресека подстабла са кореном G се, применом закона идемпотенције, могу скратити највише за:

$$\sum_{i \in U''} (\sum_{v \in V'} k_i^{v\max} - 1) \quad (7.4)$$

где је $i \in U''$ индекс примарног догађаја x_i вишеструкости веће од 1, који припада домену бар једног од посматраних посредних догађаја, а $k_i^{v\max}, v \in V'$ - максималан број понављања догађаја x_i у $P[G_v], v \in V'$, таквих да је за сваки њихов пар (x_i, x_i) минимално логичко коло G_{ii} типа И.

Дакле, прво се за сваки примарни догађај одреди појединачно максимално скраћење, када се само над његовим понављањима примени закон идемпотенције, а затим се одређује збир појединачних максималних скраћења, чиме се добија горња граница скраћења.

Пример

Посматрајмо, на пример, гранични случај када у СН постоји скуп пресека типа $C^* = x_1^{k_1} \cdot x_2^{k_2} \cdot \dots \cdot x_n^{k_n}$ и када је G вршни догађај СН коме одговара логичко коло

типа И. У Напомени 3.2 у поглављу 3.3.2 је већ речено да се над скупом пресека C^* може $\sum_{i=1}^n (k_i - 1)$ пута применити закон идемпотенције. Покажимо да је то вредност која се добија и применом (7.4).

На основу доказа Теореме 3.2., структурна функција СН је облика (3.9):

$$\sum_{C \in D[G_1]} C \cdot \sum_{C \in D[G_2]} C \cdot \dots \cdot \sum_{C \in D[G_u]} C$$

где је $k_i^v \geq 0, i=1, \dots, n, v=1, \dots, u$ број понављања примарног догађаја x_i у домену логичког кола G_v и $\sum_{v=1}^u k_i^v = k_i$. Пошто је посматрани скуп пресека облика C^* , на

основу тврђења 3.1, за сваки догађај x_i и логичко коло G_v , важи $k_i^v = k_i^{v\max}$. Применом (7.4) на посматрано СН, добија се горња граница максималног скраћења његових скупова пресека:

$$\sum_{i=1}^n \left(\sum_{v=1}^u k_i^{v\max} - 1 \right) = \sum_{i=1}^n \left(\sum_{v=1}^u k_i^v - 1 \right) = \sum_{i=1}^n (k_i - 1)$$

као што је и дато у Напомени 3.2.

Вредност (7.4) показује за колико треба повећати параметар одсецања l када се неком од метода одозго на ниже одређују минипресеци задате дужине. На тај начин се обезбеђује да се одреде и сви скупови пресека који након примене закона идемпотенције постају реда $\frac{1}{2}$. Директна метода из поглавља 1.5.1.1 се сада може модификовати увођењем одсецања скупова пресека дужих од l примарних догађаја, на следећи начин:

1. За сваки посредни догађај $G_v, v \in V'$ и сваки примарни догађај $x_i \in U''$ одредити $k_i^{v\max}$ помоћу (7.1), ако је G_v излаз логичког кола И или (7.2) ако је G_v излаз логичког кола ИЛИ;
2. Написати Булове једначине за свако логичко коло.

3. Вршити замену променљивих њиховим изразима, док се не добије вршни догађај као функција само примарних догађаја.

3.1. У свакој ДНФ структурне функције, за сваку коњункцију, одредити број чинилаца l' ;

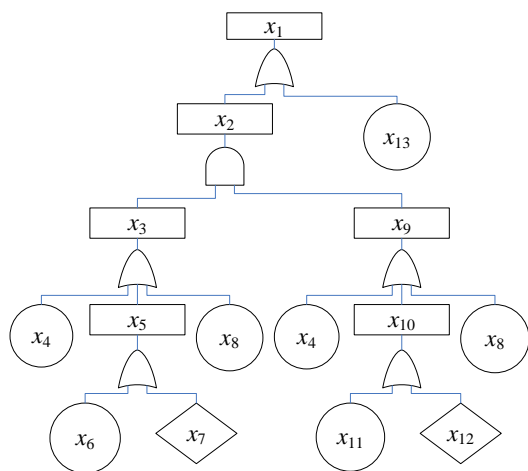
3.2. Ако је $l' \leq l$ развијати коњункцију,

3.3. Израчунати (7.4). Ако је $l' - \sum_{i \in U''} (\sum_{v \in V'} k_i^{v \max} - 1) \leq l$ развијати коњункцију, у супротном елиминисати је.

4. Извршити редукцију применом закона Булове алгебре.

Пример

У наставку ће бити илустрована модификована директна метода на примеру СН са слике 2.4, у коме је, због једноставије примене, промењено нумерисање догађаја (Слика 7.1).



Слика 7.1. СН са слике 2.4. са модификованим нумерисањем

Кораци модификоване директне методе за параметар одсецања $l = 1$ су:

$$1. x_4: k_4^{4 \max} = 1, k_4^{8 \max} = 0, k_4^{5 \max} = 0, k_4^{10 \max} = 0,$$

$$k_4^{3 \max} = \max\{k_4^{4 \max}, k_4^{5 \max}, k_4^{8 \max}\} = \max\{1, 0, 0\} = 1,$$

$$k_4^{9 \max} = \max\{k_4^{4 \max}, k_4^{10 \max}, k_4^{8 \max}\} = \max\{1, 0, 0\} = 1,$$

$$k_4^{2 \max} = k_4^{3 \max} + k_4^{9 \max} = 1 + 1 = 2, k_4^{1 \max} = \max\{k_4^{2 \max}\} = 2,$$

$$x_8: k_8^{4\max} = 0, k_8^{8\max} = 1, k_8^{5\max} = 0, k_8^{10\max} = 0, k_8^{3\max} = \max\{0, 0, 1\} = 1,$$

$$k_8^{9\max} = \max\{0, 0, 1\} = 1, k_8^{2\max} = 1 + 1 = 2, k_8^{1\max} = 2.$$

$$2. x_1 = x_2 + x_{13}, x_2 = x_3 \cdot x_9, x_3 = x_4 + x_5 + x_8, x_9 = x_4 + x_{10} + x_8, x_5 = x_6 + x_7, x_{10} = x_{11} + x_{12}$$

$$3. x_1 = x_2 + x_{13}$$

$$3.1. x_2: l' = 1, x_{13}: l' = 1,$$

$$3.2. x_2: l' \leq l \Rightarrow \text{развијати даље}, x_{13}: l' \leq l \Rightarrow \text{развијати даље},$$

$$x_1 = x_3 \cdot x_9 + x_{13}$$

$$3.1. x_3 \cdot x_9: l' = 2,$$

$$3.3. \sum_{i \in U^n} \left(\sum_{v \in V', i: x_i \in P[G_v]} k_i^{v\max} - 1 \right) = [k_4^{3\max} + k_4^{9\max} - 1] + [k_8^{3\max} + k_8^{9\max} - 1] \\ = [1 + 1 - 1] + [1 + 1 - 1] = 2$$

$$l' - \sum_{i \in U^n} \left(\sum_{v \in V', i: x_i \in P[G_v]} k_i^{v\max} - 1 \right) = 2 - 2 = 0 \leq l \Rightarrow \text{развијати даље},$$

$$x_1 = (x_4 + x_5 + x_8) \cdot (x_4 + x_{10} + x_8) + x_{13}$$

$$= x_4 x_4 + x_5 x_4 + x_8 x_4 + x_4 x_{10} + x_5 x_{10} + x_8 x_{10} + x_4 x_8 + x_{10} x_8 + x_8 x_8 + x_{13}$$

$$3.1. x_4 x_4, x_5 x_4, x_8 x_4, x_4 x_{10}, x_5 x_{10}, x_8 x_{10}, x_4 x_8, x_{10} x_8, x_8 x_8, : l' = 2,$$

$$3.3. x_4 x_4, x_8 x_8: l' - \sum_{i \in U^n} \left(\sum_{v \in V', i: x_i \in P[G_v]} k_i^{v\max} - 1 \right) = 2 - 1 = 1 \leq l \Rightarrow \text{развијати}$$

даље,

$$x_5 x_4, x_8 x_4, x_4 x_{10}, x_5 x_{10}, x_8 x_{10}, x_4 x_8, x_{10} x_8 :$$

$$l' - \sum_{i \in U^n} \left(\sum_{v \in V', i: x_i \in P[G_v]} k_i^{v\max} - 1 \right) = 2 - 0 = 2 > l \Rightarrow \text{одсећи},$$

$$x_1 = x_4 x_4 + x_8 x_8 + x_{13}$$

$$4. x_1 = x_4 + x_8 + x_{13}$$

Резултат су сви минипресеци реда 1: {P1}, {P2} и {P3}.

С обзиром да је у питању горња граница одсецања, може се десити да су међу добијеним скуповима пресека и они који су, након идемпотенције дужи од l . Такви скупови пресека се одбацују. Ова грешка издвајања скупова пресека који не задовољавају задати услов је прихватљива, за разлику од грешке неодређивања неког скупа пресека реда l .

У наставку ће бити приказано како се тврђење 7.1 може применити у одређивању минипресека задате дужине (реда) помоћу графа достижности ИПМ датог СН.

7.2.2. Одређивање параметра одсецања у ИПМ

Одређивање минипресека задате дужине, односно реда l , у ИПМ се своди на испитивање да ли грану испод посматраног чвора стабла достижности треба одсећи или даље развијати. Сваки чвор стабла достижности, односно одговарајуће маркирање M , може се посматрати као корен једног његовог подстабла. С обзиром да се у скуповима пресека, који се добијају на основу листова стабла достижности, може применити закон идемпотенције, потребно је одредити или проценити за колико ће се скратити скупови пресека у листовима подстабла са кореном M .

У Кораку 2 алгоритма ИПМ_1, ако активном чвору одговара логичко коло типа И, формира се $R_{\wedge} = \{(i, q^*, j)\}$ - скуп свих излазних места активног чвора. У складу са дискусијом из претходног поглавља, може се рећи да скуп R_{\wedge} чине места која моделирају посредне и/или примарне догађаје, за које активни чвор представља минимално логичко коло.

Посматрајмо само места која моделирају посредне догађаје. Нека су местима p_i и p_j ИПМ моделирани посредни догађаји G_v и $G_w, v, w \in V'$ датог СН. Уколико за p_i и p_j важи да су на истом путу (поглавље 6.4), односно $p_i \in^{\circ} \dots^{\circ} ({}^{\circ} p_j)$, онда је $G_w \in P[G_v]$. Како су сва места R_{\wedge} излазна места истог прелаза, она не могу бити на истом путу, односно за свака два места из R_{\wedge} важи: $p_i \notin^{\circ} \dots^{\circ} ({}^{\circ} p_j)$ и $p_j \notin^{\circ} \dots^{\circ} ({}^{\circ} p_i)$. Одатле следи да за свака два посредна догађаја $G_v, G_w, v, w \in V'$, која су моделирана местима из R_{\wedge} , важи $G_v \notin P[G_w]$ и $G_w \notin P[G_v]$.

Након паљења активног прелаза, из активног места се троши жетон, а сва места из R_{\wedge} добијају жетон, односно $M(p_i) = 0 \wedge (\forall (i, q, j) \in R_{\wedge}) M(p_j) = 1$. На тај начин се добија ново маркирање M' . У M' може постојати још места $p_j, j = 1, \dots, m$ за које

је, у некој од претходних итерација, $M(p_j)$ добило вредност 1. Сва места за које је $M'(p_j)=1$, моделирају одигравање одговарајућих догађаја. Како се у свакој итерацији, у Кораку 2 алгоритма, за $p_i = {}^{\circ}p_j$ врши промена маркирања: $M(p_i)=0$ и $M(p_j)=1$, за сваки пар места p_r и p_s , за које је истовремено $M'(p_r)=1$ и $M'(p_s)=1$, важи $p_r \notin \dots({}^{\circ}p_s)$ и $p_s \notin \dots({}^{\circ}p_r)$. Одатле следи да за догађаје G_v и G_w , који су моделирани местима p_r и p_s , важи $G_v \notin P[G_w]$ и $G_w \notin P[G_v]$.

