SOFTWARE COMPLEX
FOR AUTOMATED STRUCTURAL LOGIC MODELING AND CALCULATION
OF RELIABILITY AND SAFETY MEASURES OF CONTROL SYSTEMS

TUTORIAL
(SC ARBITR, version 1.0.1)

A. Strukov
I. Mozhaeva

Saint-Petersburg
2025

Table of Contents

INTRODUCTION

The tutorial was developed for conducting classes in the discipline "Development of emergency plans" using the SC ARBITR.

The discipline is studied during the implementation of the main educational program 20.04.01_12 "Emergency preparedness and response" (international educational program) in the direction of training (specialty) 20.04.01 "Technospheric safety".

The main tasks in the field of industrial safety at the present stage of development are

- introduction of a risk-based approach when organizing activities in the field of industrial safety;
- development of methods for analyzing and assessing the risks of accidents at industrial facilities;
- development and implementation of information technologies that allow solving complex problems of quantitative and qualitative risk analysis.

Considering risk as a combination of the probability of an event causing harm and the severity of this harm, this tutorial focuses on the study of methods for assessing the probability of a hazardous event. When analyzing the hazards associated with failures of technical devices, leak detection systems, industrial automation and control systems (IACS), and safety instrumental systems (SIS), it is recommended to analyze the technical risk, the indicators of which are determined by the appropriate methods of reliability theory. At the same time, methods for calculating the reliability of technical systems are recommended to be combined with methods for modeling accidents and quantitative assessment of the risk of accidents.

Among the many risk analysis methods used in practice, such methods as the analysis of reliability block diagrams, fault trees and event trees are singled out. In the course of performing practical tasks, students acquire practical skills in working in the software ARBITR. SC ARBITR is a program certified by the regulator

Rostekhnadzor for solving the problems of analyzing the reliability and safety of complex technical devices at hazardous production facilities, including nuclear facilities.

SC ARBITR is domestic software that implements a logical-probabilistic method based on the use of the mathematical apparatus of Boolean algebra and probability theory. In the textbook sections 1-4 describe the methodology for obtaining primary skills in working in the PC ARBITR.

Section 5 is designed for four lessons. The first lesson is devoted to the acquisition by trainees of primary skills in working in the software environment of the SC ARBITR, modeling the reliability of simple structures. The second lesson is devoted to modeling the reliability of bridge circuits and network structures. The third lesson is aimed at a deeper study of methods for analyzing the reliability of complex technical systems. The fourth lesson is devoted to the practical application of risk analysis methods for fault trees, event trees and their combinations in industrial safety problems.

To consolidate the educational material at the end of the lessons, tasks are formulated.

Appendix A provides basic terms and definitions of dependability theory. Reliability theory is based on the assumption that the time between failures of a product is a random variable, therefore many concepts from probability theory are used here – distribution function, probability density, risk function (failure rate). The description of the main indicators of reliability and availability is given.

Appendix B provides basic information about the method of reliability analysis widely used in engineering practice – the method of reliability block diagrams (RBD). The RBD method assumes that circuit elements (blocks) can be in two states – operable or failure state. The structure that is analyzed using RBD can also be in two states: either it is operational and performs the specified functions, or it is in a state of failure and cannot perform the specified functions. Therefore, to model the

system properties of the structure, the apparatus of mathematical logic is used – Boolean algebra.

The logical-probabilistic method, which underlies the SC ARBITR, combines the main theorems of probability theory and Boolean algebra.

Appendices C and D describe methods for analyzing fault trees (FTA) and event trees (ETA), which also implement a logical-probabilistic approach in solving problems of risk analysis of technical systems. The rules for building visual models are given, the description of the main logical operators used in FTA and ETA is given. Examples of building fault trees for analyzing the reliability of the lighting system and fire detection system are given. A technique for using equivalent nodes is described. An example of solving the problem of risk and efficiency analysis is considered. Examples of solving problems from the field of functional safety and scenario modeling using ETA are shown.

Appendix E describes the different importance scores for the structure elements, and provides an example of calculating the importance scores for a bridge circuit using FTA. Appendix F provides background material on common cause failure (CCF). The causes causing CCF and methods for modeling the reliability of structures taking into account CCF are described. An example of calculating the functional safety index of a duplicated channel of a safety instrumental system (SIS) is considered.

Abbreviations and symbols

SC ARBITR	Software Complex for automated structural logic modeling and calculation of reliability and safety measures of control system ARBITR
FIS	Functional Integrity Scheme
LF	Logical Function
PF	Probability Function
LCF	Logic Criterion of Functioning
LC	Logic Criteria
DNF	Disjunctive Normal Form
RBD	Reliability Block Diagram
PRA	Probability Risk Analysis
FT	Fault Tree
FTA	Fault Tree Analysis
ET	Event Tree
ETA	Event Tree Analysis
Pr (P)	Probability
R	Reliability
Q	Failure: 1–Pr(P); 1–R
Kg	Availability factor
λ	Failure rate
MCS	Minimal Cut Set
MP	Minimal Path
N_{MP}	Number of Minimal Paths
N_{MCS}	Number of Minimal Cut Sets
$N_{elem.Imp.max}$	Numbers of elements with the maximum importance value
MTTF	Mean Time To Failure
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
MRT	Mean Repair Time
RIR	Risk Increase Ratio
RRR	Risk Reduction Ratio
IACS	Industrial Automation and Control Systems
SIS	Safety Instrumental Systems

1 Basics of the software operation. Modeling of simple structures

The purpose of the part 1: to study the basics of working with the SC ARBITR, the main graphic elements used in the construction of the FIS. Learn to set the initial data of elements, modeling and calculation parameters and calculation modes.

1.1 Basics of the software operation

1.1.1 Software startup

The SC ARBITR is launched using the shortcut "SC ARBITR" (<Start>SC ARBITR) or (<Start><Programs><SPIK SZMA>SC ARBITR). The icons are shown in Figure 1.

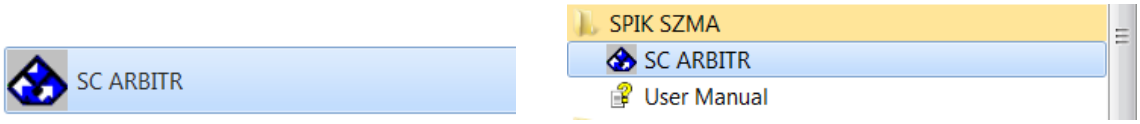


Figure1 – The SC ARBITR Startup Icons

After the Software is running, the SC ARBITR's Main window is opened (Figure 2).

A graphical tool for modeling the properties of the objects under study is the Functional Integrity Scheme (FIS).

The FIS is a directed weighted graph consisting of a set of nodes and a set of edges. The functional node of the FIS is characterized by the probability of the event realization modeled by this node. A dummy node is used to display complex logical connections and relationships between elements in the FIS and is a logical unit (I).

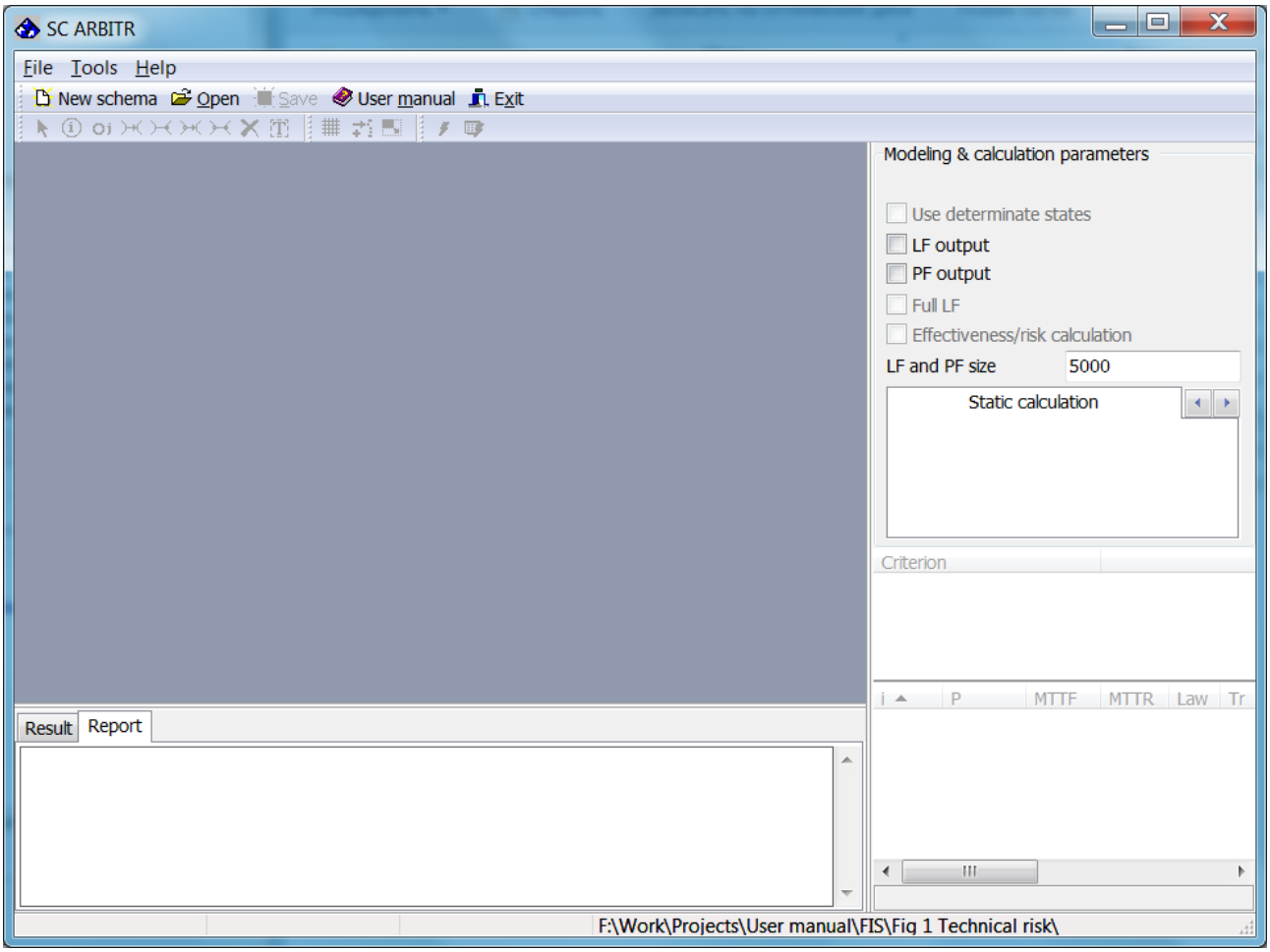
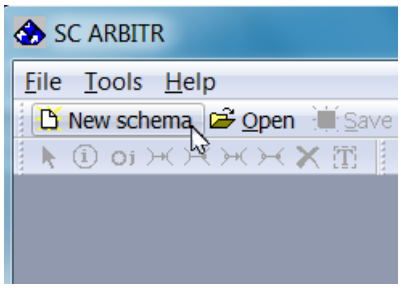


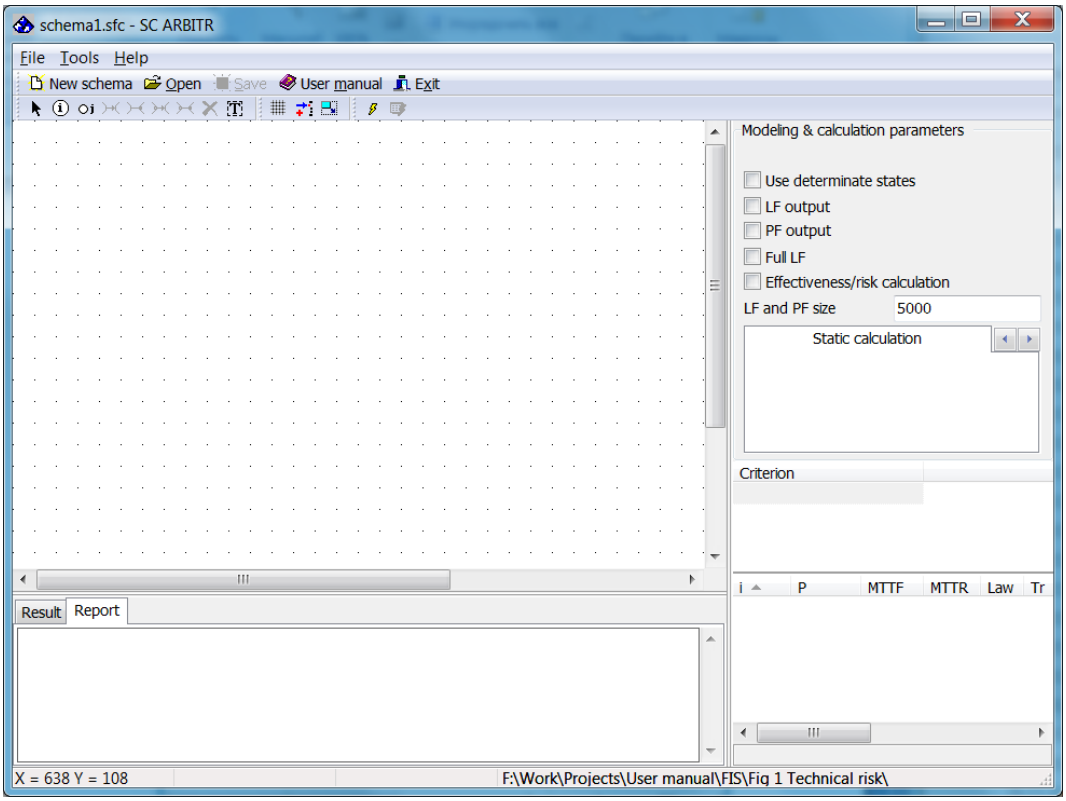
Figure 2 – SC ARBITR's main window

After launching the SC ARBITR, you must perform the following actions:

1 To build a new FIS, press the "New schema" button in the upper left part of the SC ARBIRT's window (Figure 3, a). The SC interface in the "New schema" mode is shown in Figure 3, b.



a



b

Figure 3 – Creation of a new FIS

2 To open an existing scheme – a file with the ".sfc" extension – you must click the "Open" button and select the desired file from the appropriate folder.

After creating/opening a FIS, performing modeling and calculations, the SC interface takes the form shown in Figure 4

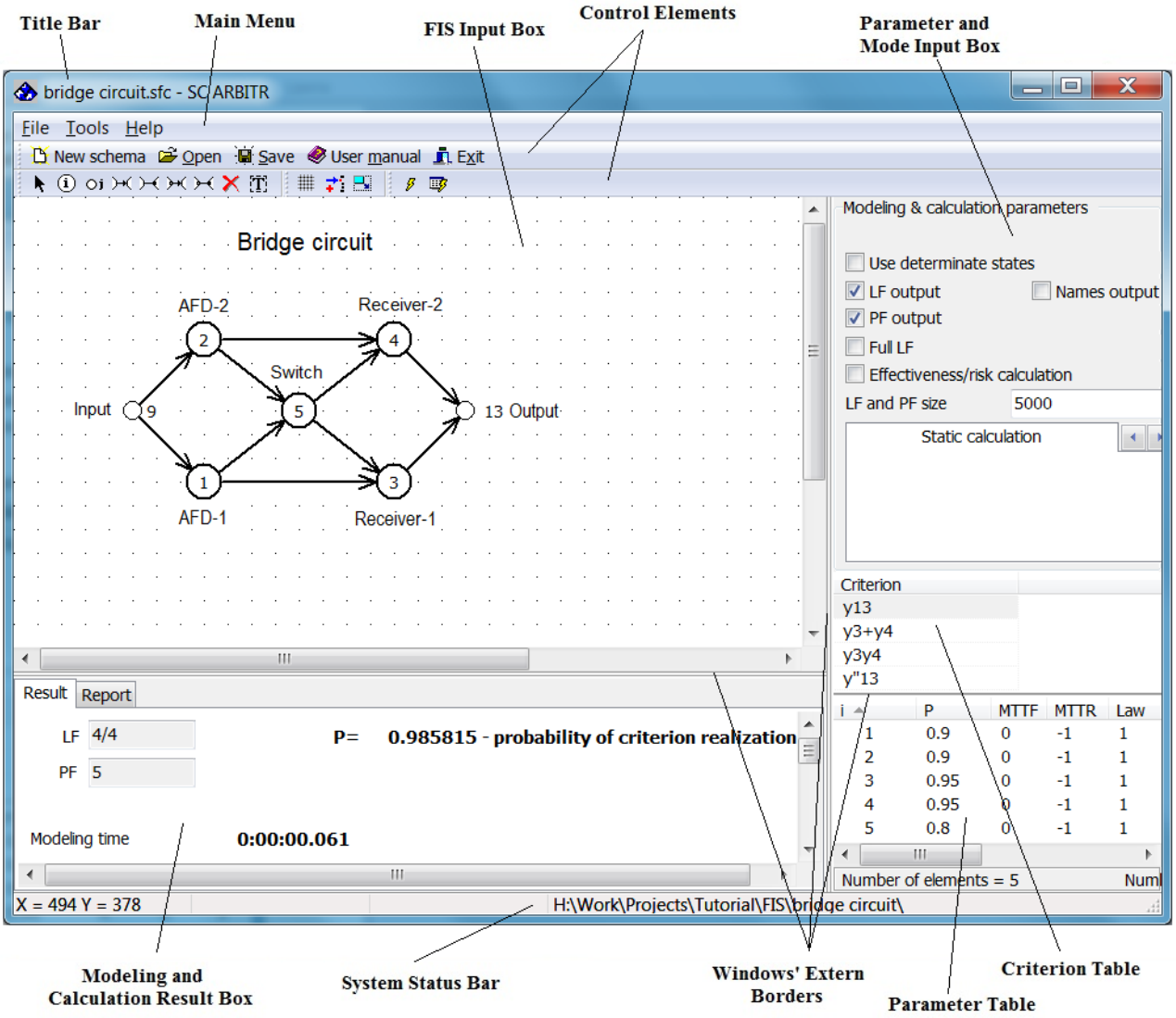


Figure 4 – The SC ARBITR's Main Window

The Software's Main window includes the following four segments:

1 Main box, located in the upper part of the Complex's Main window, consists of four bars:

- Title bar.
- Main Menu bar.
- Two control element toolbars.

2 The system's FIS Input box.

3 Box for elements' parameters input and automated modeling mode setting.

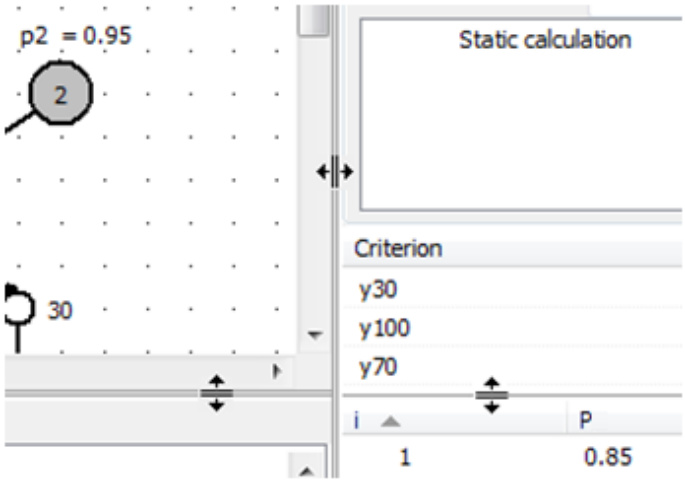
4 Modeling and calculation results output box.

1.1.2 Window Resize

To resize the Main window, drag its corner or edge. Move the pointer to the window's edge until the pointer changes to the double-headed arrow: up/down – for vertical resize, right/left – for horizontal resize. Then left holding the mouse button down, drag the border in the desired direction. Release the mouse button to fix the new window size.

The Main window is resized in two directions at once while dragging any of its corners. The pointer changes to double-headed diagonal arrow.

The Software has an option to resize vertical and horizontal borders of FIS Input Box, Parameter Input Box, and Results Output Box (Figure 4). These borders are displayed in crimson. To resize a window point to the window's border. When the cursor becomes a two-headed arrow (crHSplit), drag the window's border (right/left, up/down) to the size you want.



The scroll bar is used to guide through the contents of a window.

1.1.3 Basic Control Components

The title bar is located at the top of the Main Window (Figure 4). After the Software starts the Title bar displays the program name – **"SC ARBITR".** When the FIS is developed, saved, or opened, the Title changes into *{FIS name}* SC ARBITR. Three standard control buttons (minimize, maximize, close a window) are located at the top right-hand corner.

 – Main window Close button that closes an open window (exits the Complex).

 – This button is displayed when the Main window has a standard medium size. Click this button to expand the window to the full screen size.

 – This button is displayed when the Main window is at its full screen size. Click this button to return the window to its standard medium size. Another way to minimize/maximize the Main window is to double-click on the title bar.

 – Minimize Main window button.

The main menu bar is located under the title bar. Two toolbars with shortcuts are located under the main menu bar. These shortcuts partly duplicate the Main menu, options, ensure the FIS graph input, the LCF input, automated modeling & calculation startup.

1.1.4 The Software's Main Menu

The Complex's Main menu includes the following three items:

- File – operations with project files;
- Tools – auxiliary utilities for calculating the parameters of elements;
- Help – SC ARBITR's user manual.

Each Main menu's item can become active: point to the item and left click the mouse. Commands list of the menu item is opened. Commands lists of the main menu's items are shown in Figure 5.

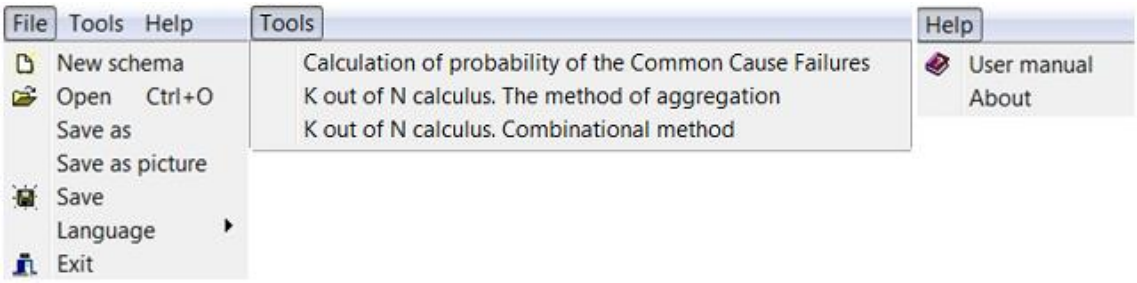


Figure 5 – Main Menu Commands

Current available commands are shown in regular type, while non-available commands are shown in shaded type. To move through the submenu list, select and call the command, use the mouse or keyboard up/down buttons.

Another way to call the menu commands is to hold down the "Alt" key and at the same time press shortcut letter key underlined letter in each menu item (Figure 5).

1.1.4.1 File Menu Commands

This menu item includes a set of file operations options.

File menu options are shown in Figure 5. On the right are the shortcut keys to call the commands without using the mouse.

File menu options' functions are shown in Table 1.

Table 1 – File menu options' functions

Option	Function
New schema	Opens a new FIS Input box (Figure 1) to develop a new FIS graph
Open Ctrl-O	Opens the standard dialog box **Open File** for the already developed and saved FIS opening
Save as	Opens the standard dialog box **Save As** for the user's file name specification and developed FIS saving
Save as picture	"FIS input box" image is saved in the .bmp file
Save	Saves changed FIS in the file with already set filename
Language	Choice of language: English Russian
Exit	Terminates the Software

1.1.4.2 Tools Menu Commands

This menu item includes a set of commands for calling auxiliary tools for element parameters calculation. These commands are shown in Table 2.

Table 2 – Tools menu options' functions

Option	Function
Calculation of probability of the common cause failures	Opens the dialog box for probabilistic parameters calculation of three standard models of common cause failures of groups of elements (Beta factor, Multiple Greek letters, and Alpha factor)
"K out of N" (KooN) calculus. Method of aggregation	Opens the dialog box for probabilistic parameters calculation of parent homogeneous combinatorial subsystems KooN (K/N) using the aggregation method
"K out of N" calculus. Combinational method	Opens the dialog box for probabilistic parameters calculation of parent heterogeneous combinatorial subsystems KooN (K/N) using the combinational method

1.1.4.3 Help Menu Commands

This menu item includes a set of SC ARBITR's help information options. These options are shown in Table 3.

Table 3 – Help menu options' functions

Option	Function
User manual	Opens the Complex's help information
About	Opens the message window About application SC ARBITR

1.1.5 Backup Main Menu Bar

This Bar is in the third line of the Main window (Figure 4) and is shown in Figure 6.

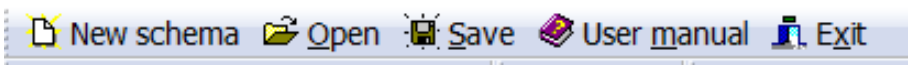


Figure 6 – File menu options shortcuts

These shortcut keys duplicate the File menu options (Figure 5 and item No. 1.4.1).

1.1.6 Toolbar (Command Shortcuts)

The shortcut keys toolbar is in the fourth line of the Main window (Figure 4). These shortcut keys are used for the FIS graph input and correction. The shortcut keys functions are shown in Table 4.

Table 4 – The shortcut keys functions

Key	Help	Function	Mode
	Select	Sets the graph mode Select	Select Mode
	Function Node	Sets the graph mode for the function node input	Node Mode
	Dummy Node	Sets the graph mode for the dummy node input	
	OR edge	Sets the graph mode for input of the edge "OR" between nodes	Edge Mode
	AND edge	Sets the graph mode for input of the edge "AND" between nodes	
	NOT-OR edge	Sets the graph mode for input of the edge "NOT-OR" between nodes	
	NOT-AND edge	Sets the graph mode for input of the edge "NOT-AND" between nodes	
	Delete	Sets the graph mode for the FIS's objects deletion	Deletion Mode
	Text	Sets the graph mode for the explanatory text input	Text Mode
	Show Grid	Opens the grid mode for the object's placement within the FIS input box	–
	Load Background Image	Opens the window for the selection of the file with background image	
	Change Window's Extent	Opens the window for the FIS input box resizing	
	Modeling and calculation	Automated generation of the logic function of system availability, polynomial of probabilistic function and calculation of system reliability and safety parameters	–
	Calculate	System reliability and safety parameters calculation	–

1.1.7 System Status Bar

The Status bar located at the very bottom of the Main window is shown in Figure 7.

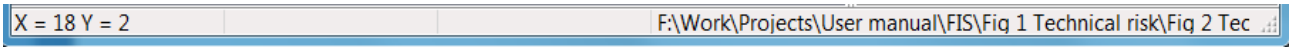


Figure 7 – The Software's Status bar

The Status bar consists of 4 sections:

- In the first section, the mouse cursor's point coordinates within the FIS input box are displayed. Point coordinates are displayed in pixels. Point of origin is in the left-hand top corner of the FIS input box. When the mouse is moved (within the FIS input box), point coordinates change.
- In the second section, current mode of the FIS graph operation is displayed. It is defined by the last pressed shortcut key. Mode names are shown in the last column of the Table 3.
- In the third section, number of the active node (selected by the user) or of the FIS graph's Connection Arrow (selected by the user) is displayed.
- In the fourth section, the full path to the working folder with project FIS files and results folders is displayed.

1.2 FIS elements: nodes, edges, text

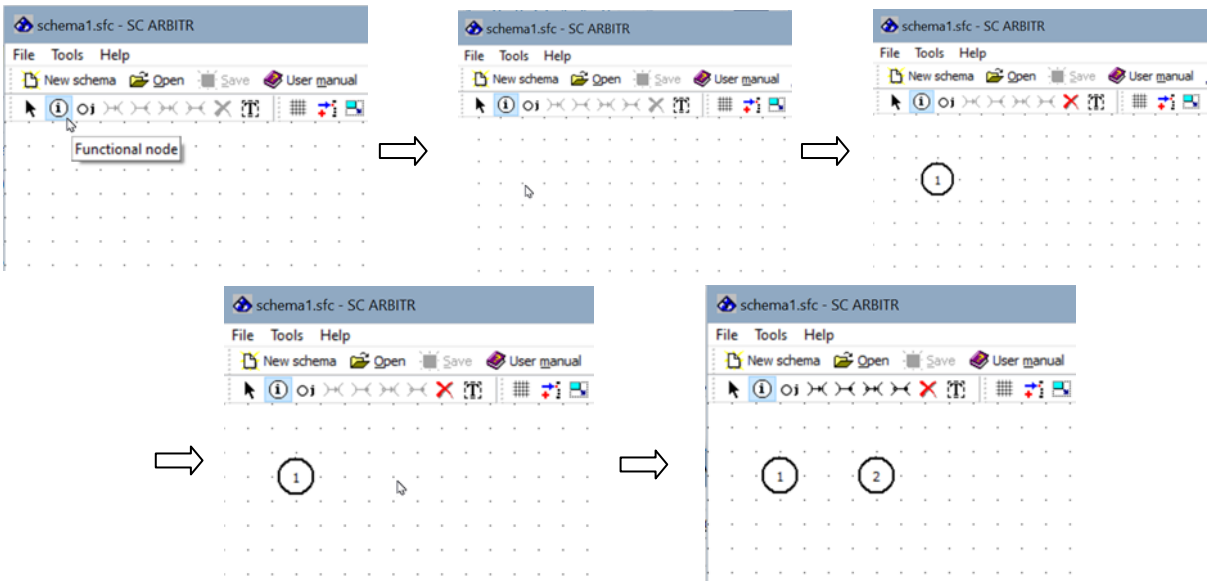
1.2.1 Adding nodes

To create new FIS, launch the SC ARBITR, press the "New schema" button (Figure 3), and start to add the nodes. To implement the option "Adding functional and dummy nodes to the FIS graph", perform the following actions:

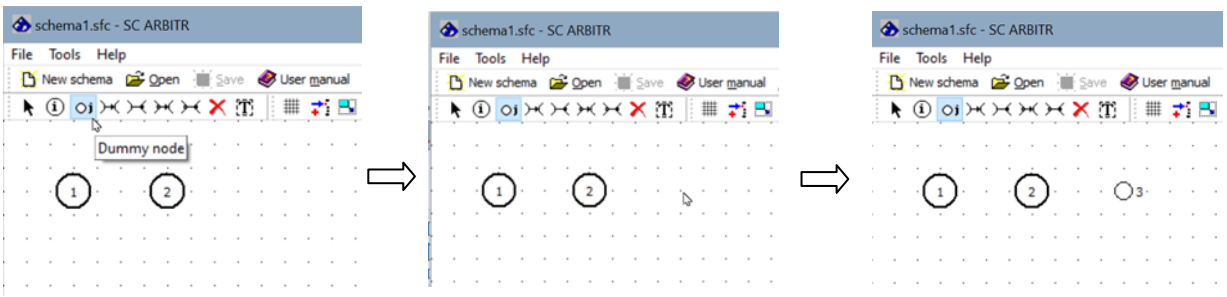
1 Press the "Functional node" or "Dummy node" button on the toolbar (Figure 3, Table 4).

