



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS**  
**TAIKOMOSIOS MATEMATIKOS KATEDRA**

**Darius Naujokaitis**

**SISTEMOS GEDIMO MODELIAVIMAS**  
**IR TIKIMYBINIS VERTINIMAS**

Magistro darbas

**Vadovas**  
**doc. dr. R. Alzbutas**

**KAUNAS, 2010**



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS**  
**TAIKOMOSIOS MATEMATIKOS KATEDRA**

**TVIRTINU**

**Katedros vedėjas**

**doc. dr. N. Listopadskis**

**2010 06 08**

**SISTEMOS GEDIMO MODELIAVIMAS**  
**IR TIKIMYBINIS VERTINIMAS**

Taikomosios matematikos magistro baigiamasis darbas

**Vadovas**

**(parašas) doc. dr. R. Alzbutas**

**2010 06 03**

**Recenzentas**

**(parašas) dr. V. Matuzas**

**2010 06 01**

**Atliko**

**FMMM 8 gr. stud.**

**(parašas) D. Naujokaitis**

**2010 05 25**

**KAUNAS, 2010**

## KVALIFIKACINĖ KOMISIJA

**Pirmininkas:** Leonas Saulis, profesorius (VGTU)

**Sekretorius:** Eimutis Valakevičius, docentas (KTU)

**Nariai:** Algimantas Jonas Aksomaitis, profesorius (KTU)

Vytautas Janilionis, docentas (KTU)

Vidmantas Povilas Pekarskas, profesorius (KTU)

Rimantas Rudzkis, habil. dr., vyriausiasis analitikas (DnB NORD Bankas)

Zenonas Navickas, profesorius (KTU)

Arūnas Barauskas, dr., vice-prezidentas projektams (UAB „Baltic Amadeus“)

**Naujokaitis D. Modelling and Probabilistic Assessment of System Failure: Master thesis in applied mathematics / supervisor dr. Alzbutas R.; Department of Applied mathematics, Faculty of Fundamental Sciences, Kaunas University of Technology. – Kaunas, 2010. – 74 p.**

## **SUMMARY**

Modelling of system failure usually is performed applying so-called fault trees. Assessment of system failure probability and failure occurrence causes usually is performed developing a minimal cut-set (MCS) and according to it evaluating probability of system failure. The last twenty years a binary decision diagram (BDD) has been applied for the exact evaluation of system failure. However, a methodology of creation and application of BDD is still under development. Till now in Lithuania for the assessment of system failure probability BDD has not been applied yet.

Presenting a performed work the main methods of development of MCS and BDD as well as features of their application is described. Also, the model and assessment of failure of Kaunas hydropower dam gates' control system is presented. For this task the different methods of development of MCS and BDD as well as corresponding software was applied. Having performed the testing calculations a comparison of results is presented too. In the work the main advantages and disadvantages of MCS and BDD application are described and newly created software "DemoITE" is introduced. The developed algorithms and universal software "DemoITE" is used for visualization of failure causes and for exact estimation of the investigated system failure probability. Also, it is proposed the order of basic events on purpose to design the least BDD.

## SANTRAUKA

Sistemų gedimo modeliavimas paprastai vykdomas taikant taip vadinamus gedimų medžius. Sistemos gedimo tikimybės ir gedimo atsiradimo priežasčių vertinimas dažniausiai atliekamas sudarant minimalių kirtimų aibę (MCS) ir pagal ją įvertinant gedimo tikimybę. Pastaruosius dvidešimt metų tiksliam gedimo tikimybės vertinimui yra taikoma dvejetainė sprendimų diagrama (BDD). Tačiau BDD sudarymo ir taikymo metodika vis dar yra tobulinama. Iki šiol Lietuvoje sistemos gedimo tikimybės vertinimui BDD dar nebuvo taikyta.

Pristatant atliktą darbą aprašomi pagrindiniai MCS ir BDD sudarymo metodai bei jų taikymo ypatybės. Taip pat pateikiamas Kauno hidroelektrinės užtvankos vartų valdymo sistemos gedimo modelis ir gedimo tikimybės vertinimas. Šiuo tikslu buvo taikomi skirtingi MCS ir BDD sudarymo metodai bei atitinkamos programinės priemonės. Atlikus bandomuosius skaičiavimus taip pat pateikiamas gautų rezultatų palyginimas. Darbe apibūdinami pagrindiniai MCS ir BDD taikymo privalumai ir trūkumai bei pristatoma naujai sukurta programinė priemonė „DemoITE“. Taikant išanalizuotus algoritmus bei sukurta programinę priemonę „DemoITE“ buvo ištirtos ir nustatytos sistemos gedimo priežastys bei tiksliai įvertinta sistemos gedimo tikimybė. Be to, pasiūlyta bazinių įvykių rangavimo principas, įgalinantis gauti mažiausią BDD struktūrą.

## TURINYS

|  |           |
|--|-----------|
| <b>Summary</b> .....   | <b>4</b>  |
| <b>Santrauka</b> .....   | <b>5</b>  |
| <b>Turinys</b> .....   | <b>6</b>  |
| <b>Lentelių sąrašas</b> .....  | <b>8</b>  |
| <b>Paveikslų sąrašas</b> .....   | <b>9</b>  |
| <b>Įvadas</b> .....  | <b>10</b> |
| <b>1. Teorinė dalis</b> .....  | <b>12</b> |
| 1.1. Sistemos gedimų medis .....                                       | 12        |
| 1.1.1. <i>Gedimų medžio sudarymas</i> .....                            | 12        |
| 1.1.2. <i>Koherentinis ir ne-koherentinis gedimų medis</i> .....       | 13        |
| 1.2. Minimalių kirtimų aibė .....                                      | 15        |
| 1.2.1. <i>Gedimų medžio išraiška Būlio algebroje</i> .....             | 16        |
| 1.2.2. <i>Baigiamojo įvykio tikimybės vertinimas taikant MCS</i> ..... | 17        |
| 1.3. Programa „OpenFTA“ .....  | 19        |
| 1.4. Dvejetainė sprendimų diagrama .....                               | 22        |
| 1.4.1. <i>Teisingumo lentelės taikymas</i> .....                       | 22        |
| 1.4.2. <i>ITE sąlygos taikymas</i> .....                               | 25        |
| 1.4.3. <i>Komponentų jungimo metodas</i> .....                         | 27        |
| 1.4.4. <i>Gedimo tikimybės vertinimas BDD struktūroje</i> .....        | 29        |
| 1.5. Programa „VisBDD“ .....   | 31        |
| <b>2. Tiriamoji dalis ir rezultatai</b> .....                          | <b>33</b> |
| 2.1. Kauno hidroelektrinė ir jos gedimų medis .....                    | 33        |
| 2.1.1. <i>Pagrindinės charakteristikos</i> .....                       | 33        |
| 2.1.2. <i>Užtvankos vartų valdymo sistema</i> .....                    | 34        |
| 2.2. Sistemos analizė taikant MCS.....                                 | 35        |
| 2.2.1. <i>Sistemos minimalių kirtimų aibė</i> .....                    | 36        |
| 2.2.2. <i>Sistemos gedimo tikimybė</i> .....                           | 36        |
| 2.2.3. <i>Sistemos vertinimas naudojant „OpenFTA“</i> .....            | 37        |
| 2.3. Sistemos analizė taikant BDD .....                                | 38        |
| 2.3.1. <i>Teisingumo lentelės taikymas</i> .....                       | 38        |
| 2.3.2. <i>ITE sąlygos taikymas</i> .....                               | 41        |
| 2.3.3. <i>Sistemos gedimo tikimybės įvertinimas</i> .....              | 42        |
| 2.3.4. <i>Sistemos BDD konstravimas taikant „VisBDD“</i> .....         | 44        |
| 2.3.5. <i>Gedimo tikimybės vertinimas taikant „DemoITE“</i> .....      | 45        |
| 2.4. Gedimo tikimybės vertinimo palyginimas .....                      | 46        |
| 2.5. Programa „DemoITE“ .....  | 48        |
| 2.5.1. <i>Realizuoti algoritmai</i> .....                              | 49        |

|   |           |
|---|-----------|
| 2.5.2. Skaičiavimo patikrinimas.....                              | 49        |
| 2.6. Bazinių įvykių rangavimas .....                              | 50        |
| <b>Diskusija.....</b>   | <b>51</b> |
| <b>Išvados.....</b>   | <b>54</b> |
| <b>Šaltiniai ir literatūra .....</b>                              | <b>55</b> |
| <b>1 Priedas. „OpenFTA“ programos rezultatai.....</b>             | <b>57</b> |
| <b>2 Priedas. Sistemos gedimų medžio teisingumo lentelė .....</b> | <b>58</b> |
| <b>3 Priedas. Duomenų failas: „ITE_Kauno_HE.txt“ .....</b>        | <b>59</b> |
| <b>4 Priedas. Programos „DemoITE“ realizacijos failai .....</b>   | <b>60</b> |
| <b>5 Priedas. Bazinių įvykių rangavimas .....</b>                 | <b>71</b> |

## LENTELIŲ SĄRAŠAS

|  |    |
|--|----|
| 1.1 lentelė Gedimų medžiuose naudojami žymėjimai .....                                       | 12 |
| 1.2 lentelė Gedimų medžio jungčių žymėjimai .....  | 16 |
| 1.3 lentelė Būlio algebros taisyklės.....  | 16 |
| 1.4 lentelė „OpenFTA“ naudojamo algoritmo rezultatai .....                                   | 21 |
| 1.5 lentelė Elektros tiekimo sistemos gedimo teisingumo lentelė .....                        | 22 |
| 2.1 lentelė Kauno HE užtvankos vartų valdymo sistemos komponentai .....                      | 35 |
| 2.2 lentelė Sistemos bazinių įvykių perkodavimas .....                                       | 44 |
| 2.3 lentelė Tikimybės vertinimo palyginimas, esant vienodoms bazinių įvykių tikimybėms ..... | 46 |
| 2.4 lentelė Tikimybės vertinimo palyginimas, esant vienodoms minimalių kirtimų tikimybėms.   | 47 |
| 2.5 lentelė BDD struktūros dydis esant skirtingiems rangavimams .....                        | 50 |



## PAVEIKSLŲ SĄRAŠAS

|  |    |
|--|----|
| 1.1 pav. Energijos tiekimo sistemos gedimas .....                              | 13 |
| 1.2 pav. Ne-koherentiškas dujų rezervuaro sistemos gedimų medis. ....          | 15 |
| 1.3 pav. „OpenFTA“ programos langai .....                                      | 20 |
| 1.4 pav. Energijos tiekimo sistemos gedimo BDD .....                           | 23 |
| 1.5 pav. BDD panaudojus (I) taisyklę.....                                      | 24 |
| 1.6 pav. BDD panaudojus (II) taisyklę .....                                    | 25 |
| 1.7 pav. BDD panaudojus (III) taisyklę.....                                    | 25 |
| 1.8 pav. BDD taikant ITE sąlygą .....  | 27 |
| 1.9 pav. Komponentų jungimo metodu gauta BDD .....                             | 28 |
| 1.10 pav. BDD struktūros tikimybinis vertinimas a) atvejui.....                | 30 |
| 1.11 pav. BDD struktūros tikimybinis vertinimas b) atvejui .....               | 30 |
| 1.12 pav. „VisBDD“ programos langas .....                                      | 31 |
| 2.1 pav. Kauno HE užtvanka.....  | 34 |
| 2.2 pav. Kauno HE užtvankos vartų valdymo sistemos gedimų medis .....          | 35 |
| 2.3 pav. Kauno HE vartų valdymo gedimų medis programoje „OpenFTA“ .....        | 38 |
| 2.4 pav. Kauno HE užtvankos vartų valdymo sistemos nekompaktiška BDD .....     | 39 |
| 2.5 pav. Kauno HE užtvankos vartų valdymo sistemos kompaktiška BDD .....       | 40 |
| 2.6 pav. BDD pagal ITE sąlygą .....  | 42 |
| 2.7 pav. Iteratyvus sistemos gedimo tikimybės vertinimas.....                  | 43 |
| 2.8 pav. Kauno HE užtvankos vartų valdymo sistema programoje „VisBDD“ .....    | 44 |
| 2.9 pav. Sistemos gedimo tikimybės vertinimas taikant programą „DemoITE“ ..... | 45 |
| 2.10 pav. Programos „DemoITE“ langas .....                                     | 48 |

## ĮVADAS

Sparčiai vystantis technologijoms įvairiose srityse (branduolinėje energetikoje, chemijos, transporto pramonėje) neišvengiama įvairių įrenginių gedimų, kurie gali sukelti sunkius padarinius, pavyzdžiui: sprogus branduoliniam reaktoriui į aplinką gali būti išmetama radioaktyvių dalelių. Gedimus gali sukelti netinkamas įrenginio eksploatavimas, įvairūs gamtos veiksniai ir kita. Sugedęs ir laiku nesutaisytas įrenginys gali pridaryti daugiau žalos nei jo teikiama nauda. Todėl labai svarbu įvertinti eksploatuojamų ir kuriamų įrengimų riziką (Augutis, Ušpuras, 2006).

Pagrindinis bendras šio darbo objektas yra sistemos gedimo tikimybė – viena iš pagrindinių rizikos vertinime naudojamų komponentų. Sistemos gedimo tikimybės vertinimas paremtas įvairių įrenginio charakteristikų kokybiniu ir kiekybiniu vertinimu. Vertinant sistemos gedimo tikimybę dažniausiai sudaromas sistemos gedimų medis (angl. k. fault tree) ir ieškoma gedimų medžio minimalių kirtimų (angl. k. minimal cut), bei vertinama minimalių kirtimų aibės MCS (angl. k. minimal cut-sets) tikimybė. Dažniausiai įvairios programos, kuriose realizuotas minimalių kirtimų aibę sudarantis ir vertinantis algoritmas, naudoja gedimo tikimybės aproksimacijas.

Naujausiuose darbuose naudojama nauja sistemos gedimo tikimybės vertinimo metodika grindžiama dvejetainėmis sprendimų diagramomis - BDD (angl. k. Binary Decision Diagrams). Žinoma, jog šis metodas teoriškai yra efektyvesnis ir tikslesnis, nes sistemos gedimo įvykių tikimybės galima gauti tiksliai ir nebūtina vertinti minimalių kirtimų aibės.

Taikomasis šio darbo objektas yra Kauno hidroelektrinės (HE) užtvankos vartų valdymo sistema. Kauno HE labai reikšminga Kauno regionui ir visai Lietuvai. Ji iš dalies reguliuoja Nemuno vandens lygį ir gamina elektros energiją. Netinkamai eksploatuojant hidroelektrinę vanduo gali pralaužti užtvanką ir užtvindydamas Nemuno vagos regionus pridaryti labai daug žalos.

Darbo tikslas:

Palyginti sistemos gedimo modeliavimo metodus ir atlikti realios sistemos (Kauno hidroelektrinės užtvankos vartų valdymo) gedimo tikimybinį vertinimą.

Darbo uždaviniai:

1. Išstudijuoti sistemos gedimo tikimybės vertinimo metodus ir taikomas programines priemones.
2. Formalizuoti sistemos gedimo tikimybės tikslaus vertinimo metodą ir aprašyti skaičiavimo algoritmą bei išplėtotas programines priemones.
3. Atlikti realios sistemos vertinimą lyginant nagrinėtus metodus.

Sprendžiant darbe užsibrėžtus uždavinius buvo sukurta programinė įranga „DemoITE“, skirta apskaičiuoti sistemos, išreikštos dvejetainė sprendimų diagrama, gedimo tikimybę.

Reikšmingiausi darbo rezultatai yra susiję su sistemos gedimo analize, siekiant kokybiškai ir kiekybiškai įvertinti sistemos gedimą, t.y. rasti sistemos gedimo priežastis ir gedimo tikimybę.

Pirma darbo dalis skirta aptarti sistemos gedimo tikimybės vertinimo problematiką ir išanalizuoti metodus, taikomus sistemos gedimui vertinti. Antroje dalyje įvertinamas Kauno hidroelektrinės vartų valdymo sistemos gedimas, įvertinama sistemos gedimo tikimybė. Palyginama vertinime naudoti metodai ir programinės priemonės.

Šio darbo temoje buvo skaitytas pranešimas „Sistemos gedimo tikimybinis vertinimas“ ir publikuotas straipsnis VII studentų konferencijoje „Taikomoji Matematika“. Be to, tema „Dvejetainės sprendimų diagramos taikymas sistemos gedimo vertinime“ bus skaitomas pranešimas ir publikuojamas straipsnis 51-oje Lietuvos matematikų draugijos konferencijoje.

## 1. TEORINĖ DALIS

### 1.1. SISTEMOS GEDIMŲ MEDIS






Gedimų medis yra vienas iš kokybinių ir kiekybinių įrankių naudojamų modeliuojant ir vertinant sistemos gedimą. Tokie medžiai plačiai taikomi vertinant įvairių pramonės (branduolinės energetikos, chemijos, transporto pramonės) objektų saugumą ir patikimumą. Taip pat gedimų medžiai taikomi kuriant įvairių įrengimų saugumo sistemas ir kita.

#### 1.1.1. GEDIMŲ MEDŽIO SUDARYMAS

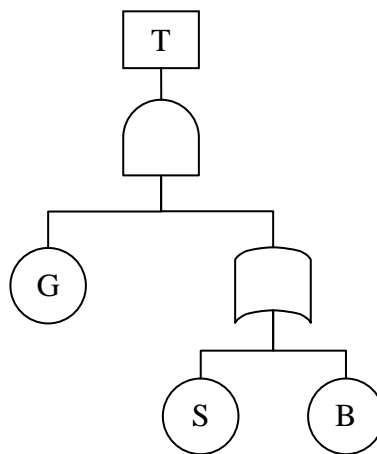
Sistemos gedimo medis sudaromas iš bazinių įvykių, loginių jungčių ir baigiamojo įvykio (angl. k. top event). Baziniai įvykiai yra įvykiai (komponentų gedimai), kuriems įvykus sistema gali sugesti. Loginės jungtys nusakomos dažniausiai naudojamais Būlio algebros loginiais operatoriais: *IR*, *ARBA* bei *NE*. Sistemos gedimą apibūdina baigiamasis įvykis, kuris išreiškiamas sujungus bazinius įvykius loginėmis jungtimis. Pagrindiniai sistemos būsenų ir jungčių žymėjimai pateikti 1.1 lentelėje.

1.1 lentelė

Gedimų medžiuose naudojami žymėjimai

| Žymėjimai   | Reikšmės             |
|---|----------------------|
|  | Bazinis įvykis A     |
|  | Baigiamasis įvykis T |
|  | <i>Jungtis IR</i>    |
|  | Jungtis <i>ARBA</i>  |
|  | Jungtis <i>NE</i>    |

Gedimų medžio pavyzdys parodytas 1.1 paveiksle. Sistemos gedimą T apibūdina trys baziniai įvykiai. Energija nebus tiekama jei suges generatorius G ir suges sistemos jungiklis S arba sistemos baterija B.



**1.1 pav. Energijos tiekimo sistemos gedimas**

Konstruojant gedimų medį svarbu žinoti, kas gali sistemai nutikti ir kada tai gali įvykti. Baziniai įvykiai turi kuo detaliau atspindėti sistemos komponentų gedimą, kad galėtume aiškiau suprasti sistemos gedimą. Gedimų medis sudaromas nuo baigiamojo įvykio žemyn link bazinių įvykių, dažnai naudojant tarpinius įvykius, apibūdinančius sistemos gedimo stadijas (NASA, 2002). Sudarant gedimų medį patariama naudoti Kumamoto ir Henley (1996) euristines rekomendacijas:

- (i) mažinti įvykio apibrėžimo abstraktumą;
- (ii) skaidyti įvykius į labiau elementarius įvykius;
- (iii) nustatyti skirtingas įvykio priežastis;
- (iv) susieti poromis priežastinius įvykius su „neapsaugotais veiksmais“. Pavyzdžiui: perkaitimas arba aušinimo netekimas susiejamas su „sistemos neišsijungimu“;
- (v) nustatyti įvykius galinčius įvykti dėl keleto priežasčių;
- (vi) tiksliai nustatyti gedimo įvykio komponentus;

Vesely ir kt. (1981) pateikia dar tris rekomendacijas, padedančias sukurti gedimų medį:

- (i) jei normalų komponento funkcionavimą galima aprašyti gedimų seka, tai tariama, kad komponentas funkcionuoja normaliai.
- (ii) plėtojant gedimų medį žemyn, pirmiausia pilnai aprašomi visi į konkrečią jungtį įeinantys įvykiai.
- (iii) gedimo įvykiai, sujungiami jungtimis, turi visada būti tinkamai apibrėžti ir jungiant jungtį su kita jungtimi reikia naudoti tarpinius įvykius.

Pirmoji Vesely ir kt. (1981) rekomendacija padeda pasiekti, kad gedimų medis būtų koherentiškas (1.1.2. poskyris), o kitos padeda suvokti sistemos scenarijų vystymosi logiką.

### **1.1.2. KOHERENTINIS IR NE-KOHERENTINIS GEDIMŲ MEDIS**

Gedimų medis vadinamas koherentišku, kai jį sudaro: baziniai įvykiai, baigiamasis įvykis ir tik jungtys *IR* bei *ARBA*.

Ankstesniame poskyryje pristatytas gedimų medžio konstravimas skirtas koherentiniams gedimų medžiams (Vesely ir kt., 1981). Tokie medžiai dažniausiai naudojami saugos (rizikos) analizėje.

Saugos analizėje naudojami ir ne-koherentiniai medžiai (Sharvia, Papadopoulos, 2008). Jungiant bazinius įvykius į ne-koherentinius medžius naudojami ne tik *IR* bei *ARBA*, bet ir Būlio algebros operatorius *NE*.

Įprastai laikoma, kad gedimų medžiuose priešingas gedimo įvykiui įvykis yra veikimo arba sėkmės įvykis, kuris nenagrinėjamas klasikinėje gedimų medžio analizėje (Bedford, Cooke, 2007). Ne-koherentiniuose gedimų medžiuose tiek komponento gedimo būseną tiek ir veikimo būseną gali sukelti sistemos gedimą.

Kad galėtume atskirti koherentinį gedimų medį nuo ne-koherentinio apibrėžkime įvykių būsenas tokiu būdu (Andrews, 2000):

$$x_i \equiv \begin{cases} 1 & \text{jei komponentas } i \text{ sugedęs} \\ 0 & \text{jei komponentas } i \text{ veikiančis} \end{cases}$$

kur  $i = 1, 2, \dots, n$ ,  $n$  – sistemos komponentų kiekis. O funkcija  $\phi$  apibrėžia sistemos būseną:

$$\phi \equiv \begin{cases} 1 & \text{sistema sugedusi} \\ 0 & \text{sistema veikianči} \end{cases}$$

kur  $\phi = \phi(x)$  ir  $x = (x_1, \dots, x_n)$ .

Tada gedimų medžio koherentiškumas apibrėžiamas tokia funkcija:

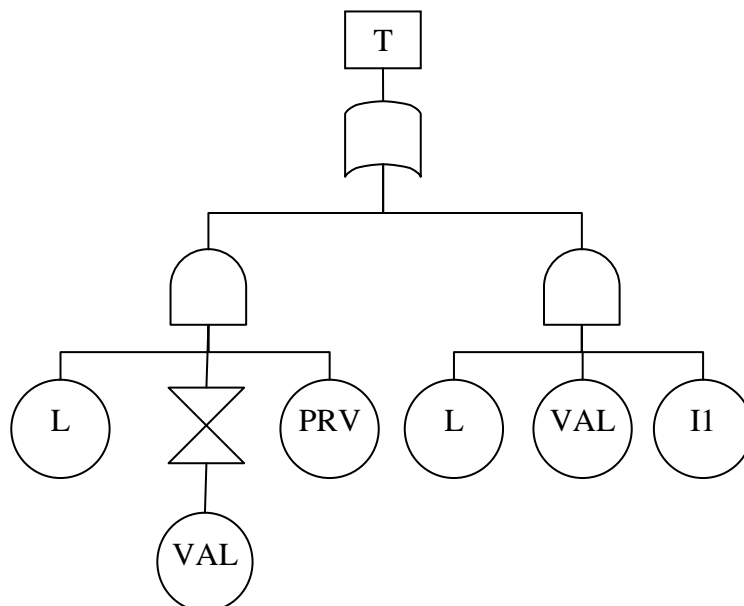
$$\phi(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) \neq \phi(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \quad \forall i. \quad (1.1)$$

Be to ši funkcinė yra nemažėjanti:

$$\phi(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) \geq \phi(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \quad \forall i. \quad (1.2)$$

Sąlyga (1.2) reiškia, kad, jei komponentas „blogės“ ( $x_i = 1$ ), tai sistemos būseną nepakis arba taip pat „blogės“, t.y. sugedus komponentui  $x_i$  sistemos būseną gali nepasikeisti arba sistema gali sugesti. Pagal (1.2) sąlygą, sugedusi sistema negali pereiti į veikimą būseną, jei suges kiti, dar nesugedę, sistemos komponentai. Kai sistema veikia, nors komponentas  $i$  yra sugedęs ir sistema sugenda dėl komponento  $i$  perėjimo į veikimo būseną (komponentas  $i$  sutaisomas ar pakeičiamas veikiančiu), tai sistema dėl komponento  $i$  yra ne-koherentiška (Remenyte, Andrews, 2009).

Ne-koherentiškas gedimų medis pavaizduotas 1.2 paveiksle (Andrews, 2001). Šio gedimų medžio komponentai yra: L - dujų nuotėkis, VAL – izoliavimo vožtuvas, I1 - uždegimo šaltinis, PRV – slėgio mažinimo vožtuvas.