С друге стране, како важи $p_r \notin \dots({}^{\circ}p_s)$ и $p_s \notin \dots({}^{\circ}p_r)$, следи да p_r и p_s припадају двома различитим гранама ИПМ. Те две гране имају заједничко место p' , које уједно представља и корен минималног подстабла ИПМ које садржи места p_r и p_s (p може бити и корен ИПМ који моделира вршни догађај СН). Како постоји маркирање M' у коме је $M'(p_r)=1$ и $M'(p_s)=1$, следи да је p' улазно место једног прелаза, односно p' моделира излаз логичког кола типа И датог СН. (Подсетимо да, пошто ИПМ има обрнуто оријентисане гране, излазном догађају датог логичког кола СН одговара улазно место прелаза ИПМ којим је моделирано посматрано логичко коло.)

Претпоставимо супротно. Нека p' моделира излаз логичког кола типа ИЛИ датог СН. У том случају p' је улазно место онолико прелаза, колико посматрано логичко коло има улазних догађаја. Постојање жетона у месту p' истовремено омогућава само један од прелаза $t \in p'^0$, што значи да се жетон из p' не може истовремено проследити у обе гране којима припадају места p_r и p_s . То даље значи да не може да се достигне маркирање M' у коме је $M'(p_r)=1$ и $M'(p_s)=1$. Дакле, минималном корену подстабла ИПМ које садржи места p_r и p_s одговара логичко коло типа И датог СН. Одатле следи да је минимално логичко коло догађаја G_v и G_w , који су моделирани местима p_r и p_s , типа И.

На основу претходне дискусије доказано је следеће тврђење:

Тврђење 7.2: У чвору M' стабла достижности ИПМ датог СН, местима $p_j, j=1, \dots, m$ за која је $M'(p_j)=1$ моделирани су догађаји $G_v, v \in V'$, чије је минимално логичко коло типа И и за које важи да је за свака два посредна догађаја $G_v, G_w, v, w \in V', G_v \notin P[G_w]$ и $G_w \notin P[G_v]$.

Нека је, како је већ означено у поглављу 7.1, S скуп свих $(k+1)$ -торки $(k, s_1, s_2, \dots, s_k)$, чији први елемент представља вишеструкост примарног догађаја x_i датог СН, а осталих k елемената индексе места $p_{s_1}, p_{s_2}, \dots, p_{s_k}$, којима је моделирано његових k_i понављања. Нека је p_j место које моделира посредни догађај G_v такво да је $M'(p_j)=1$. Тада је $k_s^{j\max}$ у ИПМ једнако $k_i^{v\max}$ у датом СН. Сада је могуће, аналогно Тврђењу 1, дефинисати горњу границу максималног скраћења скупова пресека добијених у листовима подстабла стабла достижности са кореном M' .

Напомена 7.2: У (7.4), максимално скраћење се рачуна само за $x_i, i \in U''$, односно за примарне догађаје вишеструкости веће од 1, који припадају домену бар једног од посматраних посредних догађаја. У ИПМ се овај услов своди на то да се очекивано скраћење израчунава само на основу $(k, s_1, s_2, \dots, s_k)$, таквих да постоји бар једно $M'(p_j)=1$ чији је следбеник неко од места $p_{s_1}, p_{s_2}, \dots, p_{s_k}$. Овај услов се директно може изразити на следећи начин:

$(\exists i \in \{s_1, s_2, \dots, s_k\})(\exists j \in \{1, \dots, m\})(M'(p_j)=1 \wedge p_j = {}^{\circ} \dots {}^{\circ} ({}^{\circ} p_i))$. Други, посредни начин је много једноставнији: $\sum_{j: M'(p_j)=1} k_s^{j\max} \geq 1$ и он ће бити коришћен у

формулацији Тврђења 7.3.

Сада, коришћењем дате нотације, на основу тврђења 7.1 и 7.2 и напомене 7.2, следи:

Тврђење 7.3: Нека је дато стабло достижности ИПМ датог СН и његово подстабло са кореном M' . Скупови пресека добијени у листовима подстабла са кореном M' се, применом закона идемпотенције, могу скратити највише за:

$$\sum_{s \in S: \sum_{j: M'(p_j)=1} k_s^{j \max} \geq 1} (\sum_{j: M'(p_j)=1} k_s^{j \max} - 1) \quad (7.5)$$

где је s $(k+1)$ -торка $s = (k, s_1, s_2, \dots, s_k)$ таква да постоји бар једно $M'(p_j) = 1$, чији је следбеник неко од места $p_{s_1}, p_{s_2}, \dots, p_{s_k}$, а $k_s^{j \max}$ максималан број места $p_{s_1}, p_{s_2}, \dots, p_{s_k}$ у подстаблу ИПМ са кореном p_j , $M'(p_j) = 1$, таквих да, за сваки пар места $p_{s_1}, p_{s_2}, \dots, p_{s_k}$, корен њиховог минималног подстабла у ИПМ моделира излаз из логичког кола типа И.

Напомена 7.3: Аналогно напомени 7.1, за места $p_{s_1}, p_{s_2}, \dots, p_{s_k}$ која су листови ИПМ, је $k_s^{s_r \max} = 1$, $r = 1, \dots, k$. На тај начин ће се добити тачан број понављања посматраног догађаја, ако је за неко од $p_{s_1}, p_{s_2}, \dots, p_{s_k}$ у тренутном чвору стабла достижности $M'(p_{s_r}) = 1$.

Нека је задат параметар одсецања l . Посматрајмо сада чвор M' стабла достижности ИПМ датог СН, у коме је $\sum_{j=1}^n M'(p_j) = l'$. Скупови пресека у листовима подстабла са кореном M' датог стабла достижности ће имати l' примарних догађаја или више, а применом закона идемпотенције се могу скратити највише за (7.5), односно њихова дужина може бити најмање:

$$l' - \sum_{s \in S: \sum_{j: M'(p_j)=1} k_s^{j \max} \geq 1} (\sum_{j: M'(p_j)=1} k_s^{j \max} - 1). \quad (7.6)$$

Тада су могућа три случаја:

1. $l' < l$ - чвор M' стабла достижности треба даље развијати.
2. $l' > l$ и $l' - \sum_{s \in S: \sum_{j: M'(p_j)=1} k_s^{j \max} \geq 1} (\sum_{j: M'(p_j)=1} k_s^{j \max} - 1) \leq l$ - чвор M' треба даље развијати јер

се применом закона идемпотенције могу добити скупови пресека реда l .

3. $l' - \sum_{s \in S: \sum_{j: M'(p_j)=1} k_s^{j \max} \geq 1} (\sum_{j: M'(p_j)=1} k_s^{j \max} - 1) > l$ - чвор M' треба одсећи јер ће, и након

идемпотенције, добијени скупови пресека бити дужи од l .

Поступак одсецања стабла достижности сада се састоји из следећих корака:

- Из тренутног маркирања M' извојити места p_j за које је $M'(p_j)=1$;
- За свако s одредити $\sum_{j:M'(p_j)=1} k_s^{j\max}$;
- Израчунати (7.5);
- Израчунати (7.6),
- Проверити који је од случајева 1-3) задовољен и поступити по њему.

7.2.3. Алгоритам за одређивање минипресека задате дужине

Сада је могуће формулисати алгоритам за налажење минипресека задате дужине за СН са вишеструким догађајима. Алгоритмом је обухваћен и корак 3' из поглавља 7.1. Овај алгоритам се може користити и за одређивање свих минипресека СН са вишеструким догађајем. Потребно је само задати довољно велики параметар одсецања l .

У наставку ће прво бити наведени улазни подаци и скупови који се користе у алгоритму. Они који су већ коришћени у алгоритму ИМП1 и модификацијама Корака 2 и 3 ће бити само поновљени, а нови ће бити детаљније објашњени.

Улазни подаци који се користе у алгоритму су:

- (i, q, j) , $j=1, \dots, m$ - место p_j ИПМ датог СН;
- $M_0 = [1, 0, \dots, 0]$ - почетно маркирање;
- S - скуп свих $k+1$ -торки $(k, s_1, s_2, \dots, s_k)$ вишеструкости;
- $k_s^{j\max}$ - максималан број места $p_{s_1}, p_{s_2}, \dots, p_{s_k}$ која престављају догађаје у подстаблу ИПМ са кореном p_j таквих да, за сваки пар места $p_{s_1}, p_{s_2}, \dots, p_{s_k}$, корен њиховог минималног подстабла у ИПМ моделира излаз из логичког кола типа И. За места $p_{s_1}, p_{s_2}, \dots, p_{s_k}$ која представљају примарне догађаје је $k_s^{s_r\max} = 1$;
- l - параметар одсецања.

Параметри и скупови који се користе у алгоритму су:

- k, r - индекс активног места и активног прелаза;
- R - скуп места за која важи: $p_j \in (p_k)^\circ$;
- L - скуп индекса места који су листови ИПМ у тренутној итерацији;
- R_\wedge - скуп места из R која су у И вези;
- $R \setminus R_\wedge$ - скуп места из R која су у ИЛИ вези;
- Q_\wedge - скуп места која у тренутној итерацији садрже жетон и која нису у L ;
- Q_\vee - скуп места која су у ИЛИ вези са активним местом;
- $MSP0, MSP$ и SPR - скуп скупова пресека који имају својство 7.1, 7.2 и 7.3, респективно;
- \overline{SP} - скуп одсечених чворова.

Алгоритам за одређивање минипресека кохерентног СН са вишеструким догађајима се може сажети у следеће три фазе:

- Резултат *else* гране корака 1 је један лист ИПМ или одсечени чвор стабла достижности;
- Резултат последње *else* гране корака 2 је један одсечени чвор стабла достижности;
- Резултат *else* гране корака 3 је један лист стабла достижности (један скуп пресека) са додељеним својствима 7.1-7.3;
- Резултат *else* гране корака 4 су скупови $MSP0, MSP$ и SPR , који су улаз у алгоритам редукције СНЗ, и \overline{SP} - скуп одсечених чворова.

Алгоритам ИПМ_2

Почетак:

$$(0,0,1), k := 1, r := 0, M := (1,0,\dots,0), R := \emptyset, R_\wedge := \emptyset, Q_\wedge := \emptyset, Q_\vee := \emptyset, L := \emptyset, \\ MSP0 := \emptyset, MSP := \emptyset, SPR := \emptyset, \overline{SP} := \emptyset$$

Корак 1:

$$R := \{(k, q, j) \mid q \geq r\};$$

If $R \neq \emptyset$ then Go to *Корак 2* else $L := L \cup \{j\}$; Go to *Корак 3*.