2 Then move the mouse pointer over the FIS input box (Figure 4).

3 Left-click the mouse. A functional (dummy) node will appear on the input box (Figure 8). Node number is assigned automatically.



a)



b)

Figure 8 – Adding functional nodes a) and dummy node b)

You can move the nodes on the FIS input field. To do this, activate the "Select" mode by pressing the button on the toolbar. Then move the cursor to the node to be moved, left click the mouse and, without releasing the button, move the node to a new location. Then release the mouse button. The figure 9 shows the movement of the dummy node 3.

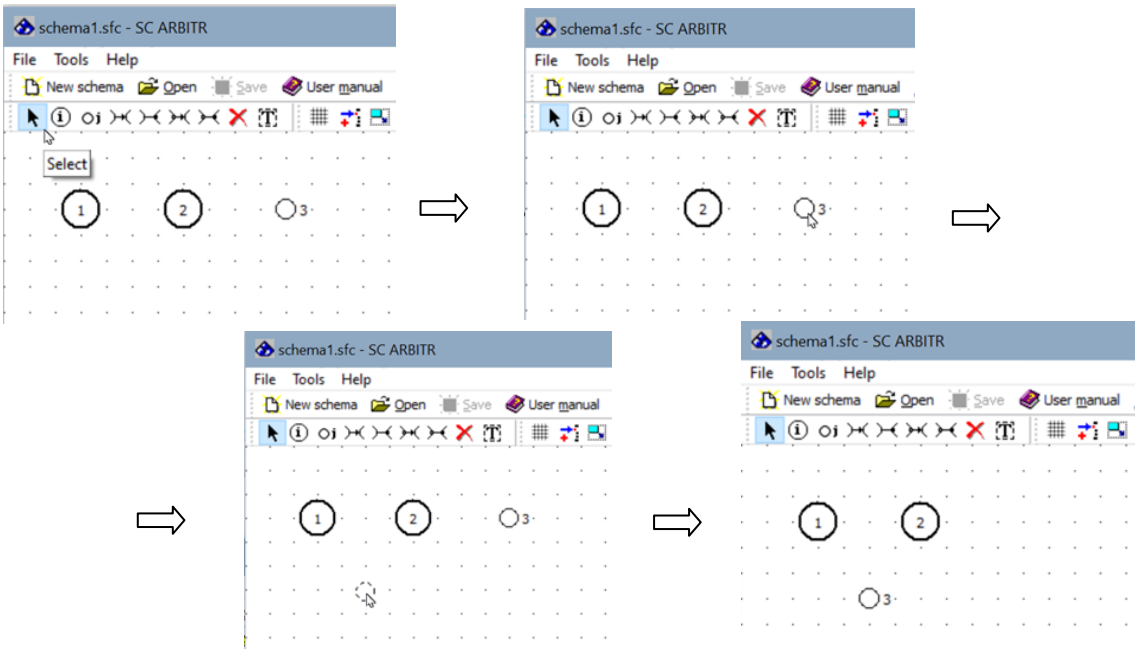
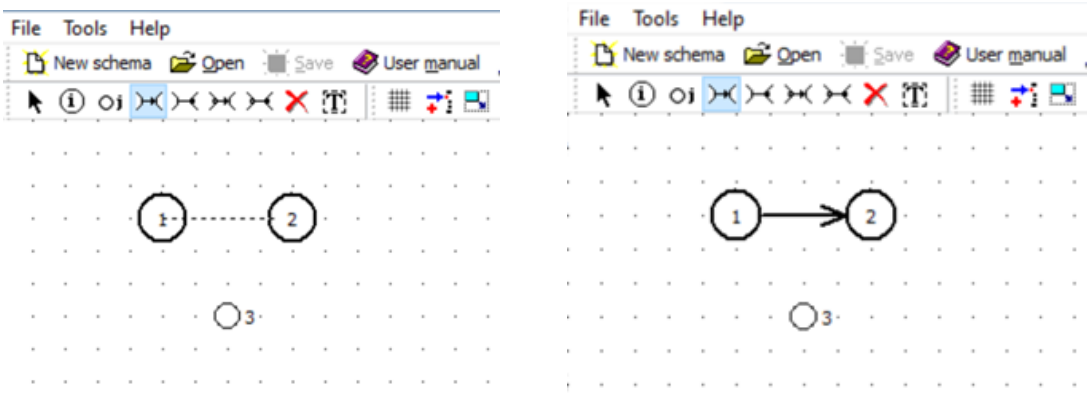


Figure 9 – The movement of the dummy node

1.2.2 Adding Connection Edges

To add an edge connecting two nodes in the FIS, click one of the edge inputs buttons on the shortcut toolbar:  (Figure 4, Table 4). Then choose graph's initial node (edge starting point), point to it, and left click the mouse in the middle of the node. Holding the left button, drag the mouse to the middle of the terminal node (edge end point), then release the button. The nodes are connected by a dotted line (future edge) (Figure 10, a) and after you release the mouse button the edge of the chosen type will appear between nodes (Figure 10, b).

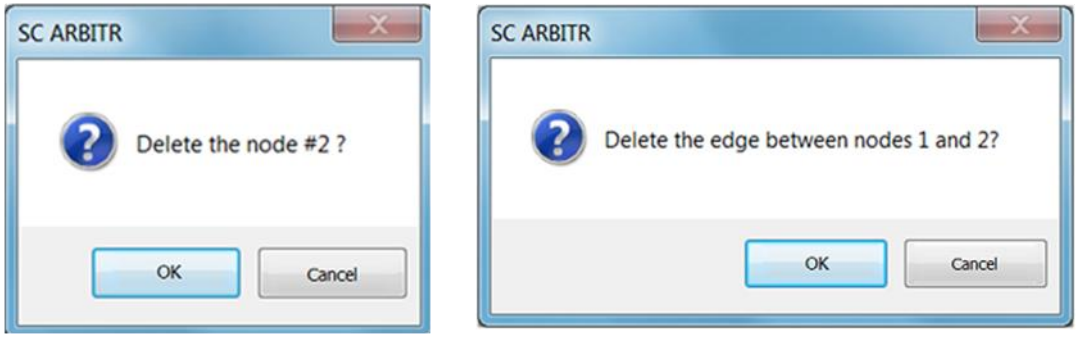


a b

Figure10 – Example of the FIS edge input

1.2.3 Deleting FIS's Nodes and Edges

To delete a node or an edge click on the shortcut toolbar (Figure 4, Table 4). The button gets fixed. Then point to the edge or node that is to be deleted and left click the mouse. The deletion confirmation dialog box is displayed (Figure 11).



a b

Figure 11 – Node (a)/ Edge (b) deletion confirmation dialog boxes

Click OK to delete the selected node / arrow. Click Cancel to cancel deletion operation.

1.2.4 Explanatory Text

Explanatory text objects are used to input titles, comments, names, etc.

1.2.4.1 Text Input

To input explanatory text into the FIS, left click on the shortcut toolbar (Figure 4). The button gets fixed, and the Text Mode is set. Moving the cursor within the FIS input box, select the text input location and left click the mouse. Text input dialog box is displayed (Figure 12).

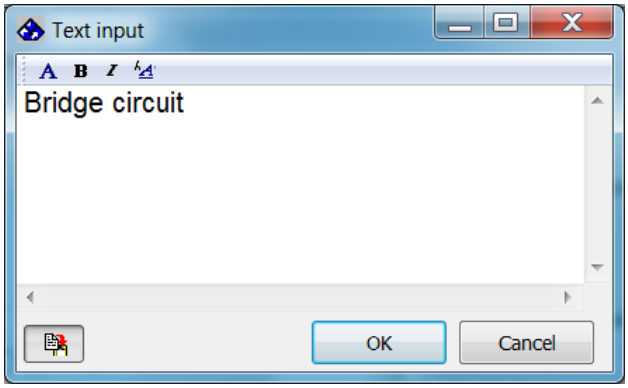


Figure 12 – Explanatory text input and editing dialog box

To select explanatory text's font type, size, and color, use the Input and editing box's control buttons . Click to open standard Font setting dialog box (Figure 13). The Font box contains all explanatory text's font type, size, and color settings.

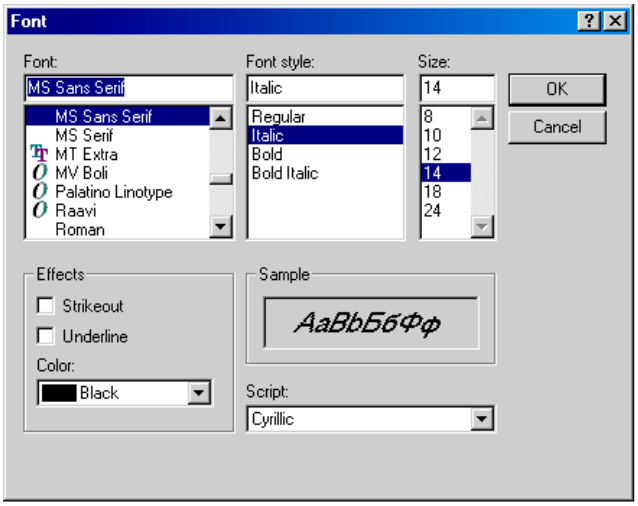


Figure 13 – Standard Font setting dialog box

The next three control buttons  are used for the quick font style setup – Bold, Italic, and Underlined types correspondingly.

Click OK in the Input and editing box to set the new font type (Figure 12). The font type can be set at any time, i.e., before, during, and after the explanatory text input.

1.2.4.2 Adding & Editing Text

Text is added using the keyboard (Figure 12). The Software's text input functions correspond to the text editor functions (input, deletion, buffer copying, and multiline editing). Example of title and explanatory text input into the FIS working box is shown in Figure 14.

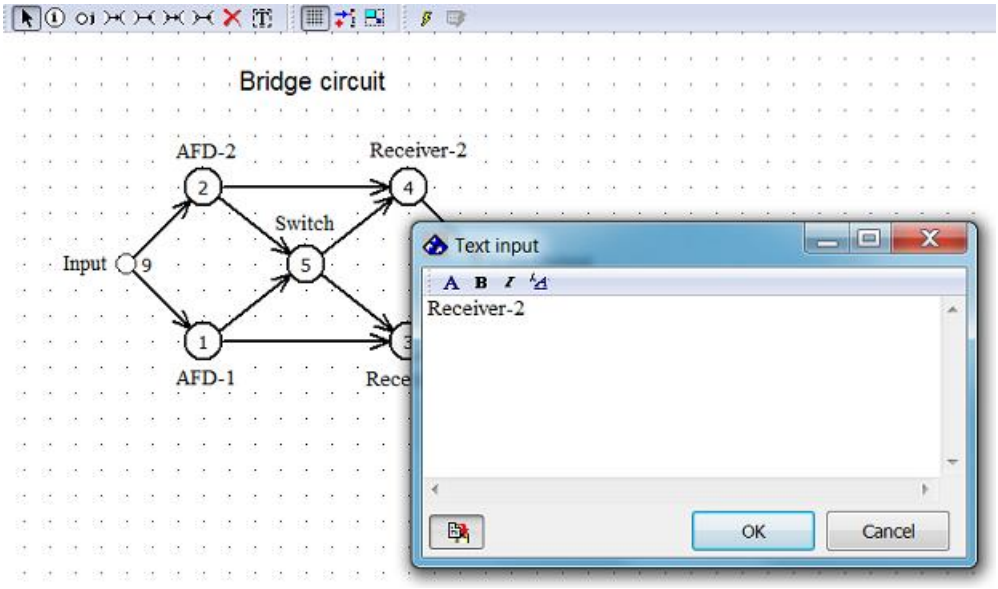


Figure 14 – Example of text input into the FIS working box

When the control button  is fixed (Figure 12), the previously saved text will be displayed in the new created editing window. This text can be edited and added to the new FIS input box. When the control button is switched off, the previously saved text will not be displayed.

To edit the text that has already been added to the FIS input box, point to the text and right-click the mouse. Context menu is displayed (Figure 15).

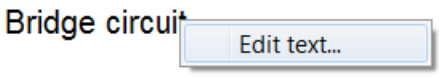


Figure 15 – Text context menu

Click the option Edit Text to open the Input and editing box with the selected text (Figure 12). Edit the text, then click OK to save changes in the FIS input box.

1.2.4.3 Deleting Text

Explanatory text deletion is like the FIS's node and arrow deletion (item No. 1.2.3). Click  on the shortcut toolbar (Figure 4, Table 4). The button is fixed. Then point to the text segment that is to be deleted and left click the mouse. Deletion confirmation dialog box is displayed (Figure 16).

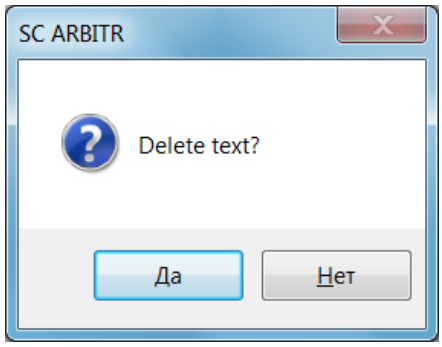


Figure 16 – Text deletion confirmation dialog box

Click Yes (Да) to delete the text. Click No (Нет) to cancel the deletion operation.

1.2.4.4 Relocating Text

Separate text can be moved to any location within the FIS input working box. Click  on the shortcut toolbar (Figure 4). The button gets fixed, and Selection Mode is set. Point to the text that is to be relocated, left click the mouse. Holding the left button drag the mouse cursor. While moving, the text is shown as a rectangle. Release the button to relocate and fix the text.

1.3 The node parameters. Modeling Mode Settings

1.3.1 Changing nodes parameters

The Software has two ways to input and edit element parameters.

I Element parameters input box

When entering new nodes in the FIS, their numbers are automatically set in ascending order. At any stage of the FIS construction, the nodes numbers can be changed.

For changing the functional node number:

1 Move the mouse pointer over the selected node.

2 Right-click the mouse. Context popup menu will appear (Figure 17, left part).

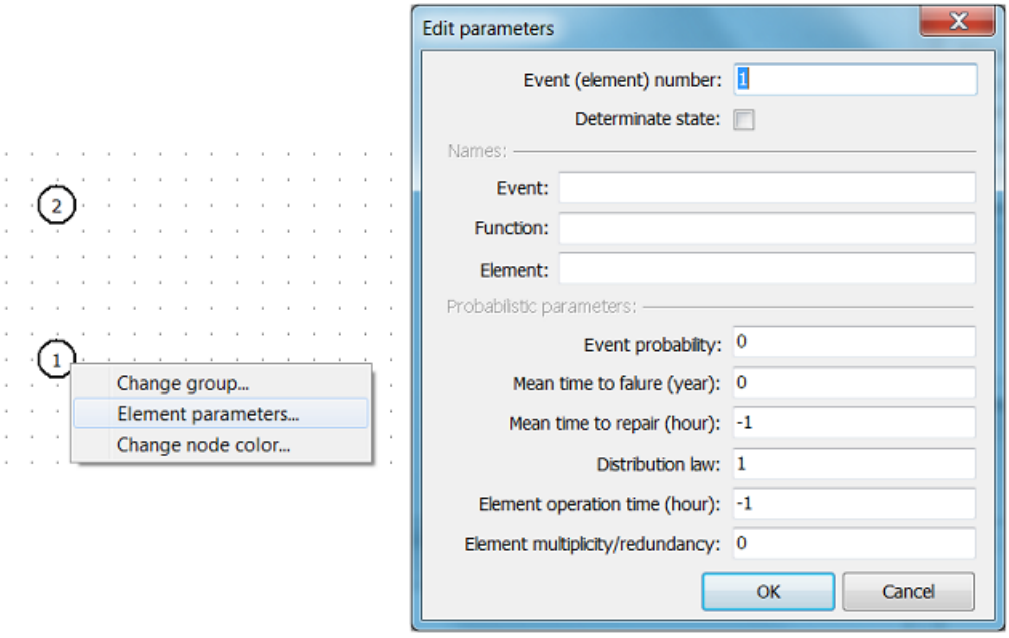


Figure 17 – Changing the functional node number

3 Select "Element parameters…" option.

4 A window for changing node parameters will appear (Figure 17, right part).

5 Place the cursor in the line "Event (element) number" and enter a new node number.

6 Click the OK button to save the node number or click Cancel otherwise.

To enter/change the event probability value of the selected functional vertex, follow steps 1–4 of the previous step and then select the line "Event probability", where the required value is entered, for example, "0.9" (Figure 18).

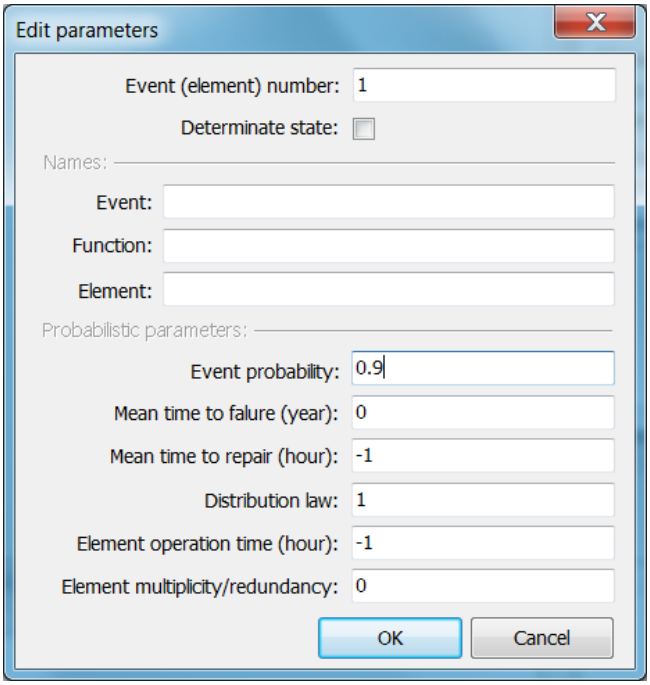


Figure 18 – Changing the event probability

The dialog box for changing the dummy node parameters is shown in Figure 19. For a dummy node, only the current number and color can be changed.

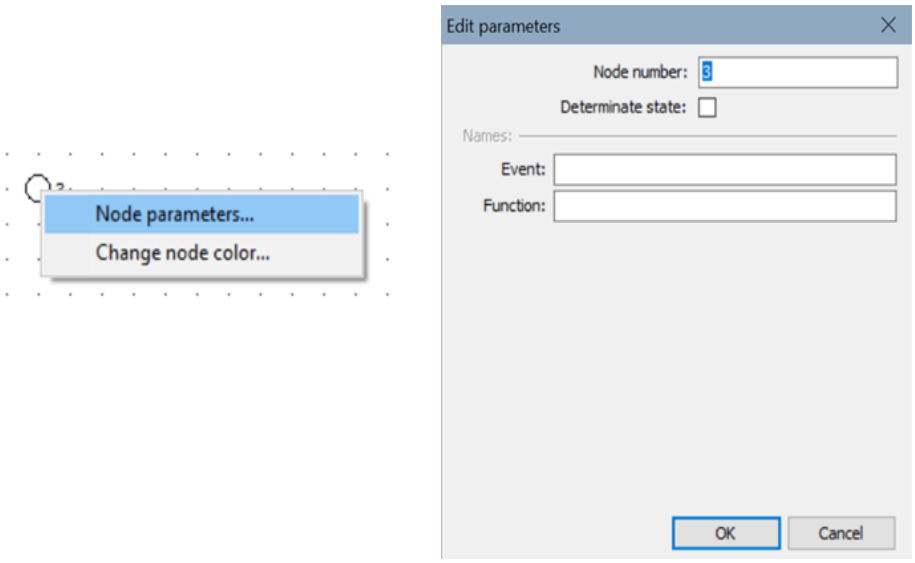


Figure 19 – Changing the dummy node parameters

II Element parameters input table

The same parameters can be added using the table located at the bottom of the Parameter and Mode input window (Figure 4, Parameters Table). Figure 20 shows a fragment of the parameter table. The value P=0.9 is entered into the table for the functional node 2.

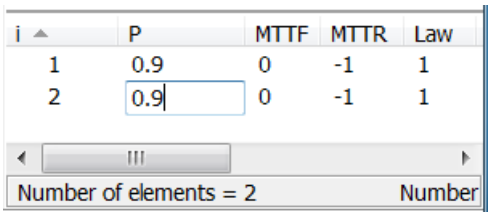


Figure 20 – Parameter table

The full table is represented in Figure 21.

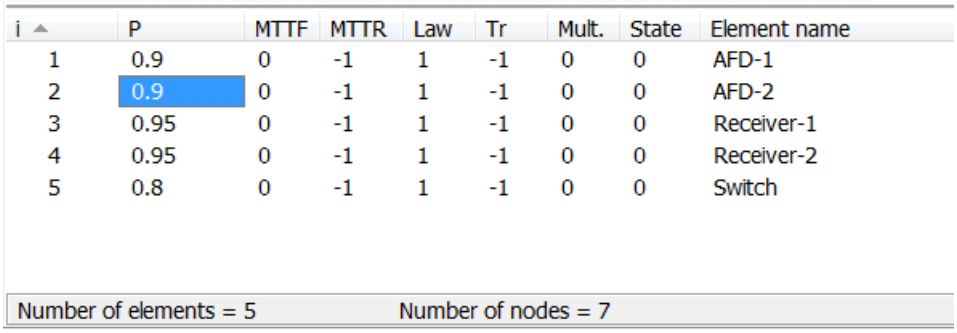


Figure 21 – Element Parameters input table

If the Table is hidden, access it using Main Window's scroll bars.

The following element parameters can be added:

i – number of the FIS's *i* nodes, that represent the system's elements;

P – static probability of the binary event result, represented in the FIS by the *i* node;

MTTF – *i* element's mean time to failures (years);

MTTR – *i* element's mean time to repair (hours); code "-1" means that the *i* element is unrecoverable;

Law – two codes are used in the present SC ARBITR version:

"0" – static probability values ("P" column) are used in calculations;

"1" – exponential law of distribution of the *i* element non-failure operation time is used with "MTTF" & "MTTR" parameters in calculations;

Tr – *i* Element lifetime (hours); code "-1" means that *i* element lifetime is considered to be equal to the whole system non-failure operating time;

Mult. – *i* element's multiplicity factor:

"0" – means that *i* element has no multiplicity factor;

integer positive number "+n" – means that *i* element represents a subsystem, consisting of "n" equitype elements with set parameters, using the AND logic (conjunctive multiplicity);

integer negative number "–n" – means that *i* element is a subsystem, consisting of "n" equitype elements with set parameters, using the OR logic (disjunctive multiplicity);

State – Determinate state of the element (1 – failed; 0 – not failed);

Element name – contains brief information about *i* element and binary event properties (the FIS nodes).

1.3.2 Changing Node Color

To make the FIS clearer and more informative, the node color can be changed. Point the mouse to the node and right-click. Near the chosen node a pop-up menu appears. Click the option "Change node color…" (Figure 22).

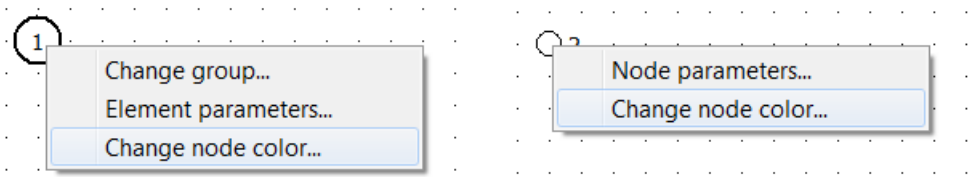


Figure 22 – Node Contextual Menu

The standard Color selection dialog box is displayed (Figure 23).

Figure 23 – Color selection dialog box

Select the node color and click OK. The dialog box closes, and the node changes its color. Click Cancel to leave the node color unchanged.

1.3.3 Modeling Modes Setting

Before modeling & calculation startup it is necessary to set modeling modes. Modeling & calculation options are located on the toolbar at the top of the Parameter and Mode input box (Figure 4). The toolbar overall view is shown in Figure 24.

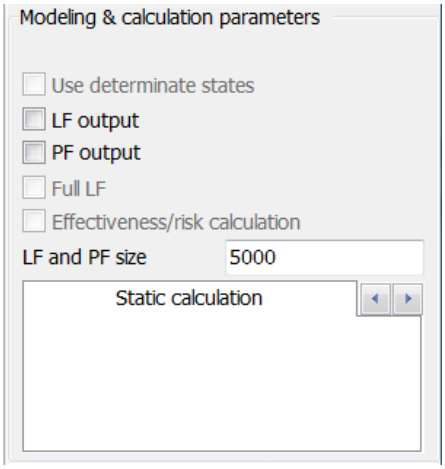


Figure 24 – Modeling and calculation parameters input box

Table 5 has parameter names, functions, and control buttons on the toolbar.

Table 5 – Modeling and computing parameters

Name	**Function**
Use determinate states	*Enabled* – analysis of the current status of the system elements, including Determinate State values
	Disabled – no analysis of the current status of system's elements is performed
LF output	*Enabled* – the explicit logic function of the system availability is added to the report in the standard disjunctive form (list of minimum paths of system functioning, minimum sections of system failures or their combinations)
	Disabled – only the size of the function of system availability (number of conjunctions) is specified in the report
PF output	*Enabled* – the system's probabilistic function is specified in the report as a polynomial
	Disabled – only the size of the probabilistic function (number of monomials) is specified in the report
Names output	*Enabled* – element name is displayed in the logic function instead of the number (available if the explicit function of system availability is enabled)
Full LF	*Available* only when the Approximate Computation window is open *Enabled* – the decomposed probabilistic function is transferred into an expanded probabilistic function
	Disabled – no transformation of the decomposed probabilistic function into the expanded probabilistic function is performed
Effectiveness/risk calculation	*Enabled* – complex mode of modeling and computing the actual efficiency value or expected damage value based on multiple criteria and performance parameters or system functioning risk parameters
	Disabled – simple mode of modeling and computing the probabilistic values for individual criteria of the system functioning
LF and PF size	Maximum allowable size of the L-function and P-function is fixed (default value is 5000)

Effectiveness/risk calculation is not compatible with a determinate analysis, since the effectiveness/risk calculation takes in consideration all set criteria. The determinate analysis takes in consideration status of the system's elements. Information on the set elements' status is saved in Table 5, Status line. Shall the element(s) fail, the system is checked to find elements which functionally failed resulting from actual elements failure. Analysis results are displayed in FIS graph as follows:

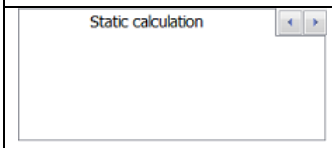
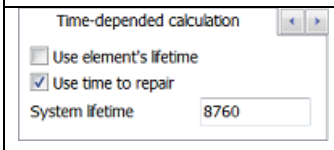
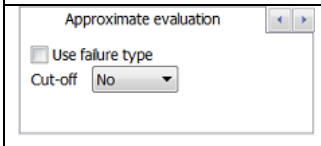
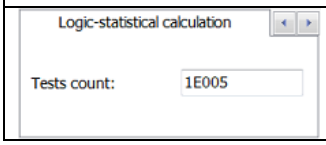
- Failed elements which are present in Table 5, Status line, are marked on the FIS in red.
- Functionally failed elements are marked on the FIS in grey.

Modeling and computing are made for one criterion (selected as a current criterion).

Cyclic switch of the computing modes is located at the bottom of the modeling and calculation panel. Select the required calculation mode with additional mode parameters using the left-right arrow buttons.

Table 6 describes possible calculation modes.

Table 6 – Calculation modes

Calculation Mode	Description
	Calculations are made based only on Pi parameters, i.e., static probability of the binary event's outcomes represented by functional nodes in the FIS. No schedule of the system's non-failure operation probability is generated
	Calculations are made based on the following set parameters: • MTTF – Mean time to failure of the system, years; • MTTR – Mean time to repair of the system's elements, hours; • Tr – System lifetime, hours. Schedules of non-failure operation, failure probability or system's availability ratio are generated. It makes sense to consider the element's run time run time only if the system is recoverable. Therefore, the "Element Run Time Recording" shall be enabled only if the "Recover Time Recording" is enabled.
	Calculation of approximate values of the system probabilistic parameters based on truncated ("Cut-off" enabled) or full ("Cut-off" disabled) monotonic logical function of system availability which represents minimum sections of failures without (Failure Type Accounting disabled) or with (Failure Type Accounting enabled) three types of element failures. No schedule of the system's non-failure operation probability is generated
	Calculation of approximate values of the system's probabilistic parameters is performed using simulation modeling method. Tests count is specified for the number of statistic tests

1.4 Adding and Editing System LCF

Logic criterion (criteria) of functioning is (are) specified in the Criterion table (Figure 4). The system's LCF (LC) input and editing table is shown in Figure 25.