**1.2 pav. Ne-koherentiškas dujų rezervuaro sistemos gedimų medis.**

Šiuo atveju dujos gali sprogti (įvykis baigiamasis įvykis T) kai:

- aptinkamas nutekėjimas ir izoliavimo vožtuvas veikia, bet slėgio mažinimo vožtuvas sugedęs;
- aptinkamas nutekėjimas, izoliavimo vožtuvas neveikia bei yra uždegimo šaltinis.

Kituose skyriuose bus nagrinėjami koherentiniai gedimų medžiai.

## 1.2. MINIMALIŲ KIRTIMŲ AIBĖ

Metodai, taikomi kokybinei ir kiekybinei gedimų medžio analizei, dažniausiai paremti gedimų medžio transformacijomis į įvairias struktūras (minimalių kirtimų aibę, dvejetainę sprendimų diagramą) ir jų analize. Dažniausiai gedimų medžio analizei naudojama minimalių kirtimų aibė (angl. k. minimal cut set - MCS), kai gedimų medis yra koherentiškas. Kirtimų aibė (angl k. cut set) yra bazinių įvykių kombinacijų (aibių) aibė. Gedimų medžio kirtimas yra tokia bazinių įvykių kombinacija, kai visi įvykiai iš šios kombinacijos įvyksta kartu, tai sistema būtinai sugenda. Minimalus kirtimas yra bazinių įvykių kombinacija sudaryta iš tokio kirtimo, kad iš kombinacijos pašalinus bet kurį bazinį įvykį likusi kombinacija nebus gedimų medžio kirtimas (Bedford, Cooke, 2007). Kitaip sakant, pašalinus bet kurį bazinį įvykį iš minimalaus kirtimo sistema nesuges – neįvyks baigiamasis įvykis. Visi skirtingi gedimų medžio minimalūs kirtimai sudaro minimalių kirtimų aibę (MCS).

Gedimų medžio pateikto 1.1 paveiksle vienas iš kirtimų yra  $\{G,S,B\}$ , bet jis nėra minimalus. Minimalus gedimų medžio kirtimas šiuo atveju yra  $\{G,B\}$ , kai jungiklis S veikiantis arba  $\{G,S\}$ , kai baterija B nesugedusi. Matome, kad 1.1 paveikslo gedimų medžio minimalių kirtimų aibė yra:

$$\{G,S\}, \{G,B\}.$$

Pastebime, kad sistema niekada nesuges, jei generatorius G nesuges.

### 1.2.1. GEDIMŲ MEDŽIO IŠRAIŠKA BŪLIO ALGEBROJE

Gedimų medį patogiau aprašyti matematinėmis išraiškomis, nei vaizduoti grafiškai, kad būtų galima jį automatizuotai analizuoti (ieškoti MCS, įvertinti baigiamojo įvykio tikimybę). Dažniausiai šiam tikslui naudojama Būlio algebra bei aibių teorija. Gedimų medžio jungčių, Būlio algebros operatorių ir aibių teorijos atitinkami žymėjimai pateikti 1.2 lentelėje.

1.2 lentelė

#### Gedimų medžio jungčių žymėjimai

| Gedimų medžio jungtis | Būlio algebros operatorius | Aibių teorijos žymėjimas |
|-----------------------|----------------------------|--------------------------|
| $X \text{ IR } Y$     | $X \cdot Y$                | $X \cap Y$               |
| $X \text{ ARBA } Y$   | $X + Y$                    | $X \cup Y$               |
| $NE X$                | $\bar{X}$                  | $X^c$                    |

Dabar galime gedimų medį, pateiktą 1.1 paveiksle, užrašyti Būlio algebroje:

$$T = G \cdot (S + B). \quad (1.3)$$

1.3 lentelė

#### Būlio algebros taisyklės

| Taisyklės pavadinimas | Išraiška   |
|-----------------------|--|
| Komutatyvumo          | $X \cdot Y = Y \cdot X$<br>$X + Y = Y + X$   |
| Asociatyvumo          | $X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$<br>$X + (Y + Z) = (X + Y) + Z$               |
| Distributyvumo        | $X \cdot (Y + Z) = X \cdot Y + X \cdot Z$  |
| Apjungimo             | $X \cdot X = X$<br>$X + X = X$   |
| Absorbcijos           | $X + X \cdot Y = X$  |
| Papildymo             | $X + \bar{X} = \Omega$<br>$\overline{(\bar{X})} = X$                                     |
| De Morgan             | $\overline{X \cdot Y} = \bar{X} + \bar{Y}$<br>$\overline{X + Y} = \bar{X} \cdot \bar{Y}$ |
| Tuščia/Pilna aibė     | $\bar{\emptyset} = \Omega$   |



Pasinaudodami Būlio algebros susistemintai pateiktomis taisyklėmis (1.3 lentelė) galime redukuoti (1.3) lygtį:

$$T = G \cdot (S + B) = G \cdot S + G \cdot B \quad (1.4)$$

ir gauti minimalių kirtimų aibę (MCS), kurios minimalūs kirtimai (1.4) lygtyje atskirti operatoriais "+", o minimalių kirtimų komponentai (baziniai įvykiai) sujungti operatoriais "·":

$$\{G, S\}, \{G, B\}.$$

Ne-koherentišką gedimų medį, pavaizduotą 1.2 paveiksle, taip pat galima aprašyti Būlio algebroje:

$$T = L \cdot \overline{VAL} \cdot PRV + L \cdot VAL \cdot I1,$$

kur pagrindiniai implikantai (prime implicants) yra:

$$\{L, \overline{VAL}, PRV\}, \quad \{L, VAL, I1\}.$$

Pagrindiniai implikantai, tai ne-koherentinio gedimų medžio tokios pačios bazinių įvykių kombinacijos, kaip ir minimalūs, koherentinio gedimų medžio, kirtimai (Sharvia, Papadopoulos, 2008; Andrews, Dunnett, 2000). Tik šiose kombinacijose gali būtų įtraukti baziniai įvykiai, kurie nėra sugedę, t.y. veikiantys baziniai įvykiai, pavyzdžiui  $\overline{VAL}$ .

Panaudojus De Morgan ir kitas Būlio algebros taisykles (1.3 lentelė), galime rasti sistemos veikimo (darbingumo) įvykį  $\overline{T}$ , kuris parodo, kokioms sąlygoms esant sistema tikrai nesuges. Taikant De Morgan taisyklę gedimų medžiui visos jungtys *ARBA* keičiamos į *IR*, taip pat visos jungtys *IR* keičiamos į *ARBA*, be to, baziniai įvykiai pakeičiami jiems priešingais įvykiais. Priešingas baziniam įvykiui įvykis aprašo veikimo būseną ir gali būti išreiškiamas prijungiant bazinį įvykį prie gedimų medžio jungtimi *NE*. Sistemos, aprašytos (1.3) lygtimi, veikimas  $\overline{T}$  išreiškiamas taip:

$$\overline{T} = \overline{G \cdot (S + B)} = \overline{G} + \overline{(S + B)} = \overline{G} + \overline{S} \cdot \overline{B} \quad (1.5)$$

t.y. sistema nesuges ( $\overline{T}$ ) jei nesuges generatorius ( $\overline{G}$ ) arba nesuges jungiklis ir baterija ( $\overline{S} \cdot \overline{B}$ ).

Kombinacijos  $\{\overline{S}, \overline{B}\}$ ,  $\{\overline{G}\}$  dar vadinamos minimaliais keliais ir sudaro minimalių kelių aibę. Minimalių kelių aibė sudaryta iš tokių minimalių kelių, kad įvykus visiems minimalaus kelio įvykiams kartu sistema tikrai veiks. Minimalių kelių aibė yra dualinė (priešinga – angl. k. dual) minimalių kirtimų aibei. Taigi, (1.4) ir (1.5) lygtys yra dualinės viena kitos atžvilgiu.

### 1.2.2. BAIGIAMOJO ĮVYKIO TIKIMYBĖS VERTINIMAS TAIKANT MCS

Ankstesniuose poskyriuose aptarėme kokybinę gedimų medžio analizę. Bet labai svarbu gedimų medį įvertinti kiekybiškai, t.y. įvertinti sistemos gedimo (baigiamojo įvykio T) tikimybę, kai baziniai įvykiai yra nepriklausomi.

Tarkime  $C_1, \dots, C_n$  yra minimalių kirtimų aibės minimalūs kirtimai, o baigiamasis įvykis  $T$  aprašomas taip:

$$T = C_1 \cup C_2 \cup \dots \cup C_n.$$

Tada, remdamiesi aibių sudėties teorema, galime įvertinti baigiamojo įvykio  $T$  tikimybę  $P(T)$  tokia lygtimi:

$$P(T) = \sum_{i=1}^n P(C_i) - \sum_{i<j} P(C_i \cap C_j) + \sum_{i<j<k} P(C_i \cap C_j \cap C_k) - \dots + (-1)^{n+1} P(C_1 \cap C_2 \cap \dots \cap C_n). \quad (1.6)$$

Kada minimalių kirtimų aibėje labai daug minimalių kirtimų, patogiau naudoti sistemos gedimo įvykio  $T$  tikimybės vertinimo aproksimacijas. Didėjant minimalių kirtimų skaičiui (1.6) lygties narių skaičius didėja eksponentiškai, t.y. (1.6) lygties narių skaičius lygus  $(2^m - 1)$ , kai  $m$  – minimalių kirtimų kiekis (NEA\_SEN\_SIN\_SMAP, 2006). Aproksimacijas ypač dažnai naudojamos programiškai realizuojant sistemos gedimo tikimybės vertinimą, bet tada sistemos gedimo tikimybė įvertinama ne tiksliai. Dažniausiai aproksimacija nusako baigiamojo įvykio  $T$  tikimybės eilę (Bedford, Cooke, 2007) ir yra didesnė arba lygi sistemos gedimo tikimybei, jei vertinami visi minimalūs kirtimai, didelių gedimų medžių atveju tai sunkiai įmanoma.

Aproksimacija gaunama atsižvelgiant į tokią nelygybę:

$$\sum_{i=1}^n P(C_i) - \sum_{i<j} P(C_i \cap C_j) \leq P(T) \leq \sum_{i=1}^n P(C_i),$$

tada įvykio  $T$  tikimybės aproksimacija yra:

$$P(T) \approx \sum_{i=1}^n P(C_i) \quad (1.7)$$

Tokia aproksimacija vadinama tiesiogine aproksimacija (Epstein, Rauzy, 2005; Vaidogas, 2003) ir duoda gerus rezultatus, kai kiekvieno (1.6) lygties nario ( $C_i$ ) tikimybė maža ( $\ll 10^{-1}$ ), kai gedimų medį sudaro daug bazinių įvykių tai ši aproksimacija duoda nemažas paklaidas (kelių ar kelių dešimčių procentų).

Galima naudoti ir kitokią aproksimaciją (NEA\_SEN\_SIN\_SMAP, 2006), kuri gaunama panaudojus De Morgan taisyklę tokiu būdu ( $C_i$  yra minimalūs kirtimai):

$$C_i + C_j = \overline{\overline{C_i + C_j}} = \overline{\overline{C_i} + \overline{C_j}}$$

$$P(\overline{C_i}) = 1 - P(C_i)$$

$$P(C_i + C_j) = P(\overline{\overline{C_i + C_j}}) = 1 - (1 - P(C_i)) \cdot (1 - P(C_j)).$$

Iš paskutinės lygties išplaukia aproksimacijos (1.8) formulė, kuri vadinama priešingąja aproksimacija:

$$P(T) = 1 - \prod_{i=1}^n (1 - P(C_i)) \quad (1.8)$$

kur  $P(C_i) = \prod_j P(C_i^j)$ , o  $j$  yra minimalaus kirtimo  $C_i$  elementas. Kai kurie autoriai (Xu, Dugan, 2006)

naudoja (1.8) aproksimacijoje mažiau lygu ženklą „ $\leq$ “.

Sistemos gedimo tikimybės vertinimo (1.6) formulę galime taikyti ir sistemos veikimo tikimybės vertinimui. Tuo atveju  $C_i$  reikštų ne minimalų kirtimą, bet minimalų kelią. Atitinkami (1.7) ir (1.8) aproksimacijos gali būti taikomos sistemos veikimo tikimybės vertinimui. Sistemos veikimo tikimybę galime gauti kitu būdu, t.y. iš vieneto atimti sistemos gedimo tikimybę.

Palyginkime tikslų baigiamojo įvykio  $T$  tikimybės vertinimą išreikštą (1.6) formulę ir sistemos gedimo tikimybės vertinimą, naudojant (1.7) ir (1.8) aproksimacijas. Palyginimui naudojamas gedimų medis apibrėžtas (1.4) lygtimi. Palyginimą atliksime a) atveju, kai bazinių įvykių  $G, S, B$  tikimybės yra 0,01 ir b) atveju, kai  $G, S, B$  tikimybės yra 0,5.

a) tiksliai įvertinama tikimybė:

$$P(T) = P(G \cap S) + P(G \cap B) - P(G \cap S \cap B) = 0,00019900,$$

atitinkamai (1.7) ir (1.8) aproksimacijos:

$$P(T) = P(G \cap S) + P(G \cap B) = 0,00020000,$$

$$P(T) = 1 - ((1 - P(G \cap S)) \cdot (1 - P(G \cap B))) = 0,00019999.$$

b) tiksliai vertinama tikimybė:

$$P(T) = P(G \cap S) + P(G \cap B) - P(G \cap S \cap B) = 0,3750,$$

atitinkamai (1.7) ir (1.8) aproksimacijos:

$$P(T) = P(G \cap S) + P(G \cap B) = 0,5000,$$

$$P(T) = 1 - ((1 - P(G \cap S)) \cdot (1 - P(G \cap B))) = 0,4375.$$

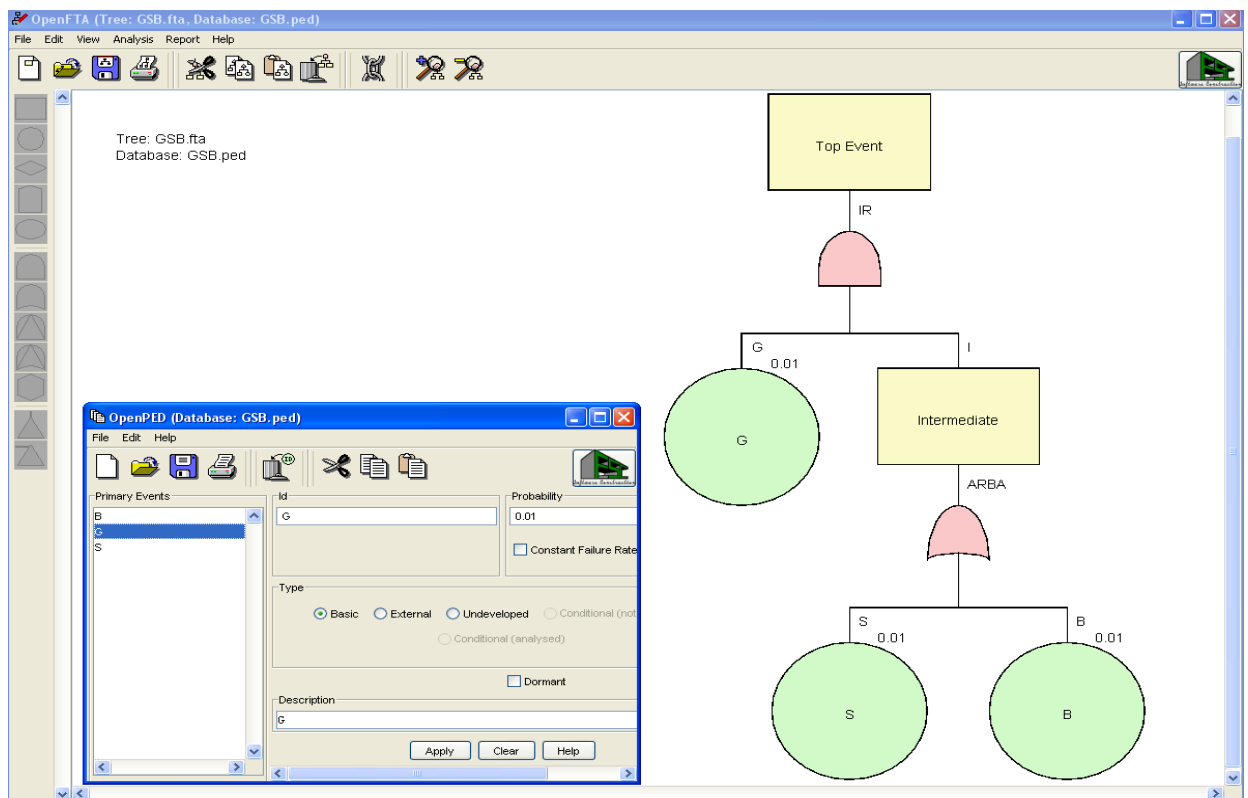
Matome, kad, kai bazinių įvykių tikimybės mažos (0,01), tai (1.7) ir (1.8) lygtys aproksimuoja tiksliai vertinamą sistemos gedimo tikimybę ((1.6) lygtis) pakankamai gerai, tokiu atveju patogiau naudoti (1.7) lygtį, nes ją lengviau realizuoti ir atliekama mažiau veiksmų. Kai bazinių įvykių tikimybės pakankamai didelės (0,5), geriau aproksimuoti (1.8) lygtimi, nes įvertis artimesnis tiksliai įvertintai tikimybei.

### 1.3. PROGRAMA „OPENFTA“

Pasaulyje sukurta daug programinių priemonių, kurių pagalba galime įvertinti gedimų medžius, daugumoje iš jų yra realizuoti algoritmai paremti minimalių kirtimų aibės vertinimu. Tokios sistemos yra „Relex“, „RAM Commander“, „RiskSpectrum FTA“, „ITEM ToolKit“, „OpenFTA“, „ASTRA 3“ (Contini, Matuzas, 2009).

Dauguma iš jų, skaičiuojant sistemos gedimo tikimybę, atmeta nereikšmingus minimalių kirtimų aibės narius. Vienos atmeta MCS kirtimus, kurių tikimybė, palyginti su baigiamojo įvykio tikimybe, labai maža. Kitos atmeta bazinius įvykius, kurių tikimybės palyginti su kitais baziniais įvykiais yra mažos. Bet toks atmetimas gali iššaukti reikšmingą tikimybės netikslumą, ypač kai atmetamų komponentų (minimalių kirtimų, bazinių įvykių) yra labai daug. Atmestų komponentų mažos tikimybės gali sumuotis ir duoti sistemos gedimo tikimybės netikslumą. Kai kurios programos atmetinėja (1.6) lygties lygius (sumas), kurie palyginti yra nereikšmingi. Viena iš tokių programų yra „OpenFTA“ (OpenFTA, 2005).

Programos langų pavyzdys pateiktas 1.3 paveiksle. Taip pat 1.3 paveiksle pavaizduota gedimų medžio, aprašyto (1.3) išraiška, realizacija.



1.3 pav. „OpenFTA“ programos langai

„OpenFTA“ programa skirta vertinti gedimų medį ir skaičiuoti baigiamojo įvykio tikimybę. Internete galima rasti nekomercinę programos versiją: „<http://www.openfta.com/>“. Programa vertina sistemos gedimų medį, pateikdama minimalių kirtimų aibę. Skaičiuoja sistemos gedimo tikimybę. Konstruojant gedimų medį galima pasirinkti įvykio tipą (bazinis, tarpinis), jungtis (*IR*, *ARBA*), kurios sujungia įvykius tarpusavyje, priskirti baziniams įvykiams tikimybės (naudojant kitą programos langą). Taip pat koreguoti sistemos gedimų medį, išsaugoti ar atspausdinti norimą gedimų medžio dalį.

Baigiamojo įvykio T tikimybę programa „OpenFTA“ vertina remiantis (1.6) lygtimi. Šia programa įvertinus 1.2.2. skyrelyje suskaičiuotas a) ir b) atvejų tikimybes, gaunamas atitinkamai toks pat rezultatas:  $1,990000 \cdot 10^{-4}$  ir  $3,750000 \cdot 10^{-4}$ .

„OpenFTA“ visada skaičiuoja pirmuosius du (1.6) lygties lygius (angl. k. terms), jei reikia, skaičiuoja N lygių, atsižvelgiant į (1.9) nelygybę (OpenFTA, 2005):

$$P_1 \equiv \sum_{i=1}^n P(C_i)$$

$$P_2 \equiv P_1 - \sum_{i=2}^n \sum_{j=1}^{i-1} P(C_i \cap C_j)$$

$$P_3 \equiv P_2 + \sum_{i=3}^n \sum_{j=2}^{i-1} \sum_{k=1}^{j-1} P(C_i \cap C_j \cap C_k)$$

...

$$\text{su kiek norima mažu } x > 0; \exists N : |P_I - P_N| < x, \quad (1.9)$$

kur  $N < I \leq n$ ,  $I = \overline{1, n}$ .

Atlikta keletas algoritmo bandymų, kai minimalių kirtimų aibė susideda iš 10 kombinacijų ( $i = \overline{1, n}$ ), atsižvelgiant į (1.9) nelygybę. Gauti rezultatai pateikti 1.4 lentelėje.

1.4 lentelė

„OpenFTA“ naudojamo algoritmo rezultatai

|                           | $C_i=0,5$                | $C_i=0,01$               | $C_i=0,01$<br>$C_2=0,9$  | $C_{1,3,5,7,9}=0,01$<br>$C_{2,4,6,8,10}=0,9$ | $C_{1,3,5,7,9}=0,001$<br>$C_{2,4,6,8,10}=0,9$ | $C_{1,3,5,7,9}=0,1$<br>$C_{2,4,6,8,10}=0,9$ |
|---------------------------|--------------------------|--------------------------|--------------------------|--|---|---|
| <b>x</b>                  | $10^{-1}$                | $10^{-6}$                | $10^{-6}$                | $10^{-6}$                                    | $10^{-6}$                                     | $10^{-2}$                                   |
| <b>Nr</b>                 | 9                        | 5                        | 6                        | 9  | 8   | 8   |
| <b>P<sub>nr</sub></b>     | $1.953125 \cdot 10^{-2}$ | $2.520000 \cdot 10^{-8}$ | $1.142400 \cdot 10^{-8}$ | $2.985255 \cdot 10^{-8}$                     | $5.921309 \cdot 10^{-9}$                      | $7.618050 \cdot 10^{-3}$                    |
| <b>P</b>                  | $9.999999 \cdot 10^{-1}$ | $9.561792 \cdot 10^{-2}$ | $9.086482 \cdot 10^{-1}$ | $9.999904 \cdot 10^{-1}$                     | $9.999900 \cdot 10^{-1}$                      | $9.996719 \cdot 10^{-1}$                    |
| <b>P<sub>tiksli</sub></b> | $9.990234 \cdot 10^{-1}$ | $9.561792 \cdot 10^{-2}$ | $9.086482 \cdot 10^{-1}$ | $9.999904 \cdot 10^{-1}$                     | $9.999900 \cdot 10^{-1}$                      | $9.999940 \cdot 10^{-1}$                    |

kur: x – kriterijus, koku tikslumu po kabelio skaičiuojama sistemos gedimo T tikimybė;

nr – paskutinis (1.6) lygties lygis, kuris įtraukiamas į gedimų medžio tikimybės vertinimą;

P<sub>nr</sub> – nr-tojo lygio tikimybė;

P – gedimų medžio tikimybė įvertinus nr lygių;

P<sub>tiksli</sub> – tiksli gedimų medžio tikimybė, įvertinus visus lygius.

Iš 1.4 lentelės matome, kad „OpenFTA“ naudojamas algoritmas skaičiuoja pakankamai tiksliai, priklausomai nuo pasirinkamo kriterijaus x dydžio. Paskutiniai skaitmenys mantisėje nėra labai reikšmingi, nes jų reikšmės mažai ką lemia lyginant mažas tikimybes.

## 1.4. DVEJETAINĖ SPRENDIMŲ DIAGRAMA

Realybėje dažnai norima tiksliai įvertinti didelės sistemos gedimo tikimybę. Įvairios programos, naudojančios minimalių kirtimų aibės vertinimą, didelio gedimų medžio tikimybei įvertinti dažniausiai taiko (1.7) ar (1.8) aproksimacija ir gauna apytikslį tikimybės įvertį. Kai naudojama (1.6) lygtis, sistemos gedimo tikimybė įvertinama tiksliai, bet ją įvertinti reikia daug laiko ir kompiuterinės atminties. Siekiant greitai ir tiksliai įvertinti sistemos gedimo tikimybę naudojamas kitas metodas, pagrįstas dvejetainės sprendimų diagramos (ang .k. binary decision diagram) taikymu. Dvejetainės sprendimų diagramos (BDD) struktūra taip pat taikoma vertinant ne-koherentiškų gedimų medžių vertinimui.

Dvejetainė sprendimų diagrama yra grafas, gautas atlikus gedimų medžio transformaciją. BDD yra sistemos, aprašytos gedimų medžiu, teisingumo lentelės grafinė išraiška, kuri aprašo sistemos gedimo ir veikimo būsenas.

Naudojant BDD lengva vertinti sistemos gedimo tikimybę, kuri visada suskaičiuojama tiksliai, priešingai nei naudojant programines priemones, naudojančias MCS metodą. Transformavus gedimų medį į BDD, gaunamas didesni sistemos gedimo vertinimo efektyvumas (neapkraunama kompiuterinė atmintis ir tiksliai skaičiuojama sistemos gedimo tikimybė) (Bedford, Cooke, 2007).

Sudarant BDD struktūrą būtina apibrėžti gedimų medžio įvykių tvarką - juos suranguoti, apie tai plačiau parašyta 1.4.1. skyrelyje.

### 1.4.1. TEISINGUMO LENTELĖS TAIKYMAS

Dvejetainė sprendimų diagrama dažnai pristatoma kaip Būlio funkcijos teisingumo lentelės kompaktiška grafinė išraiška. Kadangi gedimų medis gali būti aprašomas Būlio funkcija, tai jį galime vaizduoti naudodami dvejetainę sprendimų diagramą (Bryant, 1992, Meinel, Theobald, 1998).