Корак 2:

$$q^* := \min_{(i, q, j) \in R} q;$$

$$R_\wedge := \{(i, q^*, j)\};$$

$$R := R - R_\wedge;$$

$$M(p_k) := 0;$$

$$(\forall (i, q, j) \in R) (M(p_j) := 1; M_j := M; M(p_j) := 0)$$

$$Q_\vee := Q_\vee \cup \{(i, q, j), M_j \mid (i, q, j) \in R\};$$

$$(\forall (i, q, j) \in R_\wedge) M(p_j) := 1;$$

if $\sum_{j=1}^m M(p_j) \leq l$ then $j^* = \min_{(i, q, j) \in P_\wedge} j$; $k = j^*$; $r = q$; $R_\wedge := \emptyset$; Go to *Step 1*.

$$\text{else if } \sum_{j=1}^m M(p_j) - \sum_{\substack{s \in S: \\ \sum_{j: M^s(p_j)=1} k_s^{j \max} \geq 1}} (\sum_{j: M^s(p_j)=1} k_s^{j \max} - 1) \leq l$$

$$\text{then } j^* = \min_{(i, q, j) \in P_\wedge} j$$
; $k = j^*$; $r = q$; $R_\wedge = \emptyset$; Go to *Step 1*.

$$\text{else } (\forall (i, q, j) \in R_\wedge) M(p_j) = 0; M(p_k) = 1;$$

$$L = L + \{j\}; \overline{SP} = \overline{SP} + \{L\}; Q_\wedge = \emptyset; \text{Go to Step 4.}$$

Корак 3:

$$Q_\wedge := \{(i, q, j) \mid j = 1, j \notin L\};$$

If $Q_\wedge \neq \emptyset$ then $j^* = \min_{(i, q, j) \in Q_\wedge} j$, $k = j^*$, $r = q$, $Q_\wedge = \emptyset$, $R = \{(i, q, j^*)\}$, Go to *Корак 2*;

else if $(\forall s \in S) \sum_{i=1}^k M(p_{s_i}) = 0$ then $MSP0 = MSP0 \cup \{L\}$;

else if $(\forall s \in S) (\sum_{i=1}^k M(p_{s_i}) = k \vee \sum_{i=1}^k M(p_{s_i}) = 0)$ then $MSP = MSP \cup \{L\}$;

else $SPR = SPR \cup \{L\}$;

Корак 4:

If $Q_v \neq \emptyset$ then $j^* = \min_{\text{lex } M_j} \min_{(i,q,j) \in Q_v} j$;

$k := i; r := q; M := M_j$;

$Q_v := Q_v \setminus \{(i,q,j), M_j\}, L = \emptyset$; Go to *Корак 3*;

else Крај.

Алгоритам ИПМ_2 се разликује од алгоритма ИПМ_1 у Корацима 2 и 3. Поред тога, разлика је и у томе што се, након *then* гране корака 4 враћа на корак 3, а након тога на корак 2. Разлог за то је што, када се из Q_v изабере маркирање које ће се даље развијати, мора прво да се провери да ли ће скупови пресека у подстаблу ИПМ са кореном у новом активном чвору, након примене закона идемпотенције, задовољавати услов задатог параметра l .

Пример

У наставку ће бити илустровано како се алгоритмом ИПМ_2 добијају скупови пресека СН са слике 2.4, на основу стабла достижности ИПМ са слике 6.7.

Улазни подаци који се користе у алгоритму су:

- Тројке: (0,0,1), (1,1,2), (1,2,3), (3,3,4), (4,4,5), (4,5,6), (6,6,7), (6,7,8), (4,8,9), (3,3,10), (10,9,11), (10,10,12), (12,11,13), (12,12,14) и (10,13,15);
- Почетно маркирање: $M_0 = [1, 0, \dots, 0]^T$;
- Елементи скупа $S = \{(2,5,11), (2,9,15)\}$;

– $k_s^{j\max}$: за $s=1$: $k_1^{5\max}=1$, $k_1^{11\max}=1$, $k_1^{4\max}=1$, $k_1^{10\max}=1$, $k_1^{3\max}=2$, $k_1^{1\max}=2$ и $k_1^{j\max}=0$, $j \in \{2,6,7,8,9,12,13,14,15\}$, за $s=2$: $k_2^{9\max}=1$, $k_2^{15\max}=1$, $k_2^{4\max}=1$, $k_2^{10\max}=1$, $k_2^{3\max}=2$, $k_2^{1\max}=2$ и $k_2^{j\max}=0$, $j \in \{2,5,6,7,8,11,12,13,14\}$.

– Параметар одсецања $l=1$

Почетак: $(0,0,1)$, $k=1$, $r=0$, $M=(1000000000000000)$, $R=\emptyset$, $R_\wedge=\emptyset$, $Q_\wedge=\emptyset$,

$$Q_\vee=\emptyset, L=\emptyset, MSPO=\emptyset, MSP=\emptyset, SPR=\emptyset, \overline{SP}=\emptyset.$$

Корак 1: $R=\{(1,1,2), (1,2,3)\}$;

Корак 2: $q^* = \min_{(i,q,j) \in R} \{1,2\} = 1$; $R_\wedge=\{(1,1,2)\}$; $M(p_1)=0$; $R=\{(1,2,3)\}$;

$$M(p_3)=1, M_3=(0010000000000000), M(p_3)=0;$$

$$Q_\vee = \{[(1,2,3), M_3]\}; R = \emptyset; M(p_2) = 1,$$

$$\sum_{j=1}^{15} M(p_j) = 1 \leq l, M=(0100000000000000); j^* = 2; k = 2; r = 1, R_\wedge = \emptyset,$$

Корак 1: $R = \emptyset$, $L = \{2\}$;

пронађен је лист ИПМ

Корак 3: $Q_\wedge = \emptyset$, $M(p_5)+M(p_{11})=0$ и $M(p_9)+M(p_{15})=0$,

$$MSPO = \{\{2\}\}; \quad \# \text{ пронађен је скуп пресека са својством 1.}$$

Корак 4: $j^* = 3$; $k=1$; $r=2$; $M=(0010000000000000)$; $Q_\vee=\emptyset$, $L=\emptyset$;

Корак 3: $Q_\wedge = \{(1,2,3)\}$, $j^* = 3$; $R=\{(1,2,3)\}$; $Q_\wedge = \emptyset$;

Корак 2: $q^* = 2$; $R_\wedge=\{(1,2,3)\}$; $R = \emptyset$; $M(p_1)=0$; $Q_\vee = \emptyset$; $M(p_3)=1$;

$$M(p_3)=1, \sum_{j=1}^{15} M(p_j) = 1 \leq l, M=(0010000000000000), j^* = k = 3$$

$$; r=2; R_\wedge=\emptyset;$$

Корак 1: $R=\{(3,3,4), (3,3,10)\}$;

Корак 2: $q^* = 3$; $R_\wedge=\{(3,3,4), (3,3,10)\}$; $R = \emptyset$; $M(p_3)=0$; $Q_\vee = \emptyset$; $M(p_4)=1$; $M(p_{10})=1$,

$$(M(p_4)=1, M(p_{10})=1) \Rightarrow \sum_{j=1}^{15} M(p_j) = 2 > l,$$

$$\sum_{j=1}^{15} M(p_j) - \sum_{s \in S} \left(\sum_{j: M^s(p_j)=1} k_s^{j\max} - 1 \right)$$

$$= 2 - [(k_1^{4\max} + k_1^{10\max} - 1) + (k_2^{4\max} + k_2^{10\max} - 1)] = 2 - (1+1) = 0 \leq l$$

$$M=(000100000100000); j^* = k = 4; r=3; R_{\wedge} = \emptyset;$$

Корак 1: $R=\{(4,4,5), (4,5,6), (4,8,9)\};$

Корак 2: $q^* = 4; R_{\wedge}=\{(4,4,5)\}; R = \{(4,5,6), (4,8,9)\}; M(p_4) = 0;$

$$M_6=(000001000100000), M_9=(000000001100000);$$

$$Q_{\vee}=\{[(4,5,6),M_6],[4,8,9), M_9]\}; R = \emptyset; M(p_5) = 1;$$

$$2 - [(k_1^{5\max} + k_1^{10\max} - 1) + (k_2^{5\max} + k_2^{10\max} - 1)] = 2 - (1 + 0) = 1 \leq l$$

$$M = (000010000100000); j^* = k = 5; r=4; R_{\wedge} = \emptyset;$$

Корак 1: $R = \emptyset L = \{5\};$

пронађен је лист ИПМ

Корак 3: $Q_{\wedge}=\{(3,3,10)\}; j^* = k = 10; r = 3; R = \{(3,3,10)\}; Q_{\wedge} = \emptyset;$

Корак 2: $q^* = 3; R_{\wedge}=\{(3,3,10)\}; R = \emptyset$

$$2 - [(k_1^{5\max} + k_1^{10\max} - 1) + (k_2^{5\max} + k_2^{10\max} - 1)] = 2 - (1 + 0) = 1 \leq l$$

$$M = (000010000100000); j^* = k = 10; r = 3; R_{\wedge} = \emptyset;$$

Корак 1: $R=\{(10,9,11), (10,10,12), (10,13,15)\};$

Корак 2: $q^* = 9; R_{\wedge}=\{(10,9,11)\}; R = \{(10,10,12), (10,13,15)\}; M(p_{10}) = 0;$

$$M_{12}=(000010000001000), M_{15}=(000010000000001);$$

$$Q_{\vee}=\{[(4,5,6),M_5], [(4,8,9), M_9], [(10,10,12), M_{12}], [(10,13,15), M_{15}]\}; R = \emptyset;$$

$$M(p_{11}) = 1;$$

$$2 - [(k_1^{5\max} + k_1^{11\max} - 1) + (k_2^{5\max} + k_2^{11\max} - 1)] = 2 - (1 + 0) = 1 \leq l$$

$$M = (000010000010000); j^* = k = 11; r = 9; R_{\wedge} = \emptyset;$$

Корак 1: $R = \emptyset L = \{5, 11\};$

пронађен је лист ИПМ

Корак 3: $Q_{\wedge} = \emptyset, M(p_5) + M(p_{11}) = 2$ и $M(p_9) + M(p_{15}) = 0,$

$$MSP = \{\{5, 11\}\}; \quad \# \text{ пронађен је скуп пресека са својством 2, тј. } C^*.$$

Корак 4: $j^* = 6; k = 4; r = 5; M = M_6 = (000001000100000);$

$$Q_{\vee}=\{[(4,8,9),M_9],[10,10,12),M_{12}], [(10,13,15), M_{15}]\}; L = \emptyset;$$

Корак 3: $Q_{\wedge}=\{(4,5,6),(3,3,10)\}; j^* = k = 6; r = 5; R = \{(4,5,6)\}; Q_{\wedge} = \emptyset;$

Корак 2: $q^* = 5; R_{\wedge}=\{(4,5,6)\}; R = \emptyset; M(p_4) = 0; M(p_6) = 1;$

$$2 - [(k_1^{6\max} + k_1^{10\max} - 1) + (k_2^{6\max} + k_2^{10\max} - 1)] = 2 - (0 + 0) = 2 > l$$

$$M(p_4) = 1; M(p_6) = 0, M = (000100000100000),$$

$L = \{4\}, \overline{SP} = \{\{4\}\};$ # одсечен је један чвор

Корак 4: $j^* = 9; k = 4; r = 8; M = M_9 = (000000001100000);$

$Q_{\vee} = \{[(10,10,12), M_{12}], [(10,13,15), M_{15}]\}; L = \emptyset;$

Корак 3: $Q_{\wedge} = \{(4,8,9), (3,3,10)\}; j^* = k = 9; r = 8; R = \{(4,8,9)\}; Q_{\wedge} = \emptyset;$

Корак 2: $q^* = 8; R_{\wedge} = \{(4,8,9)\}; R = \emptyset; M(p_4) = 0; M(p_9) = 1;$

$$2 - [(k_1^{9\max} + k_1^{10\max} - 1) + (k_2^{9\max} + k_2^{10\max} - 1)] = 2 - (0 + 1) = 1 \leq l$$

$M = (000000001100000); j^* = k = 9; r = 8; R_{\wedge} = \emptyset;$

Корак 1: $R = \emptyset L = \{9\};$ # пронађен је лист ИПМ

Корак 3: $Q_{\wedge} = \{(3,3,10)\}; j^* = k = 10; r = 3; Q_{\wedge} = \emptyset;$