Figure 25 – Criterion table

Below are some guiding principles of the LCF input operation:

- A lowercase "*y*" should be added as a prefix to the integral function number, making part of the LCF, for example, "*y13*".
- If multiple integral functions are added in the LCF with a conjunctive connection, they are added without a space, for example, "*y3y4*". If there is a disjunctive connection, the disjunctive sign "+" is used, for example, "*y3+y4*".
- To set the inverse criterion, after letter "*y*" a quotation mark is placed, for example, "*y"13*".
- If it is necessary to input multiple LCFs at once, to add a new LCF point the mouse to the last line of the table (see Figure 25) and click a down-arrow key. A new line will appear, add an LCF there. To delete a selected LCF press DEL.

Damage parameter (Figure 26) is available only is Effectiveness/risk calculation mode is enabled (Figure 24, Table 5). Damage is set in conventional units and shall be standardized.

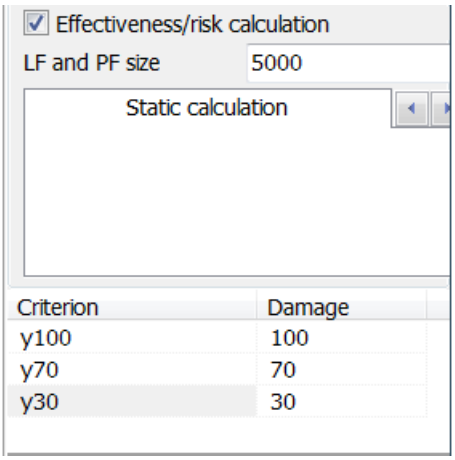


Figure 26 – LCF with damage parameter

When the Effectiveness/risk calculation mode is disabled, the modeling and calculation are applied for the selected LCF only. Subgraph calculation and modeling are performed for one LCF (selected in the table).

If an integral function number is used for the LCF that does not correspond to any main FIS node, an Incorrect LCF Error Message (Figure 27) is generated after pressing "Modeling and Calculation" (Figure 4).

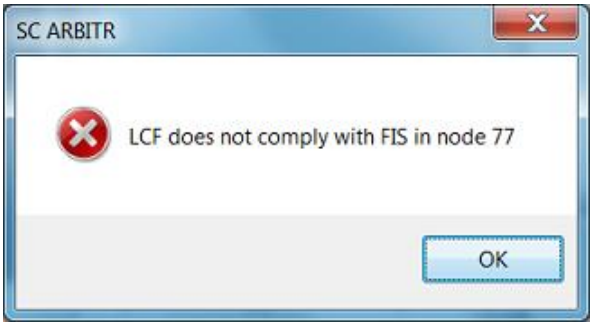


Figure 27 – Incorrect LCF Error Message

1.5 Saving/Opening a FIS

To save the FIS for the first time, click the File menu command "Save As" (Figure 4). Standard dialog box opens.

Specify the FIS name in the File Name entry line. Default filename extension ".sfc" is set. Then press "Save" button. Press Cancel button to cancel saving the FIS.

To "Save" new changes of an already saved FIS file, click the File command "Save" or click Save on the toolbar. The File name input box does not open. The FIS is saved in the opened file.

There are two ways to open an existing FIS: click the File menu command "Open" (Figure 4) or click Open on the toolbar. The standard dialog box opens.

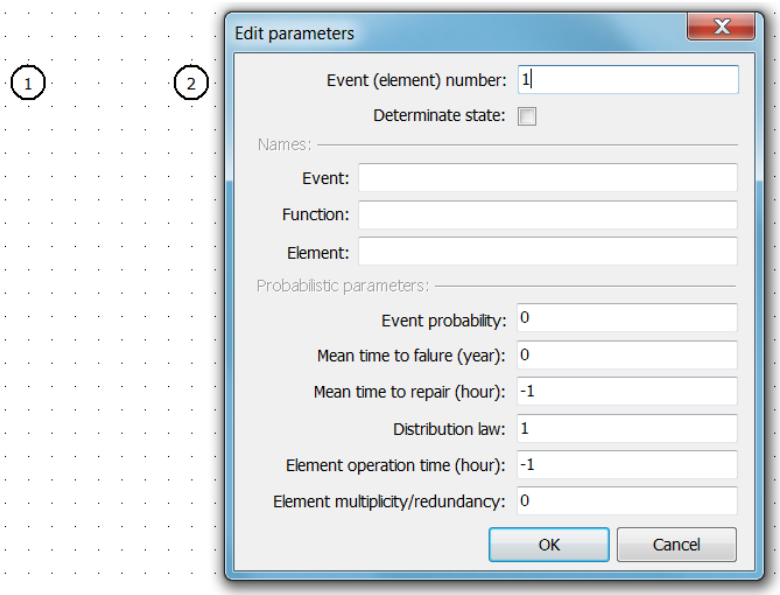
A list of previously saved FIS folders is displayed. Each folder corresponds to one FIS. Select the required folder. To open the folder, left double-click it. Select the file with "*name*.sfc" extension. Press Open button. Press Cancel button to cancel opening the file.

1.6 Multiplied nodes

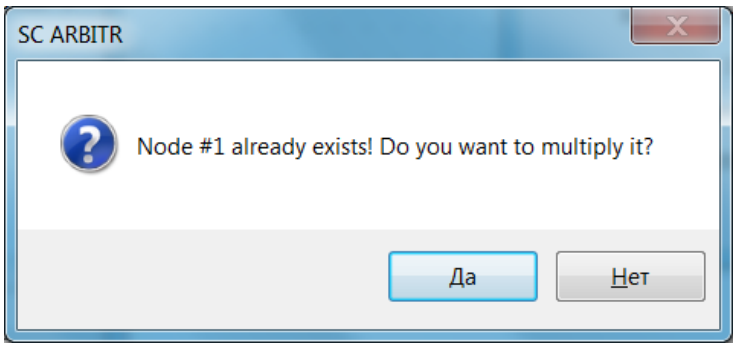
The SC ARBITER has the ability to create multiplied nodes, i.e., nodes with the same parameters. This allows you to display the same node several times on the scheme. To create a multiplied node, we need to assign to this node the number of another node that we want to duplicate.

1 Left-click the mouse on the node, select "Element parameters...". The Edit parameters window opens

2 Enter the node number to be multiply in the field "Event (element) number" and press OK button.

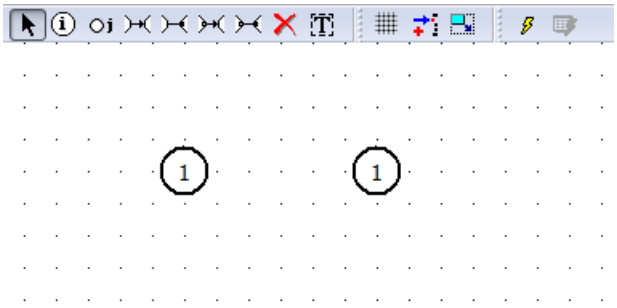


3 The node multiplication request message appears.



Click Да (Yes) to multiply node or Нет (No) otherwise.

After multiplication, the scheme will have two nodes with the same numbers.



In the Parameter Table this element becomes blue.

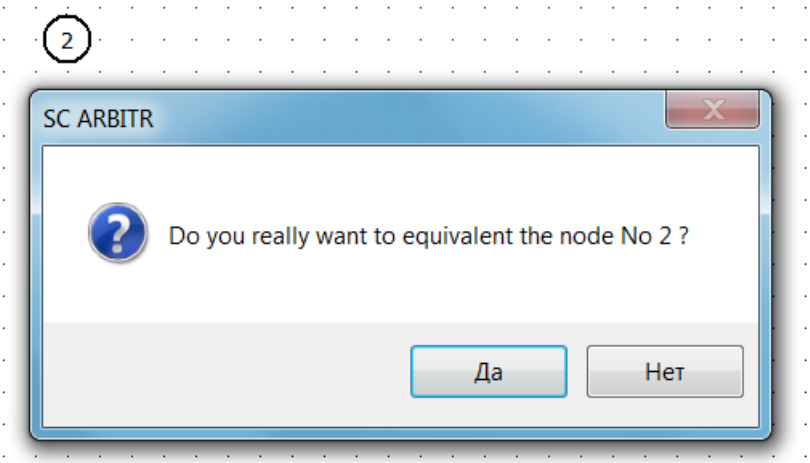
i	P	MTTF	MTTR	Law
1	0.9	0	-1	1

1.7 Equivalent nodes

Each functional node of the FIS's main graph (developed in the main FIS input box) can become equivalent. Equivalent node means that the given node is a subsystem, which structure is represented in the FIS subgraph. These FIS subgraphs are developed in the specialized Software's window.

1.7.1 Specifying equivalent nodes in the main FIS

Select the functional node within the FIS input working box that is to be equivalent. Click ⓘ on the shortcut toolbar (Figure 4). "Node Mode" of the main FIS graph is set. Then point to the selected functional node, left double-click the mouse. Confirmation request is displayed.



Click Да (Yes) to input the equivalent node. The input box of equivalent node's FIS subgraph is displayed in Figure 28.

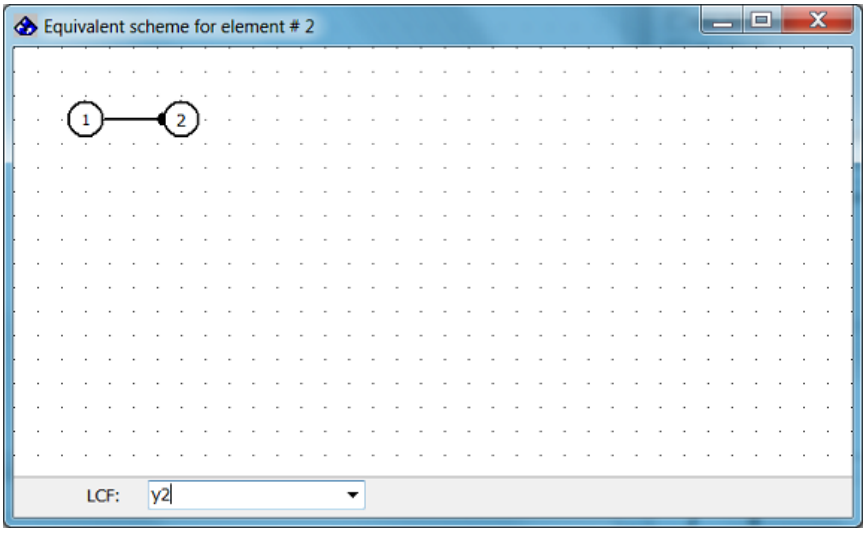


Figure 28 – The equivalent node's FIS subgraph input box

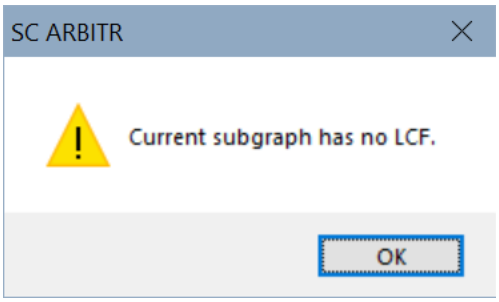
The window consists of the following three segments:

- Title bar located at the top of the window. Title bar contains the main FIS graph equivalent node's number;
- FIS subgraph input and editing field located in the middle of the window;
- Subgraph's LCF input field located at the bottom of the window.

Functional and dummy nodes, edges & explanatory text are added to the subgraph's input and editing field in the same way as into the main FIS graph. All operations are similar to those described above in item No. 1.2.

When the FIS subgraph development is completed, add the subsystem LCF to the LCF input field. In this example LCF is *y2*.

When the FIS subgraph and LCF input are completed, left-click the button  in the corner of the top right-hand window. The FIS subgraph input box closes. If the subgraph's LCF is not added before closing the window, a warning massage will appear.



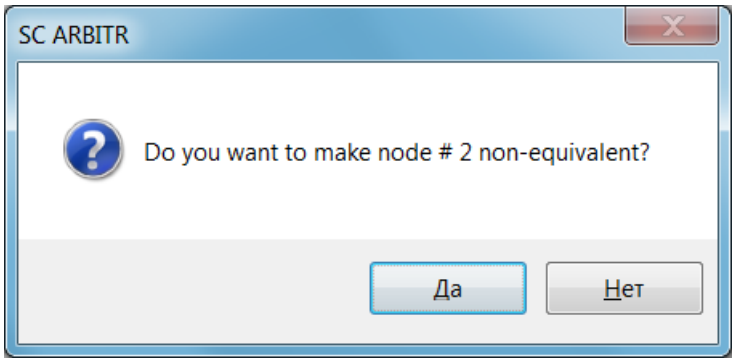
When the subgraph input box is closed, the equivalent node in the main FIS graph will be displayed as a triangle

When the equivalent node is added, the user can view and edit the node's subgraph. To set the "Node Mode" (click  on the shortcut toolbar), point to the equivalent node and double-click. The FIS input and editing field will be displayed (Figure 28). Then the FIS subgraph can be edited.

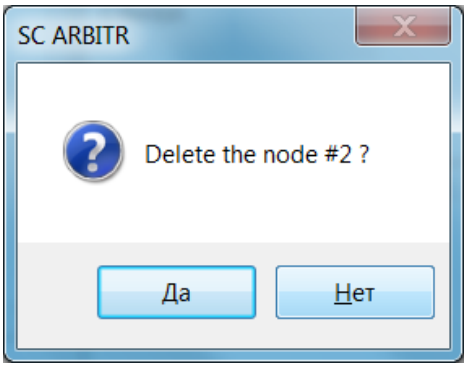
1.7.2 Deleting Equivalent Nodes

To delete an equivalent node:

- Click  on the shortcut toolbar;
- Point to the equivalent node and left-click the mouse. A confirmation request is displayed.



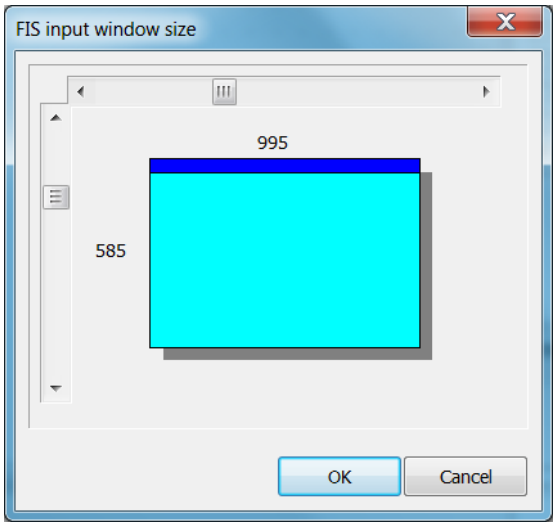
- Click No (Нет) to leave the equivalent node unchanged. Click Yes (Да) to replace the equivalent node with the main FIS graph's functional node. A confirmation request is displayed.



- Click No (Нет) to leave the functional node in FIS. To delete the functional node click Yes (Да).

1.8 FIS Input Box Resize

When a large-scale FIS is created, the default input and editing box size (995x585 pixels) may not be sufficient. To increase the FIS input workbox size, click  (Change Window's Extent) on the Toolbar (Figure 4). Dialog box will open.



Dialog box slide bars are initially set to fit the default FIS input box size. Move slide bars using the mouse to set the desired FIS input box size. While moving slide bars the Status bar numbers show the current box's size (in pixels).

When the FIS input box size is set, click OK. Dialog box closes and the FIS input box size changes. Click Cancel to keep the FIS input box size unchanged.

If the resized input box does not fit the screen, scroll bars automatically appear to display parts of the large-scale FIS.

2 Modeling of simple structures

2.1 Serial system modeling

Consider a serial system of two elements. The condition for the operability of a serial system is the operable state of all its elements.

In logical models, the outcomes of binary events are represented by logical variables x_i. We will denote by x_i the logical condition of the i-th simple event realization. The conditions for the implementation of a complex event will be denoted by y_i. The conditions for the realization of a complex event depend both on the conditions of the i-th event realization itself and on the conditions for the realization of all simple and complex events that ensure of the i-th event realization. For example, the logical condition for the operability of a serial system y_{ss} of two elements is determined by a conjunction of the form $y_{ss}=x_1x_2$.

To model a serial system, do the following:

1 Create a new FIS (part 1.2.1, Figure 3, b).

2 Add two functional nodes (Figure 8, a).

3 Set initial values: P1=0.9, P2=0.8.

4 Set modeling parameters: check the boxes "LF output" and "PF output".

The "LF output" option provides the output in the report of an analytical expression for a logical function (LF) in disjunctive normal form (DNF) – the conditions of a given system event realization (for example, system operability).

The "PF output" option provides the output in the report of an analytical expression in the form of a probability function polynomial (PF).

5 To set the conditions for the implementation by the system of the quality indicator under study and the subsequent construction of the corresponding mathematical model, it is necessary to formulate and enter one or more logical criteria (LC) into the lines of the Criterion table. LC are written in small letters.

For our example of modeling the operability of a serial system, in the Criterion table, you should enter the criterion "*y2*".

6 Select calculation mode in the Parameter and Mode Input Box (Figure 4). This example is executed in the "Static calculation" mode.

The SC ARBITR allows several ways of a serial system graphical display.

The first way to represent a serial system is to use an **OR** edge. To do this, nodes 1 and 2 are connected by an **OR** edge according to the actions described in the item No. 1.2.2 (Figure 10).

The screen interface after performing these steps is shown in Figure 29.

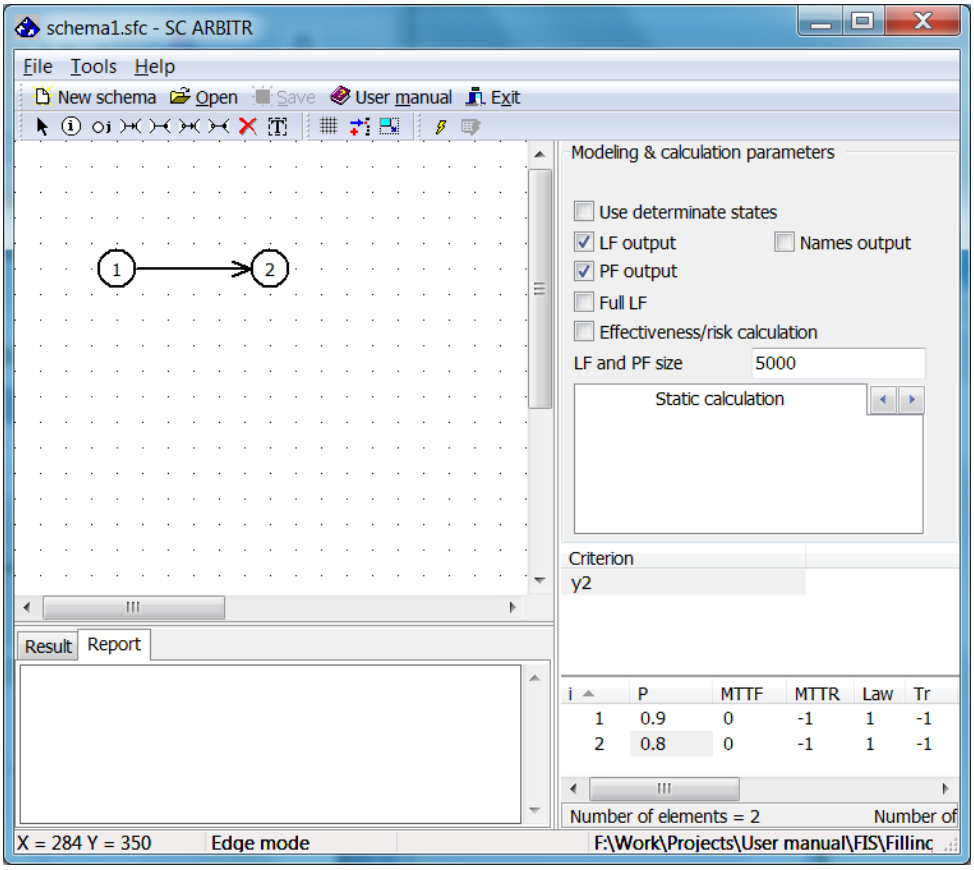


Figure 29 – Screen interface: serial system modeling. Way 1

To perform calculations, click the "Modeling and calculation" button in the shortcut keys toolbar. (ATTENTION!!! Save the FIS before making calculation.)

Brief results of the calculations are presented on the Results tab (Figure 30).

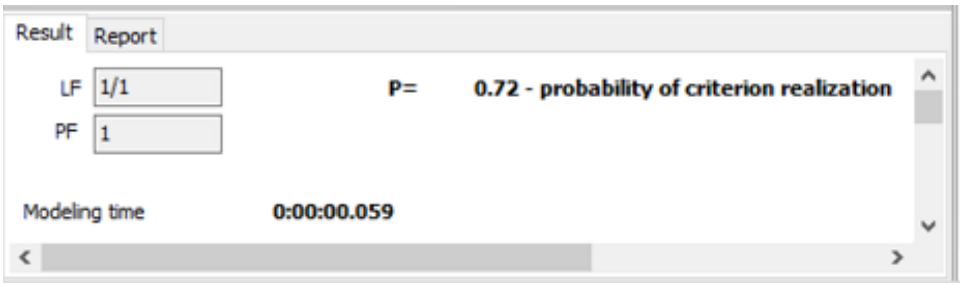


Figure 30 – Results of modeling the reliability of a serial system

The modeling results show that the number of terms in the formed LF is equal to 1 (indicated in the box LF); the number of terms in the polynomial of the probabilistic function (PF) is also equal to 1 (indicated in the box PF).

The numerical value P=0.72 is the probability of the criterion implementation, in our example it is the probability of failure-free operation of the system under study. More detailed modeling results are shown on the Report tab. A fragment of the modeling report is shown in Figure 31.

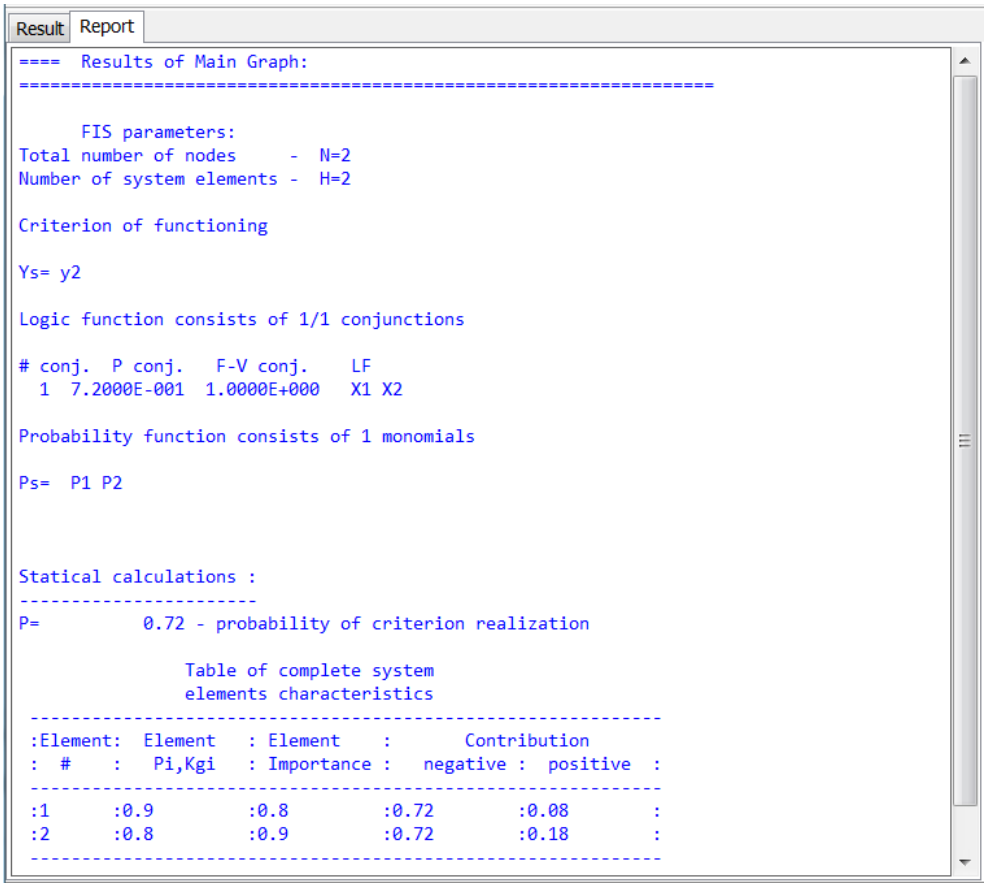


Figure 31 – Fragment of the modeling report of a serial system reliability

From the presented report, it is clear that:

1 The FIS consists of two elements, and both elements are functional nodes.

2 The modeling was carried out using the LC *y2*.

3 The system operability conditions logical function contains 1 conjunction of the form *Ys=X1X2*.

4 The probabilistic function contains 1 polynomial and given the initial data (probabilities of elements 1 and 2 failure-free operation), the probability of implementing the criterion (probability of the system failure-free operation) *Ps*=P1P2=0.9*0.8=0.72.

5 The table of complete system elements characteristics provides information about the initial data:

- Column "Element *Pi*, *Kgi*" contains information about the event probability (reliability or failure). In this example – *Pi*.
- Column "Contribution negative" provides data on how much the system measure will change if the probability of an event occurring increases to 1.0.
- The "Contribution positive" column provides data on how much the system measure will change if the probability of an event occurring drops to 0.
- The "Element Importance" column represents the element's importance, which is calculated as the sum of the absolute values of the element's contributions.

See Appendix E for details on importance.

The second way to represent a serial system is to use an **AND** edge. To do this, delete the **OR** edge (item No. 1.2.3 (Figure 11, b)) on the previous FIS (Figure 28) and then connect nodes 1 and 2 with an **AND** edge.

After pressing the button "Modeling and calculation", the calculation results and the modeling report do not change (Figure 32).

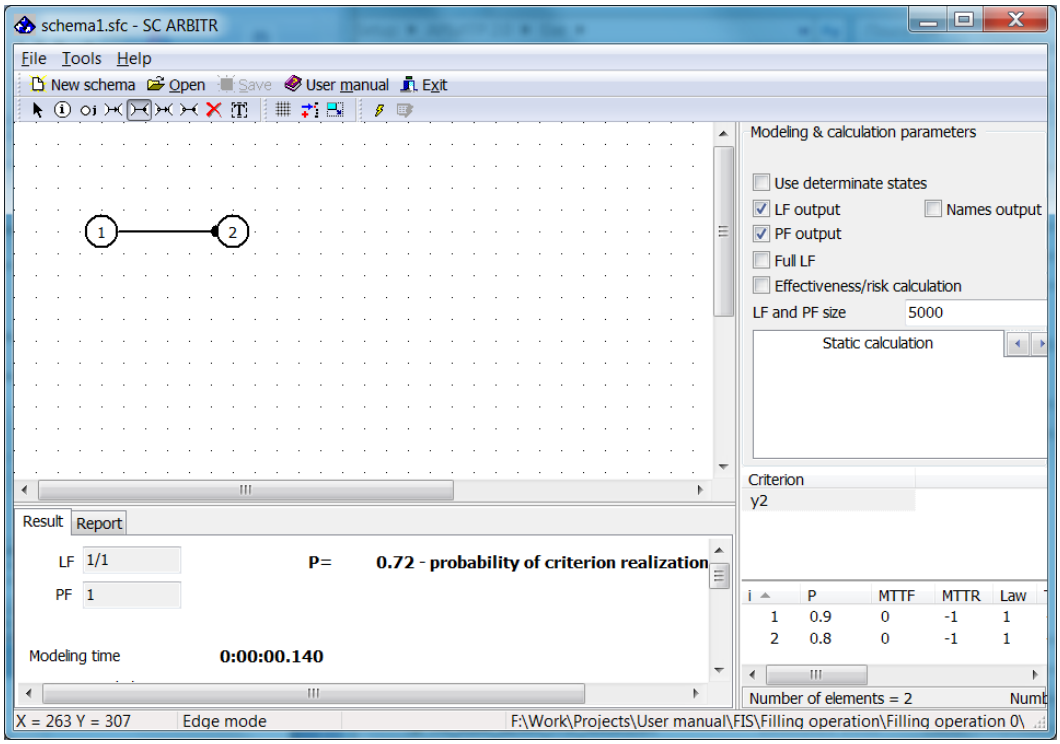


Figure 32 – Modeling the reliability of a serial system. Way 2

The third way to model the reliability of a serial system is to use an **AND** edge and a dummy node as a connector.

For this you need:

1 Add a dummy node No. 3.

2 Delete the **AND** edge between nodes 1 and 2.

3 Connect nodes 1 and 2 with **AND** edges to dummy node 3.

4 Enter the operability criterion *y3* (item No. 1.4).

After pressing the button ("Modeling and calculation"), you can make sure that the calculation results remain the same; the modeling report has slightly changed (Figure 33). Now the logical criterion for functioning is the expression *Ys=y3*.