Sudarykime teisingumo lentelę (1.5 lentelė) gedimų medžiui pavaizduotam 1.1 paveiksle. Teisingumo lentelėje reikšmė 1 rodo, kad įvykis įvyks, t.y. sistemos komponentas suges, o 0 – įvykis neįvyks, komponentas nesuges.

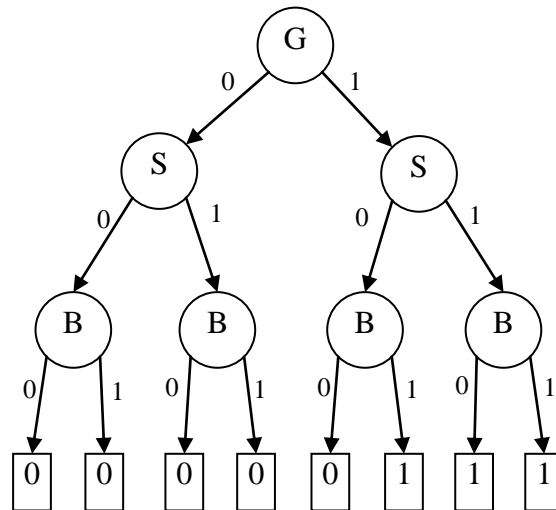
1.5 lentelė

Elektros tiekimo sistemos gedimo teisingumo lentelė

| G | S | B | T |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |

|   |   |   |   |
|---|---|---|---|
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Teisingumo lentelę galima atvaizduoti grafiškai, toks grafas (1.4 paveikslas) ir yra dvejetainė sprendimų diagrama (BDD), bet ji, kaip ir teisingumo lentelė, reikalauja daug „erdvės“, tiek vaizdavimo prasme, tiek ir realizavus programiškai - kompiuterinės atminties.



**1.4 pav. Energijos tiekimo sistemos gedimo BDD**

Dvejetainę sprendimų diagramą sudaro mazgai (baziniai įvykiai) ir šakos. Viršutinis mazgas vadinamas BDD viršūne (šaknine viršūne – angl. k. root vertex), o apatiniai mazgai, žymimi 0 arba 1, rodo sistemos būsenas (veiks arba suges), priklausomai nuo juos pasiekiančių mazgų sekos. Skaičiai prie BDD mazgo šakų rodo įvykio būseną: jei įvykis įvyko žymima 1, neįvyko – 0. BDD pavaizduota 1.4 paveiksle yra dešininė, nes 1-ai prie šakų žymimi dešinėje mazgo pusėje. Kai kurie kiti autoriai naudoja kairines BDD, kai 1-ai žymimi kairėje mazgo pusėje (Andrews, Remenyte, 2005). Šakų galuose esantys mazgai vadinami 1 (vienetine) atšaka arba 0 (nuline) atšaka (1-child, 0-child).

Kad 1.4 paveiksle pavaizduota BDD užimtų mažiau „erdvės“, ją reikia redukuoti. BDD redukcijos taisyklės pateikė Bryant (1986):

- (I) Pasikartojančių galinių mazgų šalinimas: pašalinami visi galiniai mazgai, paliekant tik po vieną nesikartojantį (0 ir 1) ir į juos atitinkamai nukreipiamos atsilaisvinusios šakos, kurios buvo nukreiptos į 0 vėl nukreipiamos į 0, kurios į 1 taip pat nukreipiamos į 1.
- (II) Pasikartojančių ne-galinių mazgų šalinimas: jei du skirtingi mazgai turi tą pačią „etiketę“, jų 0-nės atšakos identiškos ir 1-nės atšakos identiškos, tada bet kuri vieną iš dviejų mazgų pašaliname, ir į pašalintąjį mazgą buvusi nukreipta šaka nukreipiama į

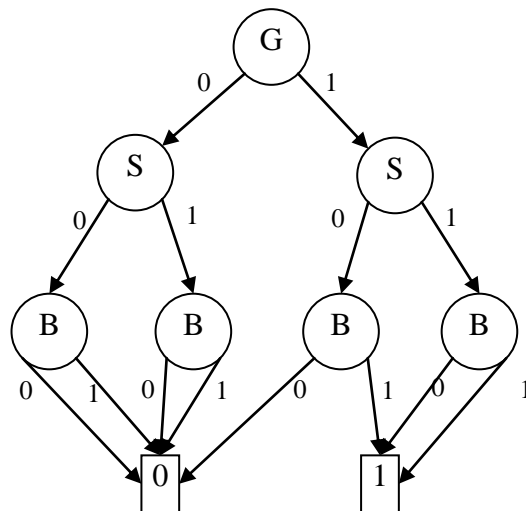
likusį mazgą. Ši taisyklė taikoma pirmiausia aukštesnio rango (angl. k. in order) mazgams, t.y. iš BDD apačios keliamą link viršūnės.

- (III) Perteklinių mazgų šalinimas: jei ne-galinio mazgo 0-nė atšaka ir 1-nė atšaka yra tas pats mazgas, tada ne-galinis mazgas pašalinamas, o į jį nukreipta šaka sujungiama su pastarojo atšaka.

Taisyklėje (II) minimas aukštesnio rango mazgas reiškia, kad jam priskirtas tvarkos numeris. Sudarant dvejetainę sprendimų diagramą gedimų medžio įvykiams pirmiausia priskiriama tvarka pagal kurią, konvertuojant gedimų medį į BDD, būtų vienareikšmiškai apibrėžiami bazinių įvykių rangai. Tvarkai nustatyti naudojamas „<“ ženklas. Žemiausio rango įvykis braižomas BDD viršūnėje.

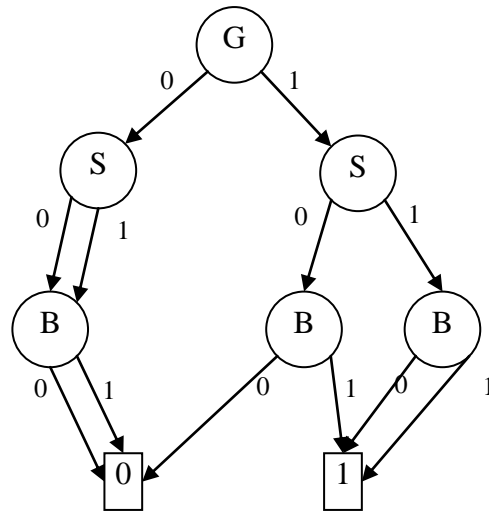
Nuo rangavimo priklauso BDD dydis – užimama „erdvė“ ir forma. Bryant (1986, 1992) parodė, kad esant  $2n$  gedimų medžio kintamųjų, priklausomai nuo rangavimo, BDD mazgų kiekis gali kisti nuo  $2n+2$  iki  $2^{n+1}$ . Geriausios tvarkos gedimų medžio įvykiams radimas yra atskiras uždavinys, kuris sprendžiamas NP-klasėje (Epstein, Rauzy, 2005; Ibanez-Llano, 2009), McMillan (1993) siūlo, kad ranguojant bazinius įvykius būtų atsižvelgiama į minimalių kirtimų aibę, tiksliau baziniai įvykiai suranguojami iš eilės pagal jų išsidėstymą minimalių kirtimų aibėje.

Pavaizduotos 1.4 paveiksle BDD bazinių įvykių tvarka yra  $G < S < B$ . Redukuokime 1.4 paveikslo BDD, panaudokime (I), (II) ir (III) taisykles, atitinkamos BDD pateiktos 1.5, 1.6 ir 1.7 paveiksluose.

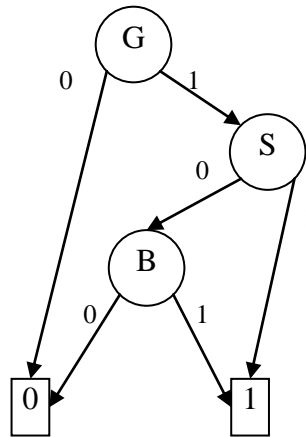


1.5 pav. BDD panaudojus (I) taisyklę





1.6 pav. BDD panaudojus (II) taisyklę



1.7 pav. BDD panaudojus (III) taisyklę

Paveiksle 1.7 pavaizduota kompaktiška BDD, kurios labiau sumažinti neįmanoma. Iš BDD struktūros sudėtinga suprasti pavaizduotos sistemos veikimą. Kai sistema yra koherentinė, tuo pačiu ir transformuojamas gedimų medis yra koherentiškas. Be to, dvejetainėje sprendimų diagramoje per vienietines šakas sujungti mazgai (baziniai įvykiai) sudaro minimalių kirtimų aibės minimalius kirtimus. Šioje BDD minimalių kirtimų aibė yra  $\{G,S\}$  ir  $\{G,0,B\}$ , t.y.  $\{G,S\}$  ir  $\{G,B\}$ . Ne-koherentiško gedimų medžio BDD struktūroje per vienietines šakas yra sujungiami mazgai sudarantys pagrindinius implikantus.

### 1.4.2. ITE SĄLYGOS TAIKYMAS

Gedimų medį transformuoti į dvejetainę sprendimų diagramą galima naudojant kitus metodus, pavyzdžiui ITE (Jeigu ... Tai ... Priešingai ... - If ... Then ... Else ...) sąlygą. Pirmiausia apibrėžkime Šenono plėtinį Būlio algebros funkcijoms (NEA\_SEN\_SIN\_SMAP, 2006; Andrews, Remenyte, 2005).

Tegul  $f(x_1, x_2, \dots, x_n)$  yra Būlio funkcija. Funkcijos  $f$  Šenono plėtinys (1.10) kintamojo  $x_i$  atžvilgiu yra

$$f(x_1, x_2, \dots, x_n) = x_i \cdot f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n) + \bar{x}_i \cdot f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \quad (1.10)$$

Tada BDD gali būti interpretuojama kaip Šenono plėtinio (Rauzy, Dutuit, 1997; Rauzy, 1996) kompaktiška išraiška, kur nenaudingi plėtiniai pašalinami ir pati funkcija išreiškiama tik vienu būdu. BDD konstravimui remiantis Šenono (1.10) plėtinio naudojama ITE (1.11) sąlyga:

$$f(x) = X \cdot f_1 + \bar{X} \cdot f_0. \quad (1.11)$$

čia  $X$  yra bazinis įvykis; funkcijos  $f_1$  ir  $f_2$  yra sumažintos Būlio funkcijos  $f$  dalys, atitinkamai, kai  $X = 1$  ir  $X = 0$ . ITE sąlyga dažniausiai aprašoma tokia išraiška  $\langle X, f_1, f_0 \rangle$ , tai reiškia: Jei  $X$  suges, tai nagrinėsime  $f_1$ , jei nesuges, nagrinėsime  $f_0$ .

Remiantis ITE apibrėžimu, dvejetainėje sprendimų diagramos struktūroje  $f_1$  prijungiama prie vienietinės mazgo  $X$  šakos, o  $f_0$  – prie nulinės mazgo  $X$  šakos. Gedimų medžio įvykiai turi būti taip pat suranguoti (NEA/SEN/SIN/SMAP, 2005). Apibrėžiant bet kokį gedimų medžio įvykį  $A$  sąlygoje ITE galime užrašyti  $A = \langle A, 1, 0 \rangle$ , t.y. Jei  $A$  (įvyks), tai 1 (suges), priešingai 0 (nesuges). Įvykiui  $A$  priešingo įvykio  $\bar{A}$  ITE sąlyga yra  $\bar{A} = \langle A, 0, 1 \rangle$ .

Naudojant ITE sąlygą, gedimų medžio transformacijai į BDD struktūrą, atsižvelgiama į dvi taisykles. Tegul  $F$  ir  $G$  yra BDD struktūros mazgai (baziniai įvykiai gedimų medyje) tokie, kad  $F = \langle X, f_1, f_0 \rangle$  ir  $G = \langle Y, g_1, g_0 \rangle$ . Tada

(I) jei  $X < Y$  (čia tikrinami mazgų  $X$  ir  $Y$  rangai), tai

$$F \text{ [op] } G = \langle X, f_1 \text{ [op] } G, f_0 \text{ [op] } G \rangle,$$

(II) jei  $X = Y$ , tai

$$F \text{ [op] } G = \langle X, f_1 \text{ [op] } g_1, f_0 \text{ [op] } g_0 \rangle,$$

kur [op] yra Būlio algebros operatorius atitinkantis gedimų medžio jungtį.

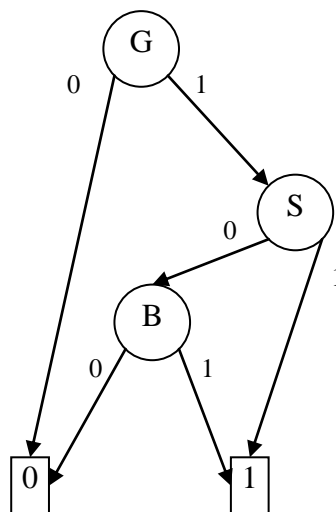
Pasinaudodami šia ITE sąlyga sudarykime gedimų medžio, pateikto 1.1 paveiksle, dvejetainę sprendimų diagramą. Pirmiausia reikia bazinius įvykius suranguoti. Panaudokime tokį patį rangavimą, kaip ir 1.4.1. poskyryje,  $G < S < B$ . Tada sistemos gedimo įvykį  $T$  galime išreikšti per (1.12) ITE sąlygą, transformuojant gedimų medį į dvejetainę sprendimų diagramą tokiu būdu:

$$\begin{aligned} T &= G \cdot (S + B) = \langle G, 1, 0 \rangle \cdot \langle \langle S, 1, 0 \rangle + \langle B, 1, 0 \rangle \rangle \\ &= \langle G, 1, 0 \rangle \cdot \langle S, 1 + \langle B, 1, 0 \rangle, 0 + \langle B, 1, 0 \rangle \rangle \\ &= \langle G, 1, 0 \rangle \cdot \langle S, 1, \langle B, 1, 0 \rangle \rangle \\ &= \langle G, 1 \cdot \langle S, 1, \langle B, 1, 0 \rangle \rangle, 0 \cdot \langle S, 1, \langle B, 1, 0 \rangle \rangle \rangle \\ &= \langle G, \langle S, 1, \langle B, 1, 0 \rangle \rangle, 0 \rangle \end{aligned} \quad (1.12)$$

Pagal gautą (1.12) ITE sąlygą nubraižome dvejetainėje sprendimų diagramoje (1.8 paveiksle). Pastebime, kad BDD 1.7 ir 1.8 paveiksluose sutampa, taip ir turi būti, nes transformavome tą patį gedimų medį skirtingais būdais, bet naudojome tą patį rangavimą. Be to, naudojant ITE sąlygą iš karto gavome dalinai redukuotą kompaktišką BDD formą ir nereikėjo naudoti teisingumo lentelės, kuri realizuojant programiškai užimtų labai daug kompiuterinės atminties.

Tokia forma yra trumpesnė ir patogesnė, ypač aprašant didelės apimties BDD (NEA/SEN/SIN/SMAP, 2006). Tokiu būdu gauta (1.12) struktūra atrodo taip:

$$T = \langle G, \langle S, 1, \langle B, 1, 0 \rangle \rangle, 0 \rangle \quad (1.13)$$



1.8 pav. BDD taikant ITE sąlygą

### 1.4.3. KOMPONENTŲ JUNGIMO METODAS

Way ir Hsia (2000) aprašė dar vieną gedimų medžio transformacijos į BDD būdą, pavadintą komponentų jungimo metodu. Šis metodas taikomas koherentiškiems gedimų medžiams, kuriuose naudojamos *IR* bei *ARBA* jungtys. Konstruojant BDD neatsižvelgiama į galimus bazinių įvykių pasikartojimus. Be to, šiam metodui nereikia iš anksto nustatyti bazinių įvykių tvarkos, priešingai nei ankstesniems dviems metodams, kur įvykius turėjome ranguoti.

Transformuojant bet kurios jungties įvykius keliaujama per gedimų medžio įvykius iš kairės į dešinę, toks jungimas tarsi suteikia baziniams įvykiams rangą, t.y. bazinių įvykių aibei priskiria tam tikrą tvarką. Be to, BDD pografiui kairėje pusėje pateikiami su daugiau jungčių ir mazgų. Kai visoms jungtims sudaromi BDD pografiai, tada galime juos jungti į pačią BDD. Toks iš apačios kylantis procesas yra baigtas, kai, atsižvelgiant į jungtį prie baigiamojo įvykio, sujungiami visi pografiai į BDD (Sakanashi, 1997). Gedimų medžio transformacijai naudojamos tokios taisyklės:

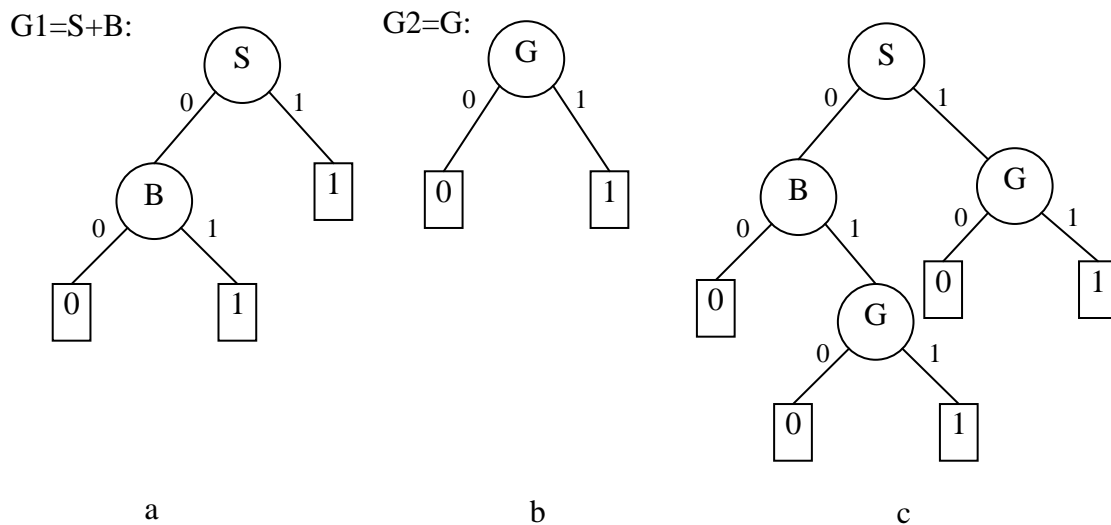
- (I) Jei du gedimų medžio įvykiai yra sujungti *IR* jungtimi, tai dvejetainėje sprendimų diagramoje atitinkami mazgai sujungiami vienetine šaka. Jei įvykiai sujungti *ARBA* jungtimi, tai BDD struktūroje jie jungiami nuline šaka.
- (II) Jei jungiami du BDD pografiai, tai vieną iš jų nustatome kaip pagrindinį BDD pografį ir taikome tokias taisykles:
  - a) Jei du BDD pografiai jungiami *IR* jungtimi, tai šalutinį pografį prijungiame vietoj kiekvieno galinio 1 mazgo prie pagrindinio BDD pografio;

- b) Jei du BDD pografiai sujungti *ARBA* jungtimi, tai šalutinį prijungiame vietoj kiekvieno galinio 0 mazgo pagrindiniame BDD pografyje.

Atlikus transformaciją tikrinama, ar nėra pasikartojančių mazgų. Jei randamas pasikartojantis mazgas, atliekamas BDD struktūros suprastinimas pagal taisykles:

- (I) Kiekvienas kelias prasideda nuo mazgo, atitinkančio pirmą kelyje pasikartojantį įvykį, ir tęsiasi iki galinio mazgo. Reikia atsižvelgti į priskirtą tvarką (iš kairės į dešinę), kad būtų išvengta pasikartojančio įvykio būsenos prieštaravimų dvejetainėje sprendimų diagramoje. Mazgą neatitinkantį minėtos tvarkos reikia pakeisti įvykiais esančiais žemiau, prie bet kurios 1 ar 0 šakos. Pavyzdžiui, jei keliauti per BDD pradedame nuo mazgo 1 šakos, tai antras mazgo pasirodymas turi būti pakeistas tolimesne BDD struktūra, prijungiant po pasikartojančio mazgo panaikinimo likusį pografį prie 1 šakos.
- (II) Jei sistemos būsena yra tokia pati nepaisant mazgo įvykio ar neįvykio, tai nereikšmingas mazgas turi būti pašalintas. Kitaip sakant, BDD struktūra, esanti toliau už minėto mazgo, turi būti prijungiama į pašalinto mazgo vietą.

Parodysime metodo veikimą (1.9 paveikslas) transformuojant 1.1 paveiksle pavaizduotą gedimų medį.



**1.9 pav. Komponentų jungimo metodu gauta BDD**

Pastebime, kad BDD forma skiriasi nuo prieš tai gautų BDD struktūrų tam pačiam gedimų medžiui. Taip yra todėl, kad komponentų jungimo metode neapibrėžiami įvykių rangai, įvykiai automatiškai pagal taisykles „gauna tvarką“ (NASA, 2002). Gautos galutinės BDD struktūros (1.9 c paveikslas) tvarka yra  $S < B < G$ . Patikrinkime algoritmą su ITE sąlyga tam pačiam gedimų medžiui, kai baziniai įvykiai suranguoti taip:  $S < B < G$ . Gauname (1.14) išraišką, kuri pilnai atitinka 1.9 c paveiksle pateiktą BDD.

$$T = G \cdot (S + B) = \langle G, 1, 0 \rangle \cdot \langle S, 1, \langle B, 1, 0 \rangle \rangle = \langle S, \langle G, 1, 0 \rangle, \langle B, \langle G, 1, 0 \rangle, 0 \rangle \rangle \quad (1.14)$$

Andrews ir Remenyte (2005) ištyrė, kad nors komponentų jungimo metodas nereikalauja bazinių įvykių rangavimo, bet turi daugiau trūkumų, nei privalumų. Komponentų jungimo metodu sudarytos BDD struktūros užima daugiau „erdvės“, reikalauja daugiau kompiuterinės atminties ir lieka pasikartojančių, nesutrauktų BDD pografių. Todėl minėti autoriai rekomenduoja, transformuojant gedimų medžius į BDD struktūrą, vietoj komponentų jungimo metodo taikyti ITE sąlygą.

#### 1.4.4. GEDIMO TIKIMYBĖS VERTINIMAS BDD STRUKTŪROJE

Transformavus gedimų medį į dvejetainę sprendimų diagramą labai svarbu įvertinti sistemos gedimo tikimybę (Dutuit, Rauzy, 2001). Kadangi gedimų medis transformuojamas pasinaudojant Šenono transformacijos (1.11) lygtimi, tai remiantis ja galime įvertinti ir sistemos, išreikštos BDD struktūra, gedimo tikimybę (1.15 lygtis):

$$P(f = 1) = P(X = 1) \cdot P(f_1 = 1) + P(X = 0) \cdot P(f_0 = 1) \quad (1.15)$$

Remiantis (1.15) formule sistemos gedimo tikimybę įvertinama tiksliai ir greitai, nereikia ieškoti ir atmetinėti nereikšmingų reikšmių, kaip kad skaičiuoja „OpenFTA“ ir kitos programos, kur vertinimas atliekamas taikant minimalių kirtimų aibės.

Naudodami (1.15) išraišką suskaičiuokime 1.8 paveiksle pavaizduotos BDD grafu sistemos gedimo tikimybę kai mazgų tikimybės, kaip ir 1.2.2. skyrelyje, a) atveju 0,01, b) atveju 0,5. Sistemos gedimo tikimybę galima skaičiuoti dviem būdais: tiesiogiai taikant (1.15) lygtį ir iteratyviai skaičiuojant nuo BDD struktūros apačios link viršūnės.

Sistemos gedimo tikimybės **tiesioginis vertinimas** atliekas taikant sąlyginę tikimybės vertinimo formulę (1.15) lygčiai (Bedford, Cooke, 2007):

$$P(T) = P(T | G = 1) \cdot P(G = 1) + P(T | G = 0) \cdot P(G = 0) = P(T | G = 1) \cdot P(G = 1) \quad (1.16)$$

Kadangi dėmuo  $P(T | G = 0) \cdot P(G = 0)$  lygus 0, t.y. sistema nesuges keliaujant per nulinę šaką iš mazgo G, todėl tokį dėmenį atmetame. Toliau pratęsiant (1.16) lygtį gauname pilną BDD struktūros tikimybės skleidinį (1.17), pagal kurį apskaičiuojame sistemos gedimo tikimybę.

$$\begin{aligned} P(T) &= P(T | G = 1) \cdot P(G = 1) \\ &= P\{T | (G, S) = (1, 1)\} \cdot P\{(G, S) = (1, 1)\} \\ &\quad + P\{T | (G, S) = (1, 0)\} \cdot P\{(G, S) = (1, 0)\} \\ &= P\{T | (G, S) = (1, 1)\} \cdot P\{(G, S) = (1, 1)\} \\ &\quad + P\{T | (G, S, B) = (1, 0, 1)\} \cdot P\{(G, S, B) = (1, 0, 1)\} \\ &= P\{(G, S) = (1, 1)\} + P\{(G, S, B) = (1, 0, 1)\} \end{aligned} \quad (1.17)$$

Sistemos gedimo tikimybės a) ir b) atvejais atitinkamai gaunamos:

$$a) P(T) = P\{(G, S) = (1, 1)\} + P\{(G, S, B) = (1, 0, 1)\} = 0,01^2 + 0,01^2 \cdot (1 - 0,99)^1 = 0,000199$$

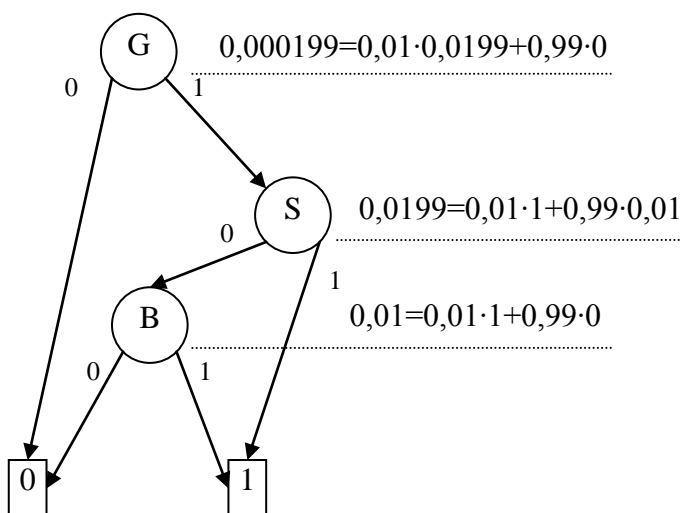
$$b) P(T) = P\{(G, S) = (1, 1)\} + P\{(G, S, B) = (1, 0, 1)\} = 0,5^2 + 0,5^2 \cdot (1 - 0,5)^1 = 0,375$$

Matome, kad sistemos gedimo tikimybės abiem atvejais sutampa su tikimybėmis suskaičiuotomis anksčiau (1.2.2. ir 1.3. skyriai).