Корак 2: $q^* = 9; R_{\wedge} = \{(10,9,11)\}; R = \{(10,10,12), (10,13,15)\}; M(p_{10}) = 0;$

$M_{12} = (000000001001000), M_{15} = (000000001000001);$

$Q_{\vee} = \{[(10,10,12), M_{12}], [(10,13,15), M_{15}], [(10,10,12), M_{12}], [(10,13,15), M_{15}]\};$

$R = \emptyset;$

$$2 - [(k_1^{9\max} + k_1^{11\max} - 1) + (k_2^{9\max} + k_2^{11\max} - 1)] = 2 - (0 + 0) = 2 > l$$

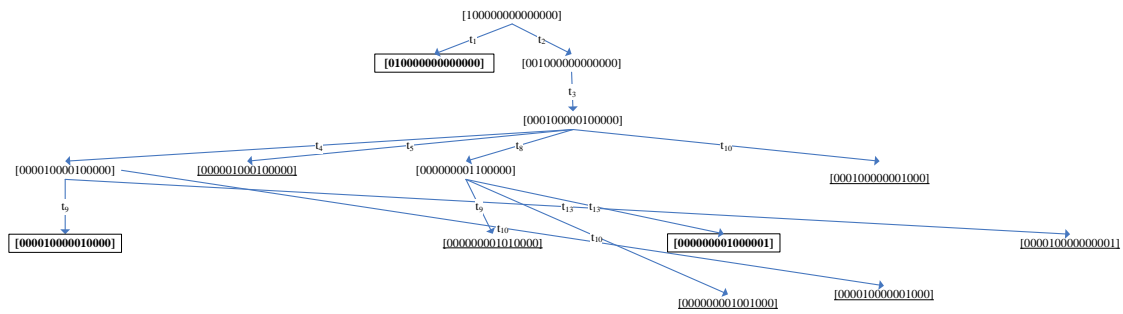
$M(p_{11}) = 0; M(p_{10}) = 1, M = (000000001100000),$

$L = \{9\}, \overline{SP} = \{\{6\}, \{9\}\};$ # одсечен је још један чвор

Даљом применом алгоритма, добија се сечење чвора стабла достижности у чворовима са маркирањима: $(000010000001000), (000000001001000), (000010000000001)$ и (000000001010000) , а у последњој итерацији алгоритма, добија се скуп пресека $\{9,15\}$ са својством 7.2, тј. скуп пресека облика C^* .

Излаз из алгоритма су скупови: $MSP_0 = \{\{2\}\}, MSP = \{\{5,11\}, \{9,15\}\}$ и $SPR = \emptyset.$

Пошто је $SPR = \emptyset$, сви добијени скупови пресека су минипресеци. Скупови пресека из MSP , након примене закона идемпотенције, постају реда 1. На тај начин су одређена сва три минипресека реда 1 СН са слике 2.4. Део стабла достизања са слике 6.12, који је тиме одређен, приказан је на слици 7.2. (одсечени чворови стабла достижности су подвучени).



Слика 7. 2. Скраћено стабло достижности ИПМ са слике 6.7.

Елемент скупа SPR , након примене закона идемпотенције, постаје реда l' и може имати једно од следећих својстава:

- $l' \leq l$ и надскуп је неког скупа пресека из MSP или SPR . Посматрани скуп пресека ће бити елиминисан применом алгорита редукције СНЗ;
- $l' \leq l$ и није надскуп ниједног скупа пресека из MSP и SPR . Да би неки од одсечених скупова пресека био подскуп посматраног скупа пресека, његов ред мора бити једнак или мање од l' . Како су сви одсечени скудови пресека реда већег од l , ниједан од њих не може бити подскуп посматраног скупа пресека. Дакле, посматрани скуп пресека је минипресек;
- $l' > l$. У том случају скуп пресека ће бити елиминисан. Овде није битно да ли је посматрани скуп пресека минипресек или не, јер је његов број примарних догађаја већи од задатог.

7.2.4. Алгоритам за одређивање $k_s^{j\max}$

Елементи $k_s^{j\max}$ се одређују на основу структуре ИПМ датог СН. У поглављу 6.5 је показано да је структура ИПМ садржана у улазним тројкама (i, q, j) , $j = 1, \dots, m$. Овде ће бити поновљене основне особине овако записане структуре ИПМ, које су потребне за даљи рад.

- Место p_j је представљено тројком (i, q, j) , где је: q индекс улазног прелаза места p_j , тј, $t_q = {}^{\circ}p_j$, i индекс улазног места прелаза t_q , тј, $p_i = {}^{\circ}t_q$, односно $p_i = {}^{\circ}({}^{\circ}p_j)$ и j индекс места p_j ;
- Корен ИПМ је $(0,0,1)$, а уколико је место p_j лист, неће постојати ниједна тројка у којој је први елемент једнак j ;
- Ако места p_j, p_k, \dots, p_u ИПМ моделирају догађаје СН који су у И вези, прва два елемента тројки које их представљају су иста, тј: $(i, q, j), (i, q, k), \dots, (i, q, u)$;
- Ако места p_j, p_k, \dots, p_u ИПМ моделирају догађаје СН који су у ИЛИ вези, само први елемент тројки које их представљају је исти, тј: $(i, q, j), (i, r, k), \dots, (i, v, u)$;

Међутим, улазним тројкама $(i, q, j), j=1, \dots, m$ су понављања истог примарног догађаја, чија је вишеструкост већа од 1, представљена различитим местима, а за одређивање $k_s^{j\max}$ је потребан податак и о вишеструкостима. Овај податак је садржан у $(k, s_1, s_2, \dots, s_k)$. Нека је P_p скуп места ИПМ, којима су моделирани примарни догађаји датог СН. Свако место из P_p може припадати само једној ($k+1$)-торки, тако да се за њих може одредити $k_s^{j\max}$ на следећи начин:

$$k_s^{j\max} = \begin{cases} 1 & \text{ако } j \in \{s_1, s_2, \dots, s_k\} \\ 0 & \text{у супротном} \end{cases}, \quad s \in S, p_j \in P_p. \quad (7.6)$$

Када је одређено $k_s^{j\max}$ за сва места из P_p , вредност овог параметра се може израчунати и за места која моделирају посредне догађаје. За сваки посредни догађај $p_g \in P \setminus P_p$ се на основу улазних тројки $(i, q, j), j=1, \dots, m$, може одредити скуп места следбеника $(p_g^{\circ})^{\circ}$:

$$(p_g^{\circ})^{\circ} = \{(g, q, j)\}. \quad (7.7)$$

Ако место $p_g \in P \setminus P_p$, моделира посредни догађај који је излаз из логичког кола типа И, онда за било који пар $(g, q', j'), (g, q'', j'')$ из $(p_g^{\circ})^{\circ}$, важи $q' = q''$. Тада је на основу (7.1) и дискусије из поглавља 7.2.2:

$$k_s^{g \max} = \sum_{j:(i,q,j) \in (p_g^o)^o} k_s^{j \max} - 1 \quad (7.8).$$

Ако место $p_g \in P \setminus P_p$, моделира посредни догађај који је излаз из логичког кола типа ИЛИ, онда за било који пар (g, q', j') , (g, q'', j'') из $(p_g^o)^o$, важи $q' \neq q''$. Тада је на основу (7.2) и дискусије из поглавља 7.2.2:

$$k_s^{g \max} = \max_{j:(i,q,j) \in (p_g^o)^o} \{k_s^{j \max}\} \quad (7.9)$$

Пошто је ИПМ нумерисана у дубину и с леве на десну страну, одређивање $k_s^{g \max}$ се врши од места са највећим индексом ка p_1 . На тај начин ће, када се рачунају $k_s^{g \max}$ за неко $p_g \in P \setminus P_p$, $k_s^{g \max}$ свих његових следбеника већ бити израчунати.

Скуп $(p_g^o)^o$ се одређује на исти начин као скуп $R = \{(k, q, j) \mid q \geq r\}$ у алгоритмима ИПМ_1 и ИПМ_2, само без додатног услова за индекс прелаза. Скупови из (7.8) и (7.9) се, такође, одређују аналогно одређивању скупова R_\wedge и $R \setminus R_\wedge$. Због тога ће се у алгоритму за одређивање $k_s^{j \max}$ за ове скупове користити исте ознаке као у ИПМ 1 и 2. Сада може да се формулише алгоритам за одређивање $k_s^{j \max}$.

Улазни подаци који се користе у алгоритму су:

- Тројке (i, q, j) , $j = 1, \dots, m$ и
- S' - скуп k -торки $s' = \{s_1, s_2, \dots, s_k\}$.

Параметри и скупови који се користе у алгоритму су:

- k, v - индекс активног места и места претходника;
- (v, r, k) - променљива типа (i, q, j) ;
- P_p - скуп листова ИПМ који моделирају примарне догађаје;
- P_g - скуп места ИПМ који моделирају посредне догађаје;
- $K = \{k(s, j), s \in S, j = 1, \dots, m\}$ - скуп параметара $k_s^{j \max}$.

Алгоритам **Kmax**

Почетак: $P_p := \emptyset$, $P_g := \emptyset$, $R := \emptyset$, $R_\wedge := \emptyset$, $K := \emptyset$

$$P_g := \{(i, q, j) \mid (\exists(j, r, k) \in P)\}$$

$$P_p := \{(i, q, j) \mid (\forall(v, r, k)) v \neq j\}$$

$$(\forall s' \in S') \wedge (\forall(i, q, j) \in P_p) \text{ if } j \in s' \text{ then } k(s', j) = 1$$

$$(\forall s' \in S') [(\forall(v, r, k) \in P_g) R = \{(i, q, j) \mid i = k\}]$$

$$\text{if } \sum_{(i, q, j) \in R} k_s^{j \max} \geq 1 \text{ then}$$

$$q^* = \min_{(i, q, j) \in R} q$$

$$R_\wedge = \{(i, q, j) \mid q = q^*\}, R = R \setminus R_\wedge$$

$$\text{if } R = \emptyset \text{ then } k_s^{k \max} = \sum_{j: (i, q, j) \in R_\wedge} k_s^{j \max} - 1$$

$$\text{else } k_s^{k \max} = \max_{j: (i, q, j) \in R_\wedge \cup R} \{k_s^{j \max}\}$$

$$\text{else } k_s^{k \max} = 0$$

$$K = K \cup \{k_s^{k \max}\}$$

Крај.

Пример

У наставку ће бити показано како се Алгоритмом **Kmax** добијају елементи скупа K за ИПМ са слике 6.7.