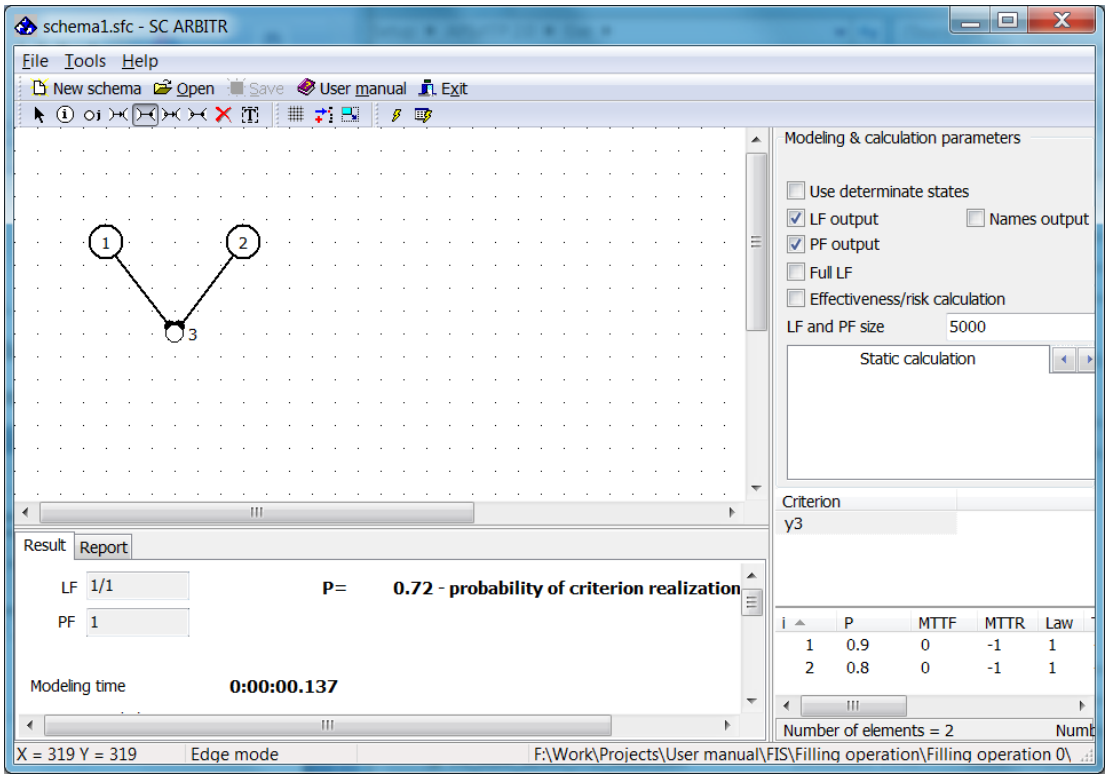


Figure 33 – Modeling the reliability of a serial system. Way 3

A significant advantage of reliability modeling on the SC ARBITR is the ability to use one FIS to assess both the probability of failure-free operation (**direct solution**) and the probability of failure (**inverse solution**).

To calculate the probability of failure of the analyzed system, in the line of the Criterion table, enter the inverse operability criterion – the failure criterion *y"3* (Figure 34). The quotation mark denotes the negation (inversion) of a Boolean variable.

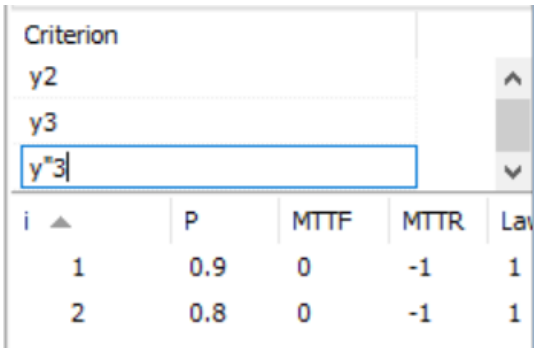


Figure 34 – Modeling the failure of a serial system

A fragment of the report with the modeling results the failure of the serial system is shown in Figure 35.

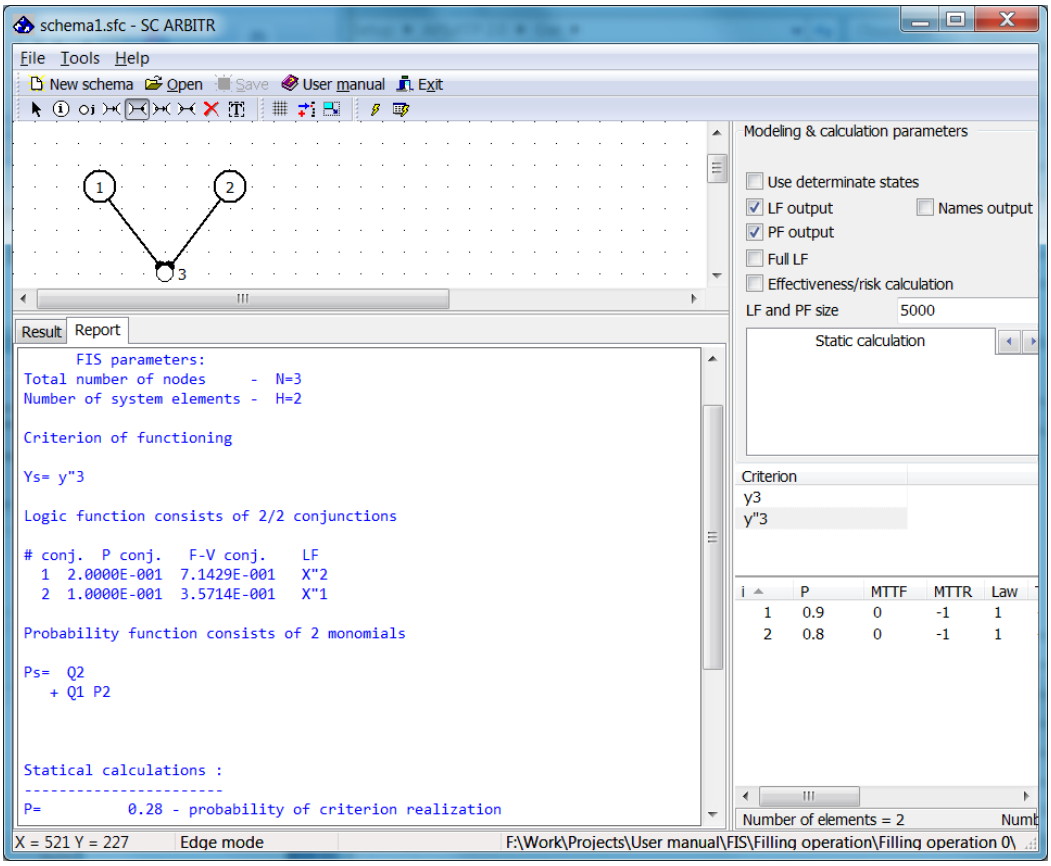


Figure 35 – Modeling the failure of a serial system. Report fragment

The logical function shown in Figure 35 can be obtained using the de Morgan rule for the LF $Y_s = X1X2$, i.e., $\bar{Y}_s = \overline{X1} \vee \overline{X2}$.

The transition to the probabilistic function was carried out after orthogonalization of the original DNF, i.e., $\bar{Y}_C = \overline{X1} \vee X1\overline{X2}$.

Attention! The report adopted a single designation of the implementing probability of a given criterion "P", regardless of the problem being solved nature.

Obtaining an inverse solution makes it possible to carry out a rigorous verification of the modeling correctness, since the sum of the probabilities of the direct and inverse solutions as the sum of the probabilities of the complete events group is equal to 1. In our case, $Pr\{y3\} + Pr\{y"3\} = 0.72 + 0.28 = 1$.

2.2 Modeling of a parallel system

Consider a parallel system consisting of two elements. The condition of parallel system operability is the operable state of at least one of the elements. In terms of the algebra of logic for monotone structures, this operability condition is written using the disjunction $Ys=X1 \vee X2$.

To graphically display the disjunction of two events, we replace the **AND** edges in Figure 31 with **OR** edges. To do this, you must perform two steps:

1 Delete **AND** edges using the button .

2 Add **OR** edges to connect functional nodes 1 and 2 with dummy node 3.

Select the LCF *y3* in the Criterion table. After pressing the button ("Modeling and calculation"), the results shown in Figure 36 will appear. The decision according to the chosen logical criterion *y3* corresponds to a logical function of the form $X1 \vee X2$.

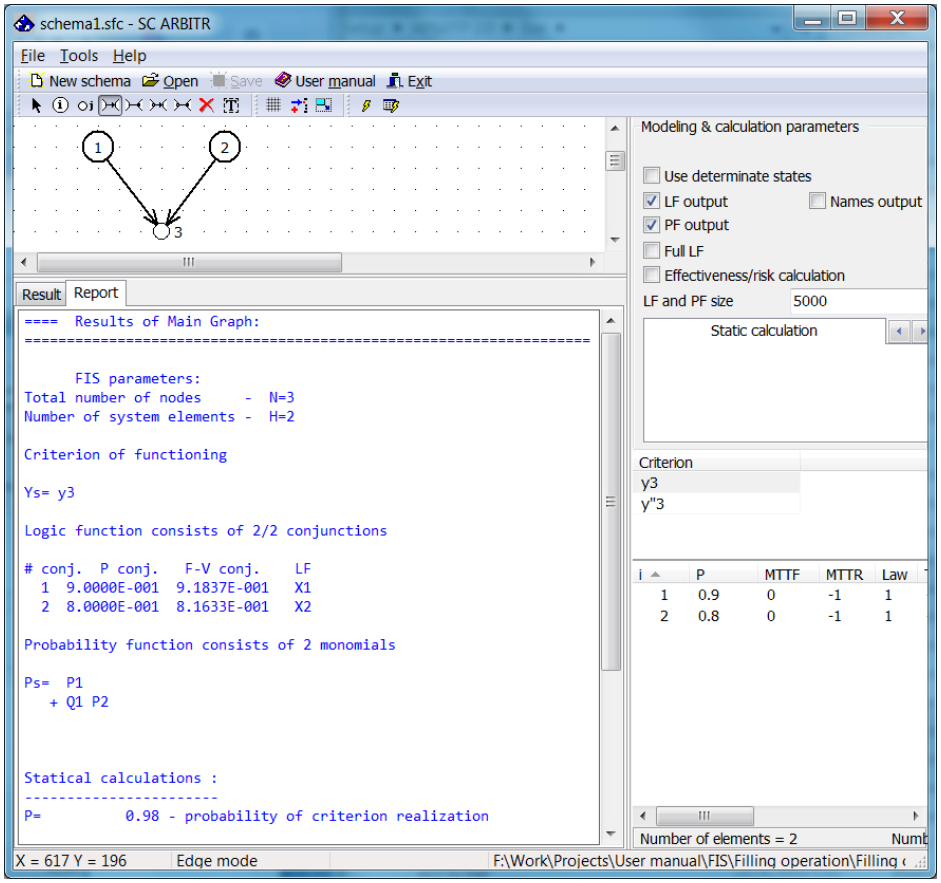


Figure 36 – Modeling the reliability of a parallel system

The probability function corresponding to this LF, which can be obtained using the simplest orthogonalization formula $A \vee B = A \vee \bar{A}B$, has the form

$$P_s = P1 + (1 - P1)P2 = P1 + Q1P2.$$

When substituting the initial data P1=0.9 and P2=0.8, the probability of non-failure operation of the parallel system will be 0.98.

It is easy to make sure that modeling according to the *Ys=y"3* criterion will lead to the formation of an LF in the form $\overline{X1 \vee X2} = X"1 \cdot X"2$. The probability function of two independent events product according to the theorem on the probability of two independent events product will have the form

$$P_s = Q1 \cdot Q2 = 0.1 \cdot 0.2 = 0.02,$$

where $Qi = 1 - Pi$ is the failure probability of the *i*-th element (*i*=1,2).

2.3 Modeling of systems with separate and whole redundancy

Let the device consist of three independent elements with the following probabilities of failure-free operation: P1=0.8; P2=0.7; P3=0.6

The **task** is to compare two options for redundancy – separate and common.

Before drawing up calculation schemes, it is necessary to delete the dummy node 3 on the previous scheme (Figure 36) using the button Delete . In this case, it can be seen that two **AND** edges are also removed.

2.3.1 Compiling a FIS of the system elements initial state

The procedure for drawing up a scheme includes the following steps:

1 Add a third system element by activating the button ("Functional node"). Place a new functional node 3 to the right of nodes 1 and 2.

2 Use the Select button to align all nodes horizontally.

3 By activating the button (**OR** edge) or the button (**AND** edge), display the series connection of three initial elements (Figure 37).

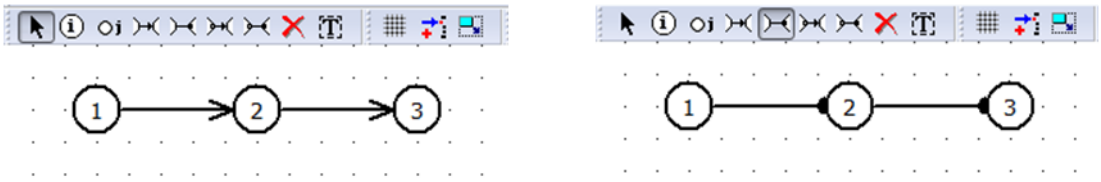


Figure 37 – Serial connection of the initial elements

To change the initial data, perform the following steps:

1 Double-click in the cell "*Pi*" of the line *i*=1 of the Parameter Table to activate the cell for entering new initial data. Enter the value P1=0.8 into this cell using the numeric keys.

2 Similarly, enter the initial data in the corresponding cells 2 (P2=0.7) and 3 (P3=0.6) (Figure 38).

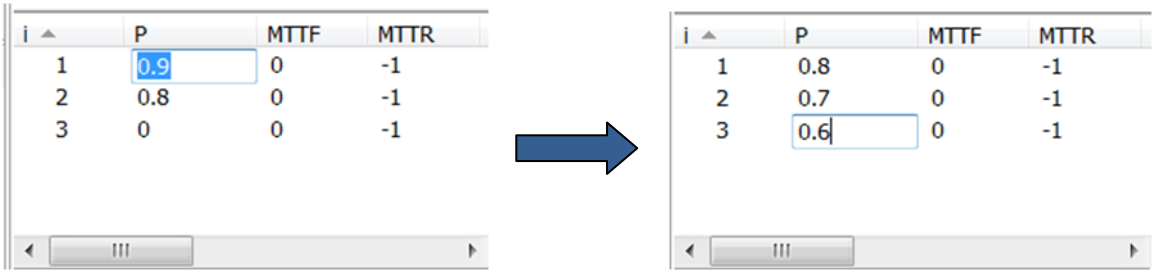


Figure 38 – Data entry for serial connection of initial elements

For a preliminary assessment of the non-failure operation probability of a non-redundant system, we add a dummy node, assigning it the number 11 and giving it the gray color (item No. 1.3.3). Then we connect the functional node 3 with the dummy node 11 with an **OR** edge (Figure 39).

To obtain modeling results, you should:

1 Change in the table "Criterion" LC – enter *y11*.

2 Press the button ("Modeling and calculation").

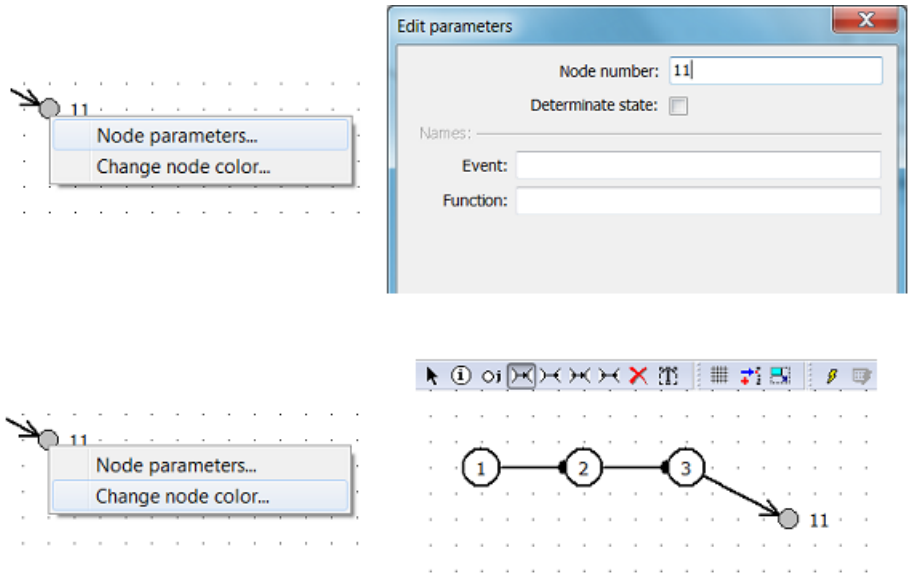


Figure 39 – Building a serial connection of the initial elements

The results shown in Figure 40 will appear on the Report tab.

Result | Report

```
====  Results of Main Graph:
=======================================================================

      FIS parameters:
Total number of nodes      -  N=4
Number of system elements -  H=3

Criterion of functioning

Ys= y11

Logic function consists of 1/1 conjunctions

# conj.  P conj.   F-V conj.    LF
  1  3.3600E-001  1.0000E+000   X1 X2 X3

Probability function consists of 1 monomials

Ps=  P1 P2 P3



Statical calculations :
-----------------------
P=         0.336 - probability of criterion realization
```

Figure 40 – The calculation results of the failure-free operation probability of the initial connection of three elements

2.3.2 FIS create for a Whole Redundancy System

Whole redundancy is the redundancy of the entire system. To do this, it is necessary to create a FIS of a series-parallel structure. The composition of this system working elements is shown in Figure 39.

To create a FIS of the whole redundancy system, the following steps should be performed.

1 Add 3 functional nodes by activating the button ⓘ. The input of a functional node occurs with each left click.

2 Connect sequentially new functional nodes 4, 5 and 6 with **AND** edges, as was done above with nodes 1, 2 and 3.

3 Add a dummy node by activating the button oj and give it the number 12. Give this node the blue color.

4 Connect functional nodes 3 and 6 with dummy node 12 by **OR** edges (Figure 41).

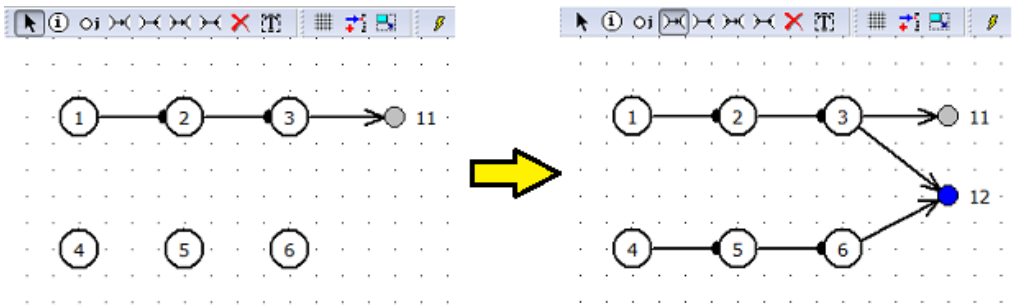


Figure 41 – Create a whole redundancy scheme

To assess the reliability value (failure-free operation probability) of a whole redundancy system, the following steps should be performed:

1 In the Parameter table in cells P4, P5 and P6, enter the reliability values of working elements, that is, P4 = 0.8; P5=0.7 and P6=0.6.

2 In the Criterion table on the next line after the record *y11*, enter the record of the new criterion *y12* (Figure 42).

After pressing the button ("Modeling and calculation"), the results shown in Figure 42 will appear on the Report tab.

In terms of Boolean algebra functions for monotone structures, the operability conditions are written using minimal paths. A minimal path is such a conjunction (logical product) of elements, none of whose components can be removed without breach conditions of the system functioning.

Recording LF in the form of disjunctive normal form (DNF) has the form: $X1X2X3 \vee X4X5X6$, which corresponds to the presence of two minimal paths in this scheme.

The probability function for the above DNF can be obtained from the formula for the probability of the independent events sum.

$$P\{Ys = 12 = True\} = P\{X1X2X3 \vee X4X5X6 = True\} =$$
$$= P1P2P3 + P4P5P6 - P1P2P3P4P5P6$$

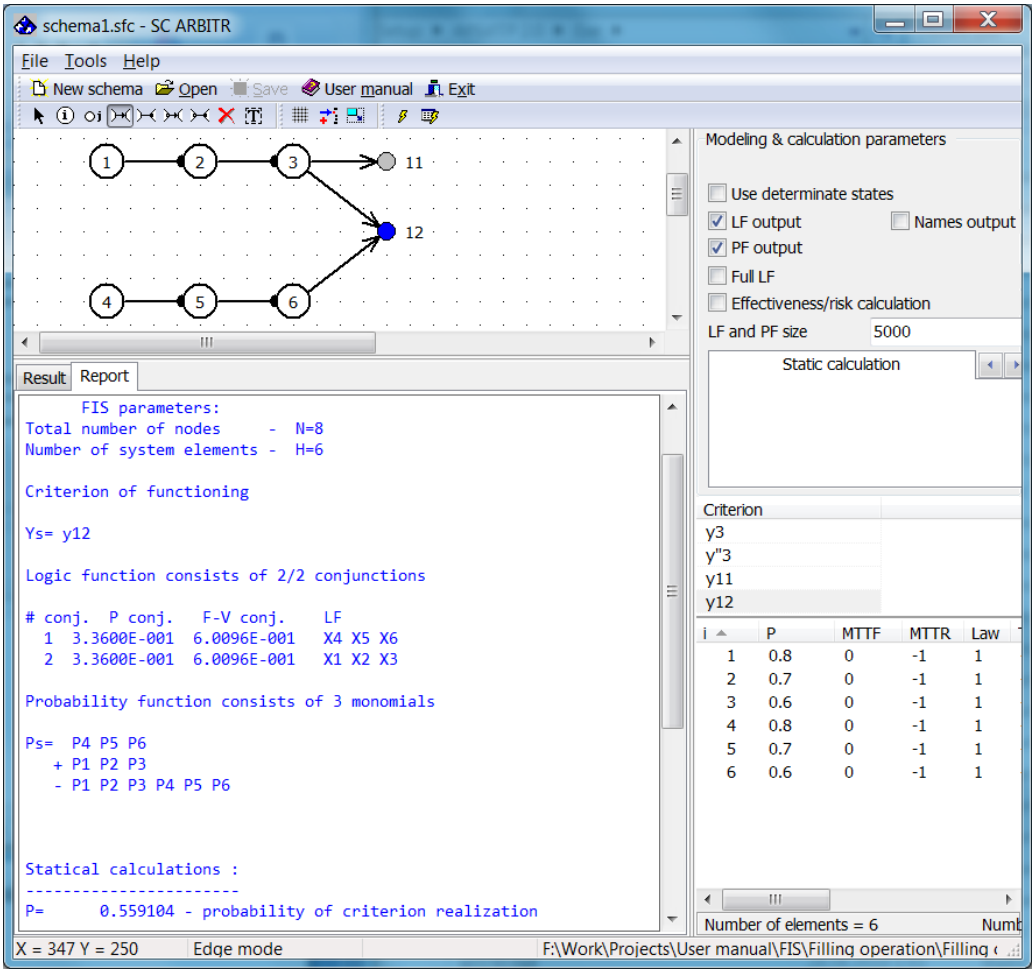


Figure 42 – The calculating results the non-failure operation probability of the system with whole redundancy

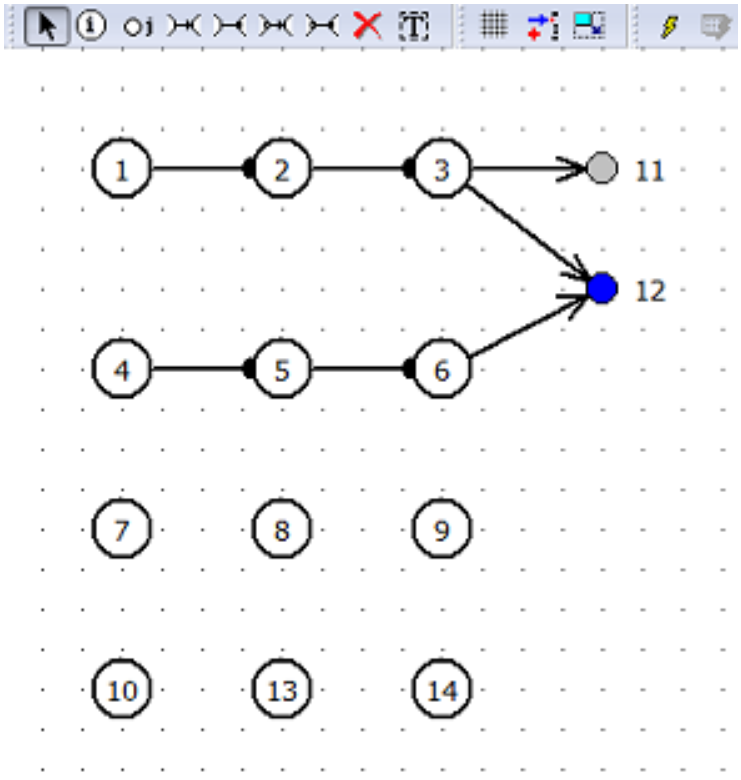
2.3.3 FIS create for a Separate Redundancy System

Since each element of the system is reserved with separate redundancy, it is necessary to create a FIS of a parallel-serial structure, the composition of the working elements of which is shown in Figure 41.

For clarity of comparison of the modeling results a non-redundant system, systems with whole and separate redundancy, we will use the procedure of "nodes multiplication", that is, we will compose the FIS of a separate redundant system on the same screen and from the same elements that were used earlier for whole redundancy.

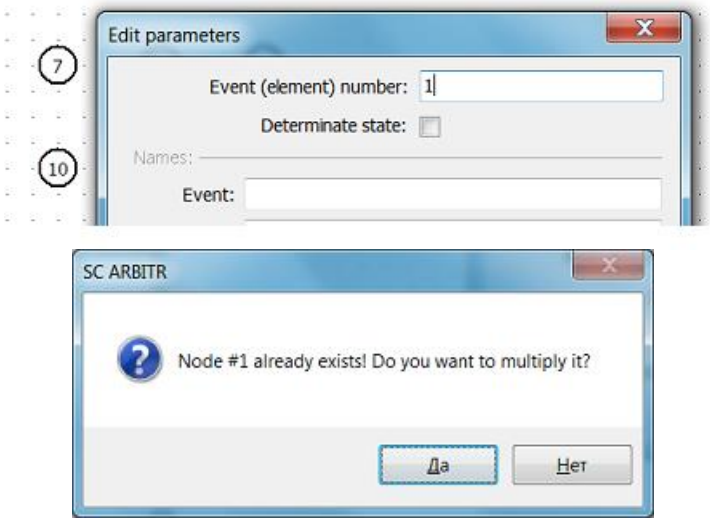
To this end, the following steps should be taken.

1 Add 6 functional nodes by activating the button ⓘ, as in the previous example (Figure 41). The input of a functional node occurs with each left click.



Change parameters of new functional nodes. They need to be given numbers 1–6.

2 For the new node No. 7, in the line "Event (element) number", put the number "1" and press the OK button. In the message that appears, "Node #1 already exists! Do you want to multiply it? ", press the Yes (Да) button.



3 Perform steps 2 for the remaining five new nodes.

After performing the indicated actions, an element base will be created on the scheme input field for compiling the FIS of a system with separate redundancy (Figure 43).

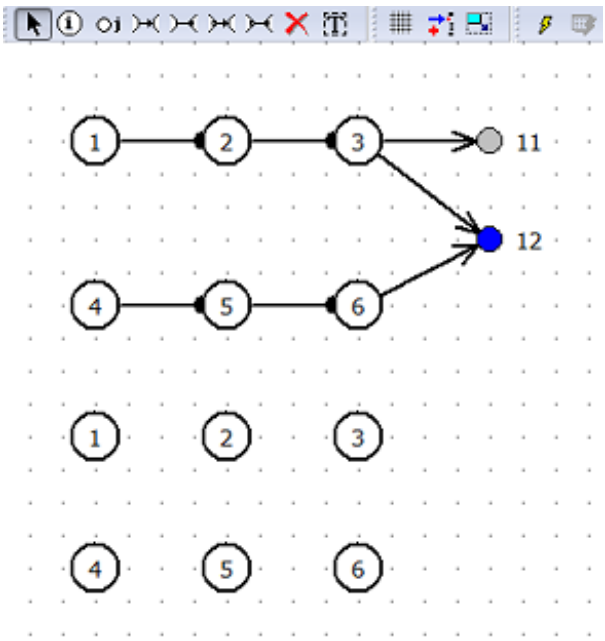


Figure 43 – Preparation of the scheme for the FIS system with separate redundancy

Compiling a FIS of the system with separate redundancy can be done in several ways, two of which are shown in Figure 44.

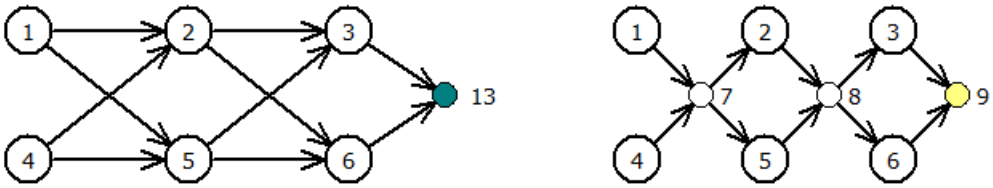


Figure 44 – Methods for compiling a FIS of the system with separate redundancy

Let's delete the dummy node 11 and all the edges in the upper part of the diagram. Let's change the number of the dummy node from 12 to 13. Add three dummy nodes 7, 8 and 9 at the bottom of the scheme. Let's connect the nodes with edges, as shown in Figure 44.

Let's enter in the "Criterion" table LC *y13* and *y9* and perform modeling and calculation. It is easy to make sure that the criteria *y13* and *y9* correspond to the same LF and PF, which can be found on the Report tab (Figure 45).