Sistemos gedimo tikimybės **iteratyvus vertinimas** atliekamas skaičiuojant dvejetainėje sprendimų diagramoje tikimybes prie mazgų nuo struktūros apačios link viršutinio mazgo. Tikimybė prie viršutinio BDD mazgo yra sistemos gedimo tikimybė. Tikimybė prie mazgo skaičiuojama atsižvelgiant į 1-nės ir 0-nės atšakų tikimybes (Bedford, Cooke, 2007). BDD pavaizduotos 1.8 paveiksle mazgo B tikimybė išreiškiama (1.18) lygtimi ir gaunama tikimybė pavaizduojama dvejetainėje sprendimų diagramoje prie B mazgo.

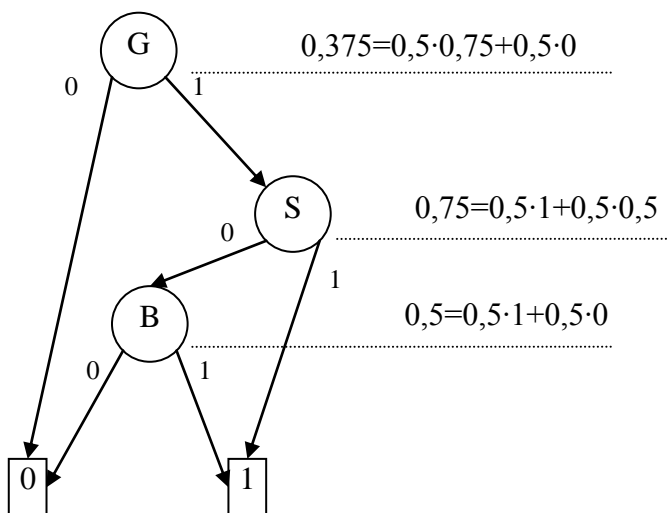
$$P_{mazgo}(B) = P(B) \cdot P(1) + (1 - P(B)) \cdot P(0) = P(B) \quad (1.18)$$

a) atvejis:



1.10 pav. BDD struktūros tikimybinis vertinimas a) atvejui

b) atvejis



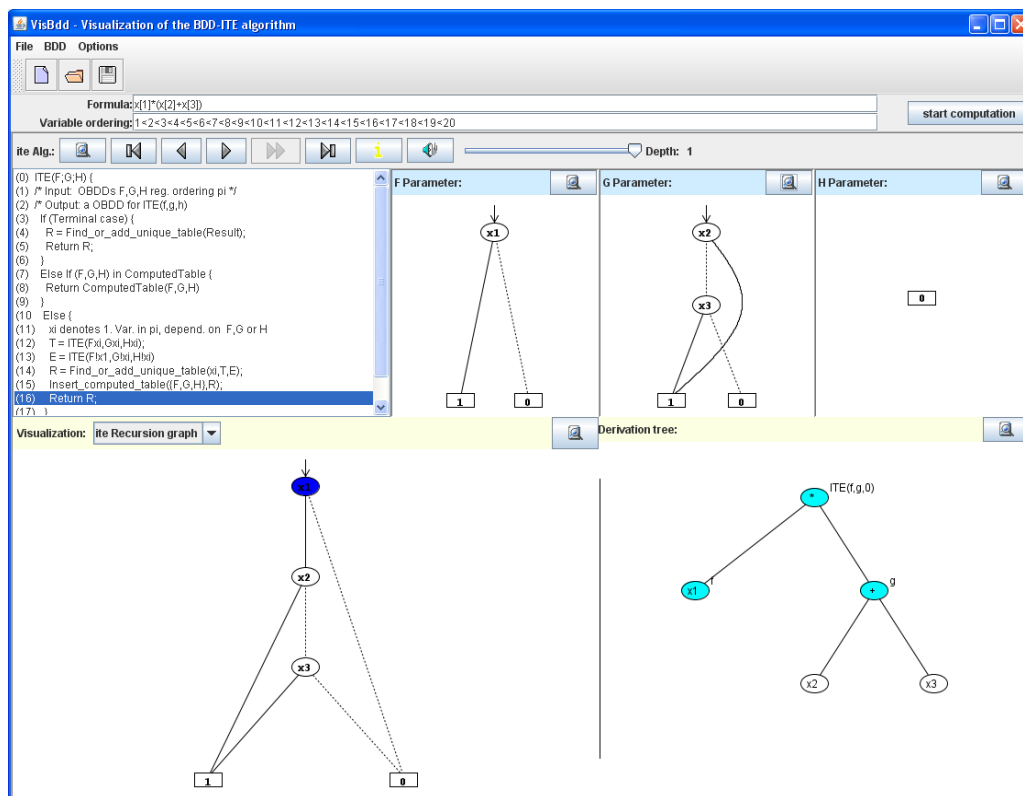
1.11 pav. BDD struktūros tikimybinis vertinimas b) atvejui

Pastebime, kad abiem atvejais (1.10 ir 1.11 paveikslai) gautos tikimybės sutampa su prieš tai šiame poskyryje (tiesioginis vertinimas) apskaičiuotomis atvejų a) ir b) tikimybėmis. Be to, iteratyvus

skaiciavimas lengviau atliekamas vertinant sistemos gedimą analitiniu būdu bei patogiau realizuojamas programiškai.

## 1.5. PROGRAMA „VISBDD“

Programa „VisBDD“ naudojama, norit pavaizduoti, kaip veikia gedimų medžio konvertavimo į dvejetainę sprendimų diagramą ITE sąlyga. Programą galima rasti internete ir nemokamai parsisiuntus naudoti (<http://bdd.hpi.uni-potsdam.de/visBDD/index.html>). Programos langas su 1.1 paveikslo gedimų medžiu pavaizduotas 1.12 paveiksle.



1.12 pav. „VisBDD“ programos langas

Programos viršuje atrašomas gedimų medžio loginė išraiška Būlio algebros simboliais ( $IR - \cdot$  bei  $ARBA - +$ ), baziniai įvykiai koduojami formatu  $x[i]$ , kur  $i$  rodo bazinio įvykio numerį, bet ne rangą. Įvykių rangai apibrėžiami antroje eilutėje. Galima pasirinkti žingsninį konvertavimą į BDD su valdymo juostos pagalba.

Tai patogi mokomoji ITE sąlygos programinė priemonė. Tyrėjas gali stebėti algoritmo žingsnius programinio kodo lange ir kartu sekti, kaip keičiasi ITE sąlygos  $\langle F, G, H \rangle$  atskiros dalys  $F$ ,  $G$  ir  $H$ . Be to, dešiniajame apatiniame programos lange visada pateikiamas įvestas gedimų medis, o kairiajame apatiniame programos lange pateikiama dvejetainė sprendimų diagrama. Paspaudus langų šonuose esančius didinimo mygtukus, gaunama tame lange nubraižytos diagramos ITE sąlyga. Pavyzdžiui, 1.12 paveiksle pateiktos dvejetainės sprendimų diagramos ITE sąlyga yra pateikiama formate  $R=(x[1],(x[2],1,(x[3],1,0)),0)$ , ji visiškai atitinka ITE struktūrą, užrašytą (1.12) arba (1.13) lygtimis, kai  $x[1]$  yra  $G$ ,  $x[2]$  yra  $S$  ir  $x[3]$  yra  $B$  bei  $G < S < B$ .

Programos pateikiama BDD yra kairinė ir mazgai pateikiami viename stulpelyje. Ištinės linijos žymi vienetines šakas, punktyrinės – nulines šakas.

„VisBDD“ programa padeda lengvai suprasti ITE sąlygą ir konvertuoja įvestą gedimų medžio loginę išraišką į BDD struktūrą. „VisBDD“ gali konvertuoti gedimų medžius jei jie sudaryti iš ne daugiau kaip 20 bazinių įvykių, todėl jos neįmanoma taikyti dideliems gedimų medžiams. Dar vienas programos trūkumas yra reikalavimas, kad baziniai įvykiai būtų perkoduoti į formatą  $x[i]$ . Be to, programa „VisBDD“ negali suskaičiuoti pavaizduotos BDD grafe sistemos gedimo tikimybės.



## 2. TIRIAMOJI DALIS IR REZULTATAI

### 2.1. KAUNO HIDROELEKTRINĖ IR JOS GEDIMŲ MEDIS

Teorinėje dalyje aprašyti metodai, naudojantys MCS, BDD struktūrą, buvo taikyti Kauno hidroelektrinės (HE) užtvankos vartų valdymo sistemos gedimo analizei. Kauno HE pasirinkau norėdamas susipažinti su jos daroma įtaka galimam Nemuno upės potvyniui. Susipažinau su šios HE pagrindinėmis savybėmis ir valdymu (2.1.1. poskyris). Atlikau Kauno HE vartų valdymo sistemos gedimų medžio kokybinę ir kiekybinę analizę. Taip pat, pasinaudodamas Kauno HE vartų valdymo sistemos gedimų medžiu, tyriaus teorinėje dalyje aprašytų metodų „elgesį“ esant įvairioms gedimų medžio bazinių įvykių tikimybėmis. Be to, analizavau BDD struktūros užimamą „erdvę“ priklausomai nuo bazinių įvykių rangavimo.

#### 2.1.1. PAGRINDINĖS CHARAKTERISTIKOS

Kauno HE yra didžiausia elektrinė Lietuvoje, naudojanti elektros gamybai atsinaujinančius išteklius, t.y. Nemuno upės vandens srautą. Tokio tipo elektrinės vadinamos vagos tipo hidroelektrinės. Kauno HE nutolusi nuo Nemuno žiočių 224 km.

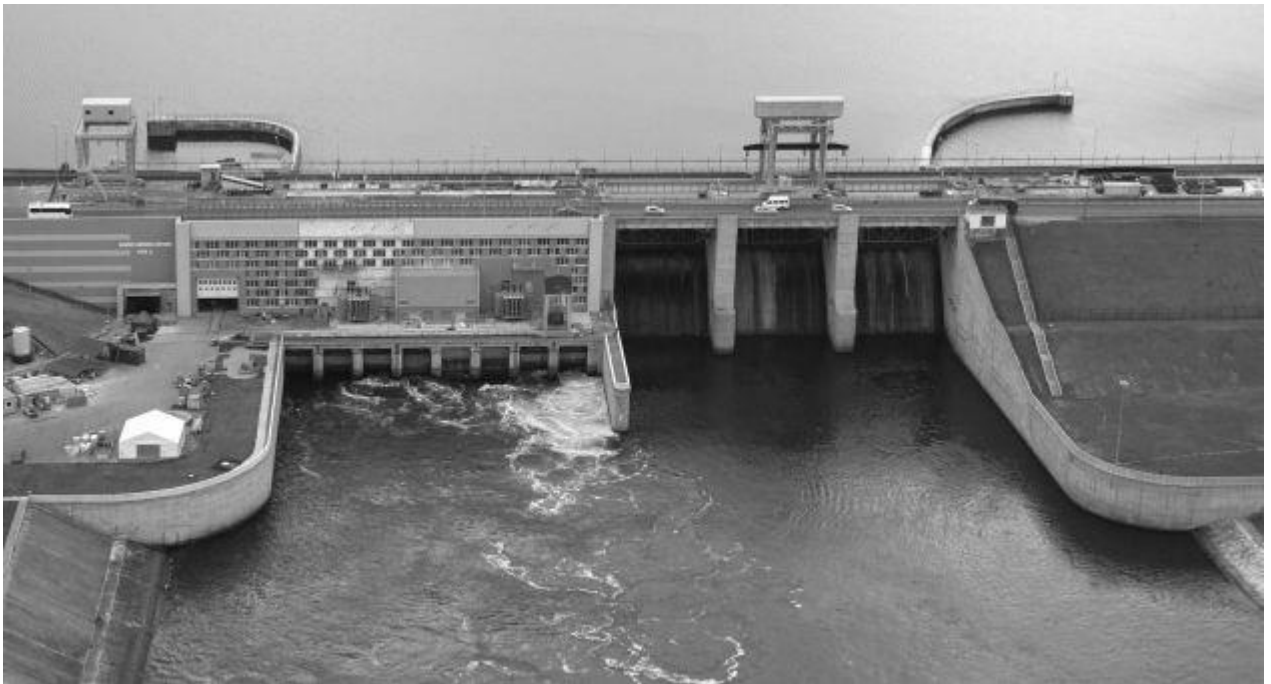
Kauno HE užtvankos statybos darbai prasidėjo 1955 metais, o per 1959-1960 metus paleistos visos keturios energijos gaminimo turbinos po 25,2 MW kiekviena, bendras HE galingumas 100,8 MW. Ji gamina 80% visos, tokiu būdu gaminamos, elektros energijos šalyje. Elektrinė kasmet pagamina apie 2% Lietuvoje suvartojamos elektros energijos. Kauno HE pagaminta elektra tiekama į 110 kV įtampos tinklą (<http://www.lpc.lt/lt>).

Pagrindiniai hidroelektrinės užtvankos duomenys:

- vartai valdomi kranais;
- maksimalus vandens pralaidumas 3990 m<sup>3</sup>/s;
- užtvankos statinių keteros altitudė 48 m;
- nominali patvankos altitudė 44 m;
- maksimali patvankos altitudė 45,6 m;
- minimali patvankos altitudė 40 m.

Užtvankus Nemuną, susidarė 63,5 km<sup>2</sup> ploto vandens baseinas – Kauno marios. Prie hidroelektrinės turbinų susidaro 20,1 m vandens perkrytis. Vidutinis Kauno marių gylis yra 7,5 m, prie užtvankos siekia 25 metrus. 2.1 paveiksle parodyta Kauno HE užtvankos vartai.

2005 m. su Švedijos bendrove ALSTOM Power Sweden AB buvo pasirašyta Kauno HE modernizavimo ir atnaujinimo rangos sutartis. Projekto tikslas – modernizuoti Kauno HE, didinant energijos tiekimo patikimumą, saugumą ir efektyvumą. Hidroelektrinės rekonstrukcijos projekto įgyvendinimui Europos Sąjunga iš struktūrinių fondų skyrė 29,2 mln. litų.

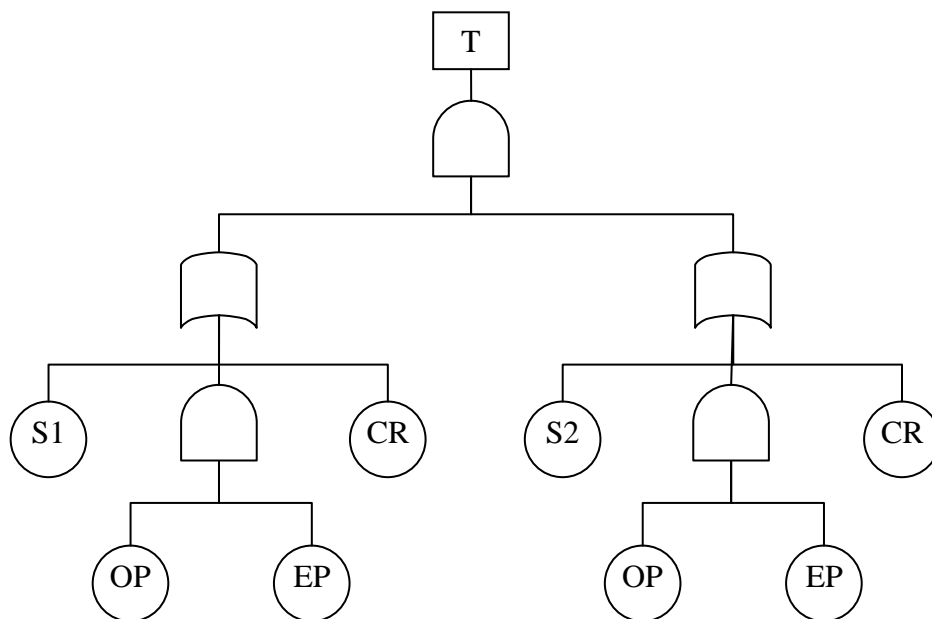


**2.1 pav. Kauno HE užtvanka**

### **2.1.2. UŽTVANKOS VARTŲ VALDYMO SISTEMA**

Kauno HE naudinga ne tik dėl elektros energijos tiekimo, bet ir saugo Nemuno vagos regioną nuo potvynio. Hidroelektrinė iš dalies reguliuoja Nemuno upės vandens srautą. Ji gali padidinti ar sumažinti upės vandens srautą už užtvankos, tokiu būdu gali būti reguliuojamas galimo potvynio vandens srautas. Reguluojant Kauno HE vandens pralaidumą pro vartus, galima sumažinti pratekančio per užtvanką vandens srautą. Sumažinus vandens srauto pratekėjimą, gali kilti vandens lygis Kauno mariose. Jei vanduo pakiltų virš maksimalios 45,6 metrų altitudės, vandens srautas gali pralaužti užtvanką ir užtvindyti didelę dalį Kauno miesto ir Nemuno upės vagos pakrantes, taip pridarydamas daug žalos Lietuvos žmonėms, aplinkai ir ekonomikai (Simaitytė, 2007).

Vanduo Kauno mariose gali pakilti dėl kelių priežasčių. Viena iš jų yra užtvankos vartų valdymo sistemos gedimas, kai negalima pakelti vartų. Tada vandens srautas gali pralaužti užtvanką ir sukelti potvynį. Kauno HE užtvankos vartų valdymo sistema pavaizduota 2.2 paveikslo gedimų medyje (Simaitytė, 2007).



**2.2 pav. Kauno HE užtvankos vartų valdymo sistemos gedimų medis**

Gedimų medžio, pateikto 2.2 paveiksle, bazinių įvykių paaiškinimai ir gedimo tikimybės pateikta 2.1 lentelėje:

**2.1 lentelė**

#### Kauno HE užtvankos vartų valdymo sistemos komponentai

| Žymėjimai | Reikšmės                           | Gedimo tikimybės  |
|-----------|------------------------------------|-------------------|
| OP        | Išorinės energijos tiekimo gedimas | $1 \cdot 10^{-3}$ |
| EP        | Avarinės energijos tiekimo gedimas | $3 \cdot 10^{-3}$ |
| CR        | Kranų bėgių gedimas                | $1 \cdot 10^{-4}$ |
| S1        | Pagrindinio kranų sistemos gedimas | $3 \cdot 10^{-4}$ |
| S2        | Atsarginio kranų sistemos gedimas  | $3 \cdot 10^{-3}$ |
| T         | Vartų valdymo sistemos gedimas     | – (vertinama)     |

Kauno HE vartų valdymo sistemos gedimų medžio baigiamojo įvykio T tikimybė 2.1 lentelėje nepateikiama, nes T yra užtvankos vartų valdymo sistemos gedimas. Įvykio T tikimybę įvertinsime kituose skyriuose naudodami pirmoje darbo dalyje išanalizuotus metodus.

## 2.2. SISTEMOS ANALIZĖ TAIKANT MCS

Kadangi 2.2 paveiksle pavaizduotas Kauno HE užtvankos vartų valdymo gedimų medis sudarytas jungiant bazinius įvykius jungtimis *IR*, *ARBA*, tai šis gedimų medis yra koherentiškas. Koherentiškų gedimų medžių analizei naudojama minimalių kirtimų aibės, todėl ir 2.2 paveiksle pavaizduotą gedimų medį analizuosime taikant metodą, paremtą minimalių kirtimų aibės vertinimu.

### 2.2.1. SISTEMOS MINIMALIŲ KIRTIMŲ AIBĖ

Pirmiausia išreiškime 2.2 paveiksle pavaizduotą gedimų medį Būlio algebros formule:

$$T = (S1 + OP \cdot EP + CR) \cdot (S2 + OP \cdot EP + CR). \quad (2.1)$$

Ieškant (2.1) formulėje aprašyto gedimų medžio minimalių kirtimų aibės, reikia šią išraišką pertvarkyti. Redukuokime Kauno HE užtvankos vartų valdymo sistemos gedimų medį (2.1) taikant 1.3 lentelėje pateiktas Būlio algebros taisykles taip, kad baigtinėje šio gedimų medžio išraiškoje (2.4) neliktų apskliaudimo ženklų, o baziniai įvykiai būtų sujungti tik „+“ bei „·“ operatoriais;

- Gedimų medžio, aprašyto (2.1) lygtimi, išskleidimas:

$$\begin{aligned} T &= (S1 + OP \cdot EP + CR) \cdot (S2 + OP \cdot EP + CR) \\ &= S1 \cdot S2 + S1 \cdot OP \cdot EP + S1 \cdot CR \\ &\quad + OP \cdot EP \cdot S2 + OP \cdot EP \cdot OP \cdot EP + OP \cdot EP \cdot CR \\ &\quad + CR \cdot S2 + CR \cdot OP \cdot EP + CR \cdot CR \end{aligned} \quad (2.2)$$

- Apjungimo taisyklės taikymas (2.2) išraiškai:

$$\begin{aligned} T &= S1 \cdot S2 + S1 \cdot OP \cdot EP + S1 \cdot CR \\ &\quad + OP \cdot EP \cdot S2 + OP \cdot EP + OP \cdot EP \cdot CR \\ &\quad + CR \cdot S2 + CR \cdot OP \cdot EP + CR \end{aligned} \quad (2.3)$$

- Absorbcijos taisyklės taikymas (2.3) išraiškai:

$$T = S1 \cdot S2 + OP \cdot EP + CR. \quad (2.4)$$

Gauta (2.4) išraiška daugiau neredukuojama. Parodome (pagal 1.2.1 poskyrį), kad Kauno HE užtvankos vartų valdymo sistemos gedimų medžio, išreikšto (2.1) – (2.4) išraiškomis, minimalių kirtimų aibė yra:

$$\{S1, S2\}, \{OP, EP\}, \{CR\}. \quad (2.5)$$

Iš minimalių kirtimų aibės matome, kad Kauno HE vartų valdymo sistema suges, jei kartu suges pagrindinis ir pagalbinis kranai, arba jei nebus tiekiama elektros energija kranams nei iš vidinio, nei iš išorinio tinklo, arba jei suges bėgiai, kuriais kranai juda. Kitų priežasčių šiai sistemai sugesti neišskiriama.

### 2.2.2. SISTEMOS GEDIMO TIKIMYBĖ

Remdamiesi gedimų medžio gedimo tikimybės vertinimu (1.2.2 poskyris) analitiškai įvertiname Kauno HE užtvankos vartų valdymo sistemos gedimo tikimybę. Įrašome į (1.6) lygtį gautas minimalių kirtimų aibės bazinių įvykių kombinacijas (2.5 išraiška):

$$\begin{aligned} P(T) &= \sum_{i=1}^3 P(C_i) - \sum_{i < j}^3 P(C_i \cap C_j) + P(C_1 \cap C_2 \cap C_3) \\ &= P(S1 \cap S2) + P(OP \cap EP) + P(CR) \\ &\quad - (P(S1 \cap S2 \cap OP \cap EP) + P(S1 \cap S2 \cap CR) + P(OP \cap EP \cap CR)) \\ &\quad + P(S1 \cap S2 \cap OP \cap EP \cap CR) \end{aligned} \quad (2.6)$$

kur  $C_i$  yra (2.5) minimalių kirtimų aibės minimalūs kirtimai.

Kadangi (2.5) išraiškos minimalių kirtimų aibės baziniai nariai yra nepriklausomi, tai (2.6) tikimybės formulę galime taip perrašyti:

$$P(T) = P(S1) \cdot P(S2) + P(OP) \cdot P(EP) + P(CR) - (P(S1) \cdot P(S2) \cdot P(OP) \cdot P(EP) + P(S1) \cdot P(S2) \cdot P(CR) + P(OP) \cdot P(EP) \cdot P(CR)) + P(S1) \cdot P(S2) \cdot P(OP) \cdot P(EP) \cdot P(CR) \quad (2.7)$$

Įrašius 2.1 lentelėje pateiktas bazinių įvykių tikimybes į (2.7) išraišką, gauname sistemos gedimo tikimybę:

$$P(T) = (3 \cdot 10^{-4} \cdot 3 \cdot 10^{-3}) + (1 \cdot 10^{-3} \cdot 3 \cdot 10^{-3}) + 1 \cdot 10^{-4} - ((3 \cdot 10^{-4} \cdot 3 \cdot 10^{-3} \cdot 1 \cdot 10^{-3} \cdot 3 \cdot 10^{-3}) + (3 \cdot 10^{-4} \cdot 3 \cdot 10^{-3} \cdot 1 \cdot 10^{-4}) + (1 \cdot 10^{-3} \cdot 3 \cdot 10^{-3} \cdot 3 \cdot 10^{-3})) + (3 \cdot 10^{-4} \cdot 3 \cdot 10^{-3} \cdot 1 \cdot 10^{-3} \cdot 3 \cdot 10^{-3} \cdot 1 \cdot 10^{-4}) = 1,0389960730027 \cdot 10^{-4}$$

Taigi, Kauno HE užtvankos vartų valdymo sistemos gedimo tikimybė yra  $1,039 \cdot 10^{-4}$ .