Улазни подаци који се користе у алгоритму су:

- Тројке: (0,0,1), (1,1,2), (1,2,3), (3,3,4), (4,4,5), (4,5,6), (6,6,7), (6,7,8), (4,8,9), (3,3,10), (10,9,11), (10,10,12), (12,11,13), (12,12,14) и (10,13,15);
- Елементи скупа $S = \{(5,11), (9,15)\}$;

Почетак: $P_p = \emptyset$, $P_g = \emptyset$, $R = \emptyset$, $R_\wedge = \emptyset$, $K = \emptyset$

$$P_g = \{(0,0,1), (1,2,3), (3,3,4), (4,5,6), (3,3,10), (10,10,12)\}$$

$$P_p = \{(1,1,2), (4,4,5), (6,6,7), (6,7,8), (4,8,9), \\ (10,9,11), (12,11,13), (12,12,14), (10,13,15)\}$$

Примарни догађаји:

$$s1 = (5,11) : k(1,2) = k(1,7) = k(1,8) = k(1,9) = k(1,13) = k(1,14) = k(1,15) = 0 \\ k(1,5) = k(1,11) = 1$$

$$s2 = (9,15) : k(1,2) = k(1,5) = k(1,7) = k(1,8) = k(1,11) = k(1,13) = k(1,14) = 0 \\ k(1,9) = k(1,15) = 1$$

Посредни догађаји:

$$s1 = (5,11)$$

$$(10,10,12) : R = \{(12,11,13), (12,12,14)\}, k(1,13) + k(1,14) = 0 \Rightarrow k(1,12) = 0$$

$$(3,3,10) : R = \{(10,9,11), (10,10,12), (10,13,15)\}, \\ k(1,11) + k(1,12) + k(1,15) = 1, q^* = 9, \\ R_\lambda = \{(10,9,11)\}, R = \{(10,10,12), (10,13,15)\} \Rightarrow k(1,10) = \max\{1, 0, 0\} = 1$$

$$(4,5,6) : R = \{(6,6,7), (6,7,8)\}, k(1,7) + k(1,8) = 0 \Rightarrow k(1,6) = 0$$

$$(3,3,4) : R = \{(4,4,5), (4,5,6), (4,8,9)\}, \\ k(1,5) + k(1,6) + k(1,9) = 1, q^* = 4, \\ R_\lambda = \{(4,4,5)\}, R = \{(4,5,6), (4,8,9)\} \Rightarrow k(1,4) = \max\{1, 0, 0\} = 1$$

$$(1,2,3) : R = \{(3,3,4), (3,3,10)\}, \\ k(1,4) + k(1,10) = 2, q^* = 3, \\ R_\lambda = \{(3,3,4), (3,3,10)\}, R = \emptyset \Rightarrow k(1,3) = 1 + 1 = 2$$

$$(0,0,1) : R = \{(1,1,2), (1,2,3)\}, \\ k(1,2) + k(1,3) = 2, q^* = 1, \\ R_\lambda = \{(1,1,2)\}, R = \{(1,2,3)\} \Rightarrow k(1,1) = \max\{0, 2\} = 2$$

$$s_2 = (9,15)$$

$$(10,10,12): R = \{(12,11,13), (12,12,14)\}, k(2,13) + k(2,14) = 0 \Rightarrow k(2,12) = 0$$

$$(3,3,10): R = \{(10,9,11), (10,10,12), (10,13,15)\},$$

$$k(2,11) + k(2,12) + k(2,15) = 1, q^* = 9,$$

$$R_{\wedge} = \{(10,9,11)\}, R = \{(10,10,12), (10,13,15)\} \Rightarrow k(2,10) = \max\{0, 0, 1\} = 1$$

$$(4,5,6): R = \{(6,6,7), (6,7,8)\}, k(2,7) + k(2,8) = 0 \Rightarrow k(2,6) = 0$$

$$(3,3,4): R = \{(4,4,5), (4,5,6), (4,8,9)\},$$

$$k(2,5) + k(2,6) + k(2,9) = 1, q^* = 4,$$

$$R_{\wedge} = \{(4,4,5)\}, R = \{(4,5,6), (4,8,9)\} \Rightarrow k(2,4) = \max\{0, 0, 1\} = 1$$

$$(1,2,3): R = \{(3,3,4), (3,3,10)\},$$

$$k(2,4) + k(2,10) = 2, q^* = 3,$$

$$R_{\wedge} = \{(3,3,4), (3,3,10)\}, R = \emptyset \Rightarrow k(2,3) = 1 + 1 = 2$$

$$(0,0,1): R = \{(1,1,2), (1,2,3)\},$$

$$k(2,2) + k(2,3) = 2, q^* = 1,$$

$$R_{\wedge} = \{(1,1,2)\}, R = \{(1,2,3)\} \Rightarrow k(2,1) = \max\{0, 2\} = 2$$

Тиме су одређени сви елементи скупа K који су коришћени у примеру из Поглавља 7.2.3.

7.3. Резултати експеримената

Проширење алгоритма ИПМ_1 кораком 3', којим се скупови пресека разврставају у скупове K_1 , K_2 и K_3 , имплементирано је у програмском језику VB (*Visual Basic*). Модификовани алгоритам је тестиран на групи СН (*Benchmark Fault Tree*) из [87] на рачунару *Intel Core 2 Duo CPU E8400* (3 GHz), *RAM 3,25 GB*.

Део добијених резултата је већ приказан у табели 3.2. у 3.3.6. Због тога ће овде, у табели 7.1, бити приказане само карактеристике одабраних СН: број логичких кола, број примарних догађаја, број места и прелаза одговарајуће ИПМ, као и број добијених скупова пресека и минипресека.

Табела 7.1. Карактеристике СН, ИПМ и резултат модификованог ИПМ_1

рб	СН	СН		ИПМ		Бр. скупова пресека	Бр. минипресека
		Бр. логичких кола	Бр. примарних догађаја	Бр. места	Бр. прелаза		
1	das9205	51	19	77	71	210160	17280
2	ftt10	175	93	394	364	690	305
3	isp9606	89	40	141	121	2292	1776
4	chinese*	60	35	95	81	$\cong 10^{10}$	392
5	chinese(ф)	25	23	48	37	272	11
6	das9202(фм)	37	31	92	90	18750	22
7	das9203(ф)	51	20	40	35	7000	1150

За СН под редним бројем 4, није било могуће добити све скупове пресека за прихватљиво време, па је због тога одређено само првих 40000 скупова пресека.

Затим је, на СН под редним бројевима 2, 3, 5 и 7, примењен алгоритам ИПМ_2, који је такође имплементиран у програмском језику VB. У нову имплементацију је укључен и алгоритам за одређивање елемената $k_s^{j\max}$, тако да су улазни подаци за ИПМ_2 исти као за модификовани ИПМ_1, а све додатне вредности се израчунавају у самом програму.

За сва четири СН одређени су минипресеци реда: 1, 2, 3, 4 и 5. Резултати одсецања стабла достижности су приказани у табели 7.2. За свако СН је дат распон вишеструкости његових примарних догађаја (l_{\min} и l_{\max}). Од резултата је приказан број минипресека са својством 7.1, 7.2 и 7.3, односно број чланова скупова K1, K2 и K3 алгоритма редукције СНЗ, који се добија за сваку вредност параметра одсецања. Поред тога, ради поређења добијених резултата, за свако СН је, на основу минипресека добијених алгоритмом ИПМ1, одређен укупан број минипресека датог реда. Ова вредност (БМ) је такође приказана у табели 7.2.

Табела 7.2. Минипресеци датог реда

СН	l_{\min}	l_{\max}		одсецање	одсецање	одсецање	одсецање	одсецање
				на 1	на 2	на 3	на 4	на 5
ftr10	1	3	БМ	57	300	305		
			К1	47	67	72		
			К2	4	4	4		
			К3	6	229	229		
isp9606	1	5	БМ	4	167	1103	1775	1776
			К1	4	60	740	867	867
			К2	0	12	12	12	12
			К3	0	95	351	896	897
chinese(ф)	2	5	БМ	0	1	2	5	11
			К1	0	0	0	0	0
			К2	0	1	2	3	3
			К3	0	0	0	2	8
das9203(ф)	2	5	БМ	0	1	19	205	1150
			К1	0	0	0	0	945
			К2	0	1	17	195	195
			К3	0	0	2	10	10

Поред скупова пресека датог реда, резултат алгоритма су и одсечена маркирања. Свако од одсечених маркирања је корен једног одсеченог подстабла стабла достижности. Постојање податка о одсеченом маркирању омогућава, ако је потребно, накнадну анализу неистраженог простора стања. Поред тога, податак о броју одсечених маркирања би могао бити показатељ ефикасности различитих модификација алгоритма одсецања ИПМ_2.

Алгоритмима ИПМ_1 и ИПМ_2, одређује се подстабло графа достижности коме одговарају секвенце паљења $\sigma = \langle t_{k_1}, t_{k_2}, \dots, t_{k_r} \rangle$ за које важи да је $k_1 < k_2 < \dots < k_r$. Другим речима, из сваког чвора графа достижности, прелази се у чвор у коме ће жетон добити место са најмањим индексом. С обзиром да је ИПМ нумерисана у дубину и с леве на десну страну, њено претраживање се врши у дубину. Уколико су места, која моделирају примарне догађаје вишеструкости веће од 1, дубоко у

структури ИПМ, сви њихови претходници ће имати велике вредности параметра $k_s^{j\max}$. Због тога ће и вредност (7.6), која представља потенцијалну дужину скупова пресека у стаблу достижности, бити мала и неће доћи до одсецања све док не постане активно неко од места са великим $k_s^{j\max}$.

Поред обилажења ИПМ по дубини, извршене су још две врсте претраживања. Применом оба начина претраживања, добијени су исти резултати као у табели 7.2. Међутим, број одсечених маркирања се разликује. У табели 7.3. је приказан број одсечених маркирања за сва три начина претраживања за три СН.

Табела 7.3. Број одсечених маркирања

СН	приступ	одсецање на 1	одсецање на 2	одсецање на 3	одсецање на 4	одсецање на 5
isp9606	П1	11	143	3	1	0
	П2	20	44	14	1	0
	П3	15	62	16	1	0
chinese(ф)	П1	73	100	144	189	181
	П2	44	63	85	115	139
	П3	60	78	95	140	158
das9203(ф)	П1	200	225	795	2470	0
	П2	10	45	440	3267	0
	П3	200	225	795	2470	0

Први начин претраживања је претраживање у дубину, алгоритмом ИПМ_2, чији су резултати одсецања приказани у редовима табеле 7.3. означеним са П1.

Други начин претраживања подразумева да се, од свих места ИПМ која могу постати активна, прво обиђе место које има највећи показатељ вишеструкости. На тај начин би се вредност (7.6) брже смањивала и може се очекивати да би се неке гране графа достижности раније одсецале. То се постиже модификацијом алгоритма, којом се, у сваком кораку, за нови активни чвор бира чвор са

максималном вредношћу израза: $\sum_{j:(i,q,j) \in R_\lambda} k_s^{j \max}$. Резултат оваквог одсецања приказан је у редовима табеле 7.3. означеним са П2.

Трећи начин претраживања ИПМ је у ширину. Овакво претраживање захтева да ИПМ буде нумерисана у ширину. У алгоритму је укинут услов да индекс новог прелаза буде већи од последњег активног и уведен скуп већ обиђених чворова. Тиме се и даље одређује само једно подстабло графа достижности. Резултати одсецања приликом овог претраживања приказани су у редовима табеле 7.3. означеним са П3.

Укупан број генерисаних чворова графа достижности једнак је збиру броја генерисаних листова, односно мртвих маркирања ИПМ, и броја одсечених чворова, тј. одсечених маркирања. С обзиром да је број листова графа достижности у сва три начина претраживања исти, ефикаснијим се може сматрати оно претраживање у коме је мањи број одсечених маркирања. На основу резултата из табеле 7.3, може се уочити да ниједан од начина претраживања не даје увек мањи број одсечених маркирања. Због тога се, у овом тренутку, може донети само прелиминарни закључак да ефикасност начина претраживања зависи од структуре СН. Даљим истраживањем би требало утврдити који елементи структуре и на који начин утичу на ову ефикасност.

Из табеле 7.2. се може видети да је у три СН из табеле 7.3, дужина, односно максималан број примарних догађаја у минипресецима, једнака пет. У случају СН *chinese*(ф), у табели 7.3. се може уочити да, и када се одређују минипресеци од пет примарних догађаја, постоје одсечена маркирања. То значи да су добијени сви минипресеци, а да није одређено цело стабло достижности. Односно, у случају када треба одредити све минипресеке неког СН, ако је позната максимална дужина његових минипресека, боље је применити алгоритам ИПМ_2 или неку од поменутих његових модификација, него алгоритам ИПМ_1.