Result | Report

```
Logic function consists of 8/8 conjunctions

# conj.  P conj.    F-V conj.    LF
  1   3.3600E-001  4.5788E-001   X2 X3 X4
  2   3.3600E-001  4.5788E-001   X4 X5 X6
  3   3.3600E-001  4.5788E-001   X1 X5 X6
  4   3.3600E-001  4.5788E-001   X1 X2 X6
  5   3.3600E-001  4.5788E-001   X1 X2 X3
  6   3.3600E-001  4.5788E-001   X2 X4 X6
  7   3.3600E-001  4.5788E-001   X3 X4 X5
  8   3.3600E-001  4.5788E-001   X1 X3 X5

Probability function consists of 8 monomials

Ps=  P4 P5 P6
   + P1 Q4 P5 P6
   + P1 P2 Q5 P6
   + Q1 P2 P4 Q5 P6
   + P1 P2 P3 Q6
   + Q1 P2 P3 P4 Q6
   + Q2 P3 P4 P5 Q6
   + P1 Q2 P3 Q4 P5 Q6
```

Figure 45 – LF and PF for a system with separate redundancy

The DNF type indicates that a single redundancy system includes a minimum of 8 paths.

The probability function *Ps* was obtained by a combined method that combines the advantages of the orthogonalization method and the application of the formula for the events sum probability.

The numerical result for a system with separate redundancy *Ps*=0.733824 confirms the advantage of separate redundancy in comparison with the whole redundancy (*Ps*=0.559). Both redundancy methods significantly increase the reliability of a non-redundant system (*Ps*=0.336).

3 Bridge Circuit Reliability Modeling

Bridge circuits belong to the class of modeling complex structures tasks. The conditions of operation (failure) these structures are not reduced to a simple combination of the operation (failure) conditions of serial and parallel structures. An example of a bridge circuit can be a fragment of the radio receiving system shown in Figure 46.

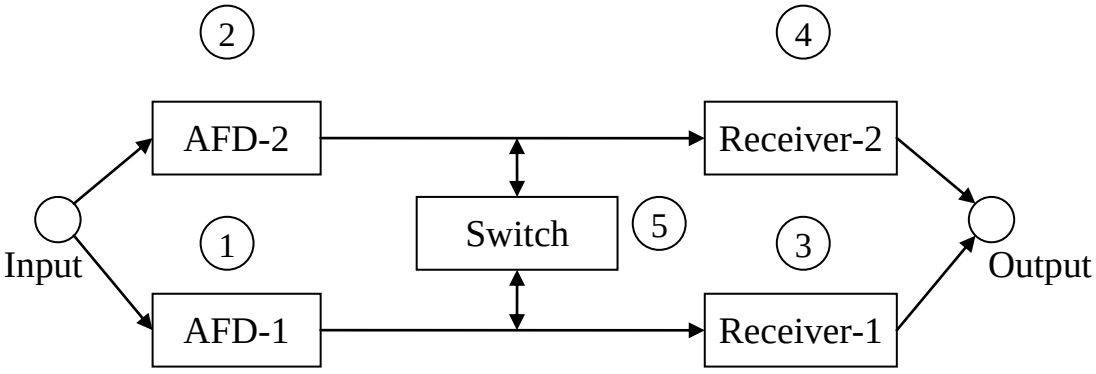


Figure 46 – The radio receiving system

In order to improve the reliability of the system, the input signal is received by a duplicated antenna-feeder system (AFD-1 and AFD-2). The signal from the antenna devices can be fed to the receivers of its own channel (Receiver-1 and Receiver-2) or through the Switch to the receivers of the adjacent channel. Depending on the tasks to be solved, the operating conditions may require the operability of both any of the two receivers, and two receivers simultaneously.

3.1 Compiling a FIS of the bridge circuit

To compile the FIS of the bridge circuit, the following steps should be performed.

1 Place the nodes in the FIS input window as shown in the left part of Figure 47.

2 Connect the nodes with OR edges as shown on the right side of Figure 47.

3 Using the "Text" button, enter the appropriate inscriptions.

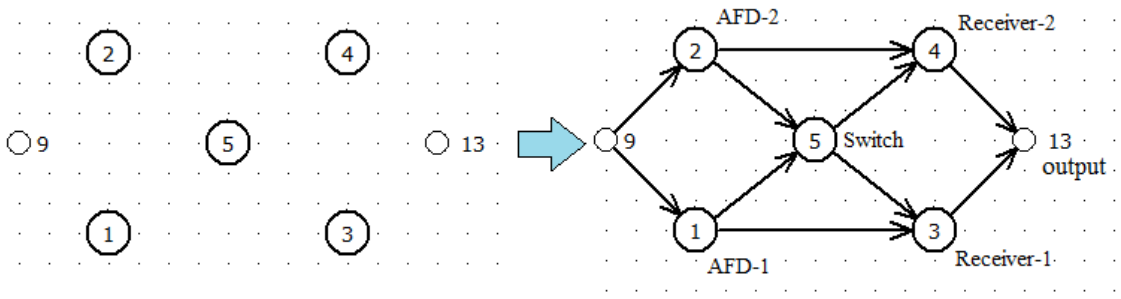


Figure 47 – Compiling a FIS of the bridge circuit

The use of dummy nodes No. 9 and No. 13 serves for the usual graphical representation of a two-terminal network with one input and one output and is not mandatory from the point of view of solving the tasks.

The FIS in the right part of Figure 47 allows modeling the bridge circuit reliability for any LC. Enter the following entries in the Criteria table:

- *y13* – LCF "Receiving a signal from at least one receiver";
- *y3+y4* – LCF "Receiving a signal from either receiver-1 or receiver-2" (without using an additional dummy node);
- *y3y4* – LCF "Signal reception from receiver-1 and receiver-2".

To carry out a numerical calculation, we introduce the following elements' reliability functions into the Parameter table (we omit the parameter t in the notation of probabilities for brevity):

p1 = 0.9; p2 = 0.9; p3 = 0.95; p4 = 0.95; p5 = 0.8.

The results of entering the initial data are shown in Figure 48.

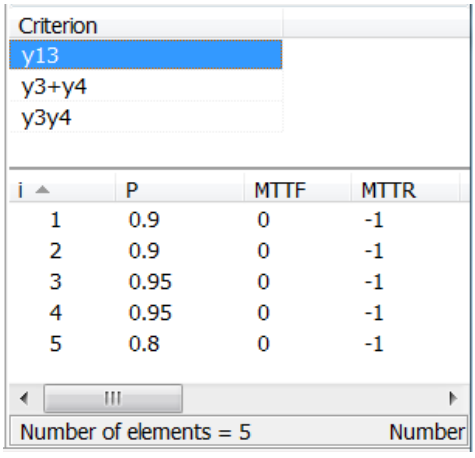


Figure 48 – Entering initial data for the bridge circuit

3.2 Calculation of reliability functions and results analysis

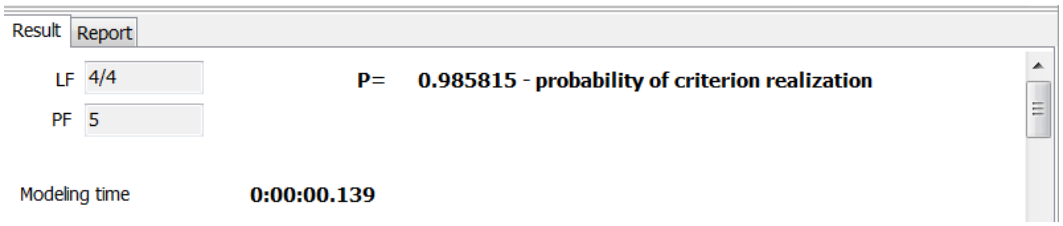
3.2.1 Calculation and results analysis for LCF *Ys = y13*

After entering the criterion *y13* and pressing the button  ("Modeling and calculation"), make sure that:

• On the Result tab (in the lower left part of the screen) the information shown in Figure 49, a appeared.

• On the Report tab (in the lower left part of the screen) the information shown in Figure 49, b appeared.

a)



b)

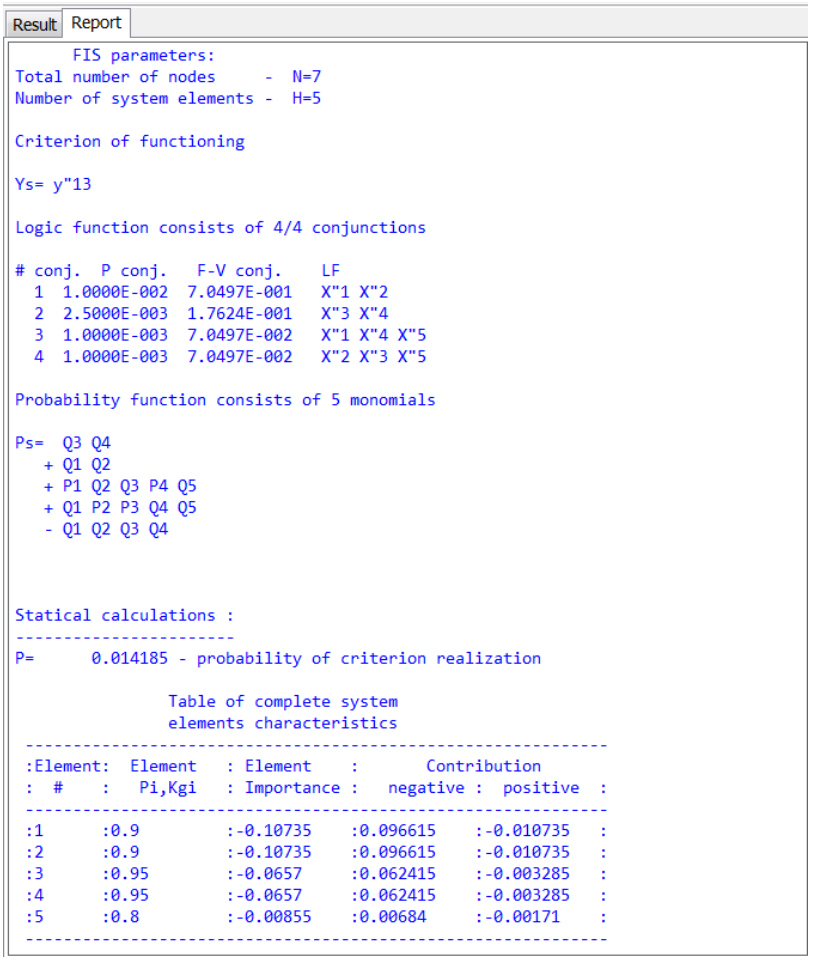


Figure 49 – Information on the Result tab a) and the Report tab b)

The report provides the following information:

1 FIS parameters (number of nodes and number of circuit elements). In our case, the number of nodes (functional and dummy) is 7, and the number of elements (functional nodes) is 5.

2 Logical criterion of functioning (LCF). The criterion by which the current calculations were carried out is displayed. In our case LCF is *y13*.

3 Logical function (LF) contains 4 conjunctions – the report section contains information on the number of conjunctions (terms) in the LF.

In operability measure modeling, the right column of the table consists of bridge circuit ***minimal paths***.

Definition: a path set in a RBD is a set of basic events whose occurrence ensures that the TOP event occurs (operability state). A path set is said to be minimal (MPS or MP) if the set cannot be reduced without losing its status as path set.

In our case, the logical function that determines the conditions of a given criterion in the traditional form can be written in the following DNF form:

$$Ys = y13 = X1X3 \vee X2X3 \vee X1X4X5 \vee X2X3X5.$$

4 **The probabilistic function** contains 5 monomials – the report section contains information about the polynomial of the probabilistic function.

This polynomial can be used as an analytical expression for estimating the reliability function of a bridge circuit.

In our case

$$Ps = P1Q2Q3P4P5 + P1P3 + P2P4 + Q1P2P3Q4P5 - P1P2P3P4,$$

when *Pi* – the *i*-th element reliability function;

Qi – the *i*-th element probability of failure, *Qi=1–Pi*.

5 **Static calculations**: the report section contains information about the quantitative results of the modeling.

In our case, P = 0.985815 is the probability of the criterion being implemented.

6 **The element characteristics table** provides information about the initial data:

- Column "Element Pi, Kgi" contains information about the event probability (reliability or failure).
- Column "Contribution negative" provides data on how much the system measure will change if the probability of an event occurring increases to 1.0.
- The "Contribution positive" column provides data on how much the system measure will change if the probability of an event occurring drops to 0.
- The "Element Importance" column represents the element's importance, which is calculated as the sum of the absolute values of the element's contributions.

3.2.2 Calculation and results analysis for LCF *Ys = y3 + y4*

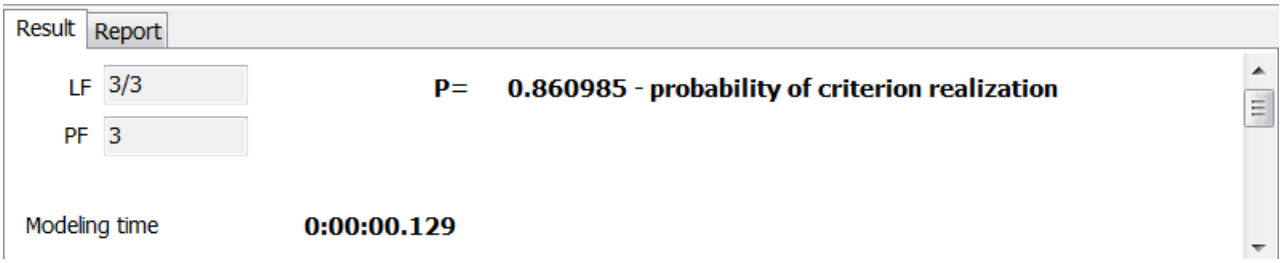
1 In the "Criterion" table, enter the logical expression *y3+y4* and press the button ("Modeling and calculation").

2 Make sure that all report data matches the data obtained for criterion *y13*.

3.2.3 Calculation and results analysis for LCF *Ys = y3y4*

1 In the "Criterion" table, enter the logical expression *y3y4* and press the button ("Modeling and calculation").

2 Make sure that the following results appear on the Result tab:



3 Make sure that the following information appears on the Report tab:

Result | Report

```
====  Results of Main Graph:
=======================================================================

      FIS parameters:
Total number of nodes      -  N=7
Number of system elements  -  H=5

Criterion of functioning

Ys= y3y4

Logic function consists of 3/3 conjunctions

# conj.  P conj.    F-V conj.    LF
  1  7.3102E-001  8.4906E-001   X1 X2 X3 X4
  2  6.4980E-001  7.5472E-001   X2 X3 X4 X5
  3  6.4980E-001  7.5472E-001   X1 X3 X4 X5

Probability function consists of 3 monomials

Ps=  P1 P2 P3 P4
   + Q1 P2 P3 P4 P5
   + P1 Q2 P3 P4 P5



Statical calculations :
-----------------------
P=      0.860985 - probability of criterion realization

               Table of complete system
               elements characteristics
 ------------------------------------------------------------
 :Element:  Element   : Element    :       Contribution
 :  #    :   Pi,Kgi   : Importance :   negative :  positive  :
 ------------------------------------------------------------
 :1      :0.9         :0.23465     :0.21118     :0.023465    :
 :2      :0.9         :0.23465     :0.21118     :0.023465    :
 :3      :0.95        :0.9063      :0.86099     :0.045315    :
 :4      :0.95        :0.9063      :0.86099     :0.045315    :
 :5      :0.8         :0.16245     :0.12996     :0.03249     :
 ------------------------------------------------------------
```

Comparison for solutions by criteria *Ys=y13* and *Ys=y3y4* of contributions and significances can be performed according to the diagrams in the section "Diagrams and chats" located on the Result tab below the simulation time information. The positive contributions diagrams according to the *Ys=y13* and *Ys=y3y4* criteria are shown in Figures 50 and 51, respectively.

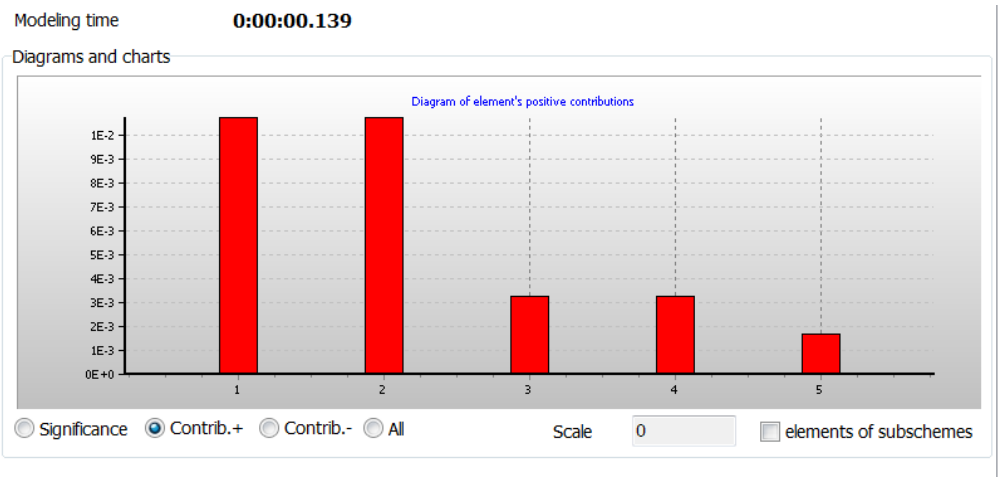


Figure 50 – The elements' positive contributions for the solution according to the criterion *Ys=y13*

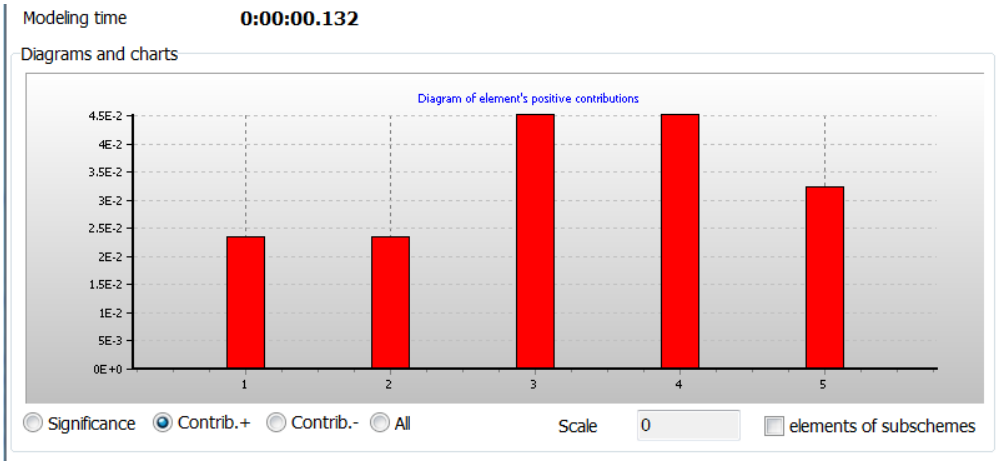


Figure 51 – The elements' positive contributions for the solution according to the criterion *Ys=y3y4*

3.3 Building a bridge circuit fault tree

A fault tree (FT) is an organized graphical representation of the conditions or other factors causing or contributing to the occurrence of a defined outcome, referred to as the "top event" (IEC 61025:2006, item No. 5.1).

Basic events in the construction of FT are circuit elements failures. Therefore, to draw up the FT of a bridge circuit, it is necessary to determine the minimal cut sets of the structure.

The minimal cut sets are the minimum sets of basic events needed to occur to cause the top event.

To find the minimal cut sets, it is necessary to perform an inverse solution of the previous task, i.e., determine the bridge circuit probability of failure. If, in the probabilistic sense, the probability of system failure is the inverse of the reliability function, that is, *Q(t)=1–P(t)*, then in logical terms, finding the opposite outcome is carried out using the unary operation of inversion (negation).

For this purpose, in the SC ARBITR, in the "Criterion" table, we will enter the LCF "*y"13*" as shown in Figure 52.

The notation "*y"13*" is equivalent to the algebraic notation for inversion "$\overline{y13}$".

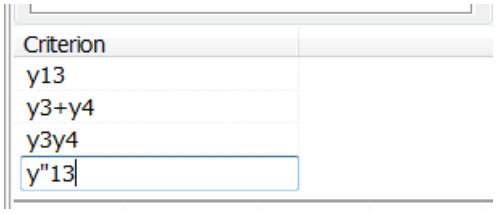
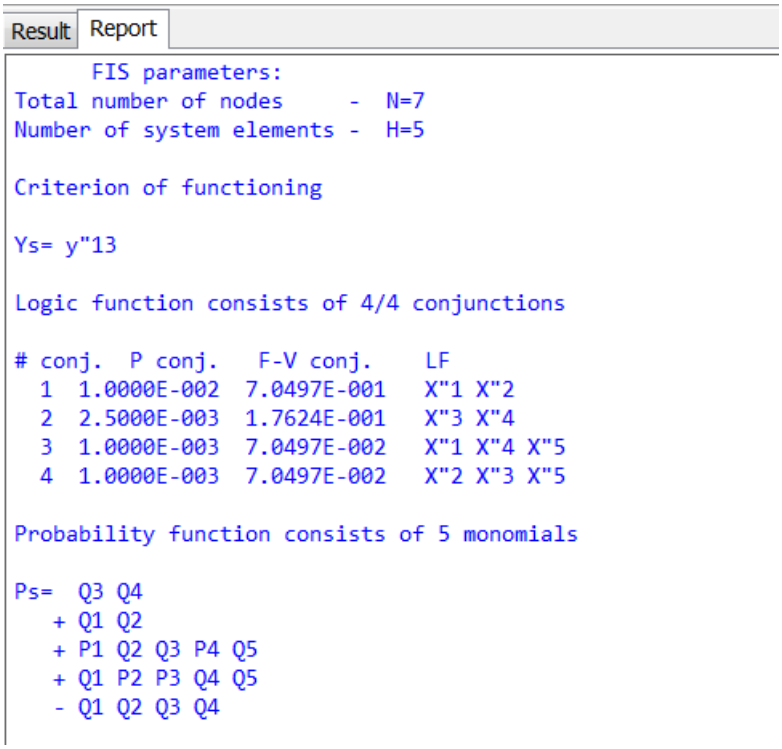


Figure 52 – Entering a criterion for solving the inverse task

At the top of the Report tab, an entry will appear:



The information in the table last column allows us to write the failure conditions of the bridge circuit in disjunctive normal form (DNF):

$$y''13 = X''1\ X''2 \vee X''3\ X''4 \vee X''2\ X''3\ X''5 \vee X''1\ X''4\ X''5.$$

The resulting DNF makes it possible to construct a FT, the top event of which is provided by the logical addition (disjunction) of four terms (conjunctions) – minimal cut sets (MCS).

Figure 53 shows an example of building a fault tree using standard graphical tools (IEC 61025:2006).

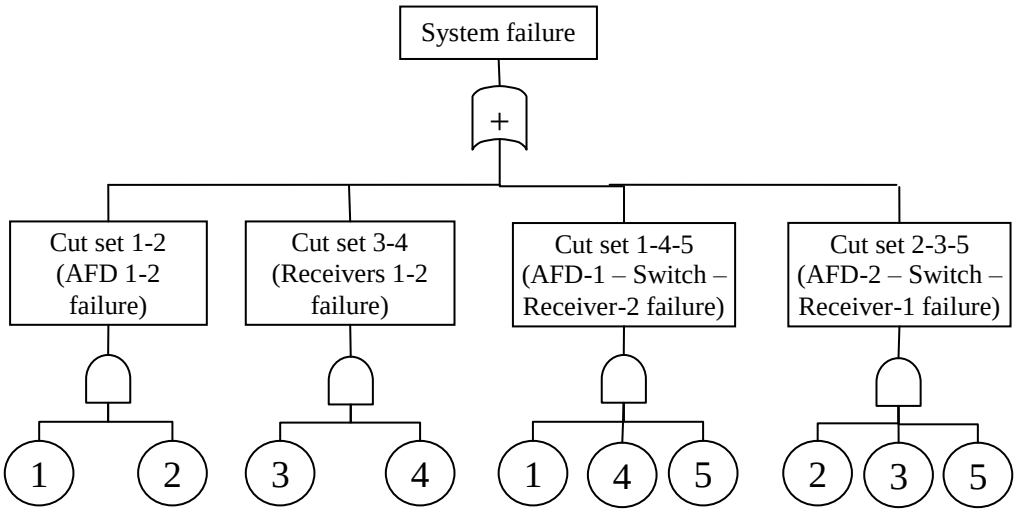


Figure 53 – FT of the bridge circuit in standard notation

When using the graphical tools of the SC ARBITR, there are several options for solving the task.

In the first case, the technology of "multiplied" events is used. Then the FT is created as follows:

1 Functional nodes are placed on the same line, corresponding to the terms of the expression for *y"13*. In this case, when creating a node with an existing number, a message appears confirming the node duplication (item No. 1.6), which must be answered in the affirmative.

2 After placing all the necessary elements on the FIS field, four dummy nodes (according to the number of disjunctions) are created to form the necessary terms

(intermediate events). At the same time, it should be remembered that the terms include negations of logical variables, so you should use the **NOT-AND** type edges.

3 The top event is formed by a dummy node, in which the previously created conjunctions are disjunctively combined.

A possible variant of the FIS for the bridge circuit fault tree is shown in Figure 54.

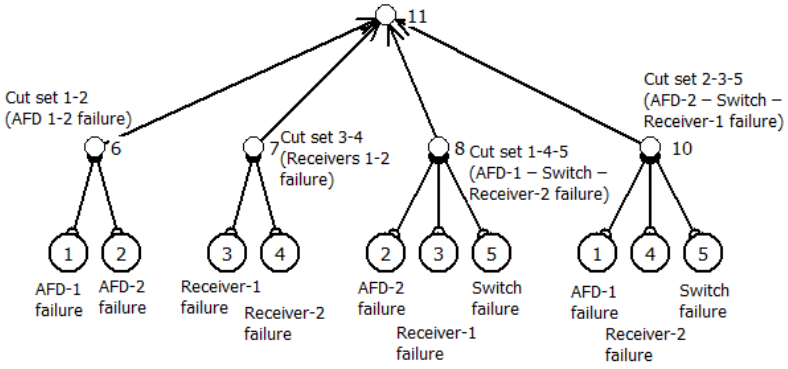


Figure 54 – The FIS of a bridge circuit in the form of a FT with nodes duplication

In the second case, the graphical tools of the SC ARBITR allow constructing an unconventional scheme, in which several logical connections can come from the basic event.

An example of construction is shown in Figure 55.

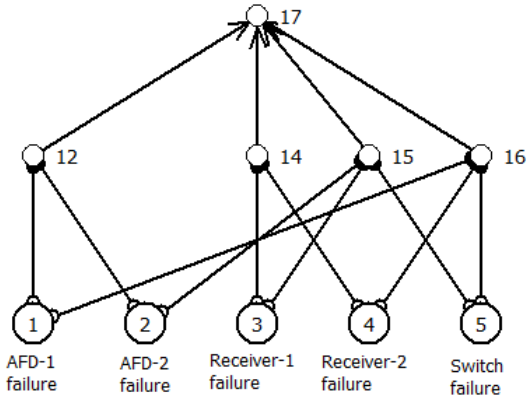


Figure 55 – The FIS of a bridge circuit without nodes duplication in the unconventional fault tree form

To analyze the indicators of the restorable bridge circuit, the following initial data should be entered:

1 Elements reliability measures in the form of mean time to failure (column "MTTF" of the initial data table) in years.

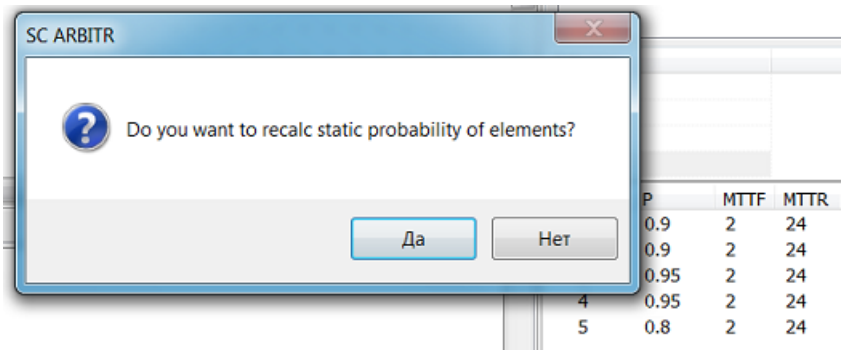
2 Elements maintainability measures in the form of mean time to restoration (column "MTTR" of the initial data table) in hours.

3 In the parameter and mode input box (Figure 4) set the checkbox on the "Use time to recover".

4 Set the required value of the system operating time (in hours).

5 If necessary, recalculate the probability *P*. To do this, press the Ctrl key and, without releasing it, left-click the mouse on any cell of the "*P*" column, then confirm the recalculation.

An example of entering initial data for a bridge circuit is shown in Figure 56, a and b.



a

i	P	MTTF	MTTR
5	0.998632010943912	2	24
1	0.998632010943912	2	24
2	0.998632010943912	2	24
3	0.998632010943912	2	24
4	0.998632010943912	2	24

b

Figure 56 – Initial data for the analysis of the restorable bridge circuit

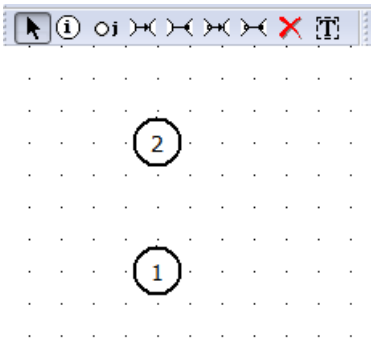
4 Additional options

4.1 Modeling system dependability using equivalent nodes

The term "reduction" in mathematical tasks defines ways to reduce the task dimension or the choice of a short form of the source data representation, such as a scheme or structure.