Įvertinkime ir palyginkime šio gedimų medžio tiesioginę ir priešingąją tikimybių aproksimacijas, išreikštas pagal (1.7) ir (1.8) formules, su tiksliai įvertinta sistemos gedimo tikimybe:

- Tiesioginė aproksimacija:

$$P(T) = \sum_{i=1}^3 P(C_i) = P(S1 \cap S2) + P(OP \cap EP) + P(CR) = (3 \cdot 10^{-4} \cdot 3 \cdot 10^{-3}) + (1 \cdot 10^{-3} \cdot 3 \cdot 10^{-3}) + 1 \cdot 10^{-4} = 1,039 \cdot 10^{-4} \quad (2.8)$$

kur  $C_i$  yra (2.5) minimalių kirtimų aibės kirtimai.

- Priešingoji aproksimacija:

$$P(T) = 1 - \prod_{i=1}^3 (1 - P(C_i)) = 1 - ((1 - P(S1 \cap S2)) \cdot (1 - P(OP \cap EP)) \cdot (1 - P(CR))) = 1 - ((1 - (3 \cdot 10^{-4} \cdot 3 \cdot 10^{-3})) \cdot (1 - (1 \cdot 10^{-3} \cdot 3 \cdot 10^{-3})) \cdot (1 - 1 \cdot 10^{-4})) = 1,0389960730027 \cdot 10^{-4} \quad (2.9)$$

kur  $C_i$  yra (2.5) minimalių kirtimų aibės kirtimai.

### 2.2.3. SISTEMOS VERTINIMAS NAUDOJANT „OPENFTA“

Programoje „OpenFTA“ sudarykime Kauno HE užtvankos vartų valdymo sistemos gedimų medį (2.3 paveikslas). Atlikus šio gedimų medžio analizę su programa „OpenFTA“ gavome tokią minimalių kirtimų aibę:

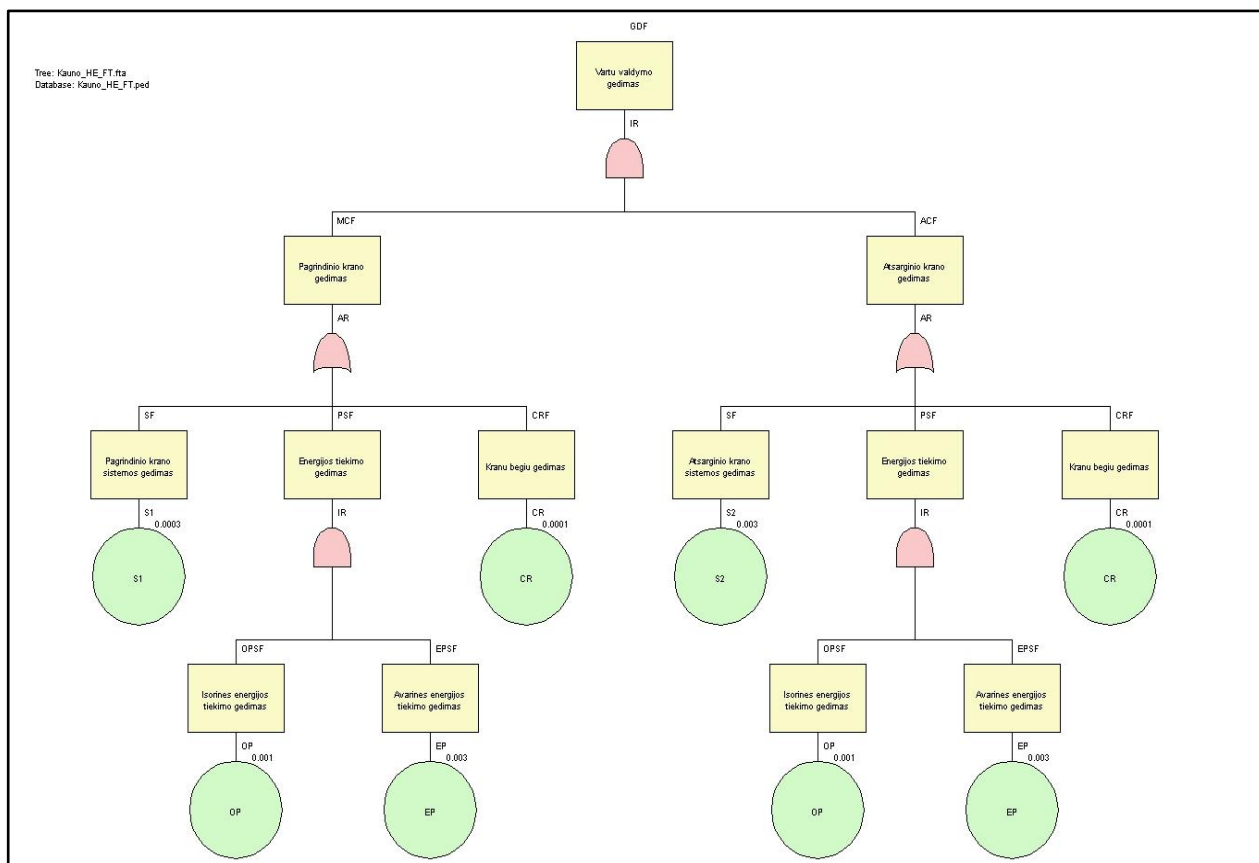
CR

EP OP

S1 S2

Kauno HE užtvankos vartų valdymo sistemos gedimo tikimybė suskaičiuota programoje „OpenFTA“:  $1.038996 \cdot 10^{-4}$ .

Atliktos šio gedimų medžio analizės pilni programos „OpenFTA“ rezultatų failai pateikti 1 priede.



2.3 pav. Kauno HE vartų valdymo gedimų medis programoje „OpenFTA“

## 2.3. SISTEMOS ANALIZĖ TAIKANT BDD

Panaudokime 1.4. skyriuje pateiktus metodus Kauno HE užtvankos vartų valdymo gedimų medžio analizei. Kadangi buvo analizuota (1.4. skyrius) keletas gedimų medžio konvertavimo į dvejetainę sprendimų diagramą metodų (Teisingumo lentelė, ITE sąlyga), tai juos panaudokime tiriamam gedimų medžiui (2.2 paveikslas) konvertuoti į BDD struktūrą. Taip pat pasinaudojant gautomis BDD, įvertinkime Kauno HE užtvankos vartų valdymo gedimo tikimybę analitiškai ir sukurta programine priemone „DemoITE“.

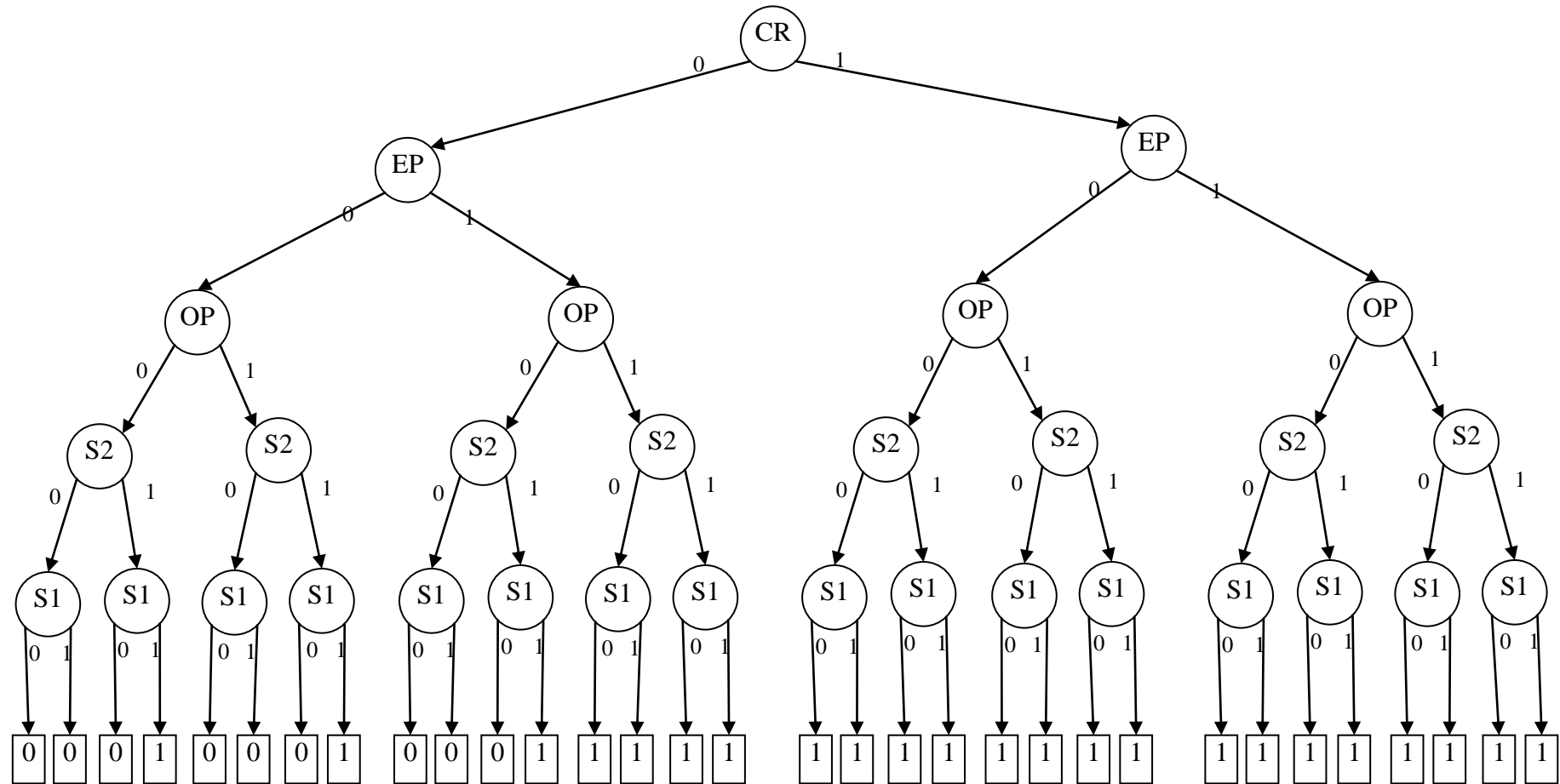
### 2.3.1. TEISINGUMO LENTELĖS TAIKYMAS

Konvertuodami gedimų medį į dvejetainę sprendimų diagramą pirmiausia turime suranguoti bazinius įvykius. Kauno HE užtvankos vartų valdymo sistemos bazinius įvykius suranguokime taip:

$$CR < EP < OP < S2 < S1.$$

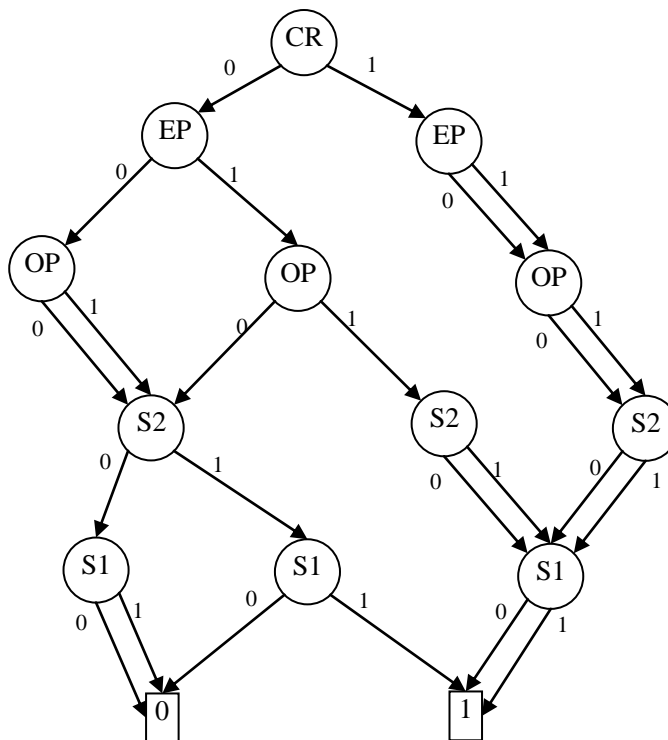
Toki rangavimą pasirinkome atsižvelgiant į 1.4.1. poskyryje pateiktą rekomendaciją.

Apibrėškime šių vartų valdymo gedimų medį teisingumo lentelėje (2 priedas) ir pateikime šios lentelės dvejetainės sprendimų diagramos struktūrą (2.4 paveikslas).

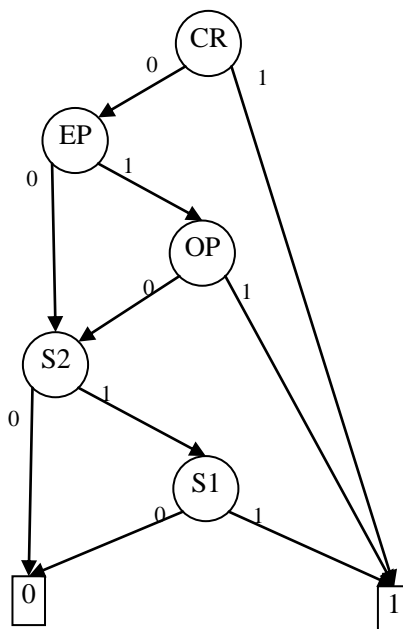


2.4 pav. Kauno HE užtvankos vartų valdymo sistemos nekompaktiška BDD

Gautoji BDD nėra kompaktiška. Pritaikę BDD struktūros redukcijos taisyklės (1.4.1. poskyris) šiai sistemos dvejetainiai sprendimų diagramai gauname kompaktišką BDD formą (2.5 b paveikslas).



a



b

2.5 pav. Kauno HE užtvankos vartų valdymo sistemos kompaktiška BDD



Galime pastebėti, kad dvejetainėje sprendimų diagramoje per vienetines šakas sujungti minimalūs sistemos gedimų medžio kirtimai (pagal 1.4.1. poskyrį), tokie patys kaip gavome (2.5) išraiškoje (2.2.1. poskyris):

$$\{CR\}, \{EP,OP\}, \{S2,S1\}.$$

Taigi, galime teigti, kad sistemos gedimų medžio konvertavimas į BDD struktūrą ir BDD redukovimas atlikti teisingai ir sudaryta BDD aprašo Kauno HE užtvankos vartų valdymo sistema.

### 2.3.2. ITE SĄLYGOS TAIKYMAS

Pasinaudodami 2.3.1. poskyryje apibrėžta Kauno HE užtvankos vartų valdymo sistemos gedimų medžio bazinių įvykių tvarka ( $CR < EP < OP < S2 < S1$ ) sudarykime dvejetainę sprendimų diagramą taikant ITE sąlygą (1.4.2. poskyris). Šio gedimų medžio konvertavimą atlikime nuosekliai kiekvienai sistemos gedimų medžio jungčiai.

$$J1 = OP \cdot EP = \langle OP,1,0 \rangle \cdot \langle EP,1,0 \rangle = \langle EP, \langle OP,1,0 \rangle, 0 \rangle$$

$$J3 = J1 = \langle EP, \langle OP,1,0 \rangle, 0 \rangle$$

$$\begin{aligned} J2 &= S1 + J1 + CR = \langle S1,1,0 \rangle + \langle EP, \langle OP,1,0 \rangle, 0 \rangle + \langle CR,1,0 \rangle \\ &= \langle CR,1,0 \rangle + \langle EP, \langle OP,1,0 \rangle, 0 \rangle + \langle S1,1,0 \rangle \\ &= \langle CR,1,0 \rangle + \langle EP, \langle OP,1,0 \rangle + \langle S1,1,0 \rangle, \langle S1,1,0 \rangle \rangle \\ &= \langle CR,1,0 \rangle + \langle EP, \langle OP,1, \langle S1,1,0 \rangle \rangle, \langle S1,1,0 \rangle \rangle \\ &= \langle CR,1, \langle EP, \langle OP,1, \langle S1,1,0 \rangle \rangle, \langle S1,1,0 \rangle \rangle \rangle \end{aligned}$$

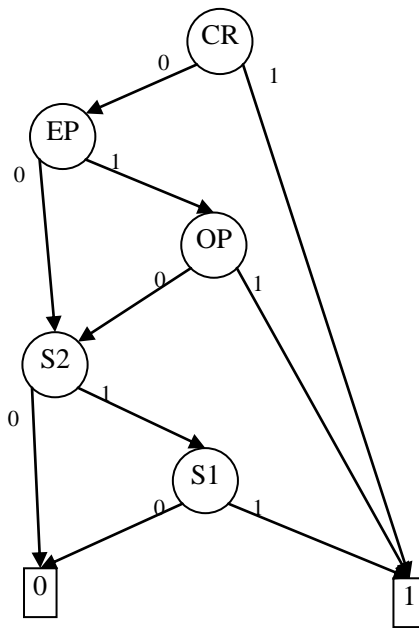
$$\begin{aligned} J4 &= S2 + J3 + CR = \langle S2,1,0 \rangle + \langle EP, \langle OP,1,0 \rangle, 0 \rangle + \langle CR,1,0 \rangle \\ &= \langle CR,1,0 \rangle + \langle EP, \langle OP,1,0 \rangle, 0 \rangle + \langle S2,1,0 \rangle \\ &= \langle CR,1,0 \rangle + \langle EP, \langle OP,1,0 \rangle + \langle S2,1,0 \rangle, \langle S2,1,0 \rangle \rangle \\ &= \langle CR,1,0 \rangle + \langle EP, \langle OP,1, \langle S2,1,0 \rangle \rangle, \langle S2,1,0 \rangle \rangle \\ &= \langle CR,1, \langle EP, \langle OP,1, \langle S2,1,0 \rangle \rangle, \langle S2,1,0 \rangle \rangle \rangle \end{aligned}$$

$$\begin{aligned} J5 &= J2 \cdot J4 = \langle CR,1, \langle EP, \langle OP,1, \langle S1,1,0 \rangle \rangle, \langle S1,1,0 \rangle \rangle \rangle \\ &\quad \cdot \langle CR,1, \langle EP, \langle OP,1, \langle S2,1,0 \rangle \rangle, \langle S2,1,0 \rangle \rangle \rangle \\ &= \langle CR,1, \langle EP, \langle OP,1, \langle S1,1,0 \rangle \rangle, \langle S1,1,0 \rangle \rangle \cdot \langle EP, \langle OP,1, \langle S2,1,0 \rangle \rangle, \langle S2,1,0 \rangle \rangle \rangle \\ &= \langle CR,1, \langle EP, \langle OP,1, \langle S1,1,0 \rangle \rangle \cdot \langle OP,1, \langle S2,1,0 \rangle \rangle, \langle S1,1,0 \rangle \cdot \langle S2,1,0 \rangle \rangle \rangle \\ &= \langle CR,1, \langle EP, \langle OP,1, \langle S1,1,0 \rangle \cdot \langle S2,1,0 \rangle \rangle, \langle S2, \langle S1,1,0 \rangle, 0 \rangle \rangle \rangle \\ &= \langle CR,1, \langle EP, \langle OP,1, \langle S2, \langle S1,1,0 \rangle, 0 \rangle \rangle, \langle S2, \langle S1,1,0 \rangle, 0 \rangle \rangle \rangle \end{aligned}$$

Tokiu būdu gaunama Kauno HE užtvankos vartų valdymo sistemos (2.10) ITE sąlyga:

$$ITE = J5 = \langle CR,1, \langle EP, \langle OP,1, \langle S2, \langle S1,1,0 \rangle, 0 \rangle \rangle, \langle S2, \langle S1,1,0 \rangle, 0 \rangle \rangle \rangle \quad (2.10)$$

Pavaizduokime (2.10) ITE struktūrą dvejetainėje sprendimų diagramoje (2.6 paveikslas). Pastebime, kad 2.5 b ir 2.6 paveiksluose vaizduojamos BDD yra tokios pačios, taip yra todėl, kad konvertuotas tas pats gedimų medis su tokiais pačiais bazinių įvykių rangais.



2.6 pav. BDD pagal ITE sąlygą

### 2.3.3. SISTEMOS GEDIMO TIKIMYBĖS ĮVERTINIMAS

Kadangi pagal du taikytus, konvertavimo į BDD struktūrą, metodus (naudojančius teisingumo lentelę ar ITE sąlygą) 2.5 b ir 2.6 paveikslų dvejetainės sprendimų diagramos yra vienodos, tai galime teigti, kad analitiškai konvertuodami sistemos gedimų medį į BDD klaidos nepadareme. Todėl galime pasinaudodami šiomis BDD įvertinti Kauno HE užtvankos vartų valdymo sistemos gedimo tikimybę. Šią tikimybę įvertinsime dviem būdais: tiesioginiu ir iteratyviu (1.4.4. poskyris).

Tiesioginis BDD struktūra(2.6 paveikslas) išreikštos sistemos gedimo tikimybės vertinimas:

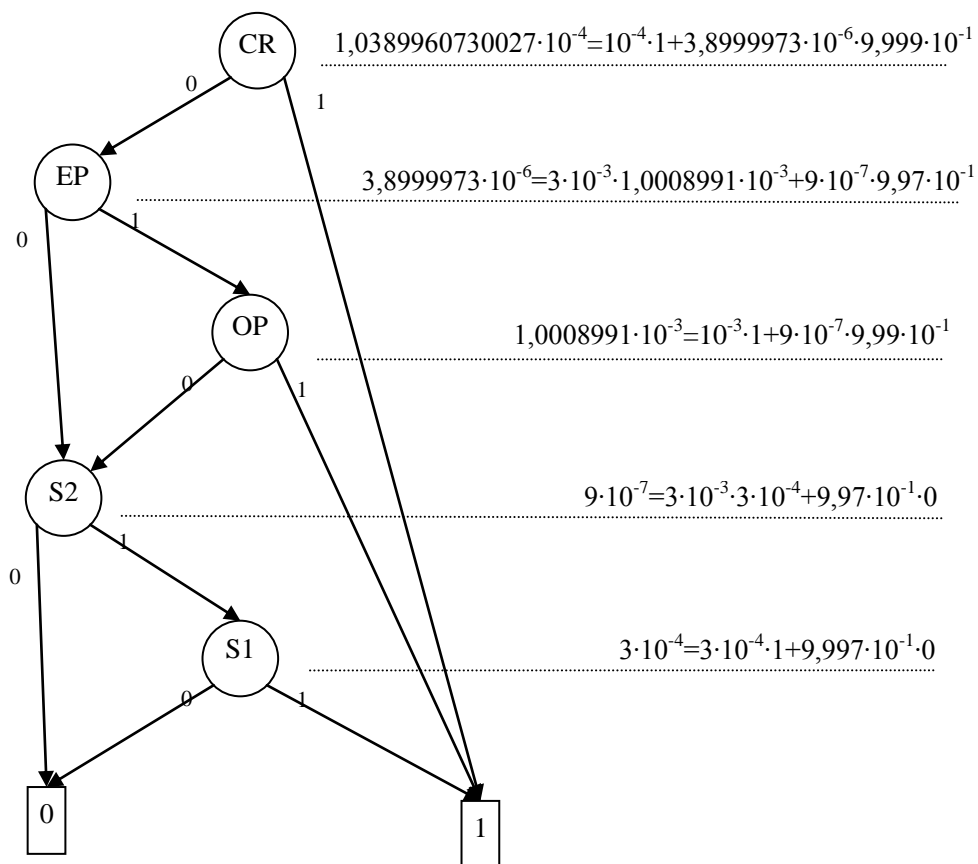
$$\begin{aligned}
 P(T) &= P(T | CR = 1) \cdot P(CR = 1) + P(T | CR = 0) \cdot P(CR = 0) \\
 &= P(T | CR = 1) \cdot P(CR = 1) \\
 &\quad + P(T | \{CR, EP\} = \{0, 1\}) \cdot P(\{CR, EP\} = \{0, 1\}) \\
 &\quad + P(T | \{CR, EP\} = \{0, 0\}) \cdot P(\{CR, EP\} = \{0, 0\}) \\
 &= P(T | CR = 1) \cdot P(CR = 1) + P(T | \{CR, EP, OP\} = \{0, 1, 1\}) \cdot P(\{CR, EP, OP\} = \{0, 1, 1\}) \\
 &\quad + P(T | \{CR, EP, OP\} = \{0, 1, 0\}) \cdot P(\{CR, EP, OP\} = \{0, 1, 0\}) \\
 &\quad + P(T | \{CR, EP, S2\} = \{0, 0, 1\}) \cdot P(\{CR, EP, S2\} = \{0, 0, 1\}) \\
 &= P(T | CR = 1) \cdot P(CR = 1) + P(T | \{CR, EP, OP\} = \{0, 1, 1\}) \cdot P(\{CR, EP, OP\} = \{0, 1, 1\}) \\
 &\quad + P(T | \{CR, EP, OP, S2\} = \{0, 1, 0, 1\}) \cdot P(\{CR, EP, OP, S2\} = \{0, 1, 0, 1\}) \\
 &\quad + P(T | \{CR, EP, S2, S1\} = \{0, 0, 1, 1\}) \cdot P(\{CR, EP, S2, S1\} = \{0, 0, 1, 1\}) \\
 &= P(T | CR = 1) \cdot P(CR = 1) + P(T | \{CR, EP, OP\} = \{0, 1, 1\}) \cdot P(\{CR, EP, OP\} = \{0, 1, 1\}) \\
 &\quad + P(T | \{CR, EP, OP, S2, S1\} = \{0, 1, 0, 1, 1\}) \cdot P(\{CR, EP, OP, S2, S1\} = \{0, 1, 0, 1, 1\}) \\
 &\quad + P(T | \{CR, EP, S2, S1\} = \{0, 0, 1, 1\}) \cdot P(\{CR, EP, S2, S1\} = \{0, 0, 1, 1\})
 \end{aligned}$$

Kadangi šios BDD (2.6 paveikslas) mazgai nepriklausomi, tai gauta išraiška supaprastėja ir sistemos gedimo tikimybę tiksliai apskaičiuojame (2.11) lygtimi:

$$\begin{aligned}
P(T) &= P(CR = 1) + P(\{CR, EP, OP\} = \{0, 1, 1\}) \\
&\quad + P(\{CR, EP, OP, S2, S1\} = \{0, 1, 0, 1, 1\}) + P(\{CR, EP, S2, S1\} = \{0, 0, 1, 1\}) \\
&= 1 \cdot 10^{-4} + (1 \cdot 10^{-4}) \cdot 3 \cdot 10^{-3} \cdot 1 \cdot 10^{-3} \\
&\quad + (1 - 1 \cdot 10^{-4}) \cdot 3 \cdot 10^{-3} \cdot (1 - 1 \cdot 10^{-3}) \cdot 3 \cdot 10^{-3} \cdot 3 \cdot 10^{-4} \\
&\quad + (1 - 1 \cdot 10^{-4}) \cdot (1 - 3 \cdot 10^{-3}) \cdot 3 \cdot 10^{-3} \cdot 3 \cdot 10^{-4} \\
&= 1,0389960730027 \cdot 10^{-4}
\end{aligned} \tag{2.11}$$

Gavome tokią pačią sistemos gedimo tikimybę, kaip ir atlikus skaičiavimus naudojant minimalių kirtimų aibę. Todėl galime teigti, kad gedimų medžio, aprašyto (2.1) lygtimi, konvertavimas į BDD atliktas teisingai.