8. ЗАКЉУЧАК

Предмет истраживања у овом раду је анализа стабла неисправности, једна од најчешће примењиваних метода у одређивању поузданости производа и система. Проблем, који је посебно изучаван, је одређивање минималних скупова пресека кохерентног СН са вишеструким догађајима. Приступ, који је коришћен у решавању поменутог проблема, заснован је на Петријевим мрежама. Постигнути доприноси и изведени закључци се могу сумирати на следећи начин:

- Предложена је нова метода за одређивање минипресека кохерентног СН са вишеструким догађајима, заснована на анализи графа достижности ИПМ. На основу једначине стања, доказано је да сва мртва маркирања ИПМ, достижна из маркирања $M = [1, 0, \dots, 0]^T$ као почетног, представљају скупове пресека датог СН. На тај начин је проблем одређивања скупова пресека сведен на проблем одређивања листова графа достижности ИПМ датог СН. При томе је доказано, такође на основу једначине стања, да није потребно одредити читав граф достижности, већ само једно његово разапињуће подстабло.
- Развијен је и имплементиран алгоритам за одређивање свих скупова пресека кохерентног СН са вишеструким догађајима на основу листова стабла достижности ИПМ, назван ИПМ_1. Алгоритам ИПМ_1 је примењен на групи проблема за тестирање.
- Анализирани су скупови пресека C^* са особином да, ако садрже неки примарни догађај чија је вишеструкост већа од 1, онда садрже сва његова понављања. Формулисана је теорема о минималности скупова пресека облика C^* . Доказано је да ови скупови јесу минимални, али да могу бити подскуп неког од неминималних скупова пресека. Такође је доказано да су скупови пресека, чији су сви примарни догађаји вишеструкости 1, минимални и да не могу бити подскуп ниједног скупа пресека. Показано је које услове структура СН треба да задовољава да би СН садржало скупове пресека облика C^* . Уколико скуп пресека

облика C^* постоји, могуће је одредити минимални број скупова пресека који га садрже, односно који ће се редуковати као његов надскуп.

– Представљен је нови алгоритам редукције АРСНЗ, који је заснован на подели фамилије скупова пресека у три групе: скупови пресека који садрже само примарне догађаје вишеструкости 1 (скуп $K1$), скупови пресека облика C^* (скуп $K2$) и остали скупови пресека (скуп $K3$). Алгоритам редукције СНЗ се може применити у свакој методи одозго наниже за одређивање минипресека. Алгоритам ИПМ_1 је модификован тако да одређује елементе скупова: $K1$, $K2$ и $K3$ и примењен на групи проблема за тестирање. Уколико скуп $K2$ није празан, број операција поређења скупова пресека се може значајно смањити ако се прво издвоје $K1$ и $K2$ – скупови минипресека, затим елиминишу скупови пресека из $K3$ који су надскупови скупова пресека из $K2$ и на крају изврши међусобно поређење преосталих скупова пресека из $K3$.

– Предложен је приступ у одређивању минипресека СН великих димензија, којим се одређују само минипресеци задатог реда (дужине). Показано је да број места која садрже жетон у сваком маркирању ИМП, односно чвору стабла достижности, не опада, идући од корена ка листовима стабла достижности. На основу тога се за СН без вишеструких догађаја, одређивање минипресека задатог реда своди на одсецање чворова стабла достижности, у којима је број жетона већи од задатог реда. Затим је анализиран параметар одсецања у СН са вишеструким догађајима. Као последица теореме о минималности скупова пресека облика C^* , може се утврдити горња граница максималног скраћења скупова пресека. За сваки чвор стабла достижности, ова граница представља максималан број примена закона идемпотенције над скуповима пресека добијених у листовима подстабла, чији је корен посматрани чвор.

– Развијен је алгоритам за одређивање минималних скупова пресека датог реда за СН са вишеструким догађајима ИПМ_2, којим се, у сваком чвору стабла достижности, одређује вредност параметра одсецања, која гарантује одређивање свих минипресека задатог реда. Алгоритам ИПМ_2 је имплементиран и примењен на групи проблема за тестирање.

– Дат је преглед напознатијих метода и приступа за: одређивање минипресека СН, редуkcију СН и редуkcију скупова пресека. Посебно су обрађене и илуcтроване методе и алгоритми за одређивање минипресека СН помоћу ПМ.

Први правац даљег истраживања биће у смеру побољшања ефикасности алгоритма ИПМ_2, другачијим претраживањем графа достижности. Наиме, предложеним алгоритмом, стабло достижности се одређује тако што прво постају активна места ИПМ са мањим индексом. Може се очекивати да ће алгоритам, којим се формира подстабло графа достижности, обилажењем прво места са већом вредношћу $k_s^{j\max}$, брже налазити минипресеке задатог реда и раније одсечати чворове стабла достижности. Поред тога, може се испитати како промена структуре СН, која не мења његову структурну функцију, утиче на ефикасност алгоритма ИПМ_2.

С обзиром на велику флексибилност теорије ПМ, предложени приступ одређивања минипресека помоћу стабла достижности ИПМ, се може проширити на више начина:

– Приступ се може проширити на некохерента СН. Постојање инхибиторских грана у ПМ, омогућава моделирање негације примарних догађаја, а функције додатних услова паљења прелаза у ПМ омогућавају моделирање операција попут: ексклузивно ИЛИ, приоритетно И и m од n логичких кола;

– ПМ вишег нивоа, посебно Обојене ПМ, пружају могућност увођења вредности жетона која, у случају АСН, може представљати вероватноћу одигравања отказног догађаја;

– Временске и стохастичке ПМ омогућавају моделирање динамичких СН, која у анализу укључују и временску димезију.

ЛИТЕРАТУРА

- [1] F Ahmad, H Huang, and X-L Wang, "Petri net modeling and deadlock analysis of parallel manufacturing processes with shared-resources," *Journal of Systems and Software*, vol. 83, no. 4, pp. 675-688, 2010.
- [2] J A Anderson, *Дискретна математика са комбинаториком*. Београд: СЕТ, 2005.
- [3] B Andrić, D Makajić-Nikolić, B Stevanović, and M Vujošević, "Modelling Inventory Control Process Using Coloured Petri Nets," in *7th Balkan Conference on Operational Research*, Constanta, 2005, pp. 25 - 28.
- [4] J Berger and L Lamontagne, "A Colored Petri Net Model for a Naval Command and Control System," *Lecture Notes in Computer Science*, vol. 691, pp. 532-541, 1993.
- [5] L Bernardinello and F Cindio, "A survey of basic net models and modular net classes, Advances in Petri Nets," The DEMON Project 1992.
- [6] J Billington and B Han, "Modelling and Analysing the Functional Behaviour of TCP's Connection Management Procedures," *International Journal on Software Tools for Technology Transfer*, pp. 269-304, 2007.
- [7] A Birolini, *Reliability Engineering: Theory and Practice*. Heidelberg: Springer, 2010.
- [8] Z Bluvband, R Polak, and P Grabov, "Bouncing failure analysis (BFA): the unified FTA-FMEA methodology," in *Annual Reliability and Maintainability Symp. IEEE*, Piscataway, 2005, pp. 463-467.
- [9] A Bobbio, G Franceschinis, R Gaeta, and L Portinale, "Parametric Fault Tree for the Dependability Analysis of Redundant Systems and Its High-Level Petri Net Semantics," *IEEE Transactions on Software Engineering*, vol. 29, no. 3, pp. 270-287, 2003.
- [10] G V Bochmann and G V Jourdan, "Testing k-Safe Petri Nets," *Lecture Notes in Computer Science*, vol. 5826, pp. 33-48, 2009.
- [11] T B Boffey, *Graph Theory in Operations Research*. London: Macmillan Press, 1982.
- [12] B J Bowles and E C Pelaez, "Application of fuzzy logic to reliability engineering," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 435-449, 1995.
- [13] W Brauer and W Reisig, "Carl Adam Petri and Petri Net," *Informatik-Spektrum*, vol. 29, no. 5, pp. 369-374, 2006.
- [14] M Broy and E Denert, *Software Pioneers*. Berlin: Springer-Verlag, 2002.
- [15] J Campos, G Chiola, and M Silva, "Properties and Performance Bounds for

- Closed Free Choice Synchronized Monoclass Queueing Networks," *IEEE Transactions on Automatic Control*, vol. 36, no. 12, pp. 1368-1381, 1991.
- [16] C Capellmann, H Dibold, and U Herzog, "Using High-Level Petri Nets in the Field of Intelligent Networks," *Application of Petri Nets to Communication Networks, Lecture Notes in Computer Science*, vol. 1605, pp. 1-36, 1999.
- [17] J Cardoso and B Pradin-Chezalviel, "Logic and Fuzzy Petri Nets," in *Manufacturing and PN, XVIII International Conference on Application and Theory of PN*, Toulouse, 1997.
- [18] J A Carrasco and V Sune, "An algorithm to find minimal cuts of coherent fault-trees with event-classes, using a decision tree," *IEEE Transactions on Reliability*, vol. 48, no. 1, pp. 31 – 41, 1999.
- [19] K Z Chakib, "An improved minimal cut set algorithm," *The International Journal of Quality and Reliability Management*, vol. 13, no. 2, pp. 114-132, 1996.
- [20] A Cheng, J Esparza, and J Palsberg, "Complexity results for 1-safe nets," *Theoretical Computer Science*, vol. 147, no. 1-2, pp. 117-136, 1995.
- [21] C H Cheng, Y W Wu, and M C Tsai, "Applying α -cut Approach to Analyze Fuzzy System Reliability," in *VI IFSA World Congress*, Sao Paulo, 1995, pp. 101-104.
- [22] P L Clemens. (2002) Fault Tree Analysis. [Online]. <http://www.fault-tree.net/by-title.htm>
- [23] S Contini and V Matuzas, "Analysis of large fault trees based on functional decomposition," *Reliability Engineering and System Safety*, vol. 96, pp. 383–390, 2011.
- [24] O Coudert and J Madre, "Fault Tree Analysis: 1020 Prime Implicants and Beyond," in *In Proc. Annual Reliability and Maintainability Symposium*, Atlanta, 1993, pp. 240 - 245.
- [25] S Dinghua, "A Unified Algorithm for Computer-aided Fault Tree Analysis," *Scientia Sinica*, vol. XXVII, no. 7, 1984.
- [26] J Dugan, S Bavuso, and M Boyd, "Dynamic fault tree models for fault computer system," *IEEE Transactions on Reliability*, vol. 41, pp. 363-377, 1992.
- [27] C A Ericson II, "Fault Tree Analysis - A History," in *Proceedings of the 17th International System Safety Conference*, Orlando, 1999, pp. 87-96.
- [28] C A Ericson II, *Hazard Analysis technique for System Safety*. New Jersey: John Wiley & Sons, 2005.
- [29] C A Ericson, "What do you do when you run out of computer," in *Proceedings of the 19th international safety conference*, 2001.
- [30] M.H C Everdij and H.A P Blom. (2008) Safety Methods Database. [Online]. <http://www.nlr.nl/documents/flyers/SATdb.pdf>