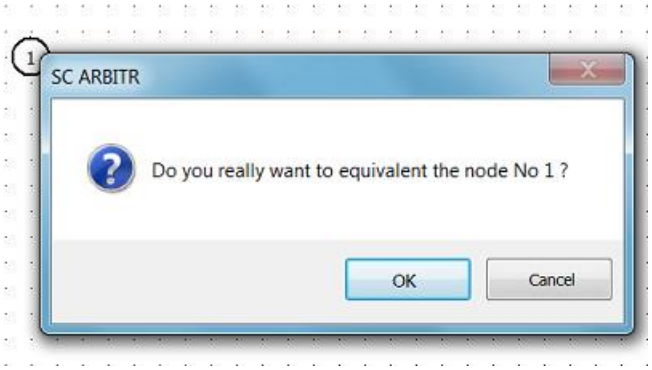
In SC ARBITR, it is possible to reduce the representation of the original FIS of a complex or large-sized structure. For this, the apparatus of equivalent nodes is used.

Create a FIS from two functional nodes.



Make each node equivalent (creating an equivalent node is described in item No. 1.7.1).

Consider an example of using equivalent nodes to calculate the reliability of a system with whole redundancy (Figure 41). First, let's create an equivalent node No. 1.



In subschema 1, we will create a scheme corresponding to the non-redundant part of the system with whole redundancy (Figure 57). Then we will create an equivalent

node 2 and construct a subschema corresponding to the second non-redundant part of the system with a whole redundancy (Figure 58).

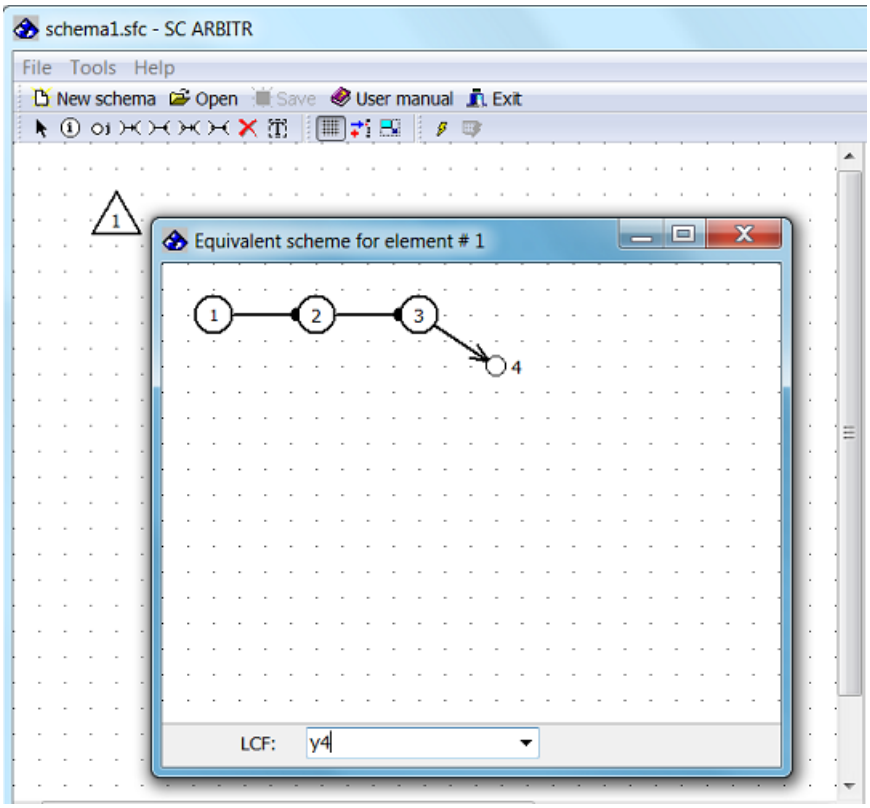


Figure 57 – Creating the FIS of equivalent node 1

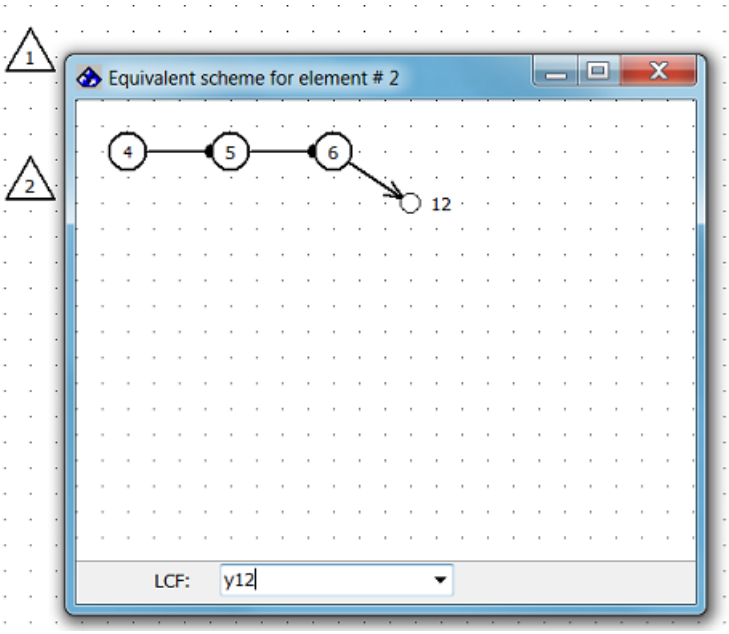


Figure 58 – Creating the FIS of equivalent node 2

In the main window, we'll complete the construction of a system scheme with whole redundancy using equivalent nodes (Figure 59).

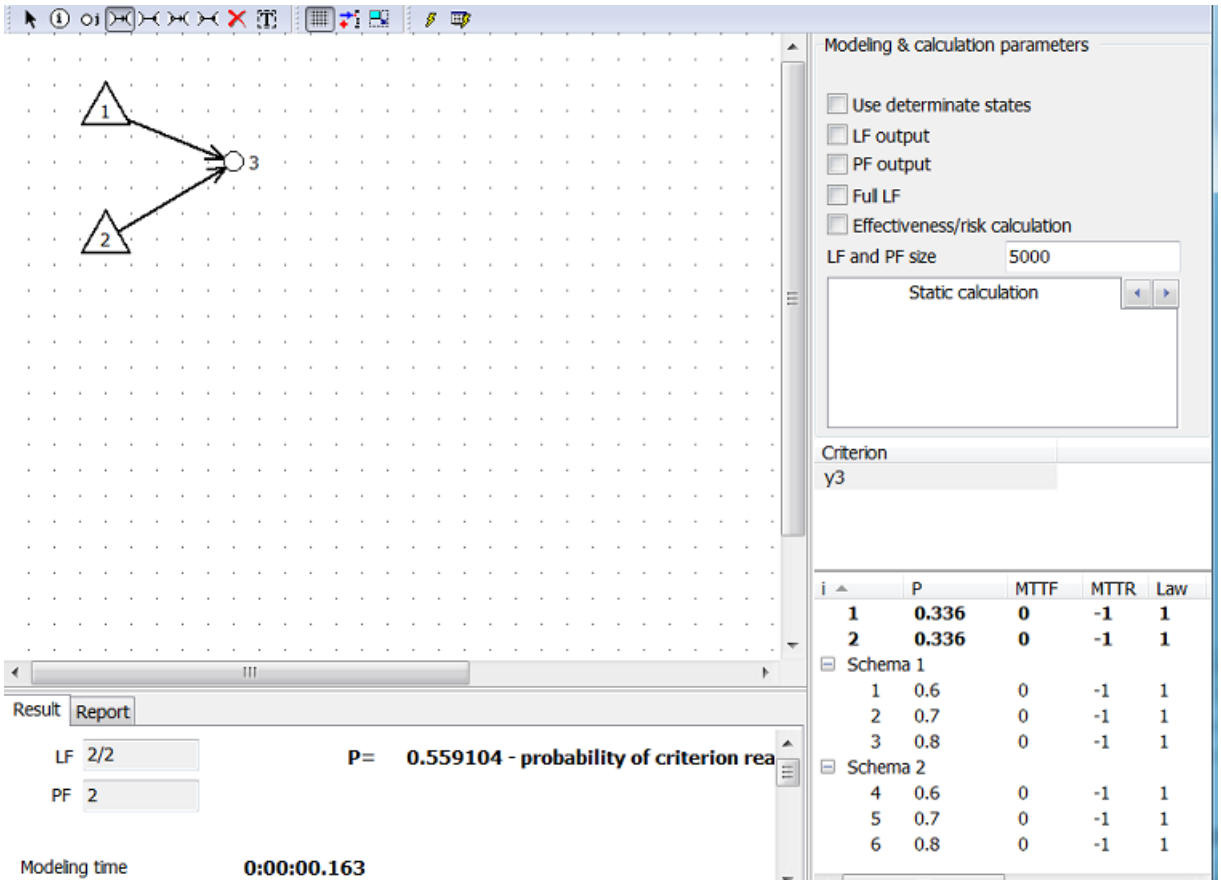


Figure 59 – FIS of the system with whole redundancy using equivalent nodes

Reducing the elements number when presenting a design model on an interface screen is sometimes called a reduction operation.

As can be seen from Figure 59, the results of modeling a system with whole redundancy using equivalent nodes coincide with the results shown for the non-reduced structure in Figure 42.

4.2 Reliability modeling of "K out of N" structures

To model the reliability of "K out of N" structures, it is possible to compose three types of FIS.

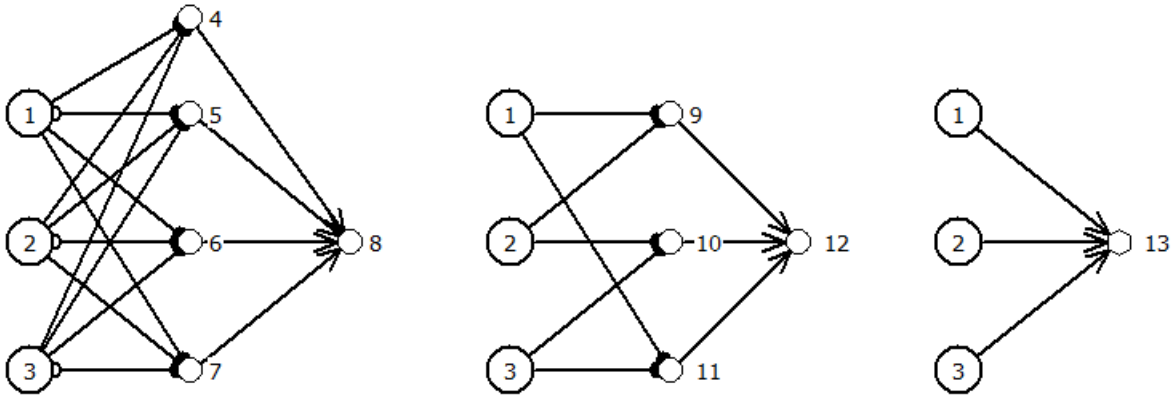
4.2.1 Compiling a complete FIS

Compilation of a complete FIS is based on the reproduction of a logical function of a given structure in the form of an unreduced disjunctive normal form (DNF).

For example, for a "2 out of 3" structure, such a DNF looks like this:

$$Y = X_1X_2X_3 \vee \bar{X}_1X_2X_3 \vee X_1\bar{X}_2X_3 \vee X_1X_2\bar{X}_3.$$

Figure 60,a shows the FIS that implements this DNF (LCF *y8*).



a b c

Figure 60 – Variants of the FIS for structure "2 out of 3"

4.2.2 Compiling a minimum DNF

Application of the Boolean algebra reduction and absorption rules, as well as analysis of the LCF *y8* solution report show that the reduced (minimal) DNF in this case has the form:

$$Y = X_1X_2 \vee X_2X_3 \vee X_1X_3.$$

Figure 60,b shows the FIS that implements this reduced DNF (LCF *y12*).

The most convenient way to compose a FIS for modeling the reliability of "K out of N" structures is to use the "Node parameters..." option of a dummy node. Figure 61 shows the pop-up window "Node parameters..." when you right-click the mouse for the dummy node No. 13. In the line "Determinate state" in the cells "K"

and "N", the required parameters of the structure "K out of N" are entered. In our example, there is "2 out of 3".

After entering the structure parameters, the dummy node will have the shape of a hexagon (Figure 60, c).

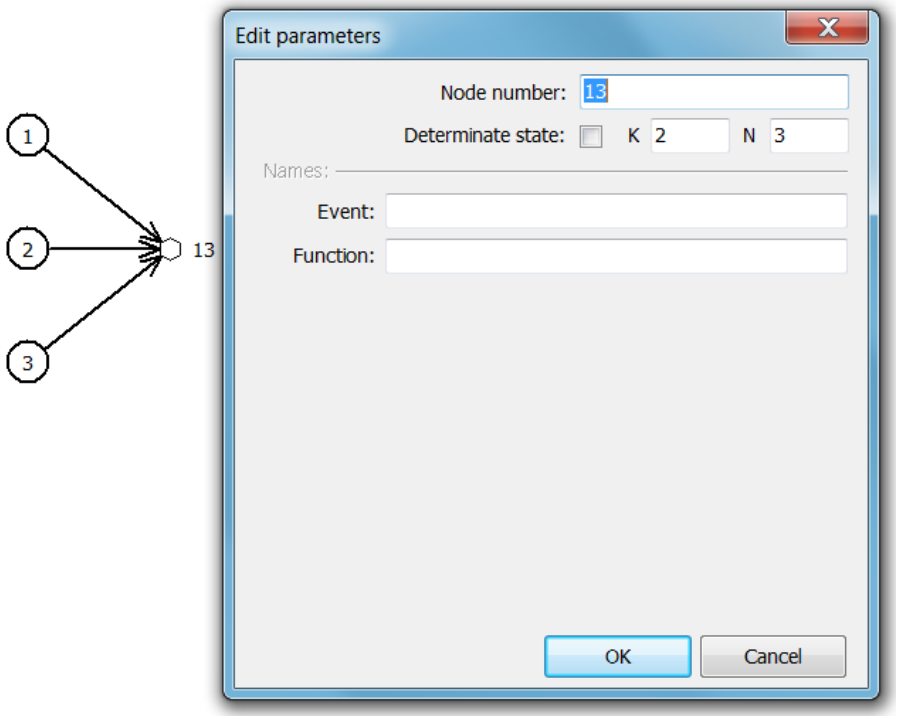


Figure 61 – Pop-up window "Node parameters..."

Attention: The implementation of the "K out of N" structure with K=N is equivalent to the implementation of the elements' conjunctive connection (connection by **AND**) and is not used to avoid visual perception of such a structure as connections by **OR**.

4.3 Applying the element parameter "Element multiplicity"

The description of the element parameter "Element multiplicity" is given in item No. 1.3.1. Using this parameter allows you to significantly reduce the number of functional nodes, reflecting a serial or parallel connection of identical elements in terms of reliability.

Let's look at a few examples.

Example 1. A serial elements connection.

The circuit section consists of a series (in terms of reliability) connection of 5 resistors and 10 capacitors. The reliability function of the resistor is *Ri*=0.97, and of the capacitors is *Rj*=0.94.

Figure 62 shows two ways of the described structure graphic representation.

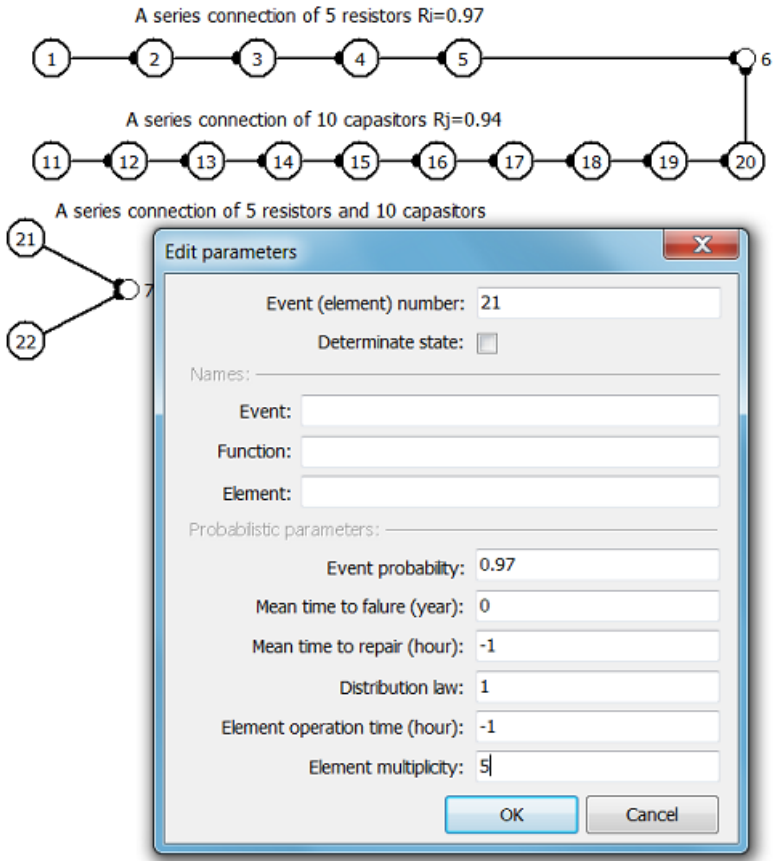


Figure 62 – Applying the "Multiplicity" option for a serial structure

The upper part of Figure 62 shows the circuit without using the "Multiplicity" parameter. The lower part of the figure shows the same structure using the "Multiplicity" parameter (the figure shows the window for changing for the element No. 21 parameters, in which the multiplicity of the element "5" is indicated in the bottom line). It should be noted that the nodes with the multiplicity parameter >1 are indicated as one element in the report (Figure 63).

```
Criterion of functioning

Ys= y7

Logic function consists of 1/1 conjunctions

# conj.  P conj.    F-V conj.     LF
  1  4.6253E-001  1.0000E+000   X21 X22

Probability function consists of 1 monomials

Ps=  P21 P22


Statical calculations :
-----------------------
P= 0.46252712523 - probability of criterion realization
```

Figure 63 – A fragment of the report according to the scheme of Figure 62

Example 2. A parallel elements connection.

The device uses redundancy with a multiplicity of "2/1" (two redundant elements, one main). Reliability function of one element is R=0.6.

Figure 64 shows two ways of the described structure graphic representation.

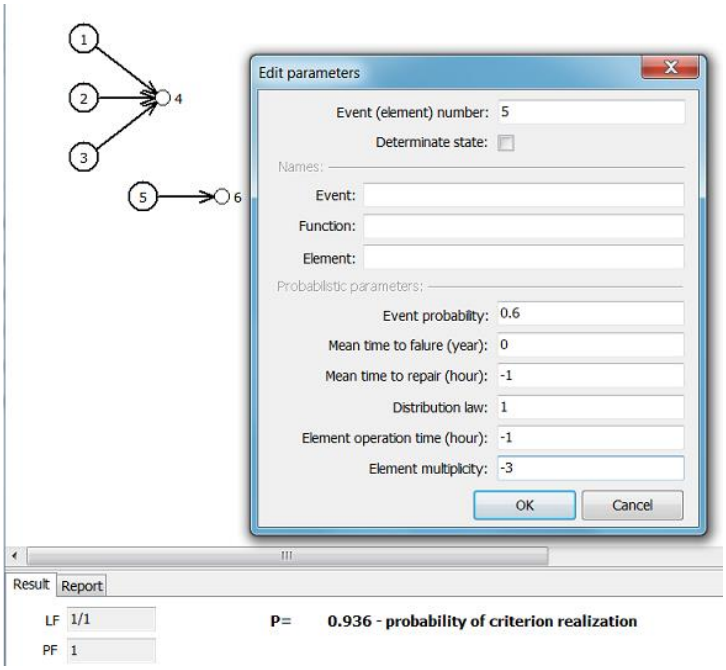


Figure 64 – Applying of the "Multiplicity" parameter for a parallel structure

The right part of the figure shows the window for changing the element No. 5 parameters, in which the element multiplicity "-3" is indicated in the bottom line. It

should be noted that in this case, in the report, the node with the multiplicity parameter <1 is indicated as one element (Figure 65).

```
Criterion of functioning

Ys= y6

Logic function consists of 1/1 conjunctions

# conj.  P conj.   F-V conj.    LF
  1  9.3600E-001  1.0000E+000   X5

Probability function consists of 1 monomials


Statical calculations :
-----------------------
P=         0.936 - probability of criterion realization
```

Figure 65 – A fragment of the report according to the scheme of Figure 64

4.4 Option "Sorting of initial data"

When filling in the element parameter table, by default, the sorting of the initial data is carried out in ascending order of the functional nodes' numbers. In this case, in the cell of the "*i*" column heading, a triangle sign with the top directed upwards ▲ is placed (Figure 66, a).

i ▲	P	MTTF
7	0	33.6
8	0	148
9	0	148
10	0	20.6
11	0	20.6
12	0	150
13	0	150
14	0	150
15	0	150
16	0	14.7
17	0	14.7
18	0	71.5
19	0	71.5
20	0	14.7
21	0	148
22	0	148
77	0	33.6

a

i	P	MTTF ▲
16	0	14.7
17	0	14.7
20	0	14.7
520	0	14.7
10	0	20.6
11	0	20.6
7	0	33.6
77	0	33.6
18	0	71.5
19	0	71.5
118	0	71.5
8	0	148
9	0	148
21	0	148
22	0	148
12	0	150
13	0	150

b

Figure 66 – Sorting of initial data: a – by default by the nodes' numbers;
b – by increasing the mean time to failures (MTTF)

To sort the source data by other parameters, select the header cell of the desired column and left-click in it the mouse (Figure 66, b). When you click again in the header of the selected column, the sorting direction is reversed (the icon appears – sorting in descending order).

4.5 Option "Recalculation of static probability"

To obtain the value of the static probability – the reliability function (excluding the restoration time) or the availability factor of the element (including the restoration time) – after entering the value of the mean time to failures in the dynamic calculation mode, press the Ctrl key and, without releasing Ctrl, left-click the mouse in any cell of the "*Pi*" column (Figure 67, a). The screen will display the inscription "Do you want to recalc static probability of elements?" (Figure 67, b).

When confirming the option, the program will automatically recalculate the value of the mean time to failure:

• in the mode excluding time to recovery – into the element reliability function according to the formula $Pi = exp(-t / MTTFi)$, where t is the system operating time indicated on the "Dynamic calculation" tab of the parameter and mode input box (Figure 67, c);

• in the mode, including time to recovery – to the availability factor according to the formula $Kgi = MTTFi / (MTTFi + MTTRi)$.

Figure 67, d shows a fragment of the screen interface after sorting the initial data in descending order of the elements' reliability function values.

Attention:

1 Recalculation of the static probability when changing the parameter "System time" when performing calculations is not carried out automatically.

2 If the parameter of an element in the "Law" column is equal to "0" (i.e., a static probability is set), then recalculation for this element is not performed.

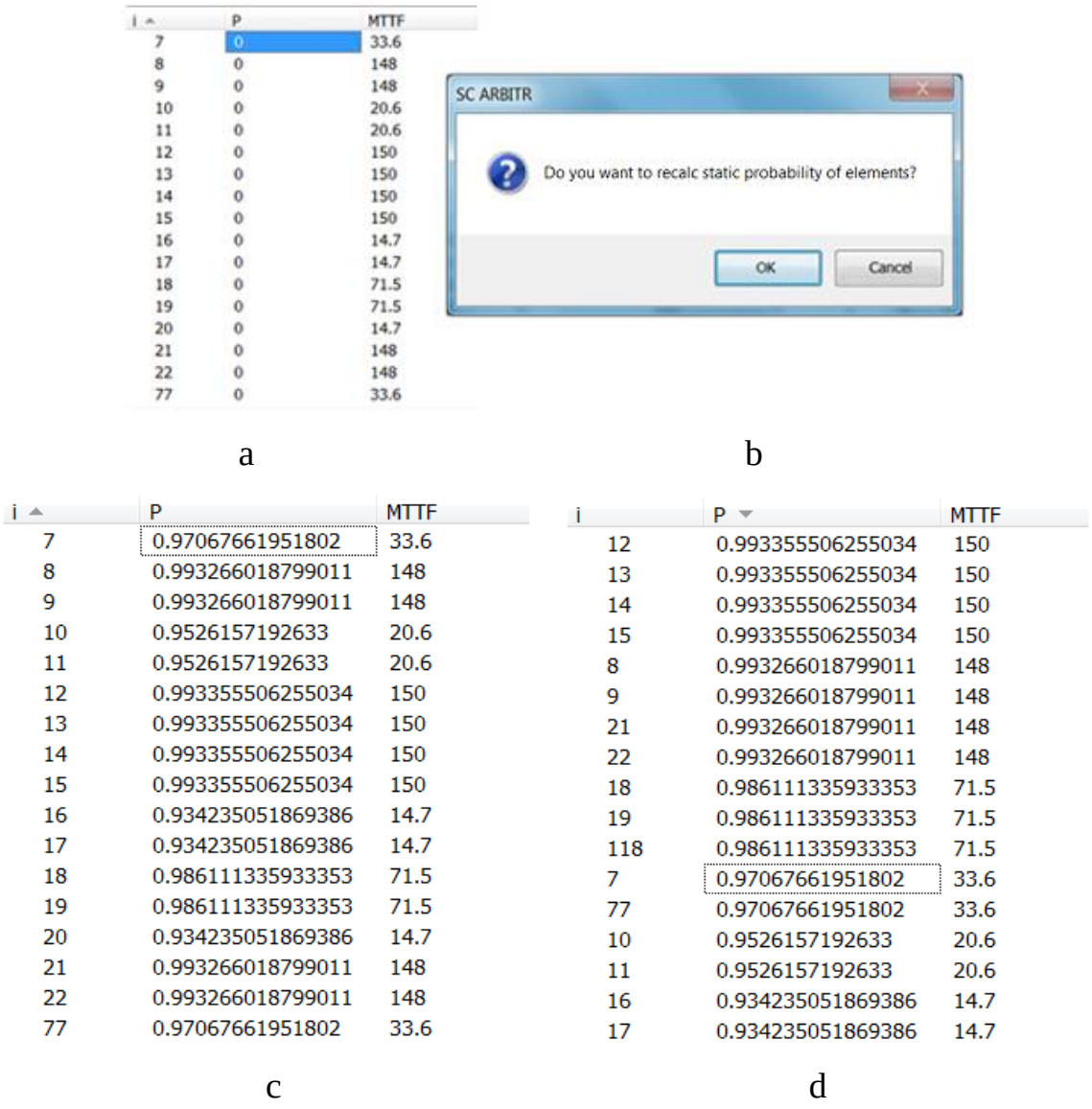
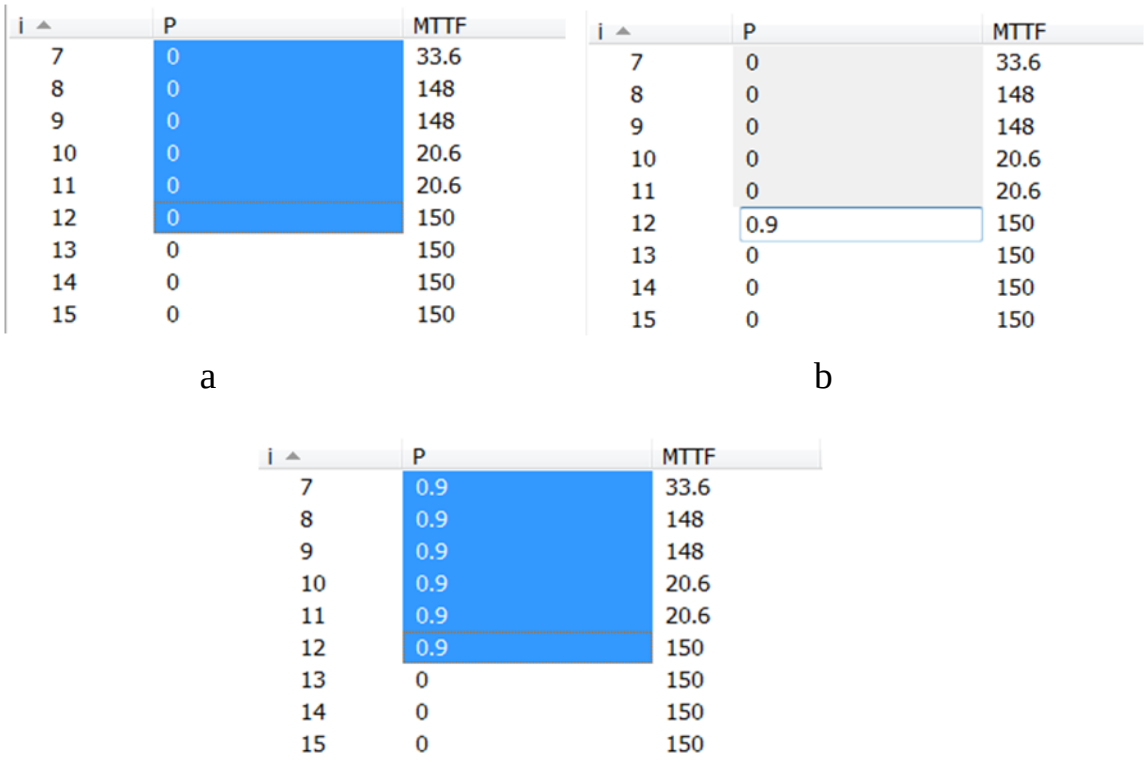


Figure 67 – Recalculation of static probability and sorting in descending order

4.6 Option "Quick input of initial data"

If it is necessary to enter the same values of the elements' parameters (*MTTF*, *P*, *MTTR*) for several elements of the circuit at once, select the corresponding cells in the column of the parameters table by pressing the up/down arrows while pressing the Shift key (Figure 68, a). Or, with the Shift key pressed, by the left-click mouse, select the first and last element from the block of elements that have the same value. Then, releasing the Shift key, enter the required parameter value. If the actions are performed correctly, all other cells of the selected range, except for the last one, will

be highlighted in gray (Figure 68, b). After entering the parameter value, press the Enter key (Figure 68, c).



i	P	MTTF
7	0	33.6
8	0	148
9	0	148
10	0	20.6
11	0	20.6
12	0	150
13	0	150
14	0	150
15	0	150

a

i	P	MTTF
7	0	33.6
8	0	148
9	0	148
10	0	20.6
11	0	20.6
12	0.9	150
13	0	150
14	0	150
15	0	150

b

i	P	MTTF
7	0.9	33.6
8	0.9	148
9	0.9	148
10	0.9	20.6
11	0.9	20.6
12	0.9	150
13	0	150
14	0	150
15	0	150

c

Figure 68 – Quick input of the elements' parameters

4.7 Option "Calculate"

The "Calculate" option can be performed after receiving the modeling result. The option allows you to recalculate the results when the initial data changes without constructing the logical and probabilistic functions. Using this option is convenient when solving large-scale or complex problems that require significant time to build the logical and probabilistic functions. The option is executed after pressing the active button "Calculate" on the toolbar.