Iteratyvus sistemos gedimo tikimybės vertinimas:



2.7 pav. Iteratyvus sistemos gedimo tikimybės vertinimas

Iteratyviu būdu tiksliai įvertinus Kauno HE užtvankos vartų valdymo sistemos gedimo tikimybę gauname jos įvertį lygų  $1,0389960730027 \cdot 10^{-4}$ . Pastebime, kad jis sutampa su tiesioginiu vertinimu suskaičiuota tikimybė, bet analitiškai vertinant sistemos gedimo tikimybę dvejetainėje sprendimų diagramoje patogiau taikant iteratyvų skaičiavimą.

### 2.3.4. SISTEMOS BDD KONSTRAVIMAS TAIKANT „VISBDD“

Pasinaudojant programa „VisBDD“ sudaroma Kauno HE užtvankos vartų valdymo sistemos BDD. Kadangi šioje programoje baziniai įvykiai žymimi išraiška  $x[i]$ . Perkoduojame 2.1 lentelėje pateiktus bazinius įvykius:

2.2 lentelė

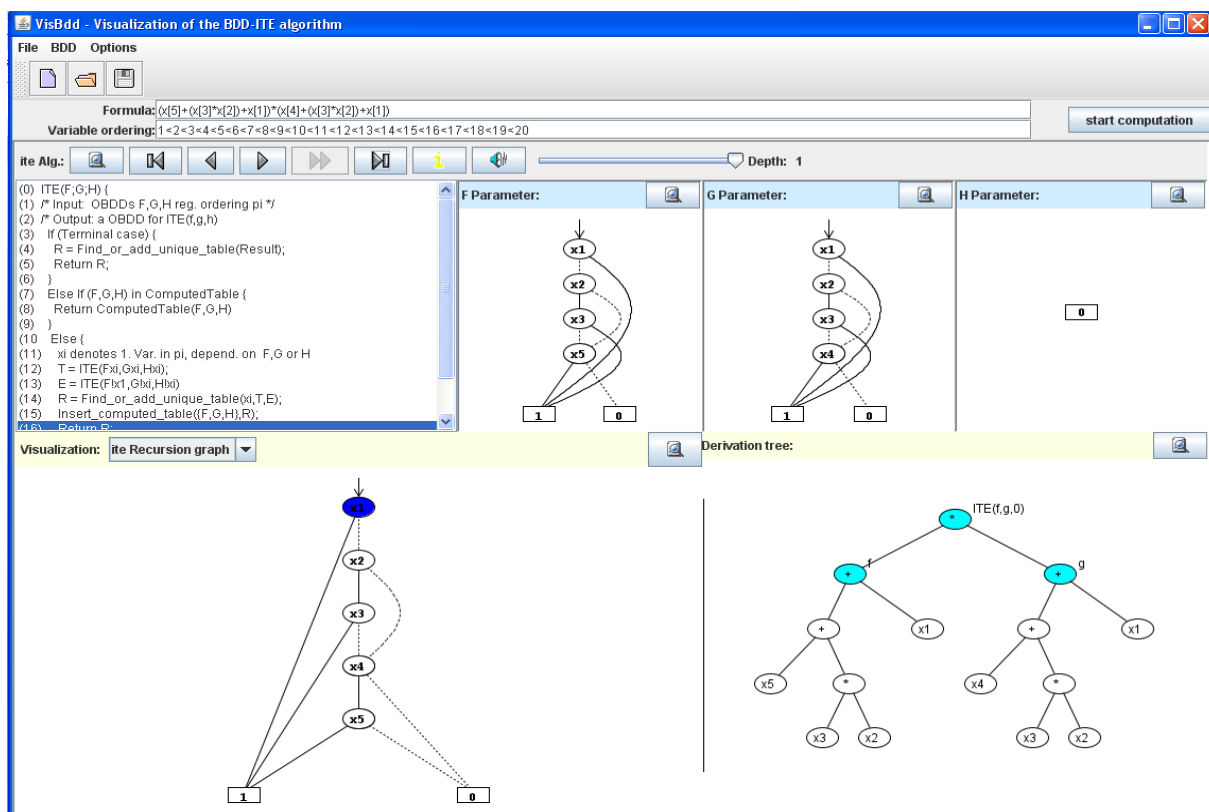
#### Sistemos bazinių įvykių perkodavimas

| Žymėjimas programoje VisBDD | Žymėjimas | Gedimo tikimybės  |
|-----------------------------|-----------|-------------------|
| $x[1]$                      | CR        | $1 \cdot 10^{-4}$ |
| $x[2]$                      | EP        | $3 \cdot 10^{-3}$ |
| $x[3]$                      | OP        | $1 \cdot 10^{-3}$ |
| $x[4]$                      | S2        | $3 \cdot 10^{-3}$ |
| $x[5]$                      | S1        | $3 \cdot 10^{-4}$ |

Tokiu būdu perkoduoti baziniai įvykiai (2.2 lentelė) atitinka 2.3.1. poskyryje apibrėžtą rangavimą:

$$x[1] < x[2] < x[3] < x[4] < x[5].$$

Į programą „VisBDD“ (2.8 paveikslas) suvedama (2.1) išraiška su perkoduotais žymėjimais ir parenkama aukščiau parodytą bazinių įvykių tvarka.



2.8 pav. Kauno HE užtvankos vartų valdymo sistema programoje „VisBDD“

Taikant programą „VisBDD“ sistemos gedimų medžių konvertavimą į BDD gaunama tokia pati BDD kaip ir 2.3.1. bei 2.3.2. poskyriuose, tik mazgai yra viename stulpelyje ir vienetinės šakos nukreiptos į kairę. „VisBDD“ pateikiama ITE (2.12) struktūra visiškai atitinka (2.10) struktūrą:

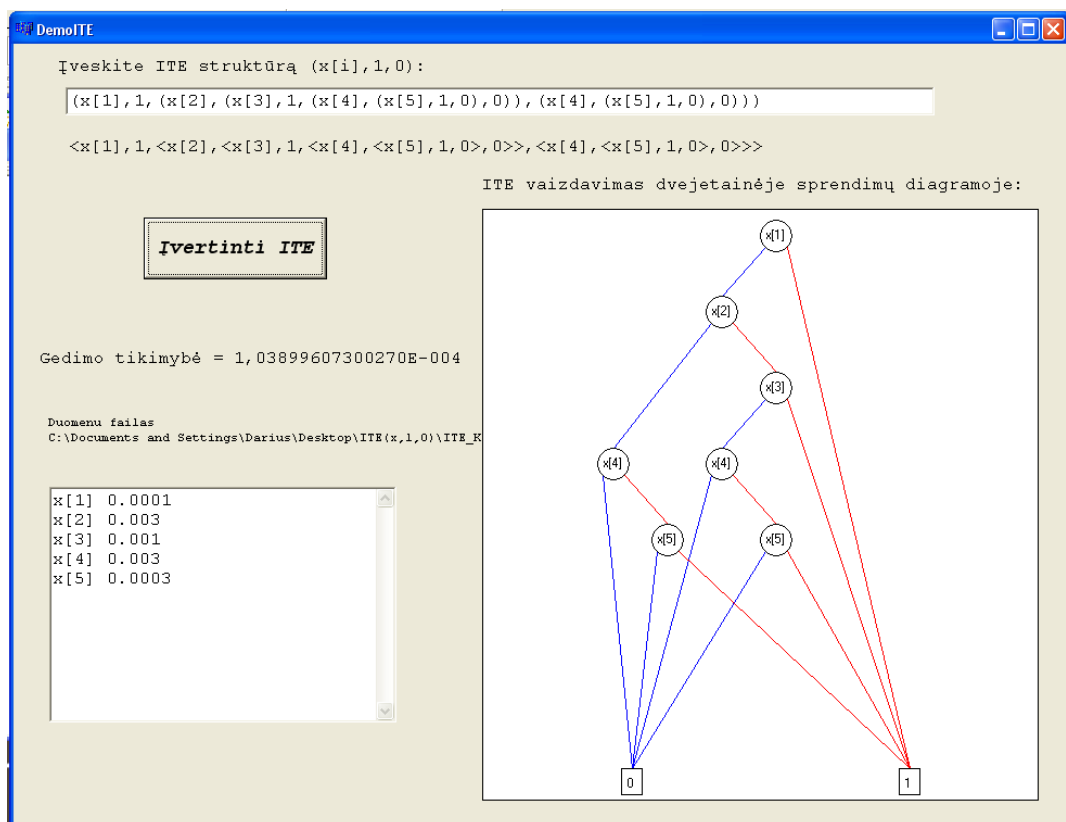
$$R=(x[1],1,(x[2],(x[3],1,(x[4],(x[5],1,0),0)),(x[4],(x[5],1,0),0))) \quad (2.12)$$

Taigi, programa „VisBDD“ atlikę Kauno HE užtvankos vartų valdymo sistemos gedimų medžio konvertavimą gavome tokią pačią BDD kaip ir analitiškai konvertuojant (2.3.1. ir 2.3.2. poskyriai).

### 2.3.5. GEDIMO TIKIMYBĖS VERTINIMAS TAIKANT „DEMOITE“

Kadangi programa „VisBDD“, sukurta konvertuoti gedimų medžius į BDD, negali įvertinti sistemų gedimo tikimybių, buvo sukurta programa „DemoITE“ (plačiau 2.5 skyrius). Programa sistemos gedimo tikimybei vertinti naudoja ITE sąlygą, kurią pateikia programa „VisBDD“, ir atskirą bazinių įvykių gedimo tikimybių duomenų failą.

Su „DemoITE“ programa atliktas Kauno HE užtvankos vartų valdymo sistemos gedimo tikimybės vertinimas (2.9 paveikslas). Tikimybė skaičiuota pagal (2.12) ITE sąlygą, naudojant sistemos bazinių įvykių gedimo tikimybes (2.1 lentelė) įrašytas į duomenų failą „ITE\_Kauno\_HE.txt“. Gauta sistemos gedimo tikimybė yra  $1,0389960730027 \cdot 10^{-4}$ . Ji visiškai sutampa su analitiškai įvertintomis sistemos gedimo tikimybėmis (2.3.3. poskyris).



2.9 pav. Sistemos gedimo tikimybės vertinimas taikant programą „DemoITE“

Programos ITE dešiniajame lange (2.9 paveikslas) pavaizduota grafiška (2.12) ITE sąlyga išreikšta dešininėje dvejetainėje sprendimų diagramoje. Ši BDD nėra pilnai redukuota į kompaktišką.

Pastebėta, kad keliais būdais (taikant teisingumo lentelę, ITE, „VisBDD“) konvertavus Kauno HE užtvankos vartų valdymo sistemos gedimų medį į dvejetainę sprendimų diagramą, kai rangavimas toks pat (2.3.1. poskyris), gaunama ta pati BDD struktūra. Taip pat įvertinus sistemos gedimo tikimybę, ji visais atvejais (tiesiogiai, iteratyviai, „DemoITE“) buvo tokia pati  $1,0389960730027 \cdot 10^{-4}$ .

## 2.4. GEDIMO TIKIMYBĖS VERTINIMO PALYGINIMAS

Ankstesniuose skyriuose gauta, kad sistemos gedimo tikimybė vertinant įvairiais būdais nesiskiria. Todėl patikrinta, ar nesiskirs sistemos gedimo tikimybė esant kitokioms bazinių įvykių tikimybėms. Analizei naudota Kauno HE užtvankos vartų valdymo sistema. Sistemos gedimo tikimybė vertinta, naudojant analitinį MCS vertinimą pagal (2.7) formulę ir aproksimacijas išreikštas (2.8), (2.9) lygtimis bei naudojant programas „OpenFTA“ ir „DemoITE“. Programoje „DemoITE“ naudojamos ITE sąlygos bazinių įvykių rangavimas toks pat, kaip 2.3.1. poskyryje. Tyrimo rezultatai pateikti 2.3 lentelėje. Bazinių įvykių tikimybės 2.3 lentelėje parinktos taip, kad visų bazinių įvykių tikimybės būtų (kiekvienu atveju) vienodos.

2.3 lentelė

Tikimybės vertinimo palyginimas, esant vienodoms bazinių įvykių tikimybėms

| Gedimo tikimybės vertinimo būdai | Bazinių įvykių tikimybės          |                      |                      |                      |                      |
|----------------------------------|-----------------------------------|----------------------|----------------------|----------------------|----------------------|
|                                  | CR                                | EP                   | OP                   | S2                   | S1                   |
|                                  | $9,99 \cdot 10^{-1}$              | $9,99 \cdot 10^{-1}$ | $9,99 \cdot 10^{-1}$ | $9,99 \cdot 10^{-1}$ | $9,99 \cdot 10^{-1}$ |
| Analitinis                       | $9,99999996003999 \cdot 10^{-1}$  |                      |                      |                      |                      |
| Tiesioginė aproksimacija         | 2,995002                          |                      |                      |                      |                      |
| Priešingoji aproksimacija        | $9,99999996003999 \cdot 10^{-1}$  |                      |                      |                      |                      |
| „OpenFTA“                        | $9,999999 \cdot 10^{-1}$          |                      |                      |                      |                      |
| „DemoITE“                        | $9,99999996003999 \cdot 10^{-1}$  |                      |                      |                      |                      |
|                                  | $5,0 \cdot 10^{-1}$               | $5,0 \cdot 10^{-1}$  | $5,0 \cdot 10^{-1}$  | $5,0 \cdot 10^{-1}$  | $5,0 \cdot 10^{-1}$  |
| Analitinis                       | $7,1875 \cdot 10^{-1}$            |                      |                      |                      |                      |
| Tiesioginė aproksimacija         | 1,0                               |                      |                      |                      |                      |
| Priešingoji aproksimacija        | $7,1875 \cdot 10^{-1}$            |                      |                      |                      |                      |
| „OpenFTA“                        | $7,187500 \cdot 10^{-1}$          |                      |                      |                      |                      |
| „DemoITE“                        | $7,1875 \cdot 10^{-1}$            |                      |                      |                      |                      |
|                                  | $1 \cdot 10^{-10}$                | $1 \cdot 10^{-10}$   | $1 \cdot 10^{-10}$   | $1 \cdot 10^{-10}$   | $1 \cdot 10^{-10}$   |
| Analitinis                       | $1,0000000002 \cdot 10^{-10}$     |                      |                      |                      |                      |
| Tiesioginė aproksimacija         | $1,0000000002 \cdot 10^{-10}$     |                      |                      |                      |                      |
| Priešingoji aproksimacija        | $1,00000008274037 \cdot 10^{-10}$ |                      |                      |                      |                      |
| „OpenFTA“                        | $1,000000 \cdot 10^{-10}$         |                      |                      |                      |                      |
| „DemoITE“                        | $1,0000000002 \cdot 10^{-10}$     |                      |                      |                      |                      |

Visais atvejais (analitiškai, „OpenFTA“, „DemoITE“) suskaičiavus sistemos gedimo tikimybę (2.3 lentelė) gavome tokias pačias gedimo tikimybes. Pastebime, kad įvertintos gedimo tikimybės

aprosimacijos ne visada sutampa su tikslia sistemos gedimo tikimybe. Kai bazinių įvykių tikimybės didelės ( $9,99 \cdot 10^{-1}$ ,  $5,0 \cdot 10^{-1}$ ), tai tiesioginė aproksimacija labai skiriasi nuo tiksliai vertinamos sistemos gedimo tikimybės ir priešingai, kai bazinių įvykių tikimybės mažos ( $1 \cdot 10^{-10}$ ), tai skiriasi priešingą aproksimacija įvertinta sistemos gedimo tikimybė nuo tiksliai įvertintos.

Kadangi analitiškai vertinant (pagal (1.6) formulę) skaičiuojamos minimalių kirtimų tikimybių sumos, o atliktame tyrime (2.3 lentelė) minimalių kirtimų tikimybės skiriasi:

$$P(C_1 = CR) = 1 \cdot 10^{-10}$$

$$P(C_2 = EP \cap OP) = 1 \cdot 10^{-20}$$

$$P(C_3 = S1 \cap S2) = 1 \cdot 10^{-20}$$

Todėl vertėtų įvertinti sistemos gedimo tikimybę, kai minimalių kirtimų tikimybės yra tokios pačios. Tokiu būdu pakeitus bazinių įvykių tikimybės, dar kartą įvertiname sistemos gedimo tikimybę (2.4 lentelė).

2.4 lentelė

#### Tikimybės vertinimo palyginimas, esant vienodoms minimalių kirtimų tikimybėms

| Gedimo tikimybės vertinimo būdai   | Bazinių įvykių tikimybės          |                           |                       |                           |                       |
|------------------------------------|-----------------------------------|---------------------------|-----------------------|---------------------------|-----------------------|
|                                    | CR                                | EP                        | OP                    | S2                        | S1                    |
|                                    | $9,99 \cdot 10^{-1}$              | $9,995 \cdot 10^{-1}$     | $9,995 \cdot 10^{-1}$ | $9,995 \cdot 10^{-1}$     | $9,995 \cdot 10^{-1}$ |
| <b>Minimalių kirtimų tikimybės</b> | $9,99 \cdot 10^{-1}$              | $\sim 9,99 \cdot 10^{-1}$ |                       | $\sim 9,99 \cdot 10^{-1}$ |                       |
| Analitinis                         | $9,999999990005 \cdot 10^{-1}$    |                           |                       |                           |                       |
| Tiesioginė aproksimacija           | 2,9970005                         |                           |                       |                           |                       |
| Priešingoji aproksimacija          | $9,999999990005 \cdot 10^{-1}$    |                           |                       |                           |                       |
| „OpenFTA“                          | $9,999999 \cdot 10^{-1}$          |                           |                       |                           |                       |
| „DemoITE“                          | $9,999999990005 \cdot 10^{-1}$    |                           |                       |                           |                       |
|                                    | $5,0 \cdot 10^{-1}$               | $7,07 \cdot 10^{-1}$      | $7,07 \cdot 10^{-1}$  | $7,07 \cdot 10^{-1}$      | $7,07 \cdot 10^{-1}$  |
| <b>Minimalių kirtimų tikimybės</b> | $5,0 \cdot 10^{-1}$               | $\sim 5,0 \cdot 10^{-1}$  |                       | $\sim 5,0 \cdot 10^{-1}$  |                       |
| Analitinis                         | $8,749244885995 \cdot 10^{-1}$    |                           |                       |                           |                       |
| Tiesioginė aproksimacija           | 1,499698                          |                           |                       |                           |                       |
| Priešingoji aproksimacija          | $8,749244885995 \cdot 10^{-1}$    |                           |                       |                           |                       |
| „OpenFTA“                          | $8,749245 \cdot 10^{-1}$          |                           |                       |                           |                       |
| „DemoITE“                          | $8,749244885995 \cdot 10^{-1}$    |                           |                       |                           |                       |
|                                    | $1 \cdot 10^{-10}$                | $1 \cdot 10^{-5}$         | $1 \cdot 10^{-5}$     | $1 \cdot 10^{-5}$         | $1 \cdot 10^{-5}$     |
| <b>Minimalių kirtimų tikimybės</b> | $1 \cdot 10^{-10}$                | $\sim 1 \cdot 10^{-10}$   |                       | $\sim 1 \cdot 10^{-10}$   |                       |
| Analitinis                         | $2,9999999997 \cdot 10^{-10}$     |                           |                       |                           |                       |
| Tiesioginė aproksimacija           | $3,0 \cdot 10^{-10}$              |                           |                       |                           |                       |
| Priešingoji aproksimacija          | $3,00000024822111 \cdot 10^{-10}$ |                           |                       |                           |                       |
| „OpenFTA“                          | $3,000000 \cdot 10^{-10}$         |                           |                       |                           |                       |
| „DemoITE“                          | $2,9999999997 \cdot 10^{-10}$     |                           |                       |                           |                       |

Šioje 2.4 lentelėje suskaičiavus sistemos gedimo tikimybės (analitiškai, „OpenFTA“, „DemoITE“) taip pat nesiskiria. O įvertintos gedimo tikimybės aproksimacijas, jos kaip ir 2.3 lentelėje ne visada sutampa su tiksliai įvertinta sistemos gedimo tikimybe, t.y. kuo minimalių kirtimų tikimybės

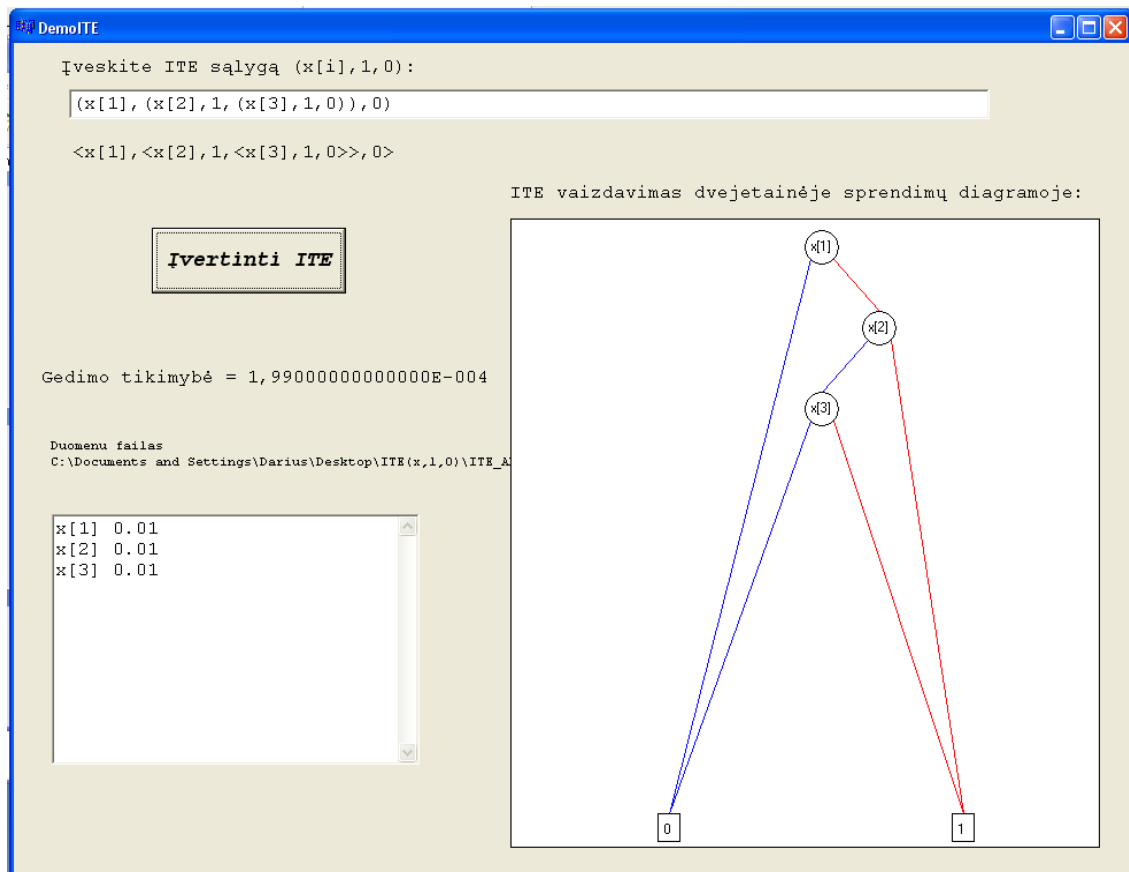
didesnės ( $9,99 \cdot 10^{-1}$ ,  $5,0 \cdot 10^{-1}$ ), tuo tiesioginė aproksimacija labiau skiriasi nuo tiksliai įvertintos gedimo tikimybės ir priešingai, kuo minimalių kirtimų tikimybės mažesnės ( $1 \cdot 10^{-10}$ ), tuo labiau skiriasi priešingą aproksimacija įvertinta sistemos gedimo tikimybė nuo įvertintos tiksliai.

## 2.5. PROGRAMA „DEMOITE“

Sukurta programa „DemoITE“ tiksliai vertina sistemos, aprašytos ITE sąlyga, gedimo tikimybę. Programa sukurta su programavimo kalba „C++ Builder 6“. Programos lango pavyzdys pateiktas 2.10 paveiksle. Pradedant dirbti su programa pirmiausia reikia į langelį: „Įveskite ITE( $x[i],1,0$ ) sąlygą“ įvesti duomenis, t.y. tiriamos sistemos ITE sąlygą, pavyzdžiui:

$$(x[1],(x[2],1,(x[3],1,0)),0). \quad (2.13)$$

Taip pat turi būti sukurtas ITE sąlygos elementų (mazgų, t.y. bazinių įvykių) gedimo tikimybių failas su plėtinium „\*.txt“. Duomenų failo pavyzdys „ITE\_Kauno\_HE.txt“ pateiktas 3 priede.