- [31] J B Fussell, E B Henry, and N H Marshall, *MOCUS a Computer Program to Obtain Minimal Sets from Fault Trees*. Idaho Falls: ANCR 1156 Aerojet Nuclear Company, 1974.
- [32] H Genrich, R Koffner, and K Voss, "Executable Petri Net Models for the Analysis of Metabolic Pathways," *International Journal on Software Tools for Technology Transfer*, vol. 3, no. 4, pp. 394-404, 2001.
- [33] C Girault and R Valk, *Petri Nets for Systems Engineering A Guide to Modelling, Verification, and Applications*. Heidelberg: Springer-Verlag, 2001.
- [34] L González, D García, and B Galván, "An Intrinsic Order Criterion to Evaluate Large, Complex Fault Trees," *IEEE Transactions on reliability*, vol. 53, no. 3, pp. 297-305, 2004.
- [35] S Gordon and J Billington, "Analysing a Missile Simulator with Coloured Petri Nets," *International Journal on Software Tools for Technology Transfer*, vol. 2, pp. 144-159, 1998.
- [36] F Gottschalk, W.M P van der Aalst, M H Jansen-Vullers, and H.M W Verbeek, "Protos2CPN: Using Colored Petri Nets for Configuring and Testing Business Processes," *International Journal on Software Tools for Technology Transfer*, vol. 10, no. 1, pp. 95-110, 2008.
- [37] T Greene, T Brown, and L Route, "Control in AUTOVON Electronic Switching Centers," *IEEE Transactions on Communication Technology*, vol. 17, no. 4, pp. 442 – 446, 1969.
- [38] D F Haasl, "Advanced Concepts in Fault Tree Analysis," in *Proceedings of the System Safety Symposium*, Seattle, 1965.
- [39] Y Y Haines, *Risk Modeling, Assessment, and Management. 3rd Edition*. New Jersey: John Wiley & Sons, 2008.
- [40] S Heitsch, M Köhler, M Martens, and D Moldt, "High-level Petri Nets for a Model of Organizational Decision Making," in *21st International Conference on Application and Theory of Petri Nets*, Aarhus, 2000.
- [41] HPNS. (1998) High-level Petri Nets Standard, Letter Ballot Summary of CD 15909. [Online]. <http://www.petrinets.org>
- [42] H Z Huang, X Tong, and M J Zuo, "Posbist fault tree analysis of coherent systems," *Reliability Engineering and System Safety*, vol. 84, pp. 141–148, 2004.
- [43] G S Hura and J W Atwood, "The Use of Petri Nets to Analyze Coherent Fault-Trees," *IEEE Transaction on Reliability*, vol. 37, pp. 469-474, 1988.
- [44] ISO/IEC. (1997) High-level Petri Nets – Concepts, Definitions and Grafical Notation, Committee Draft ISO/IEC 15909, Version 3.4. [Online]. <http://www.petrinets.org>
- [45] L Jansen, M Meyer, Z Horste, and E Schnieder, "Technical Issues in Modelling

- the European Train Control System," in *Proceedings of the Workshop on Practical Use of Coloured Petri Nets and Design/CPN*, Aarhus, 1998, pp. 103-115.
- [46] K Jensen, *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. Heilderbeg: Springer-Verlag, 1997.
- [47] K Jensen and L M Kristensen, *Coloured Petri Nets. Modelling and Validation of Concurrent Systems*. Heidelberg: Springer, 2009.
- [48] C W Johnson, *Failure in Safety-Critical Systems: A Handbook of Accident and Incident Reporting*. Glasgow: University of Glasgow Press, 2003.
- [49] D B Kececioglu, *Reliability Engineering Handbook*. Lancaster: DEStech Publications, Inc, 2002.
- [50] A Keisner and J Olds, "Reliability Analysis Technique Comparison, as Applied to the Space Shuttle. AE8900 Special Project," Atlanta, 2003.
- [51] D E Knuth, *The Art of Computer Programming, Volume 4, Generating All Tree*. Massachusetts: Addison-Wesley, 2006.
- [52] L M Kristensen and K Jensen, "Specification and Validation of an Edge Router Discovery Protocols for Mobile Ad Hoc Networks. In Integration of Software Specification Techniques for Applications in Engineering," *Notes in Computer Science*, pp.. Springer-Verlag, vol. 3147, pp. 248-269, 2004.
- [53] P M Kwantes, "Design of Clearing and Settlement Operations: A Case Study in business Process Modelling and Evaluation with Petri Nets," in *Proceedings of the 7th Worskhop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, Aarhus , 2006, pp. 217-236.
- [54] S Laftit, J M Proth, and X Xie, "Optimization of Invariant Criteria for Event Graphs," *IEEE Transactions on Automatic Control*, vol. 37, no. 5, pp. 547-555, 1992.
- [55] C Y Lee, "Representation of Switching Circuits by Binary-Decision Programs," *Bell Systems Technical Journa*, vol. 38, pp. 985-999, 1959.
- [56] K H Lee and J Favrel, "Hierarchical Reduction Method for Analysis and Decomposition of Petri Nets," *IEEE Transactions on Systems, Man. And Cybernetics*, vol. SMC-15, no. 2, pp. 272-281, 1985.
- [57] J Li, X Dai, Z Meng, J Dou, and X Guan, "Rapid design and reconfiguration of Petri net models for reconfigurable manufacturing cells with improved net rewriting systems and activity diagrams," *Computers & Industrial Engineering*, vol. 57, no. 4, pp. 1431-1451, 2009.
- [58] N Limnios, *Fault Tree*. Wiltshire: ISTE Ltd, 2007.
- [59] N Limnios and R Ziani, "An Algorithm for Reducing Cut Sets in Fault-Tree Analysis," *IEEE Transactions on Reliability*, vol. 35, no. 5, pp. 559 – 562, 1986.

- [60] B Lindstrom and L Wells, "Simulation Based Performance Analysis in Design/CPN," in *CPN'98, First Workshop and Tutorial on Practical Use of Coloured Petri Nets and Design/CPN*, 1998.
- [61] T S Liu and S B Chiou, "The application of Petri nets to failure analysis," *Reliability Engineering and System Safety*, vol. 57, pp. 129-142, 1997.
- [62] H Liu, Z Jiang, Y Richard, and K Fung, "Performance modeling, real-time dispatching and simulation of wafer fabrication systems using timed extended object-oriented Petri nets," *Computers & Industrial Engineering*, vol. 56, no. 1, pp. 121-137, 2009.
- [63] D Lyon. (2001) On the Use of Stochastic petri Nets for the Performance of Real-time Nth Order Stochastic Composition. [Online]. <http://www.docjava.com/pub/document/cmj>
- [64] D Makajić-Nikolić, B Panić, and M Vujosević, "Bullwhip Effect and Supply Chain Modelling and Analysis using CPN Tools," in *Proceedings of the Fifth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, Aarhus, 2004, pp. 219-234.
- [65] D Makajić-Nikolić and M Vujošević, "Minimal Cut Sets Generation Using Reverse Petri Nets," in *10th BALCOR*, Thessaloniki, 2011, pp. 362-369.
- [66] R Manian, J B Dugan, D Coppit, and K J Sullivan, "Combining various solution techniques for dynamic fault tree analysis of computer systems," in *3rd IEEE International High-Assurance Systems Engineering Symposium (HASE'98)*, Washington, 1998, pp. 21-28.
- [67] Jr, R J McGraw, "Petri Net and Fault Tree Analysis: Combining Two Techniques for a Software Safety Analysis on an Embedded Military Application," Naval Postgraduate School, Monterey, Master's thesis 1989.
- [68] V Mogulothu, H H Ammar, K Lateef, and T Nikzadeh. "Performability Analysis of the Commanding Component of NASA's Earth Observing System" *The 10th Int. Conf. on Parallel and Distributed Computing*, New Orleans, Oct. 1997
- [69] M K Molloy, "Performance Analysis Using Stochastic Petri Nets," *IEEE Transactions on Computers*, vol. C-31, no. 9, pp. 913-917, 1982.
- [70] K H Mortensen and V O Pinci, "Modelling the Work Flow of a Nuclear Waste Management Program," *Lecture Notes in Computer Science*, vol. 815, pp. 376-395, 1994.
- [71] T Murata, "Petri Nets: Properties, Analysis and Applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541-580, 1989.
- [72] H Nguyen, *Transactions on Rough Sets V*. Heidelberg: Springer, 2006.
- [73] H Oberheid, "A Coloured Petri Net Model of Cooperative Arrival Planning in Air Traffic Control," in *Proceedings of the Seventh Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, Aarhus, 2006, pp. 177-

- [74] P.D T O'Connor, *Practical reliability engineering. - 4th ed.* Chichester: John Wiley and Sons, 2002.
- [75] F Ortmeier and G Schellhorn, "Formal Fault Tree Analysis – Practical Experiences," *Electronic Notes in Theoretical Computer Science*, vol. 185, pp. 139–151, 2007.
- [76] A Perkusich, L M Araújo, R S Coelho, K C Gorgônio, and A.J P Lemos, "Design and Animation of Colored Petri Nets Models for Traffic Signals," in *Proceedings of the 2nd Workshop on Practical Use of Coloured Petri Nets and Design/CPN*, Aarhus, 1999, pp. 99-118.
- [77] M Pesic and W.M P van der Aalst, "Modeling Work Distribution Mechanisms Using Colored Petri Nets," *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 3-4, pp. 327-352, 2007.
- [78] C A Petri, "Communication with automata, Supplement 1 to technical report RADC-TR-65-377, vol. 1, Translated by Greene Jr CF from Kommunikation mit Automaten," 1966.
- [79] C A Petri, "Fundamentals of a Theory of Asynchronous Information Flow," in *1st IFIP World Computer Congress*, Munich , 1963, pp. 386-390.
- [80] PNW. (2010) Petri Nets World. [Online]. <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>
- [81] M Praveen and K Lodaya, "Analyzing Reachability for Some Petri Nets With Fast Growing Markings," *Electronic Notes in Theoretical Computer Science*, vol. 223, pp. 215-237, 2008.
- [82] L Priese and H Wimmel, *Petri-Netze*. Heidelberg : Springer, 2008.
- [83] J M Proth and X Xie, *Petri Nets: A Tool For Design And Management Of Manufacturing Systems*. New York: John Wiley & Sons, 1996.
- [84] P Racloz and D Buchs. (2008) *Reseaux de Petri: Verication des propri_et_es*. [Online]. https://smv.unige.ch/teaching/old-courses/Formal_tools_09_10/files/RdPVerifProp.pdf
- [85] D M Rasmuson and N H Marshal, "FATRAM: A Core Efficient Cut-Set Algorithm," *IEEE Transaction on Reliability*, vol. R-27, no. 4, pp. 250-253, 1978.
- [86] M Rausand and A Høyland, *System Reliability Theory; Models, Statistical Methods and Applications*. New York: Wiley, 2004.
- [87] A Rauzy. (2010) A Benchmark of Boolean Formulae. [Online]. <http://iml.univ-mrs.fr/~arauzy/aralia/benchmark.html>
- [88] A Rauzy, "New algorithms for fault tree analysis," *Reliability Engineering and System Safety*, vol. 40, pp. 203-221, 1993.
- [89] K A Reay, *Efficient fault tree analysis using binary decision diagrams*.

Loughborough: Loughborough University, 2002.