5 Practical lessons "Development of emergency plans" using the SC ARBITR

5.1 Lesson 1. SC ARBITR. Modeling of simple structures

Data, name: _____, _____ Variant no._____

	Curriculum element. Tutorial page numbers	Implementation	Notes. Answers by task options
1	Basics of the software operations ***(item No. 1)***		Software startup. Windows resize. Toolbar. FIS elements: nodes, edges, text
2	Node parameters. Modeling mode settings. Adding/editing system LCF ***(items No. 1.3-1.4, Appendix A)***		Changing node parameters Changing node color Modeling mode settings Adding and editing system LCF
3	Reliability Block Diagram – RBD (***Appendix B***)		
4	Modeling of simple structures. Serial system modeling ***(item No. 2.1)***		Direct and inverse solution of modeling tasks
5	Modeling of simple structures. Parallel system modeling ***(item No. 2.2)***		
6	Modeling of system with separate and whole redundancy ***(item No. 2.3)***		One FIS is created (the method of nodes multiplication is used)
7	Task 1-1 solving with a teacher (***task 1-1***, ***Appendix B, E***)		R = ; N_{MP}= ; Q = ; N_{MCS}= ;
8	Task 1-2 solving (***task 1-2***) (by himself)		R = ; N_{MP}= ; Q = ; N_{MCS}= ; $MTTFsys$= $N_{elem.Imp.max}$:
9	Task 1-3 solving (***task 1-3***) (by himself)		R = ; N_{MP}= ; Q = ; N_{MCS}= ; $MTTFsys$= $N_{elem.Imp.max}$:
10	Reliability modeling of "K out of N" ***(item No. 4.2)***		

N_{MP} – the minimal paths number;

N_{MCS} – the minimal cut sets number

$N_{elem.Imp.max}$ – the number of elements with maximum importance.

5.1.1 Task 1-1. RBD, static calculation

The block diagram of the Industrial automatic control system (*IACS*) is shown in Figure 5.1.1. Information is transmitted from node *I* (Input) to node *O* (Output).

Variants of the initial data on the reliability of the system elements – the probability of failure-free operation *Pi(t)* – are given in Table 5.1.1.

Task content:

1. Create a functional integrity scheme (*FIS*).
2. Perform reliability modeling to calculate reliability indicators in the "Static calculation" mode.
3. Compile a report of the modeling results.

The report has to contain:

- The value of the system failure-free operation probability *R*;
- The value of the system failure probability *Q*;
- The number of minimal paths N_{MP} and minimal cut sets N_{MCS};
- Numbers of elements with the maximum importance value $N_{elem.Imp.max}$.

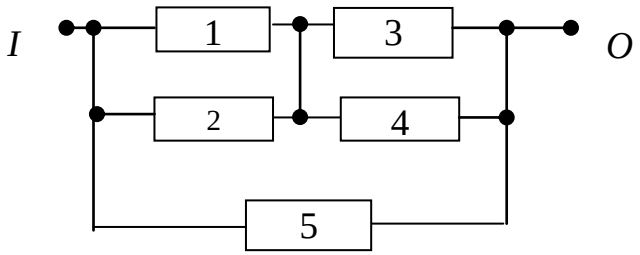


Figure 5.1.1 – Structural scheme of the information system

Table 5.1.1 – Initial data for task 1-1

Element/ Variant	1	2	3	4	5
1	0,8	0,8	0,85	0,95	0,9
2	0,95	0,8	0,85	0,95	0,9
3	0,8	0,9	0,85	0,95	0,9
4	0,8	0,9	0,95	0,95	0,9
5	0,8	0,8	0,95	0,8	0,9
6	0,8	0,8	0,85	0,8	0,8
7	0,8	0,8	0,85	0,8	0,95
8	0,6	0,6	0,85	0,8	0,95
9	0,7	0,75	0,85	0,8	0,92
10	0,78	0,79	0,85	0,83	0,92

5.1.2 Task 1-2. RBD, time-depended calculation

The block diagram of the *IACS* is shown in Figure 5.1.2. Information is transmitted from node *I* to node *O*.

Variants of the initial data on the reliability of the system – failure rate λi (1 per hour) – are given in table 5.1.2.

System elements lifetime t=8760(hour).

Task content:

1. Create a functional integrity scheme (*FIS*);
2. Perform reliability modeling to calculate reliability indicators in the "Time-depended calculation" mode without "Use element's lifetime" and "Use time to repair".
3. Compile a report of the modeling results.

The report has to contain:

- The value of the system failure-free operation probability *R(t)*;
- The value of the system failure probability *Q(t)*;
- The value of the system Mean Time To Failure (in hours and years) *MTTFsys*;
- The number of minimal paths N_{MP} and minimal cut sets N_{MCS};
- Numbers of elements with the maximum importance value $N_{elem.Imp.max}$.

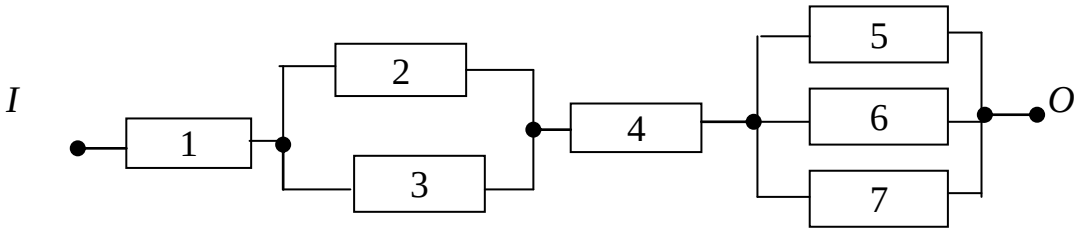


Figure 5.1.2 – Structural scheme of the information system

Table 5.1.2 – Initial data for task 1-2

Element/ Variant	1	2	3	4	5	6	7
1	1,0E-06	2,0E-06	2,0E-06	4,0E-06	5,0E-06	5,0E-06	5,0E-06
2	7,0E-06	2,0E-06	2,0E-06	4,0E-06	5,0E-06	5,0E-06	5,0E-06
3	1,0E-06	4,0E-06	4,0E-06	4,0E-06	5,0E-06	5,0E-06	5,0E-06
4	1,0E-06	2,0E-06	2,0E-06	4,0E-06	6,0E-06	6,0E-06	6,0E-06
5	1,0E-06	1,0E-06	1,0E-06	4,0E-06	6,0E-06	6,0E-06	6,0E-06
6	4,0E-06	1,0E-06	1,0E-06	4,0E-06	6,0E-06	6,0E-06	6,0E-06
7	1,0E-06	2,0E-06	2,0E-06	4,0E-06	3,0E-06	3,0E-06	3,0E-06
8	1,0E-06	8,0E-06	8,0E-06	4,0E-06	3,0E-06	3,0E-06	3,0E-06
9	6,0E-06	2,0E-06	2,0E-06	1,0E-06	5,0E-06	5,0E-06	5,0E-06
10	7,0E-06	2,0E-06	2,0E-06	4,0E-06	1,0E-06	1,0E-06	1,0E-06

5.1.3 Task 1-3. Automatic information system modeling

Structural scheme of the automatic information system (*AIS*) is shown in Figure 5.1.3. Information is transmitted from node *I* to node *O*.

Variants of the initial data on the reliability of the elements – Mean time to failure *MTTF* (h) – are given in Table 5.1.3.

System elements lifetime t=500(h).

Task content:

1. Create a functional integrity scheme (*FIS*);
2. Perform reliability modeling to calculate reliability indicators in the "Time-depended calculation" mode without "Use element's lifetime" and "Use time to repair".
3. Compile a report of the modeling results.

The report has to contain:

- The value of the system failure-free operation probability *R(t)*;
- The value of the system failure probability *Q(t)*;
- The value of the system Mean Time To Failure (in hours and years) *MTTFsys*;
- The number of minimal paths N_{MP} and minimal cut sets N_{MCS};
- Numbers of elements with the maximum importance value $N_{elem.Imp.max}$.

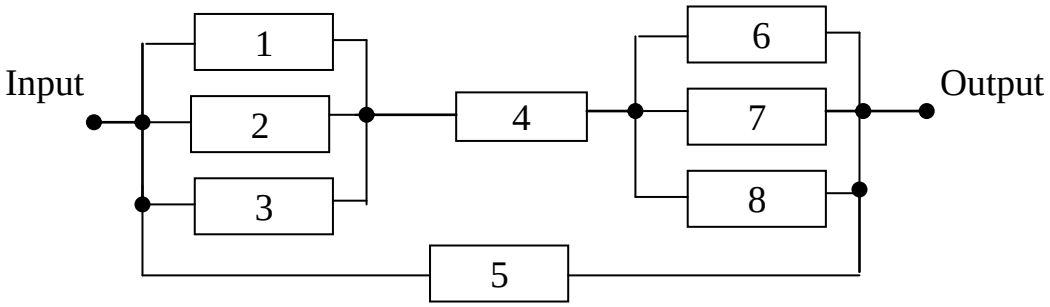


Figure 5.1.3 – Structural scheme of the automatic information system

Table 5.1.3 – Initial data for task 1-3

Element/ Variant	1	2	3	4	5	6	7	8
1	5000	5000	5000	10000	8000	4000	4000	4000
2	1000	1000	1000	10000	8000	4000	4000	4000
3	1000	1000	1000	10000	8000	8000	4000	4000
4	5000	5000	5000	10000	1000	4000	4000	4000
5	1000	1000	1000	10000	10000	4000	4000	4000
6	5000	5000	5000	10000	8000	40000	40000	40000
7	50000	50000	50000	10000	8000	4000	4000	4000
8	45000	45000	45000	10000	2000	100000	100000	100000
9	5000	5000	5000	10000	12000	1000	1000	1000
10	1000	1000	1000	10000	15000	8000	4000	4000

5.2 Lesson 2. Bridge circuit reliability modeling. Modeling of the net structure.

Data, name: _____, _____ Variant no._____

	Curriculum element. Tutorial page numbers	**Implementation**	**Notes. Answers by task options**
1	Compiling a *FIS* of the bridge circuit. Solution of direct and inverse tasks. Analysis of modeling results. ***(items No. 3.1-3.2)***		*R=y13=* $N_{elem.Imp.max}$*:* *R=y3 y4=* $N_{elem.Imp.max}$*:* *R=y3+y4=* $N_{elem.Imp.max}$*:*
2	Building a bridge circuit fault tree. Present *FT* in the form of 2 types *FIS*: - with nodes multiplication (traditional *FT* form); - *FIS* without nodes multiplication ***(item No. 3.3)***		*Q=y11=* *R=y"11=* *Q=y17=* *R=y"17=*
3	Studying Fault Tree modeling modes on the example of bridge circuit reliability analysis: 1. Approximate calculation for: q_i=0.01, q_i=0.1, q_i=0.5. Comparison with the exact result. 2. Application of the logic-statistical method. Comparison with the exact result. 3. Reliability analysis of a reparable bridge circuit. Compare the modeling results of the reparable scheme with the modeling mode without repair		1. [*qi*=0.01]: $Q_{exact}(y13)$= $Q_{approx}(y13)$= [*qi*=0.1]: $Q_{exact}(y13)$= $Q_{approx}(y13)$= [*qi*=0.5]: $Q_{exact}(y13)$= $Q_{approx}(y13)$= 2. Interval evaluations: [*pi*=0.99]: $R_{static}(y13)$= ± [*pi*=0.9] : $R_{static}(y13)$= ± [*pi*=0.5]: R_{static}=*y13*= ± 3. *Kg=* *MRT=* *MTBF=* R_{repar}(8760)= *MTTF=* $R_{non\ repar}$(8760)=
4	Task 2-1 solving (***task 2-1***) Structure "Simple network".		*R(t=8760) =* N_{MP}= *Q(t=8760) =* N_{MCS}= *MTTF =*
5	Task 2-2 solving (***task 2-2***) Structure "Network ARPA" (by himself)		*P(t) =* *Q(t)=* N_{MP}= N_{MCS} =

5.2.1 Task 2-1. Simple network

The block diagram of the network reliability ("Simple network 1") is shown in Figure 5.2.1[1].

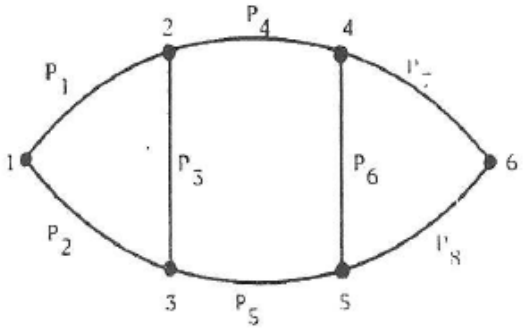


Figure 5.2.1 – Simple network 1

The network structure, presented in the form of a directed graph, implements the transfer of information from node No. 1 to node No. 6. Vertical edges between network nodes (communication channels) are bidirectional.

All network nodes are absolutely reliable.

Graph edges (communication channels between nodes) are characterized by a finite probability *Pi(t)*. Variants for the values of the reliability function of communication channels *Pi(t)*, *i*=1–8 , *t*=8760 hours are given in Table 5.2.1.

Task content:

1. Create a functional integrity scheme (FIS);
2. Perform reliability modeling to calculate reliability indicators in the "Time-depended calculation" mode without "Use element's lifetime" and "Use time to repair".
3. Compile a report of the modeling results.

The report has to contain:

- The value of the system failure-free operation probability *R(t)*;
- The value of the system failure probability *Q(t)*;

[1] .B.Misra, "An Algorithm for the Reliability Evaluation of Redundant Network", *IEEE Trans. Reliability*, vol R-19, No.4, 1970, P.146–151.

- The value of the system Mean Time To Failure (in hours and years) *MTTFsys*;
- The number of minimal paths N_{MP} and minimal cut sets N_{MCS};
- Numbers of elements with the maximum importance value $N_{elem.Imp.max}$.

Table 5.2.1 – Reliability indicators of the "Simple network 1" scheme

Variant	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
1	0.9	0.95	0.93	0.86	0.98	0.88	0.9	0.91
2	0.89	0.93	0.92	0.87	0.9	0.9	0.95	0.92
3	0.92	0.95	0.9	0.8	0.85	0.93	0.95	0.9
4	0.9	0.96	0.87	0.93	0.9	0.96	0.92	0.95
5	0.93	0.95	0.9	0.92	0.8	0.95	0.92	0.94
6	0.88	0.98	0.89	0.92	0.88	0.92	0.94	0.93
7	0.95	0.9	0.83	0.9	0.79	0.82	0.9	0.91
8	0.9	0.8	0.7	0.6	0.7	0.8	0.87	0.92
9	0.91	0.9	0.97	0.9	0.96	0.83	0.96	0.91
10	0.85	0.92	0.92	0.95	0.84	0.79	0.9	0.9

5.2.2 Task 2-2. ARPA network

The reliability block diagram of the reduced ARPA network is shown in Figure 5.2.2[2].

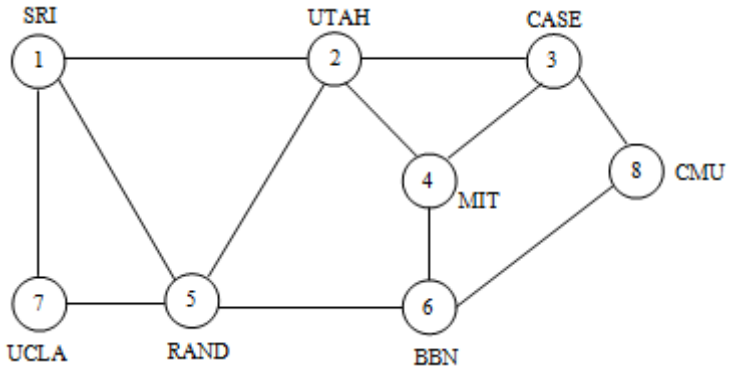


Figure 5.2.2 – Reduced ARPA network

The network structure, represented as a directed graph, implements the transfer of information from node No. 7 of UCLA (University of California at Los Angeles) to node No. 8 of CMU (Carnegie Mellon University).

In Figure 5.2.2, network nodes *n1÷ n8* are absolutely reliable.

Edges between nodes *i* and *j* of the network (communication channels) have a finite reliability, given in the form of reliability function for a given time *Pij(t)=Pij*. The non-horizontal lines in Figure 5.2.2 can be bidirectional **if necessary**.

The values of the communication channels reliability function *Pij* are given in Table 5.2.2.

Task content:

1. Create a functional integrity scheme (FIS);
2. Perform reliability modeling to calculate reliability indicators in the "Static calculation" mode.
3. Compile a report of the modeling results.

[2] L.Fratta, U.G.Montanari, "A Boolean Algebra Method for Computing the Terminal Reliability in a Communication Network", *IEEE Trans. Circuit Theory*, vol CT-20, 1973 May, P. 203–211.

The report has to contain:

- The value of the system failure-free operation probability *R*;
- The value of the system failure probability *Q*;
- The number of minimal paths N_{MP} and minimal cut sets N_{MCS};
- Numbers of elements with the maximum importance value $N_{elem.Imp.max}$.

Table 5.2.2 – Reliability indicators *Pij* of the reduced ARPA network

Var.	P_{12}	P_{15}	P_{71}	P_{23}	P_{24}	P_{25}	P_{34}	P_{38}	P_{46}	P_{56}	P_{68}	P_{75}
1	0.9	0.81	0.981	0.729	0.81	0.81	0.729	0.9	0.9	0.81	0.91	0.9
2	0.89	0.93	0.92	0.87	0.9	0.9	0.95	0.92	0.9	0.95	0.93	0.86
3	0.92	0.95	0.9	0.8	0.85	0.93	0.95	0.9	0.93	0.95	0.93	0.86
4	0.9	0.96	0.87	0.93	0.9	0.96	0.92	0.95	0.96	0.95	0.93	0.86
5	0.93	0.95	0.9	0.92	0.8	0.95	0.92	0.94	0.9	0.95	0.93	0.86
6	0.88	0.98	0.89	0.92	0.88	0.92	0.94	0.93	0.8	0.9	0.95	0.92
7	0.95	0.9	0.83	0.9	0.79	0.82	0.9	0.91	0.9	0.93	0.95	0.9
8	0.9	0.8	0.7	0.6	0.7	0.8	0.87	0.92	0.94	0.8	0.85	0.93
9	0.91	0.9	0.97	0.9	0.96	0.83	0.96	0.91	0.92	0.84	0.75	0.83
10	0.85	0.92	0.92	0.95	0.84	0.79	0.9	0.9	0.89	0.99	0.87	0.9

The reduced ARPA network for the problem of assessing the reliability of a two-terminal network "UCLA (University of California at Los Angeles) – CMU (Carnegie Mellon University)" was developed in 1973 by Italian specialists L. Fratta and U. Montanari.

Initially, ARPANET included 4 switching nodes IMP (IMP – Interface Message Processors, in modern terminology – a router) in the following switching nodes:

1. University of California, Los Angeles – UCLA, where the SDS Sigma 7 computer was located.
2. Stanford Research Institute (SRI, No. 2), which housed the first SDS 940 host computer, named "Genie".
3. University of California, Santa Barbara (UCSB), which housed the Interactive Mathematics Center, equipped with an IBM 360/75.
4. University of Utah (UTAH, No. 3), where I. Sutherland worked on the DEC PDP-10.

The first message over the ARPANET was transmitted on 10/29/1969 at 10:30 pm from the UCLA node to the SRI node. The message consisted of one word "login". Due to a computer malfunction, only the first two letters "l" and "o" were transmitted. After about an hour, the problem was fixed and the message was successfully transmitted. A permanent link between UCLA and SRI nodes was established on 12/5/1969.

In March 1970, the ARPANET reached the East Coast of the United States when the IMP of the Cambridge University, Massachusetts was connected to the network. By June 1970, 9 IMPs were connected, by December 1970 – 13, and by September 1971 – 18. This ensured that 23 universities and government agencies were included in the network of host computers. Approximately at the same time, the scheme presented by L.Fratta, U.G.Montanari in their 1973 article was developed.

By 1981, there were 213 hosts on the ARPANET, and then a new host came online about every 20 days.

In 1983 U.S. Department of Defense Communications Agency used some of the ARPANET equipment for military purposes, organizing the MILNET network.

The ARPANET information network ceased to exist in June 1990.

5.3 Lesson 3. Reliability of complex technical systems

Data, name: _____, _____ Variant no._____

	Curriculum element. Tutorial page numbers	Implemen-tation	Notes. Answers by task options
1	**Task 3-1** solving (***task 3-1***) Reliability Analysis of the Ship Power System		R_I = R_{II} = *LFdp*= *LFip*=
2	**Task 3-2** solving (***task 3-2***) Analysis of the reliability of the structure "ARPA Network" using the method of serial-parallel reduction		*R*=
3	Solving problems and examples at the request of the trainees		

R_I , R_{II} – values of the probability of failure-free operation for two variants of the initial data;

LFdp, *LFip* = dimensions of the logical function in solving direct and inverse problems.

5.3.1 Task 3-1. Reliability Analysis of the Ship Power System

The ship power system (SPS) consists of three generators (G) of the same capacity (x1, x2, x3), three main switchboards (MSB) (x4, x6, x9), three connectors (Con) (x5, x7, x8) between MSB and six secondary switchboards (SSB) (x10 – x15) (Figure 5.3.1).

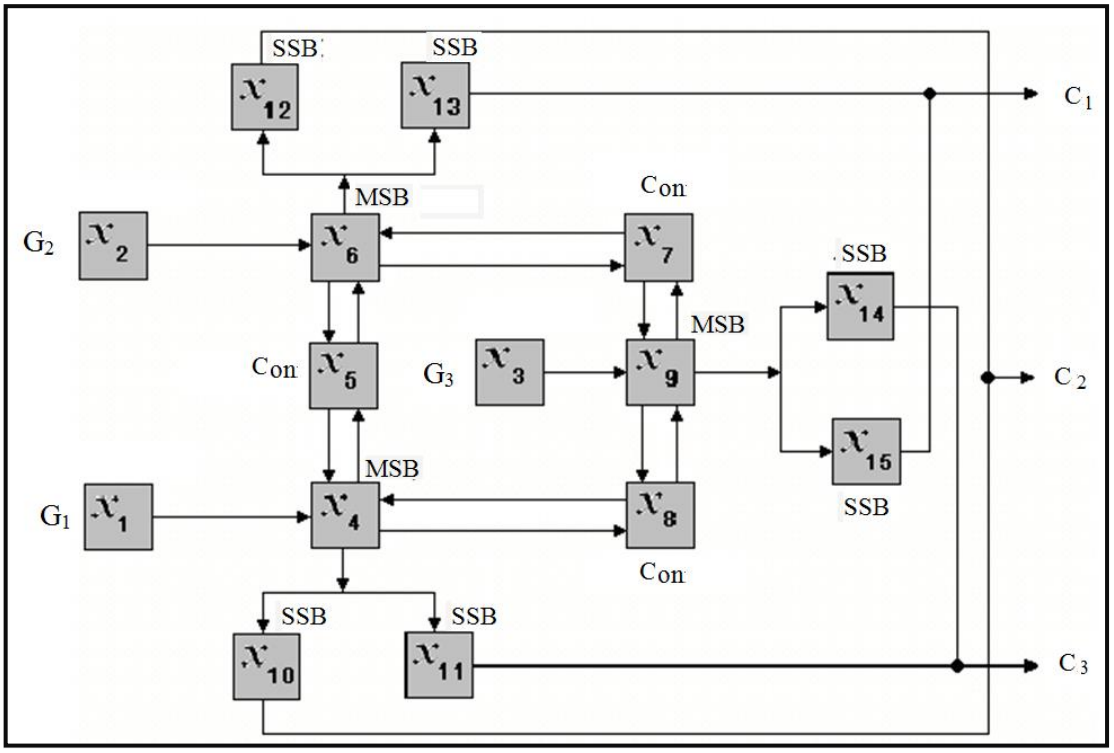


Figure 5.3.1 – Structural scheme of SPS

Each generator is connected to the corresponding MSB. To connect the load to the main switchboard, two SSBs are switched. The load is connected to two SSB connected to different main switchboards.

To increase the reliability of power supply, connectors are installed in the circuit, which allow connecting voltage between the main switchboards.

The task of the SPS is to provide uninterrupted simultaneous power supply to three groups of critical users C_1, C_2 and C_3, each of which can be connected to one of the two STSs. It is also known that for the simultaneous supply of all three groups of users, the power of one generator is sufficient, and there are no restrictions on the capacity of either the main switchboard or the connectors between them.

Task content:

1. Draw up functional integrity schemes (FIS) for modeling the reliability of SPS.

2. Calculate the SPS reliability indicators in the "Static calculation" mode for two options for the values of the probabilities of failure-free operation specified in Table 5.3.1.

3. Compile a report on the simulation results.

The report must contain:

- values of the probability of failure-free operation for two variants of the initial data (Table 5.3.1);
- the number of shortest success paths for successful operation and minimal cut sets;
- dimensions of the logical function in solving direct and inverse problems;
- analysis of 2-3 shortest success paths and 2-3 minimal cut sets (MCS) to verify the correctness of the FIS compilation;
- analysis of the ranked values of the contributions and significance of the SPS equipment groups.

Table 5.3.1 – The variants of the initial data

Group of elements	№ of elements	Ri (Variant 1)	Ri (Variant 2)
MSB	4,6,9	0.8	0.9
SSB	10 – 15	0.85	0.9
Generators	1-3	0.9	0.9
Connectors	5,7,8	0.7	0.9

5.3.2 Task 3-2. Analysis of the reliability of the structure "ARPA Network" using the method of serial-parallel reduction

Figure 5.3.2 shows a topology of the ARPA network, which included 21 terminals in the early 1970s. The network structure, represented as a directed graph, implements the transfer of information from the terminal (node) No.20 UCLA (University of California at Los Angeles) to the terminal (node) No.21 CMU (Carnegie Mellon University).

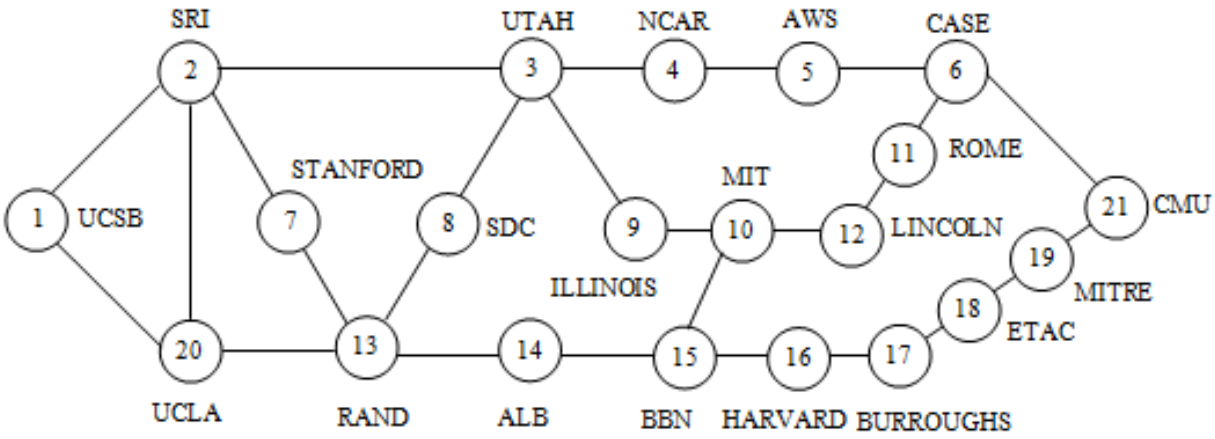


Figure 5.3.2 – Topology of the ARPA computer network

Terminals are connected with each other by 26 communication channels, network nodes No.1 ÷ No.21 are absolutely reliable.

Edges between nodes *i* and *j* of the network (communication channels) have a finite reliability, given in the form of probabilities of failure-free operation for a given time $Pij(t) = Pij$.

The task of analyzing the reliability of this structure is to carry out a step-by-step procedure of series-parallel reduction in order to minimize a finite number of circuit elements.

The term "reduction" in mathematical problems defines methods for reducing the dimension of a problem or choosing a short form for representing initial data, for example, diagrams or structures.

In problems of reliability analysis, the procedure of serial-parallel reduction consists in replacing the series or parallel connection of circuit elements with one element that is equivalent from the point of view.

In the SC ARBITR there is a possibility of visual display of the results of reduction of the original FIS of a complex or large-sized structure. For this purpose, the apparatus of equivalent nodes is used.

Task content:

1. Draw up a functional integrity scheme (FIS) for modeling the reliability of the ARPA network (Figure 5.3.2), which contains 26 elements (communication lines between terminals).

2. Simulate the reliability of the ARPA network in the "Static calculation" mode for the values of the probabilities of failure-free operation of the elements $Pij = 0.9$.

3. Perform serial-parallel reduction of the original FIS, reducing the number of FIS elements to 12 (Figure 5.3.3) and using the graphical apparatus of equivalent nodes of the SC ARBITR.

4. Compile a report on the simulation results.

The report must contain:

- results of ARPA network reliability modeling for full and reduced FIS;
- table of results of step-by-step serial-parallel reduction (Table 5.3.2). The numbers of SFC nodes in Table 1 are given as an example.