2.10 pav. Programos „DemoITE“ langas

Įvedus ITE sąlygą ir paspaudus mygtuką „Įvertinti ITE“ programa atidaro duomenų failo pasirinkimo mygtuką. Vartotojui pasirinkus duomenų failą programa patikrina ir perrašo ITE sąlygą žemiau įvedimo langelio formatu  $\langle x[1], \langle x[2], 1, \langle x[3], 1, 0 \rangle \rangle, 0 \rangle$ . Jei ITE sąlyga bloga, tai programa išveda klaidos pranešimą su tekstu: „Neatitinka ITE formatas ( $x[i], 1, 0$ )!“. Jei nepasirenkamas duomenų failas, tai programa išveda priminimo pranešimą: „Pasirinkite duomenų failą!“. Jei duomenų



faile nėra kurio nors mazgo gedimo tikimybės, tada išvedamas pranešimas: „[vykio x[i] tikimybės nėra duomenų faile!“.

Kai įvesti duomenys atitinka programos reikalavimus, tai „DemoITE“ skaičiuoja ir pateikia sistemos gedimo tikimybę (prie laukelio „Gedimo tikimybė =“). Taip pat parodo mazgų tikimybių duomenų failą lange „Duomenų failas“ ir nubraižo grafinę ITE sąlygą, t.y. dvejetainę sprendimų diagramą, lange „ITE vaizdavimas dvejetainėje sprendimų diagramoje“. Programos lange pavaizduota BDD nėra kompaktiškos formos. Panaudojus BDD redukcijos taisykles (1.4.1. poskyris) gaunama kompaktiška dvejetainė sprendimų diagrama.

### 2.5.1. REALIZUOTI ALGORITMAI

Programos realizacijos kodas pateiktas 4 priede. Pagrindiniai programos metodai yra: *NuskaitytiITE*, *SkliaustuSutapimas*, *GedimoTikimybe*, *BraizytiBDD*.

Programos „DemoITE“ metodas *NuskaitytiITE* yra rekurentiškas, t.y. kai metodas daug kartų kreipiasi pats į save. Metodas nuskaito ITE sąlygą iš programos lango, priskiria ITE struktūros mazgams gedimo tikimybes iš duomenų failo ir sudaro BDD struktūrą. Sudaryta BDD naudojama kituose programos metoduose (*GedimoTikimybe*, *BraizytiBDD*), skirtuose vertinti sistemos gedimo tikimybę ir braizyti dvejetainę sprendimų diagramą.

Metodas *SkliaustuSutapimas* tikrina programos lange pateiktos ITE sąlygos korektiškumą, t.y. tikrina ar įvesta ITE sąlyga turi (x[i],1,0) formatą.

Metodas *GedimoTikimybe* vertina sistemos, aprašytos ITE sąlyga, gedimo tikimybę. Metode realizuotas iteratyvus sistemos gedimo tikimybės vertinimas (1.4.4. poskyris).

Metodas *BraizytiBDD* programos lange „ITE vaizdavimas dvejetainėje sprendimų diagramoje“ nubraižo įvestos ITE sąlygos BDD struktūrą.

### 2.5.2. SKAIČIAVIMO PATIKRINIMAS

Kad sukurtą „DemoITE“ programą būtų galima taikyti, sistemos apibrėžtos ITE sąlyga gedimo tikimybės vertinimui, buvo patikrintas gedimo tikimybę skaičiuojantis programos metodas (algoritmas) *GedimoTikimybe*. Metode *GedimoTikimybe* naudojamas iteratyvus sistemos gedimo vertinimas buvo realizuotas panaudojus rekursiją. Tokiu būdu metodas sistemos gedimo tikimybę vertina nuo aukščiausią rangą turinčių mazgų (nuo BDD apačios), kol pasiekiamas BDD viršutinis mazgas, kurio tikimybę įvertinus gaunama sistemos gedimo tikimybė.

Programos „DemoITE“ tikimybės vertinimo adekvatumas patikrintas suskaičiavus (1.12) ITE išraiška aprašytos sistemos gedimo tikimybę a) ir b) atvejais (1.2.2. poskyris).

- a) sistemos gedimo tikimybė yra  $1,99 \cdot 10^{-4}$ .
- b) sistemos gedimo tikimybė yra  $3,75 \cdot 10^{-1}$ .

Įvertintus sistemos gedimą gauname tokias pačias gedimo tikimybes kaip ir ankstesniuose, sistemos, aprašytos (1.12) išraiška, gedimo tikimybes skaičiavimuose (1.4.4. poskyris). Todėl galime teigti, kad programa „DemoITE“ tiksliai vertina sistemos, aprašytos ITE sąlyga, gedimo tikimybę ir yra tinkama naudoti tokiu būdu aprašytų sistemų gedimo tikimybes vertinimui.

## 2.6. BAZINIŲ ĮVYKIŲ RANGAVIMAS

Patikrinkime ar naudojamas Kauno HE užtvankos vartų valdymo sistemos bazinių įvykių rangavimas: CR<EP<OP<S2<S1, naudojamas 2. darbo dalyje (2.3. skyrius), yra optimalus, t.y. ar taikant minėtą rangavimą gaunama mažiausia BDD struktūra (iš mažiausiai elementų sudaryta BDD). Tuo tikslu įvertinkime BDD mazgų kiekį esant skirtingiems bazinių įvykių rangavimams (2.5 lentelė).

**2.5 lentelė**

### BDD struktūros dydis esant skirtingiems rangavimams

| Bazinių įvykių tvarka | BDD mazgų kiekis |
|-----------------------|------------------|
| CR<EP<OP<S2<S1        | 7                |
| CR<OP<EP<S1<S2        | 7                |
| OP<EP<CR<S1<S2        | 7                |
| S1<S2<CR<OP<EP        | 8                |
| S1<S2<OP<EP<CR        | 8                |
| S2<S1<EP<OP<CR        | 8                |
| CR<S1<OP<S2<EP        | 10               |
| CR<OP<S1<S2<EP        | 10               |
| S1<CR<OP<S2<EP        | 11               |
| S2<OP<S1<CR<EP        | 12               |

Tyrimą atlikome naudodami programą „VisBDD“ (1.5. skyrius). Detalūs šio tyrimo rezultatai (bazinių įvykių perkodavimas ir rangavimas, gauta ITE sąlyga, nubraižytas BDD grafas bei suskaičiuotas BDD mazgų kiekis) pateikta 5 priede, o 2.5 lentelėje pateikta koncentruota informacija, rodanti, kaip skiriasi BDD struktūra esant skirtingiems bazinių įvykių rangavimams.

Matome, kad esant (2.3 poskyryje) bazinių įvykių rangavimui (CR<EP<OP<S2<S1) BDD struktūra yra mažiausia (sudaryta iš 7 mazgų), todėl 2.3 skyriuje gauta Kauno HE užtvankos vartų valdymo sistemos BDD struktūra užima mažiausiai „erdvės“. Kadangi sistema sudaryta iš 5 bazinių įvykių ( $5=2n$ ), tai sistemos BDD mazgų kiekis priklausomai nuo rangavimo (Bryant, 1992) kinta tarp 7 ( $7=2n+2$ ) ir 12 ( $12 \approx 2^{n+1} = 11,314$ , apvalinama į viršų), tai galime pastebėti ir 2.5 lentelėje.

## DISKUSIJA

Taikant sistemos gedimo tikimybei įvertinti skirtus skirtingus skaičiavimo metodus darbe kokybiškai ir kiekybiškai įvertinta Kauno HE užtvankos vartų valdymo sistemos gedimo tikimybė. Vertinimui naudojome sudarytą bendrą sistemos gedimo modelį, kuris buvo išreikštas gedimų medžiu (2.2 paveikslas).

Kokybiškai analizuojant Kauno HE užtvankos vartų valdymo sistemą išskyrėme tokias sistemos komponentų kombinacijas, kad sugedus bent vienos kombinacijos komponentams kartu sistema tikrai suges, t.y. neveiks užtvankos vartų valdymas. Šias kombinacijas nusako minimalūs kirtimai (1.2. skyrius) kurie buvo gauti sistemos gedimų medžio (2.1) išraiškai taikant 1.2.1. poskyryje pateiktas Būlio algebros taisykles. Gautos kombinacijos: {CR}, {OP,EP}, {S1,S2} reiškia, kad Kauno HE negalės valdyti užtvankos vartų:

- jei bus pažeisti bėgiai, kuriais važinėja kranai;
- jei nebus elektros energijos tiekimo kranams;
- jei kartu suges abu, Kauno HE vartus valdantys (kilnojantys), kranai.

Šių kombinacijų teisingumą patvirtina ir 2.3.1. poskyryje sudaryta dvejetainė sprendimų diagrama, kurioje per vienetines šakas sujungti mazgai atitinka gautus minimalius kirtimus.

Kiekybiškai analizuojant Kauno HE užtvankos vartų valdymo sistemą įvertinome sistemos gedimo tikimybę. Kiekybinę analizę atlikome keliais būdais: naudojome minimalių kirtimų aibės analitinį gedimo tikimybės vertinimą (2.2.2. poskyris); taikėme dvejetainės sprendimų diagramos (2.7 paveikslas) gedimo tikimybės vertinimą; taip pat sistemos gedimo tikimybę vertinome su programomis „OpenFTA“ (2.2.3. poskyris) ir „DemoITE“ (2.3.5. poskyris). Gautos gedimo tikimybės įverčio reikšmės ( $1,039 \cdot 10^{-4}$ ) visiškai sutapo tarpusavyje. Taigi, Kauno HE užtvankos vartų valdymo sistemos gedimo tikimybė yra  $1,039 \cdot 10^{-4}$ , tai reikštų, kad vartų valdymo sistema apytikriai sugenda vieną kartą per dešimt tūkstančių metų. Bet šios sistemos gedimo padariniai gali būti katastrofiški (2.1 skyrius), t.y. sugedus vartų valdymui, kai jie yra uždaryti, pradėtų kilti Kauno marių vanduo, kuris gali persilieti per užtvankos viršų ir/arba išlaužti užtvanką, tokiu būdu užtvindydamas dalį Kauno miesto ir rajono (Simaityte, 2007).

Darbe patikrinome kaip kinta sistemos gedimo tikimybės įverčio tikslumas, esant skirtingoms bazinių įvykių tikimybėms ir vertinimą atliekant skirtingais skaičiavimo būdais. Panašus tyrimas buvo pateiktas Naujokaičio (2010). Tyrimui panaudojome Kauno HE užtvankos vartų valdymo sistemą. Sistemos gedimo tikimybę vertinome analitiškai pagal (1.6) lygtį bei programomis „OpenFTA“ ir „DemoITE“. Taip pat apskaičiavome sistemos gedimo tikimybės įverčio (1.7) ir (1.8) aproksimacijas. Bazinių įvykių tikimybės 2.3 lentelėje keitėme nuo artimos vienetai iki artimos nuliui, o 2.4 lentelėje bazinių įvykių tikimybės keitėme taip, kad visų minimalių kirtimų tikimybės kistų nuo artimos

vienetui iki artimos nuliui. Visais atvejais (2.3 ir 2.4 lentelėse) įvertintos sistemos gedimo tikimybės, t.y. analitinio vertinimo, programų „OpenFTA“ bei „DemoITE“, įverčiai sutapo. Todėl galime užtikrintai teigti, kad, kai sistema sudaryta iš nedaug komponentų, programos „OpenFTA“ ir „DemoITE“ tiksliai vertina sistemos gedimo tikimybę. Ta taip pat patvirtina teorinė ypatybė, jog bendru atveju dvejetainės sprendimų diagramos pagrindu sudarytas algoritmas, sistemos gedimo tikimybę įvertina tiksliai.

Iširtos gedimo tikimybės aproksimacijos ne visada buvo artimos tiksliai įvertintai sistemos gedimo tikimybei. Kai bazinių įvykių tikimybės artimos vienetai tai (1.8) priešingoji aproksimacija tiksliai įvertina sistemos gedimo tikimybę, o tiesioginė (1.7) aproksimacija tikimybę vertina labai netiksliai, be to, aproksimacijos įvertis gali būti didesnis už vieneta. Kai bazinių įvykių tikimybės artimos nuliui, tai tiesioginė aproksimacija yra mažesnė už priešingąją aproksimaciją ir tiksliau įvertinama sistemos gedimo tikimybę. Todėl gedimo tikimybės vertinimui taikant aproksimacijas būtina atsižvelgti į bazinių įvykių tikimybių dydį. Tikimybės aproksimacijos dažniausiai taikomos, kai gedimų medis sudarytas iš daug bazinių įvykių ir turi daug minimalių kirtimų (Bedford, Cooke, 2007; NEA/SEN/SIN/SMAP, 2006), bet tokiu atveju aproksimacija gali reikšmingai skirtis nuo tikrosios tikimybės vertės. Tad norint tiksliai įvertinti sistemos gedimo tikimybę, patartina taikyti tikslaus vertinimo metodą, paremtą BDD struktūros vertinimu.

Keliais būdais atlikus Kauno HE užtvankos vartų valdymo sistemos gedimų medžio konvertavimą į dvejetainę sprendimų diagramą (teisingumo lentelė, ITE sąlyga, programa „VisBDD“) galime pastebėti, kad pasirinkus tą pačią bazinių įvykių tvarką, gauname tokią pačią BDD. Matome, kad konvertavimas pagrįstas ITE sąlyga yra naudojamas programoje „VisBDD“. Be to, analitinis ITE sąlygos taikymas gedimų medžio konvertavimui yra efektyvesnis nei taikant teisingumo lentelę, nes gauta BDD gali užimti mažiau kompiuterinės atminties vietos, šiuo atveju nereikia saugoti didelės teisingumo lentelės.

Konvertavus Kauno HE užtvankos vartų valdymo sistemos gedimų medį į BDD esant skirtingam bazinių įvykių rangavimui (2.6 skyrius) gauname skirtingas BDD struktūras (5 priedas). Kai baziniai įvykiai ranguojami atsižvelgiant į McMillan (1993) siūlomą rangavimą (1.4.1. poskyris), t.y. rangai iš eilės priskiriami baziniams įvykiams esantiems minimaliuose kirtimuose, neatsižvelgiant į minimalių kirtimų išdėstymą. Matome, kad kai  $CR < EP < OP < S2 < S1$  arba  $CR < OP < EP < S1 < S2$  arba  $OP < EP < CR < S1 < S2$ , tai BDD mazgų skaičius yra 7 (2.5 lentelė), o kai  $S1 < S2 < OP < EP < CR$  arba  $S2 < S1 < EP < OP < CR$  arba  $S1 < S2 < CR < OP < EP$ , tai BDD mazgų skaičius yra 8 (2.5 lentelė). Kai baziniai įvykiai suranguoti neatsižvelgiant į McMillan (1993) siūlomą rangavimą, tai galime gauti BDD struktūrą sudarytą iš 12 mazgų (2.5 lentelė) nors bazinių įvykių yra tik 5. Taigi, konvertuojant

gedimų medį į BDD struktūrą siūlau taikyti rangavimo principą (praplečiant McMillan (1993) rangavimo būdą):

- pirmiausia siūlau išrikiuoti minimalių kirtimų aibės minimalius kirtimus taip, kad minimalūs kirtimai turintys pasikartojančių bazinių įvykių (kai bazinis įvykus gedimų medyje yra naudojamas ne vieną kartą) būtų minimalių kirtimų aibės pradžioje;
- tada iš eilės (pradedant nuo mažiausio minimalaus kirtimo elementų) suranguoti visus bazinius įvykius.

Jei bazinis įvykis yra keliuose minimaliuose kirtimuose, tai rangas jam suteikiamas pirmą kartą jį sutikus minimalių kirtimų aibėje ir toliau jam suteiktas rangas nekinta. Matome, kad išrikiavus minimalių kirtimų aibės kombinacijas  $\{CR\}$ ,  $\{EP,OP\}$ ,  $\{S2,S1\}$  arba  $\{CR\}$ ,  $\{OP,EP\}$ ,  $\{S1,S2\}$  arba  $\{OP,EP\}$ ,  $\{CR\}$ ,  $\{S1,S2\}$ , t.y.  $CR < EP < OP < S2 < S1$  arba  $CR < OP < EP < S1 < S2$  arba  $OP < EP < CR < S1 < S2$ , gauname BDD sudarytą iš 7 mazgų (2.5 lentelė)

Išnagrinėjus programoje „OpenFTA“ (1.3 skyrius) naudojamą (1.6) lygtimi pagrįstą gedimo tikimybės vertinimo algoritmą galime teigti, kad mažų gedimų medžių tikimybės vertinimas yra kiek norima tikslus (1.4 lentelė). Realizuojant šį algoritmą galima pasirinkti tikslumo kriterijaus  $x$  dydį (1.9) nelygybėje ir, tokiu būdu parinkti norimą tikimybės tikslumą. Tikslumo kriterijus  $x$  padeda išvengti (1.6) lygtyje taikomų skaičiavimo tikslumą įtakančių kombinacijų lygių, kurių įtaka vertinamai tikimybei yra palyginti maža. Kadangi 1.4 lentelėje šiuo algoritmu įvertintos gedimo tikimybės (priklausomai nuo pasirinkto tikslumo kriterijaus  $x$  reikšmės) nesiskyrė nuo tikslios tikimybės, tai galime teigti, kad programa „OpenFTA“ pakankamai tiksliai vertina sistemos gedimo tikimybę.

## IŠVADOS

1. Darbe išanalizuota ir susistemintai pateikta sistemos gedimo dažniui suskaičiuoti taikoma metodika leidžia skirtingais būdais (taikant minimalių kirtimų aibę arba dvejetainę sprendimų diagramą) analitiškai įvertinti įvairių sistemų gedimo (arba veikimo) tikimybę.
2. Aprašytas dvejetainės sprendimų diagramos taikymo algoritmas ir išplėtos programinės priemonės leidžia tiksliai įvertinti įvairių sistemų gedimo tikimybę.
3. Sukurta programinė priemonė „DemoITE“ įgalina vizualizuoti sistemą nusakančią ITE sąlygą ir tiksliai vertinti sistemos gedimo tikimybę (sistemos gedimo dažnį per metus).
4. Atlikus kokybinę Kauno HE užtvankos vartų valdymo sistemos analizę nustatytos minimalios sistemos gedimą sukeliančios komponentų kombinacijos (baziniai įvykiai):
  - jei bus pažeisti bėgiai, kuriais važinėja kranai;
  - jei nebus elektros energijos tiekimo kranams;
  - jei kartu suges abu, Kauno HE vartus valdantys (kilnojančios), kranai.
5. Atlikus kiekybinę Kauno HE užtvankos vartų valdymo sistemos analizę tiksliai įvertinta sistemos gedimo tikimybė per metus, kurios įvertis yra  $1,039 \cdot 10^{-4}$ .
6. Pasiūlytas bazinių įvykių rangavimo principas įgalinantis gauti mažiausią galimą BDD struktūrą.

## ŠALTINIAI IR LITERATŪRA

1. Andrews, J. D. and S. J. Dunnett, Event Tree Analysis Using Binary Decision Diagrams, IEEE Trans. Reliability, 49(2), 2000, p.p. 230–238.
2. Andrews, J. D., "To Not or NOT to Not", Proceedings of the 18th International System Safety Conference, 2000.
3. Andrews, J. D., Remenyte, R., Fault Tree Conversion to Binary Decision Diagrams, Proceedings of the 23rd International System Safety Conference, 2005.
4. Andrews, J. D., The use of not logic in fault tree analysis, Quality and Reliability Engineering International, Vol. 17, 2001, pp. 143-150.
5. Augutis, J., Ušpuras E. Technologijų rizika. Lietuvos energetikos institutas, 2006, 247 p.
6. Bedford, T., Cooke, R., Probabilistic Risk Analysis: Foundations and Methods, Cambridge: Cambridge University Press, Fifth printing 2007.
7. Bryant, R. E., Graph-Based Algorithms for Boolean Function Manipulation, IEEE Trans. Computers C-35, no. 8, 1986, p.p. 677-690.
8. Bryant, R. E., Symbolic Boolean manipulation with ordered binary decision diagrams. ACM Computing Surveys, 1992, pp. 293–318.
9. Contini, S., Matuzas, V., ASTRA 3.0: Test Case Report, Luxembourg, Publications Office of the European Union, 2009.
10. Dutuit, Y. ir Rauzy, A., "Exact and Truncated Computations of Prime Implicants of Coherent and Non-Coherent Fault Tree within Aralia", Preprint submitted to Elsevier Preprint, 2001.
11. Epstein, S. and Rauzy, A. Can we trust PRA? Reliability Engineering System Safety, 88(3), 2005, p.p. 195–205.
12. Ibanez-Llano, C., Melendez, E., Nieto F., and Rauzy, A., "Minimal cutsets-based reduction approach for the use of binary decision diagrams on probabilistic safety assessment fault tree models", Journal Risk and Reliability, Vol. 223 Part O, 2009.
13. Įmonės „Lietuvos energija“ administracijos tinklalapis [interaktyvus] [žiūrėta 2010 03]. Prieiga per internetą: <http://www.lpc.lt/lt>.
14. Kumamoto, H. and Henley, E, Probabilistic Risk Assessment and Management for Engineers and Scientists, IEEE Press, Piscataway, NJ, 1996.
15. Lietuvos energija administracijos tinklalapis [interaktyvus] [žiūrėta 2010 03]. Prieiga per internetą: <http://bdd.hpi.uni-potsdam.de/visBDD/index.html>.
16. McMillan, K. L., Symbolic Model Checking. Kluwer Academic Publication, 1993.
17. Meinel, C. and Theobald, T., "Algorithms and Data Structures VLSI Design", Springer-Verlag Berlin Heidelberg, 1998.

18. NASA, Fault Tree Handbook with Aerospace Applications, version 1.1. NASA, August 2002.
19. Naujokaitis, D., "Sistemas gedimo tikimybinis vertinimas", VIII studentų konferencija Taikomoji matematika, 2010.
20. NEA/SEN/SIN/SMAP, "Task Group on the CSNI Safety Margins Action Plan (SMAP)", 2006.
21. OpenFTA, Version 1.0 User Manual, Formal Software Construction Limited, 2005.
22. Programos „OpenFTA“ tinklalapis [interaktyvus] [žiūrėta 2010 01]. Prieiga per internetą: <http://www.openfta.com>.
23. Rauzy, A., "A Brief Introduction to Binary Decision Diagrams". European Journal of Automation, Vol 30, No.8, 1996.
24. Rauzy, A., Dutuit, Y., Exact and truncated computations of prime implicants of coherent and noncoherent fault tree. Reliability engineering and system safety 58, 1997, p.p. 127-144.
25. Rauzy, A., New algorithms for fault tree analysis, Reliability Engineering and System Safety, 1993, pp. 203-211.
26. Remenyte-Prescott, R. and Andrews, J. D., "An Efficient Real-time Method of Analysis for Non-coherent Fault Trees", Quality and Reliability Engineering International, 2009, p.p. 129–150.
27. Sakanashi, H., Highchi, T., Iba, H. and Kakazu, Y., "Evolution of Binary Decision Diagrams for Digital Circuit Design using Genetic Programming", Springer Berlin/Heidelberg, Vol. 1259/1997, 1997.
28. Sharvia, S. Papadopoulos, Y., Non-coherent Modelling in Compositional Fault Tree Analysis, Proceedings of the 17th World Congress, The International Federation of Automatic Control, Seoul, Korea, July 6-11, 2008.
29. Simaityte, J., "Flood discharge control based on risk assessment", Doctoral dissertation, 2007.
30. Vaidogas, E. R., "Techninių sistemų rizikos analizė", Technologija, 2003, 148 p.
31. Vesely, W.E., Goldberg, F.F., Roberts, N.H. and Haasi, D.F., The Fault Tree Handbook, US Nuclear Regulatory Commission, NUREG 0492, 1981.
32. Way, Y. S., Hsia, D. Y., A simple component-connection method for building binary decision diagrams encoding a fault tree, Reliability Engineering and System Safety, 2000, pp. 59-70.
33. Xu, H. and Dugan, J. B., "Combining Dynamic Fault Trees and Event Trees for Probabilistic Risk Assessment", Proceedings of Annual Reliability and Maintainability Symposium, 2006, pp. 214–219.