- [90] R Remenyte-Prescott and J D Andrews, "An enhanced component connection method for conversion of fault trees to binary decision diagrams," *Reliability Engineering and System Safety*, vol. 93, pp. 1543-1550, 2008.
- [91] (2011) Report of Japanese Government to the IAEA Ministerial Conference on Nuclear Safety - The Accident at TEPCO's Fukushima Nuclear Power Stations. [Online]. <http://www.ornl.gov/sci/nsed/outreach/presentation/2011/Beatty.pdf>
- [92] Z Rochdi, B Driss, and T Mohamed, "Industrial systems maintenance modeling using Petri nets," *Reliability Engineering and System Safety*, vol. 65, no. 2, pp. 119-124, 1999.
- [93] T Runge, "Application of Coloured Petri Nets in Systems Biology," in *Proceedings of the Fifth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, Aarhus, 2004, pp. 77-96.
- [94] S Sankaranarayanan, H Sipma, and Z Manna, "Petri Net Analysis Using Invariant Generation Verification: Theory and Practice," *Lecture Notes in Computer Science*, vol. 2772, pp. 188-189, 2004.
- [95] G Scheschonk and M Timpe, "Simulation and Analysis of a Document Storage System," *Lecture Notes in Computer Science*, vol. 815, pp. 454-470, 1994.
- [96] K Schmidt, "Computation of Invariants for Algebraic Petri Nets," in *Proceedings of the CS & P Workshop*, Nieborow, 1993.
- [97] S N Semanderes, "ELRAFT: A computer program for the Efficient Logic Reduction Analysis of Fault Tree," *IEEE Transaction on Nuclear Science*, vol. 18, no. 1, pp. 481-487, 1971.
- [98] D M Shalev and J Tiran, "Condition-based fault tree analysis (CBFTA): A new method for improved fault tree analysis (FTA), reliability and safety calculations," *Reliability Engineering & System Safety*, vol. 92, no. 9, pp. 1231-1241, 2007.
- [99] R M Shapiro, "Validation of a VLSI Chip Using Hierarchical Coloured Petri Nets," *Journal of Microelectronics and Reliability, Special Issue on Petri Nets*, pp. 667-687, 1991.
- [100] R M Sinnamon and J D Andrews, "New approaches to evaluating fault trees," *Reliability Engineering and System Safety*, vol. 58, pp. 89-96, 1997.
- [101] R H Sloan and U Buy, "Reduction rules for time Petri nets," *Acta Informatica*, vol. 33, pp. 687-706, 1996.
- [102] E Sloane and B Gelhot, "Applications of the Petri net to simulate, test, and validate the performance and safety of complex, heterogeneous, multi-modality patient monitoring alarm systems Engineering in Medicine and Biology Society," in *IEMBS '04*, 2004.
- [103] (2011) SRPS EN 61025:2011. [Online]. http://www.iss.rs/standard/?natstandard_document_id=34234

- [104] M Stamatelatos and W Vesely, *Fault Tree Handbook with Aerospace Applications*. Washington: NASA Headquarters, 2002.
- [105] D Stanton and D White, *Constructive Combinatorics*. New York: Springer-Verlag, 1986.
- [106] P V Suresh, A K Babar, and V V Raj, "Uncertainty in Fault Tree Analysis: A Fuzzy Approach," *Fuzzy Sets and Systems*, vol. 83, pp. 135-141, 1996.
- [107] H.L T Tanaka, F S Fan, and K Toguchi, "Fault-Tree Analysis by Fuzzy Probability," *IEEE Transactions on Reliability*, vol. R-32, no. 5, pp. 453-457, 1983.
- [108] A Törn. (2008) Modeling primitives. [Online]. <http://users.abo.fi/atorn/Petri/P22.html>
- [109] (1975) US Nuclear Regulatory Commission: Reactor Safety Study: An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants, Appednix II. [Online]. http://www.professormcgill.com/blog/WASH1400/WASH1400-Appendix_II.pdf
- [110] M Uzam, "The use of the Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems," *The International Journal of Advanced Manufacturing Technology*, vol. 23, no. 3-4, pp. 204-219, 2004.
- [111] VB. (2010) [Online]. <http://msdn.microsoft.com/en-us/vstudio/hh388568>
- [112] W E Vesely, F F Goldberg, N H Roberts, and D F Haasl, *Fault Tree Handbook.:* NURREG-0492, U.S. Nuclear Regulatory Commission, 1982.
- [113] M Vujošević, D Makajić-Nikolić, and M Strak, "Fuzzy Petri Net Based Reasoning for the Diagnosis of Bus Condition," in *Proceedings of NEUREL 2004: Seventh Seminar on Neural Networks Applications in Electrical Engineering*, Belgrade, 2004, pp. 225-229.
- [114] Y S Way and D Y Hsia, "A simple component-connection method for building decision diagrams encoding a fault tree," *Reliability and System Safety*, vol. 70, pp. 59-70, 2000.
- [115] R R Willie, *Computer-aided Fault Tree Analysis*. Berkeley: OR Center, 1978.
- [116] D B Wheeler, J S Hsuan, R R Duersch, and G M Roe, "Fault Tree Analysis Using Bit Manipulation," *IEEE Transactions on Reliability*, vol. R-26, no. 2, pp. 9-99, 1977.
- [117] Y Wu, L Xie, and Y Yue, "Study of Fault Analysis Technology by Means of Petri Nets," *International Journal of Performability Engineering*, vol. 6, no. 3, pp. 269-277, 2010.
- [118] X Yang, J Li, W Liu, and P Guo, "Petri Net model and Reliability Evaluation for Wind Turbine Hydraulic Variable Pitch Systems," *Energies*, vol. 4, pp. 978-997, 2011.

- [119] S Yang and T Liu, "Failure Analysis for an Airbag Inflator by Petri Nets," *Quality and Reliability Engineering International*, vol. 13, no. 3, pp. 139-151, 1997.
- [120] D A Zaitsev, "Decomposition-based Calculation of Petri net Invariants," in *Proceedings of Workshop on Token based computing (ToBaCo), 25-th International conference on application and theory of Petri nets*, Bologna, 2004, pp. 79-83.
- [121] K Zurowska and R Deters, "Overcoming Failures in Composite Web Services by Analysing Colored Petri Nets," in *Proceedings of the Eighth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, vol. 87-106, Aarhus, 2007.
- [122] М Вујошевић, "Анализа стабла неисправности: преглед основних појмова и техника," *Техника*, vol. 38, no. 11, pp. 1546-1555, 1983.
- [123] М Вујошевић, "Управљање ризицима као сегмент интегрисаног система менаџмента," *Total Quality Management & Excellence*, vol. 36, no. 4, 2008.
- [124] В Е Котов, *Сети Петри*. Москва: Наука, 1984.
- [125] С Крчевинац, М Чангаловић, В Ковачевић-Вујчић, М Мартић, and М Вујошевић, *Операциона истраживања*. Београд: ФОН, 2004.
- [126] Д Макајић-Николић, Б Андрић, М Станојевић, and П Павловић, "Симулација рада семафора помоћу петријевих мрежа," in *XXXIII Симпозијум о операционим истраживањима*, Бања Ковиљача, 2006, pp. 433-436.
- [127] Д Макајић-Николић and М Вујошевић, "Анализа стабла неисправности када су вероватноће примарних догадаја фази бројеви," in *XXVII SYM-OP-IS*, Београд, 2000, pp. 313-317.
- [128] Д Макајић-Николић, М Вујошевић, and Б Васић, "Примена обојених петријевих мрежа у моделирању система одржавања аутобуса," in *SYM-OP-IS*, 2001.
- [129] Д Макајић-Николић, М Вујошевић, and Н Николић, "Одређивање минималних скупова пресека помоћу Петријевих мрежа," in *XXXI SYM-OP-IS*, Фрушка Гора, 2004, pp. 217-220.
- [130] Н Небојша and Д Макајић-Николић, "Решавање проблема распоређивања у производним системима применом Петријевих мрежа," in *СимОрг*, Златибор, 2004.
- [131] Н Новокмет, Д Макајић-Николић, and М Вујошевић, "Моделирање процеса пријема и испоруке пошиљака помоћу петријевих мрежа," in *XXXIII SYM-OP-IS*, Златибор, 2007, pp. 385-388.
- [132] Б Панић, Б Лончар, and Д Макајић-Николић, "Примена Петријевих мрежа у моделирању процеса издавања часописа YUJOR," in *SYM-OP-IS'02*, Тара, 2002, pp. V-1-V-5.

- [133] Љ Папић, "Основне концепције и модели у анализи стабла отказа," in *DQM, зборник радова*, Чачак, 2004, pp. 6-36.
- [134] В Петровић, *Теорија графова*. Нови Сад: ПМФ, 1998.
- [135] Р Петровић, М Вујошевић, and Д Петровић, *Оптимизација редундантних система*. Београд: Саобраћајни факултет, 1993.
- [136] И Петровић, Д Макајић-Николић, and М Вујошевић, "Моделирање и симулација процеса пријема робе код велепродавца воћа применом петријевих мрежа," in *XXXII SYM-OP-IS*, Врњачка Бања, 2005, pp. 373-376.
- [137] (2009) ТР. [Online]. http://www.ef.uns.ac.rs/Download/menadzment_rizikom_master/2009-11-05_teorija_rizika.pdf
- [138] Д Цветковић, *Теорија графова и њене примене*. Београд: Научна књига, 1990.
- [139] Д Цветковић and С Симић, *Одабрана поглавља из дискретне математике*. Београд: Академска мисао, 2002.
- [140] Д Цветковић et al., *Комбинаторна оптимизација: Математичка теорија и алгоритми*. Београд: ДОПИС, 1996.

БИОГРАФИЈА

Драгана Макајић-Николић рођена је 29.11.1969. године у Неготину. Основну и средњу школу завршила је у Зајечару са одличним успехом. Факултет организационих наука уписала је 1990. године. Дипломирала је 1994. године на смеру за менаџмент са средњом оценом 8.97 (осам, 97/100) и дипломским радом, на предмету Операциона истраживања, оцењеним оценом 10. Исте године уписала је последипломске студије, смер операциона истраживања, Положила је све планом предвиђене испите са просечном оценом 9.89 (девет, 89/100), а јуна 2002. године одбранила је магистарску тезу под називом: „Примена обојених Петријевих мрежа у реинжењерингу пословних процеса”, под менторством проф.др Мирка Вујошевића.

Од 01.04.1995. до 01.04.1997. године радила је на Факултету организационих наука као сарадник обдарен за научно-истраживачки рад на предмету Операциона истраживања. Од маја 1997. до октобра 1998. године учествовала је на пројекту "Оптимизација асортимана у ICN Југославија" на коме су ангажовани чланови Лабораторије за операциона истраживања Факултета организационих наука. Од октобра 1998. до јуна 2003. године радила на Факултету организационих наука као асистент-приправник на предмету Операциона истраживања, а од јуна 2003. године као асистент на области Операциона истраживања.

Учествовала је на пет научно истраживачких пројекта који су финансирани од стране Министарства за науку и технологију. Такође, била је члан организационог одбора две домаће и једне међународне научне конференције у организацији ФОН-а. Ангажована је као сарадник на пројекту мастер студија за индустријско инжењерство који је реализован у сарадњи ФОН-а и Централне школе из Париза.

Прилог 1.

Изјава о ауторству

Потписана Драгана Макаић-Николић

број индекса 432/08

Изјављујем

да је докторска дисертација под насловом

НОВИ ПРИСТУП АНАЛИЗИ ПОУЗДАНОСТИ СИСТЕМА ПРИМЕНОМ

ИНВЕРЗНИХ ПЕТРИЈЕВИХ МРЕЖА

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

Потпис докторанда

У Београду, 30.04.2012.

Д. Макаић-Н.

Прилог 2.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора Драгана Макајић-Николић

Број индекса 432/08

Студијски програм _____

Наслов рада НОВИ ПРИСТУП АНАЛИЗИ ПОУЗДАНОСТИ СИСТЕМА
ПРИМЕНОМ ИНВЕРЗНИХ ПЕТРИЈЕВИХ МРЕЖА

Ментор проф. др Мирко Вујошевић

Потписана Драгана Макајић-Николић

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду**.

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, 30.04.2012.

Д. Макајић-Н.

Прилог 3.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

НОВИ ПРИСТУП АНАЛИЗИ ПОУЗДАНОСТИ СИСТЕМА ПРИМЕНОМ
ИНВЕРЗНИХ ПЕТРИЈЕВИХ МРЕЖА

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство

2. Ауторство - некомерцијално

3. Ауторство – некомерцијално – без прераде

4. Ауторство – некомерцијално – делити под истим условима

5. Ауторство – без прераде

6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

Потпис докторанда

У Београду, 30.04.2012.

2. Markić A.