# 1 PRIEDAS. „OPENFTA“ PROGRAMOS REZULTATAI

## Minimal Cut Sets

=====

Tree : Kauno\_HE\_FT.fta

Time : Thu May 13 10:34:43 2010

Method : Algebraic

No. of primary events = 5

Minimal cut set order = 1 to 5

Order 1:

1) CR

Order 2:

1) EP OP

2) S1 S2

Order 3:

Order 4:

Order 5:

Qualitative Importance Analysis:

| Order | Number |
|-------|--------|
|-------|--------|

-----

|   |   |
|---|---|
| 1 | 1 |
|---|---|

|   |   |
|---|---|
| 2 | 2 |
|---|---|

|   |   |
|---|---|
| 3 | 0 |
|---|---|

|   |   |
|---|---|
| 4 | 0 |
|---|---|

|   |   |
|---|---|
| 5 | 0 |
|---|---|

|     |   |
|-----|---|
| ALL | 3 |
|-----|---|

## Probabilities Analysis

=====

Tree : Kauno\_HE\_FT.fta

Time : Thu May 13 10:36:57 2010

Number of primary events = 5

Number of minimal cut sets = 3

Order of minimal cut sets = 5

Unit time span = 1.000000

Minimal cut set probabilities :

|   |       |               |
|---|-------|---------------|
| 1 | CR    | 1.000000E-004 |
| 2 | EP OP | 3.000000E-006 |
| 3 | S1 S2 | 9.000000E-007 |

Probability of baigiammm level event (minimal cut sets up to order 5 used):

|         |                |                               |
|---------|----------------|-------------------------------|
| 1 term  | +1.039000E-004 | = 1.039000E-004 (upper bound) |
| 2 terms | -3.927000E-010 | = 1.038996E-004 (lower bound) |
| 3 terms | +2.700000E-016 | = 1.038996E-004 (upper bound) |

Exact value : 1.038996E-004

Primary Event Analysis:

| Event | Failure contrib. | Importance |
|-------|------------------|------------|
| CR    | 1.000000E-004    | 96.25%     |
| EP    | 3.000000E-006    | 2.89%      |
| OP    | 3.000000E-006    | 2.89%      |
| S1    | 9.000000E-007    | 0.87%      |
| S2    | 9.000000E-007    | 0.87%      |

## 2 PRIEDAS. SISTEMOS GEDIMŲ MEDŽIO TEISINGUMO LENTELĖ

| CR | EP | OP | S2 | S1 | T |
|----|----|----|----|----|---|
| 0  | 0  | 0  | 0  | 0  | 0 |
| 0  | 0  | 0  | 0  | 1  | 0 |
| 0  | 0  | 0  | 1  | 0  | 0 |
| 0  | 0  | 0  | 1  | 1  | 1 |
| 0  | 0  | 1  | 0  | 0  | 0 |
| 0  | 0  | 1  | 0  | 1  | 0 |
| 0  | 0  | 1  | 1  | 0  | 0 |
| 0  | 0  | 1  | 1  | 1  | 1 |
| 0  | 1  | 0  | 0  | 0  | 0 |
| 0  | 1  | 0  | 0  | 1  | 0 |
| 0  | 1  | 0  | 1  | 0  | 0 |
| 0  | 1  | 0  | 1  | 1  | 1 |

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

### 3 PRIEDAS. DUOMENŲ FAILAS: „ITE\_KAUNO\_HE.TXT“

x[1] 0.0001

x[2] 0.003

x[3] 0.001

x[4] 0.003

x[5] 0.0003

## 4 PRIEDAS. PROGRAMOS „DEMOITE“ REALIZACIJOS FAILAI

### *ITE\_BDD.h*

```
//-----
/*          Programa ITE
           Darius Naujokaitis
           Kauno Technologijos Universitetas,
           Fundamentaliųjų mokslų fakultetas,
           FMMM-8 grupė.

Programa skirta tiksliai vertinti sistemos, pateiktos ITE struktūra (x[i],1,0),
gedimo tikimybę, bei parodyti ITE struktūros BDD.

           2010 metai.

*/
//-----

#ifndef ITE_BDDH
#define ITE_BDDH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Dialogs.hpp>
#include <ExtCtrls.hpp>
#include <string>
#include <fstream>
#include <iomanip>
using namespace std;
//-----
class TBDD{
public:
    char *pav;
    double tik;
    TBDD *nulis;
    TBDD *vienas;
```

```

    TBDD() { pav = new char[4]; pav = "\0"; tik = 0; nulis = vienas = NULL;};
    ~TBDD() { delete []pav; tik = 0; nulis = vienas = NULL; };
};

```

```

struct Tikimybes{
    double tik;
    char *pav;
};

```

```

class TForm1 : public TForm
{
__published:    // IDE-managed Components
    TEdit *Edit1;
    TLabel *Label1;
    TButton *Button1;
    TLabel *Label2;
    TButton *Button2;
    BaigiammenDialog *OpenDialog1;
    TMemo *Memo1;
    TLabel *Label3;
    TLabel *Label4;
    TLabel *Label5;
    TLabel *Label6;
    TImage *Image1;
    void __fastcall Button1Click(TObject *Sender);
    void __fastcall Button2Click(TObject *Sender);
private:    // User declarations
public:    // User declarations
    TBDD *ite;
    char *bdd;
    Tikimybes *prob;
    int ii;
    double gedimas;

```

```

AnsiString DuomenuFailas;
TBDD * NuskaitytiITE(char *eil);
int SkliaustuSutapimas(char *e);
String Spausdinti(TBDD * nod);
void NuskaitytiTikimybes();
AnsiString DuomenuFailoVardas();
void NuskaitytiFaila();
double IvykioTikimybe(char *pv);
double GedimoTikimybe(TBDD * nod);
void BraizytiBDD(TBDD * nod, int x, int y);
void Apskritimas(char *pv, int x, int y);

__fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif

ITE_BDD.cpp
//-----
/*          Programa ITE
          Darius Naujokaitis
          Kauno Technologijos Universitetas,
          Fundamentaliųjų mokslų fakultetas,
          FMMM-8 grupė.
          Programa skirta tiksliai vertinti sistemos, pateiktos ITE struktūra (x[i],1,0),
          gedimo tikimybę, bei parodyti ITE struktūros BDD.
          2010 metai.
*/
//-----

#include <vcl.h>
#pragma hdrsbaigiammm

```

```

#include "ITE_BDD.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    Image1->Canvas->Brush->Color = clWhite;
}
//-----
void TForm1::BraizytiBDD(TBDD * nod, int x, int y){
    TBDD *vx = nod;
    int yra, bus1, bus0;

    if ((vx->pav[0] != '1') && (vx->pav[0] != '0')) {
        yra = String(vx->pav[2]).ToInt();
        if ((vx->vienas != NULL) && (vx->vienas->pav[0] != '1') && (vx->vienas->pav[0] != '0'))
        {
            bus1 = String(vx->vienas->pav[2]).ToInt();
            BraizytiBDD(vx->vienas,x+(50*(bus1-yra)),y+(70*(bus1-yra)));
        }
        if ((vx->nulis != NULL) && (vx->nulis->pav[0] != '1') && (vx->nulis->pav[0] != '0')) {
            bus0 = String(vx->nulis->pav[2]).ToInt();
            BraizytiBDD(vx->nulis,x-(50*(bus0-yra)),y+(70*(bus0-yra)));
        }
        Image1->Canvas->MoveTo(x+25,y+25);
        Image1->Canvas->Pen->Color = clRed;
        if(vx->vienas->pav[0] == '1')
            Image1->Canvas->LineTo((Image1->Width*3/4+10),(Image1->Height)-30);
        else if (vx->vienas->pav[0] == '0')
            Image1->Canvas->LineTo((Image1->Width/4+10),(Image1->Height)-30);
        else Image1->Canvas->LineTo(x+15+50*(bus1-yra),y+70*(bus1-yra));
    }
}

```

```

Image1->Canvas->MoveTo(x+5,y+25);
Image1->Canvas->Pen->Color = clBlue;
    if(vx->nulis->pav[0] == '1')
        Image1->Canvas->LineTo((Image1->Width*3/4)+10,(Image1->Height)-30);
    else if (vx->nulis->pav[0] == '0')
        Image1->Canvas->LineTo((Image1->Width/4)+10,(Image1->Height)-30);
    else Image1->Canvas->LineTo(x+15-50*(bus0-yra),y+70*(bus0-yra));
Image1->Canvas->MoveTo(x,y);
Image1->Canvas->Pen->Color = clBlack;
Apskritimas(vx->pav, x, y);
}
}
//-----
void TForm1::Apskritimas(char *pv, int x, int y){
    if (pv[0] == '1' || pv[0] == '0') {
        Image1->Canvas->Rectangle(x,y,x+20,y+25);
        Image1->Canvas->TextOutA(x+5,y+7,String(pv));
    } else {
        Image1->Canvas->Ellipse(x,y,x+30,y+30);
        Image1->Canvas->TextOutA(x+5,y+7,String(pv));
    }
    Image1->Repaint();
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    int ilg = Edit1->Text.Length();
    char *e;
    String sp = "\0";
    ite = NULL;
    e = new char[ilg];

    Image1->Canvas->Rectangle(0,0,Image1->Width,Image1->Height);
    if (ilg > 0) strncpy(e, Edit1->Text.c_str(), ilg);

```



```

else e[0] = '\0';
if (strlen(e) > 0) {
    NuskaitytiTikimybes();
    if (DuomenuFailas != "") ite = NuskaitytiITE(e);
}
else ShowMessage("Įveskite ITE struktūrą (x[i],1,0)!");
if (ite != NULL){
    sp = Spausdinti(ite);
    gedimas = GedimoTikimybe(ite);
    BraizytiBDD(ite,(Image1->Width/2),10);
    Apskritimas("1",(Image1->Width*3/4),(Image1->Height)-30);
    Apskritimas("0",(Image1->Width/4),(Image1->Height)-30);
    Label2->Caption = sp;
    Label5->Caption = FloatToStrF(gedimas, ffExponent, 15, 3);
    ite->~TBDD();
    delete []e;
}
else if (DuomenuFailas != "") ShowMessage("Neatitinka ITE formatas (x[i],1,0)!");
}
//-----
double TForm1::GedimoTikimybe(TBDD * nod){
    TBDD *vx = nod;
    double p, f1, f0;
    p = f1 = f0;
    if ((vx->pav[0] == '1') || (vx->pav[0] == '0')) {
        p = vx->tik;
    } else {
        if (vx->vienas != NULL)
            f1 = vx->tik * GedimoTikimybe(vx->vienas);
        if (vx->nulis != NULL)
            f0 = (1 - vx->tik) * GedimoTikimybe(vx->nulis);
        p = f1 + f0;
    }
    return p;
}

```

```

}
//-----
String TForm1::Spausdinti(TBDD * nod){
    TBDD *vx = nod;
    String e, e1, e0, israiska;
    e.Delete(0,e.Length());
    e1.Delete(0,e1.Length());
    e0.Delete(0,e0.Length());
    israiska.Delete(0,israiska.Length());
    if ((vx->pav[0] == '1' || (vx->pav[0] == '0')) {
        israiska = String(vx->pav);
    } else {
        e = String(vx->pav);
        if (vx->vienas != NULL)
            e1 = Spausdinti(vx->vienas);
        if (vx->nulis != NULL)
            e0 = Spausdinti(vx->nulis);
        israiska += AnsiString::StringOfChar('<',1) + e + AnsiString::StringOfChar(',',1) + e1 +
AnsiString::StringOfChar(',',1) + e0 + AnsiString::StringOfChar('>',1);
    }
    return israiska;
}
//-----
void TForm1::NuskaitytiTikimybes(){
    DuomenuFailas = DuomenuFailoVardas();
    if (DuomenuFailas == "")
        ShowMessage("Pasirinkite duomenų failą!");
    else {
        Memo1->Lines->LoadFromFile(DuomenuFailas);
        Label6->Caption = "Duomenu failas\n" + DuomenuFailas;
        NuskaitytiFaila();
    }
}
//-----

```

```

void TForm1::NuskaitytiFaila(){
    int i = 0;
    int j = 0;
    char e[4];
    double ee;
    ifstream fd(DuomenuFailas.c_str());
    while (!fd.eof()){
        i++;
        fd >> e >> ee;
    }
    fd.close();
    if (i == 0) ShowMessage("Duomenų failas tuščias!");
    else {
        prob = new Tikimybes[i];
        ifstream fs(DuomenuFailas.c_str());
        while (!fs.eof()){
            prob[j].pav = new char[4];
            fs >> prob[j].pav >> prob[j].tik;
            prob[j].pav[4] = '\0';
            j++;
        }
        fs.close();
    }
    ii = j;
}

//-----
AnsiString TForm1::DuomenuFailoVardas(){
    OpenFileDialog->Filter = "Text failai (*.txt)*.txt";
    if (OpenFileDialog->Execute() && FileExists(OpenFileDialog->FileName))
        return OpenFileDialog->FileName;
    else return "";
}

//-----
TBDD * TForm1::NuskaitytiITE(char *eil){

```

```

TBDD *nod = NULL;
char *mazg;
char *e1, *e0, *ec, *s1, *s0;
int ilg, ilg1, ilg0;
ilg = ilg1 = ilg0 = 0;
ilg = SkliaustuSutapimas(eil);
if (ilg > 0){
    ec = new char [ilg-1];
    strcpy(ec, eil);
    strcpy(ec, ec+1);
    ec[ilg-1] = '\0';
    mazg = new char[4];
    strncpy(mazg, ec, 4);
    mazg[4] = '\0';
    nod = new TBDD();
    nod->pav = new char[4];
    strcpy(nod->pav,mazg);
    nod->pav[4] = '\0';
    nod->tik = IvykioTikimybe(nod->pav);

    strcpy(ec, ec+5);
    ilg1 = SkliaustuSutapimas(ec);
    s1 = new char[ilg1];
    strncpy(s1, ec, ilg1);
    s1[ilg1] = '\0';

    if ((s1[0] == '1') || (s1[0] == '0')) {
        nod->vienas = new TBDD();
        nod->vienas->pav = new char[4];
        strcpy(nod->vienas->pav,s1);
        nod->vienas->pav[1] = '\0';
        if (s1[0] == '1') nod->vienas->tik = 1;
    } else {
        nod->vienas = NuskaitytiITE(s1);
    }
}

```

```

    }

    strcpy(ec, ec+(1+ilg1));
    ilg0 = SkliaustuSutapimas(ec);
    s0 = new char[ilg0];
    strncpy(s0, ec, ilg0);
    s0[ilg0] = '\0';

    if ((s0[0] == '0') || (s0[0] == '1')) {
        nod->nulis = new TBDD();
        nod->nulis->pav = new char[4];
        strcpy(nod->nulis->pav,s0);
        nod->nulis->pav[1] = '\0';
        if (s0[0] == '1') nod->nulis->tik = 1;
    } else {
        nod->nulis = NuskaitytiITE(s0);
    }
} else {
    return nod;
}
delete []mazg;
delete []ec;
delete []s1;
delete []s0;
return nod;
}
//-----
double TForm1::IvykioTikimybe(char *pv){
    for(int i = 0; i < ii; i++){
        if (!strcmp(prob[i].pav,pv)) return prob[i].tik;
    }
    String pr = "[vykio " + String(pv) + " tikimybės nėra duomenų faile!";
    ShowMessage(pr);
    return 0;
}

```

```
}  
//-----  
int TForm1::SkliaustuSutapimas(char *e){  
    int at = 0;  
    int uz = 0;  
    int i = 0;  
    if (e[0] == '(') {  
        at = 1;  
        while (at > uz && i < strlen(e)) {  
            i++;  
            if (e[i] == '(') at++;  
            if (e[i] == ')') uz++;  
        }  
        i++;  
        if (at != uz) i = 0;  
    } else i = 0;  
    if (e[0] == '1') i = 1;  
    if (e[0] == '0') i = 1;  
    if ((at*9 + 1) != i) i = 0;  
    return i;  
}  
//-----  
void __fastcall TForm1::Button2Click(TObject *Sender)  
{  
    Close();  
}  
//-----
```

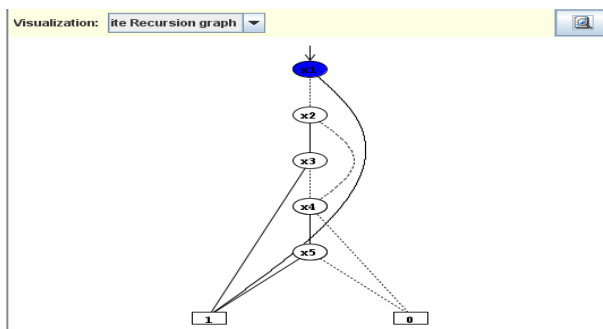
## 5 PRIEDAS. BAZINIŲ ĮVYKIŲ RANGAVIMAS

Šiame priede pateikta pirmose dviejose eilutėse bazinių įvykių tvarka (rangavimas), toliau prie tokio rangavimo gauta ITE sąlyga, nubraižytas BDD grafas ir pateiktas BDD mazgų kiekis.

Kai  $x[1] < x[2] < x[3] < x[4] < x[5]$ ,

t.y.  $CR < EP < OP < S2 < S1$ .

Tai  $R = (x[1], 1, (x[2], (x[3], 1, (x[4], (x[5], 1, 0), 0)), (x[4], (x[5], 1, 0), 0)))$

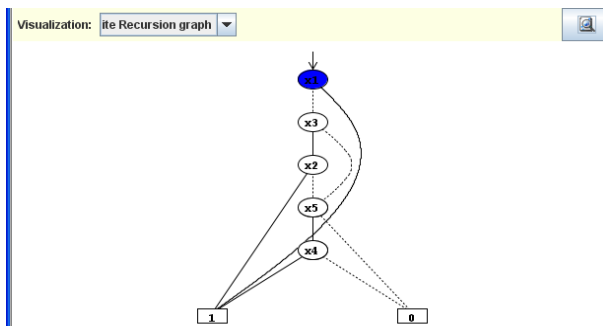


Mazgų skaičius: 7.

Kai  $x[1] < x[3] < x[2] < x[5] < x[4]$ ,

t.y.  $CR < OP < EP < S1 < S2$ .

Tai  $R = (x[1], 1, (x[3], (x[2], 1, (x[5], (x[4], 1, 0), 0)), (x[5], (x[4], 1, 0), 0)))$

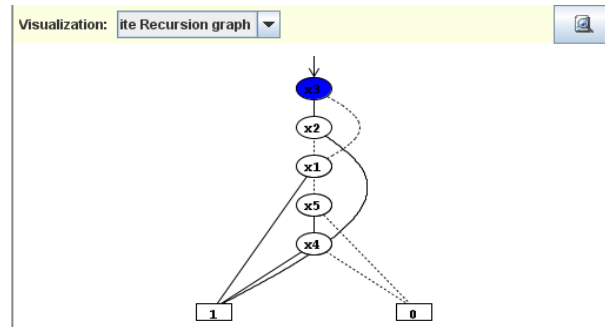


Mazgų skaičius: 7.

Kai  $x[3] < x[2] < x[1] < x[5] < x[4]$ ,

t.y.  $OP < EP < CR < S1 < S2$ .

Tai  $R = (x[3], (x[2], 1, (x[1], 1, (x[5], (x[4], 1, 0), 0))), (x[1], 1, (x[5], (x[4], 1, 0), 0)))$

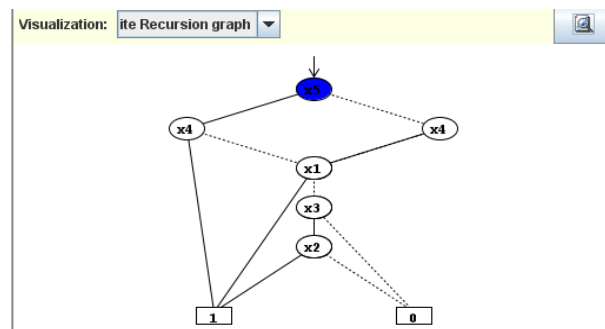


Mazgų skaičius: 7.

Kai  $x[5] < x[4] < x[1] < x[3] < x[2]$ ,

t.y.  $S1 < S2 < CR < OP < EP$ .

Tai  $R = (x[5], (x[4], 1, (x[1], 1, (x[3], (x[2], 1, 0), 0))), (x[4], (x[1], 1, (x[3], (x[2], 1, 0), 0))), (x[1], 1, (x[3], (x[2], 1, 0), 0)))$

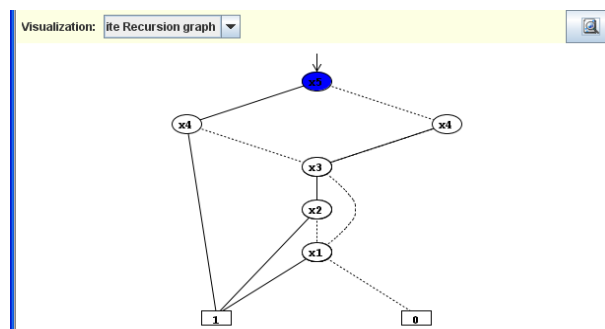


Mazgų skaičius: 8.

Kai  $x[5] < x[4] < x[3] < x[2] < x[1]$ ,

t.y.  $S1 < S2 < OP < EP < CR$ .

Tai  $R = (x[5], (x[4], 1, (x[3], (x[2], 1, (x[1], 1, 0), (x[1], 1, 0))), (x[4], (x[3], (x[2], 1, (x[1], 1, 0), (x[1], 1, 0))), (x[3], (x[2], 1, (x[1], 1, 0), (x[1], 1, 0))))$



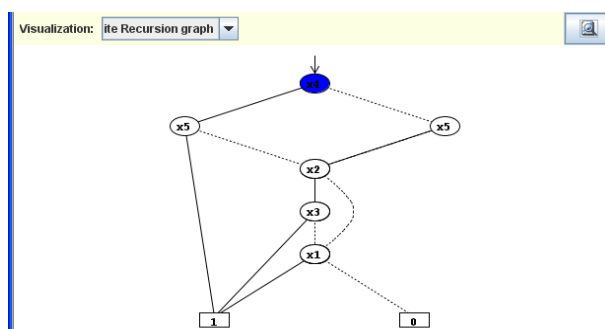
Mazgų skaičius: 8.



Kai  $x[4] < x[5] < x[2] < x[3] < x[1]$ ,

t.y.  $S2 < S1 < EP < OP < CR$ .

Tai  $R = (x[4], (x[5], 1, (x[2], (x[3], 1, (x[1], 1, 0)), (x[1], 1, 0))), (x[5], (x[2], (x[3], 1, (x[1], 1, 0)), (x[1], 1, 0)), (x[2], (x[3], 1, (x[1], 1, 0)), (x[1], 1, 0)))$

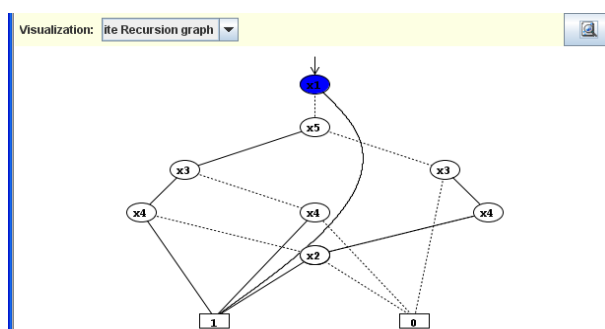


Mazgų skaičius: 8.

Kai  $x[1] < x[5] < x[3] < x[4] < x[2]$ ,

t.y.  $CR < S1 < OP < S2 < EP$ .

Tai  $R = (x[1], 1, (x[5], (x[3], (x[4], 1, (x[2], 1, 0)), (x[4], 1, 0)), (x[3], (x[4], (x[2], 1, 0), (x[2], 1, 0)), 0)))$



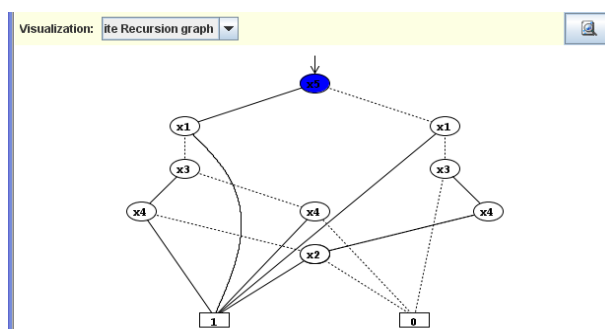
Mazgų skaičius: 10.

Kai  $x[5] < x[1] < x[3] < x[4] < x[2]$ ,

t.y.  $S1 < CR < OP < S2 < EP$ .

Tai

$R = (x[5], (x[1], 1, (x[3], (x[4], 1, (x[2], 1, 0)), (x[4], 1, 0))), (x[1], 1, (x[3], (x[4], (x[2], 1, 0), (x[2], 1, 0)), 0)))$

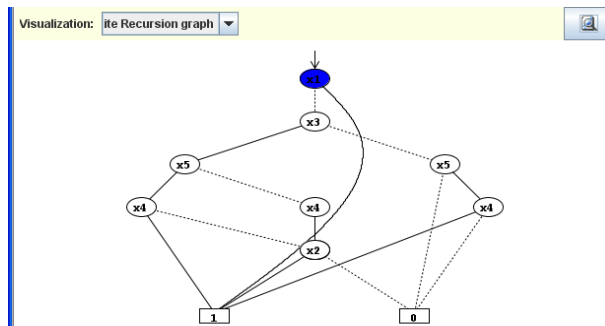


Mazgų skaičius: 11.

Kai  $x[1] < x[3] < x[5] < x[4] < x[2]$ ,

t.y.  $CR < OP < S1 < S2 < EP$ .

Tai  $R = (x[1], 1, (x[3], (x[5], (x[4], 1, (x[2], 1, 0))), (x[4], (x[2], 1, 0), (x[2], 1, 0))), (x[5], (x[4], 1, 0), 0))$

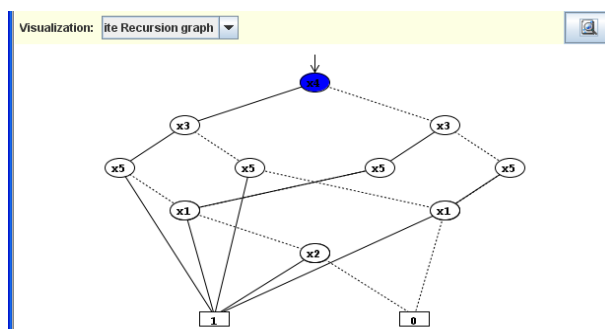


Mazgų skaičius: 10.

Kai  $x[4] < x[3] < x[5] < x[1] < x[2]$ ,

t.y.  $S2 < OP < S1 < CR < EP$ .

Tai  $R = (x[4], (x[3], (x[5], 1, (x[1], 1, (x[2], 1, 0))), (x[5], 1, (x[1], 1, 0))), (x[3], (x[5], (x[1], 1, (x[2], 1, 0), (x[1], 1, (x[2], 1, 0))), (x[5], (x[1], 1, 0), (x[1], 1, 0))))$



Mazgų skaičius: 12.