Table 5.3.2 – Results of serial-parallel reduction

N	communication channels (Figure 1)	communication channels (Figure 2)	N nodes of the full FIS	N nodes of the reduced FIS	*Pj* (p_i=0.9)
1	SRI-UTAH	SRI-UTAH	9	9	0.9
2	SRI-Stanford Stanford-RAND	SRI-RAND	4 5	4	0.81
3	UCSB- SRI UCLA-UCSB UCSB - SRI	UCLA-SRI	1 2 3	1	0.981
…	…	…	…	…	…
12	BBN-HARVARD Harvard-Burrough Burrough-ETAC ETAC-MITRE MITRE-CMU	BBN-CMU	17 18 19 20 21	2	0.59049

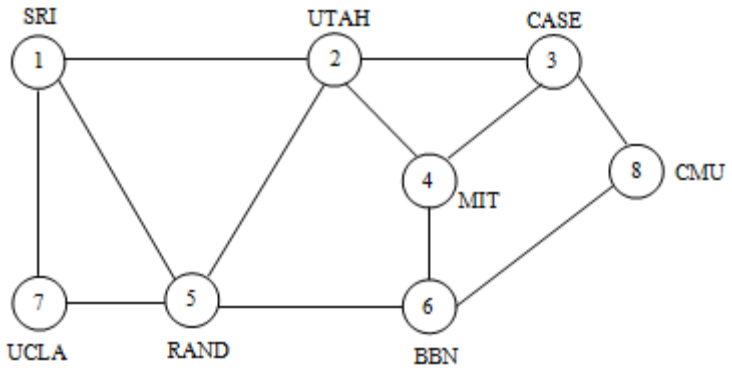


Figure 5.3.3 – Reduced ARPA network

Lesson 4. Risk analysis of complex technical systems

Data, name: _____, _____ Variant no._____

	Curriculum element. Tutorial page numbers	Implemen-tation	Notes. Answers by task options
1	Theory and application of fault tree analysis (FTA): AND, OR, NOT operations (***item No. 1.2.2***) Construction of fault tree in SC ARBITR. Examples C1, C2. Gates AND, OR (*App.C, P.127-128*) Example C3. Lighting system (*App. C, p.129-131*) Example C4. Fire detector system (*App. C, p.131-135*) **Task 4-1.** The death of a person from electrical shock **Task 4-2**. Fault tree for a fire in a storage tank		
2	Theory and application of event tree analysis (ETA): Construction of event tree in SC ARBITR. Mode "Effectness/Risk" (***item No. 1.3.3***). Combination of FTA and ETA. Together with the teac her, example D1, Safety barriers (*App.D, p.141-142*), example D2, Risk analysis of pedestrian injuries (*App.D, p.142-146*) example D3, Event Tree Analysis Method in functional safety problems (*App.D, p.146-151*). **Task 4-3**. Scenario modeling of fire risk **Task 4-4**. IACS functional safety analysis		
3	Solving problems and examples at the request of the trainees		

5.3.3 Task 4-1. The death of a person from electric shock

Description of example:

The death of a person from electric shock can occur when his body is included in an electrical circuit with a current sufficient to inflict damage. Therefore, for an accident to occur (TOP event X1), the simultaneous existence of three events is necessary:

1 Intermediate event "X10" means the presence of potentially high voltage on the body of the electrical installation.

In turn, the event "X10" can be a consequence of any of the two events – the prerequisites "X11" and "X12", where "X11" is a decrease in the insulation resistance of the current-carrying parts, "X12" is the contact of the current-carrying parts with the installation case.

2 Event "X20" means the appearance of a person on a conductive base connected to the ground.

The event "X20" is also determined by both of two prerequisites: "X21" – the entry of a person onto a conductive base, "X22" – touching the grounded elements of the room with the unprotected surface of the human body.

3 The event "X30" is the result of one of 3 target functions: "X31" – the need for repair, "X32" – the need for maintenance, "X33" – the use of the electrical installation for its intended purpose, or the normal operation of the installation.

Task content:

1 Construct a fault tree according to the description of the example.

2 Perform a calculation of the probability of electric shock to a person using the initial data given in Table 5.4.1.

Table 5.4.1 – Initial data for the task 4-1 solution

N event	Event content	Probability
11	decrease in the insulation resistance	0.2
12	contact of the current-carrying parts with the case	0.3
21	appearance of a person on a conductive base	0.1
22	touching the grounded elements of the room	0.05
31	the need for repair	0.2
32	the need for maintenance	0.3
33	intended purpose, or the normal operation	0.4

3 Check the correctness of the construction of the fault tree according to the physical content of the minimum sections.

4 Carry out calculations in the mode of approximate evaluation and compare the results with calculations in the static mode.

5 Check in the Excel or in PC ARBITR the correctness of the calculations of the importance factors F-V, risk reducing ratio and risk increasing ratio.

6 Perform probability calculations using cutoff mode at level 1·E-03. Explain the reasons for changing the calculation results.

5.3.4 Task 4-2. Fault tree for a fire in a storage tank

This example is to illustrate the "fire" top event identification procedure.

Figure 5.4.1 shows a typical storage tank of flammable materials. A centrifugal pump is used to pump the materials to supply other processes. Here the process controls and safe guards are not shown since they are not related to fire event identification.

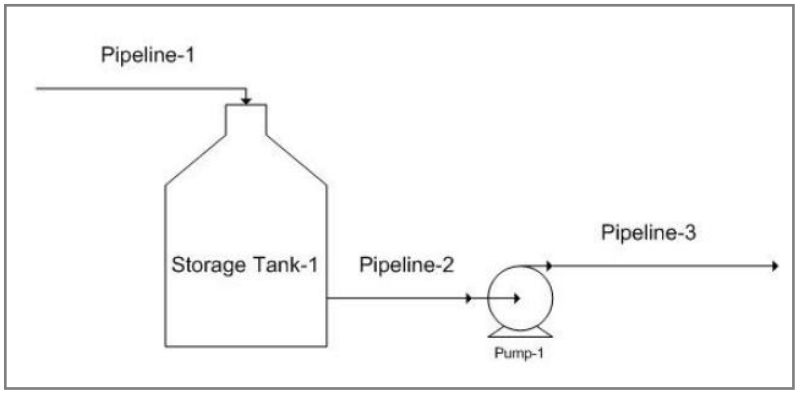


Figure 5.4.1 – Typical storage tank of flammable materials

Description

Assume that storage tank and centrifugal pump are the only two potential sources of fire in this process.

A fire in a storage tank can occur if an ignition source is present and the containment of the tank ruptures.

Destruction of the containment of a storage tank for combustible materials can occur due to leakage due to rupture or due to corrosion.

Two types of fire protection are implemented on the storage tank. Tank storage fire will occur in case of failure of both types of protection. Tank protection No. 1 failure can be due to either failure of the pre-acting valve, or failure of the fuse, or human error. Tank protection No. 2 failure may occur due to a stuck valve of the dry chemical system or operator error of the system.

A pump fire will occur in the event of a fire and failure of two types of protection – Overheating Event (OHE) and Parameter Deviation Event (PDE).

A fire at the pump may occur if there is an ignition source and leakage of combustible materials due to contact with air.

Task content:

1 Construct a fault tree according to the description of the example.

2 Perform a calculation of the fire probability using the initial data given in Table 5.4.2.

The initial data for the quantitative assessment of the fire probability are given in the Table 5.4.2.

Table 5.4.2 – Initial data for the quantitative assessment of the fire probability

No.	Event	Probability
1	Storage tank rupture	1.0E-05
2	Leak externally at storage tank	2.0E-04
3	Ignition Source of storage tank	2.0E-05
4	Fusible link storage tank	5.0E-04
5	Pre-active valve fails	3.0E-04
6	Human error	1.0E-02
7	Dry chemical valve stuck	8.0E-04
8	Operator fails to respond	1.0E-02
9	Leak externally at pump	2.0E-03
10	Ignition Source of pump	5.0E-03
11	Parameter Deviation Event (PDE)	3.0E-03
12	Overheating Event (OHE)	4.0E-03

5.3.5 Task 4-3. Scenario modeling of fire risk

A dangerous event in production is considered – the occurrence of a fire due to engine overheating. To model scenarios for the development of a dangerous event, a combined tree of failures and events is used (the "Cause and effect" method). The consequences of a possible fire are losses, designated from C_0 to C_4.

C_o – losses are $3,000 if an engine breaks down and production downtime is 2 hours.

C_1 – if there is a local fire, medium damage to the equipment is caused and there are 24 hours of downtime, then the loss will be $39,000.

C_2 – if a fire occurs and medium damage is caused to the equipment and there will be 1 month of downtime, then the losses will amount to $1.74 million.

C_3 – if a fire occurs and significant damage is caused to the equipment and downtime for a very long time, then the losses will amount to $20 million.

C_4 – if a major fire occurs and there are wounded or dead from the plant's personnel, then the losses will amount to $50 million.

The initiating event "engine overheating" has a probability of 0.88. The probability that an engine overheating will cause a fire is 0.02.

In the event of a fire, a hand-held fire extinguisher can be used, the probability of failure of which is 0.037. The probability of incorrect human actions is 0.1.

To extinguish a fire, a fire extinguishing apparatus can be used, the probability of failure of which is 0.04, and the probability of failure of the fire extinguishing apparatus control system is 0.011.

Also, fire extinguishing equipment can be used to extinguish a fire, the probability of failure of which is 0.0109, and the probability of failure of the fire extinguishing equipment control system is 1E-05.

Task content:

It is necessary to develop an event tree that describes various scenarios for the occurrence and development of a fire. Elements of the event tree can be equivalent

events that are used to model failures of a hand-held fire extinguisher, equipment and fire extinguishing equipment.

Modeling should be performed with the calculation of efficiency and risk. According to the simulation results, fill in the columns of the Table 5.4.3.

Table 5.4.3 – The modeling results

Scenario	**Effects, E ($)**	**Events probability, Pr**	**Risk (E*Pr)**
C_0	3 000		
C_1	39 000		
C_2	1.744 million		
C_3	20 million		
C_4	50 million		

5.3.6 Task 4-4. IACS functional safety analysis

To ensure the safe operation of a hazardous production facility, an industrial automatic control system (IACS) is used. The IACS consists of a distributed control system (DCS) and an instrumental safety system (SIS).

In case of failure of the DCS, the safety functions are performed by the SIS. The DCS consists of a sensor, a logical device and a final (actuator) element. The SIS also consists of a sensor, a logical device and a final (actuator) element, but all elements of the system are redundant.

Task content:

1 Construct a combined tree of accident risk events at a hazardous production facility. Modeling of DCS and SIS failures is recommended to be carried out using fault trees. Calculate the probability of an accident without using the SIS system and using the SIS system. Initial data for modeling:

- $qs = 0.03$ – probability of sensor failure;
- $q_L = 0.001$ – *probability* of logical device failure;
- $q_{FE} = 0.07$ – *probability* of failure of the final (actuator) element;
- $\beta s = 10\%$ – *coefficient* of beta-model of common causes failures (CCF) for the subsystem of redundant sensors;
- $\beta_L = 5\%$ – *coefficient* of beta-model CCF for the subsystem of redundant logical devices;
- $\beta_{FE} = 10\%$ – *coefficient* of beta-model CCF for the subsystem of redundant final (actuator) elements.

2 Calculate the risk reduction factor as the ratio of the probability of an accident without using the SIS system and using the SIS system.

Appendix A

Basic concepts of the theory of dependability

In 2006, at a joint meeting of the International Electrotechnical Commission (IEC) and International Organization for Standardization (ISO), it was determined: Dependability is one of the qualities of the process and product.

Quality – a set of product properties that determine its suitability to satisfy certain needs in accordance with its purpose.

Quality implies a certain cost of resources to achieve it, which will allow achieving the absence of failures.

It is important that quality is one of the risk reduction tools.

Dependability as one of the components of quality is a complex property, which in turn consists of the properties of reliability, availability, maintainability and maintenance support.

It is important that dependability is one of the risk reduction tools too (Figure A.1).

Figure A.1 – Quality and dependability

The main definitions of reliability are given in the international dictionary IEC 50 (191) and GOST 27.002–2015.

Dependability: The property of an object to maintain over time the ability to perform the required functions in given modes and conditions of use, maintenance, storage and transportation.

Reliability: The property of an object to continuously maintain the ability to perform the required functions for some time or operating time in specified modes and conditions of use.

Maintainability: The property of an object that is its ability to maintain and restore the state in which the object is able to perform the required functions, through maintenance and repair.

Availability: The property of an object, which consists in its ability to be in a state in which it can perform the required functions in the specified modes and conditions of use, maintenance and repair, assuming that all the necessary external resources are provided.

Failure: An event consisting in a violation of operational state of an object.

Operational state: The state of an object in which it is capable of performing the required functions.

The main assumption of reliability theory is that failure is a random repeated event. Therefore, the theory of reliability is based on the theory of probability.

Probability of failure-free operation (reliability function): The probability that, within a given operating time, an object will not fail.

Lifetime: Duration or amount of work of an object.

Mean time to failure (MTTF): The mathematical expectation of an object's time to failure.

Mean time between failures (MTBF): The mathematical expectation of an object's time between failures.

Failure rate: The conditional density of the probability of an object failure occurring, determined under the condition that no failure has occurred before the considered point in time.

Mean time to repair (MTTR) – mathematical expectation of repair time.

Availability factor: The probability that an object will be in an operational state at a given time.

Since failure is a random event, time to failure as a continuous random variable is described by the distribution function.

The distribution function *F(t)* is an integral characteristic of a continuous random variable or an integral distribution law. The distribution function prescribes to each value of a random variable the probability of its realization (Figure A.2).

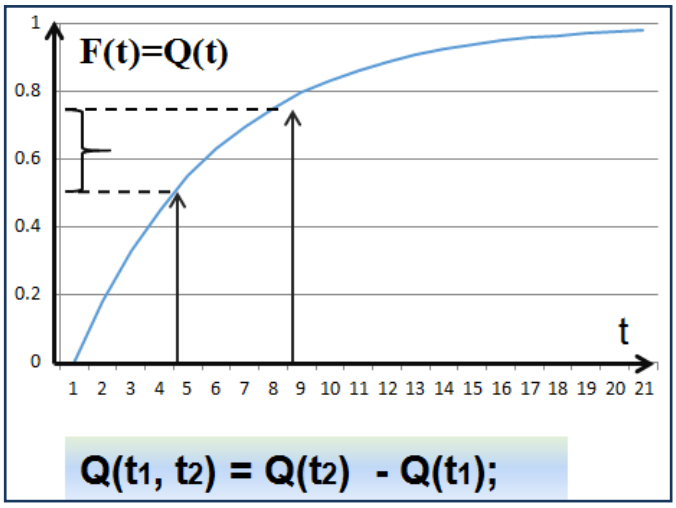


Figure A.2 – Failure probability Q(t) and distribution function F(t)

The distribution function will be called the failure probability and denoted *Q(t)*. The probability of failure is an increasing function – the longer the operating time of the product, the higher the probability of failure.

The probability of failure on a time interval $Q(t_1,t_2)$ is calculated as the difference between the values of the distribution function at the ends of the time interval

$$Q(t_1, t_2) = Q(t_2) - Q(t_1).$$

Attention! The entry $Q(t)$ means that this is the probability of failure on the interval $(0,t)$.

Reliability function is the probability that a device will not fail within a given operating time: $R(t) = 1 - Q(t)$.

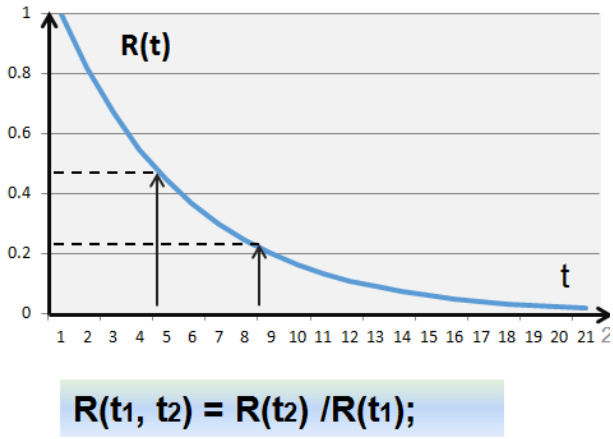


Figure A.3 – Reliability function $R(t)$

The reliability function on a time interval $R(t_1,t_2)$ is calculated as a quotient of division

$$R(t_1, t_2) = R(t_2)/R(t_1).$$

The entry $R(t)$ means that this is the reliability function on the interval $(0,t)$.

If $F(t)$ is an integral characteristic of random operating time to failure, then the probability density function $f(t)$ is a differential characteristic of a random variable:

$$f(t) = \frac{dF(t)}{dt} = -\frac{dR(t)}{dt}.$$

The failure rate is defined as the ratio

$$\lambda(t) = \frac{f(t)}{1-F(t)} = \frac{f(t)}{R(t)} = -\frac{\frac{dR(t)}{dt}}{R(t)}.$$

A typical bath type function of the failure rate for the product life cycle is shown in Figure A.4.

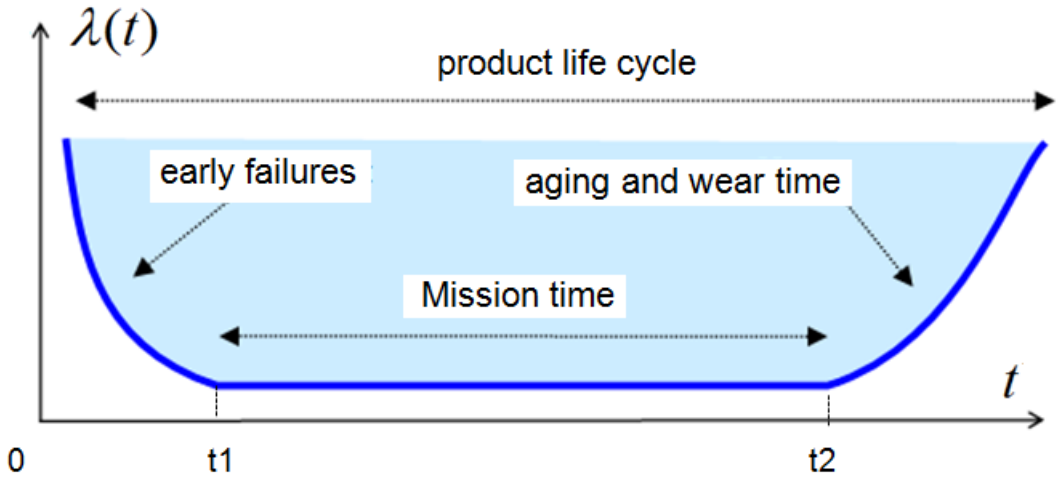


Figure A.4 – Function of the failure rate

Mean time to failure (MTTF) as the mathematical expectation of an object's time to failure is calculated as the first central moment of a random variable *t* (time to failure)

$$To = MTTF = \int_0^\infty tf(t)\partial t.$$

Graphically, the MTTF is the area under the curve of the reliability function (Figure A.5).

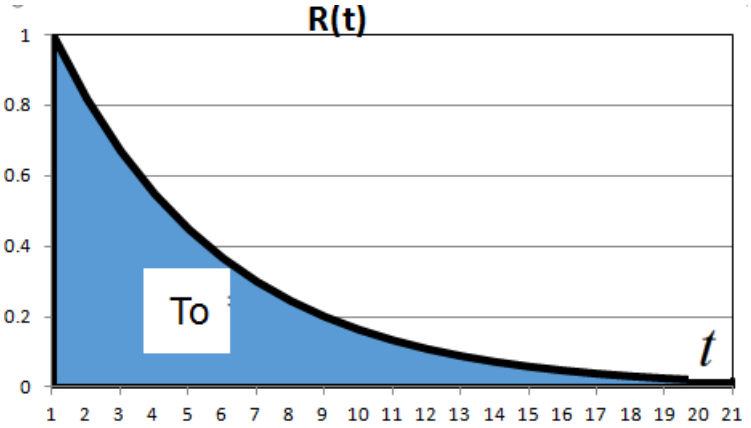


Figure A.5 – Graphical interpretation of MTTF (To)

The main failure model in reliability theory is the exponential failure model with the distribution function of time to failure

$$F(t) = 1 - \exp(-\lambda t),$$

where λ – failure rate.

Reliability function is

$$R(t) = 1 - F(t) = \exp(-\lambda t).$$

For exponential failure model

$$MTTF = \frac{1}{\lambda}.$$

For highly reliable systems (for $\lambda t << 1$) based on Taylor series expansion

$$Q(t) \approx \lambda t.$$

When analyzing the reliability of repairable technical items, the availability factor is calculated. The expression for calculating the factor can be obtained on the basis of the Markov model

$$K_g = \frac{MTTF}{MTTF+MRT} + \frac{MRT}{MTTF+MRT} \exp[-\left(\frac{1}{MTTF} + \frac{1}{MRT}\right) t].$$

In practice, the expression for the stationary availability factor is often used

$$K_g = \frac{MTTF}{MTTF+MRT}.$$

Appendix B

Reliability Block Diagrams and Boolean Methods

There are many different reliability analysis methods. Fault trees (FT) and reliability block diagrams (RBD) are both symbolic analytical logic techniques that can be applied to analyze system reliability and risk related characteristics. In PC ARBITR, for modeling fault trees and RBD, the same graphic symbols are used – vertices and arcs. That's why most of the logical constructs in a fault tree diagram can also be modeled with a RBD.

Block diagrams are widely used in engineering and science and exist in many different forms. They can also be used to describe the interrelation between the components and to define the system. When used in this fashion, the block diagram is then referred to as a reliability block diagram.

RBD is a directed acyclic graph (i.e., a graph without loops) representing logical relationships between the success state of a system and the success state of its constituent blocks. This logical structure is mainly represented by simple serial and parallel graphical structures.

A reliability block diagram is a graphical representation of the components of the system and how they are reliability-wise related (connected). It should be noted that this may differ from how the components are physically connected. RBD of a simplified computer system with a redundant fan configuration is shown on Figure B.1.

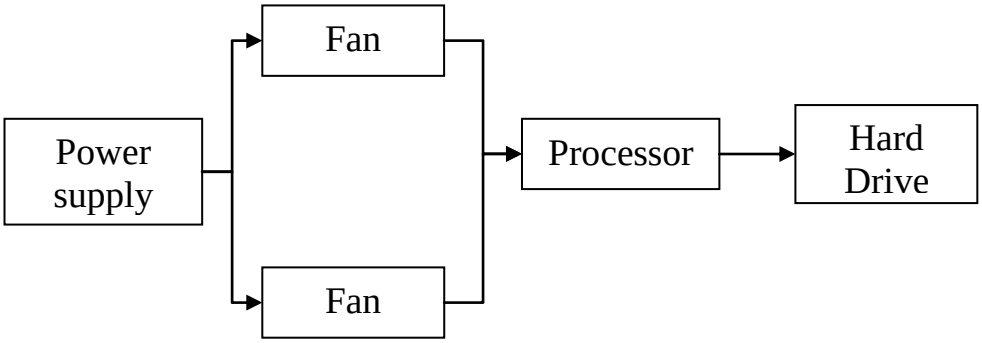


Figure B.1 – RBD of a Simplified computer system

RBDs are constructed out of blocks. The blocks are connected with direction lines that represent the reliability relationship between the blocks.

A block is usually represented in the FIS by a functional or equivalent node. In a reliability block diagram, such blocks represent the component, subsystem or assembly with probabilistic characteristics.

One of the most important assumptions is that the elements of a system (or the blocks that represent them in an RBD) can only exist in one of two states: up or down (failure).

It is also assumed that failures and repairs of individual units are statistically independent events.

Since RBD describes the logical relationships necessary to describe the functional state of the system, it does not necessarily reflect the way the hardware is physically connected, although RBD usually takes into account the physical connections in the system to the maximum extent possible.

If the functioning of the system requires that all blocks function, then in the corresponding structural diagram of reliability, all blocks should be connected in series, as shown in Figure B.2.

Such systems will be called serial systems (in the sense of reliability).

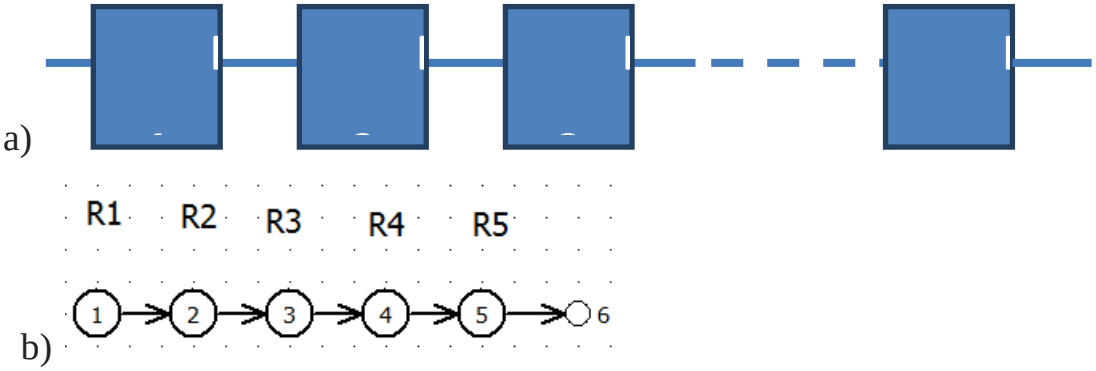


Figure B2 – a) configuration of serial system; b) FIS of serial system

Reliability function of a serial system is calculated by the formula

$$Rs(t) = \prod_{i=1}^{n} r_i(t).$$

For the exponential failure model, the reliability function of a serial system is calculated by the formula

$$Rs(t) = \prod_{i=1}^{n} e^{-\lambda_i t} = e^{-\sum_{i=1}^{n} \lambda_i t} = e^{\Lambda_s t},$$

where $\Lambda_s = \sum_{i=1}^{n} \lambda_i$ – system failure rate.

MTTF of serial system calculated by the formula

$$MTTFs = \frac{1}{\Lambda_s} = \frac{1}{\sum_{i=1}^{n} \lambda_i}.$$

For calculate serial system probability of failure we must use Puancare-Silvester formula (inclusion-exclusion formula). For example, system probability of failure for three units calculated by the formula

$$Qs = q_1 + q_2 + q_3 - q_1 q_2 - q_1 q_3 - q_2 q_3 + q_3\, q_2 q_3.$$

If $\lambda t \ll 1$, then $Qs = q_1 + q_2 + q_3$.

If, in accordance with the definition of system success/failure, the failure of one component or unit does not affect the operation of the system, use another type of system block diagram. For example, if the entire serial chain is duplicated, then the structural diagram of the reliability of the system is shown in Figure B.3.

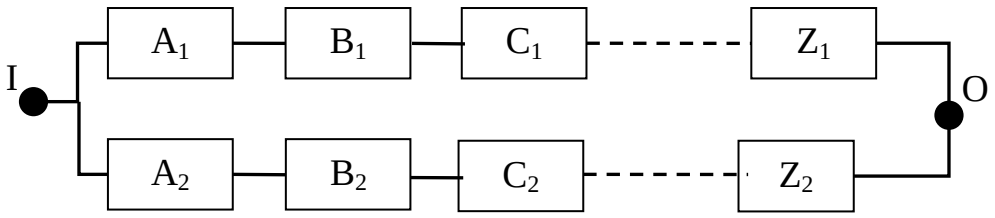


Figure B.3 – system with whole redundancy (per channel redundancy)

If each block of the serial chain is duplicated, then the structural diagram is as shown in Figure B.4. Structural diagrams of this type are called parallel or "parallel model".

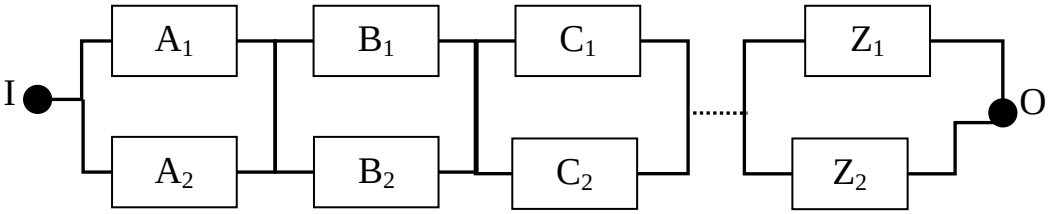


Figure B.4 – Separate redundant system (element redundant)

Often there is a need to model a system, the definition of success of which states that the functioning of the system requires the functioning *m* or more parallel elements. The block diagram of the reliability of such a system takes the form shown in Figures B.5.

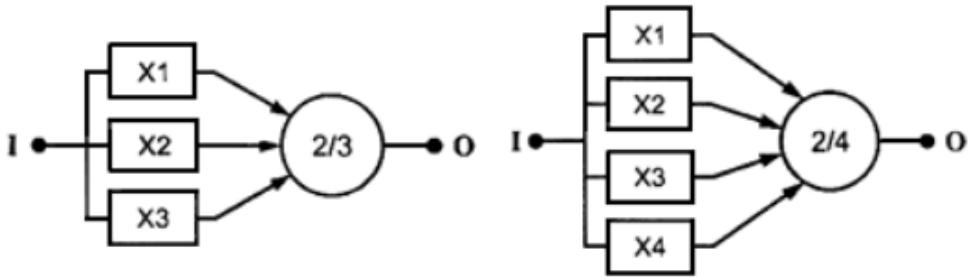


Figure B.5 – Examples of systems "K out of N"

Truth tables and Boolean algebra formulas can be used to analyze RBD (IEC 710678:2006 Analysis techniques for dependability – Reliability block diagram and Boolean methods).

The SC ARBITR uses the basic theorems of Boolean algebra.

Theorem 1. Any function of the algebra of logic with n arguments can be represented in the following form:

$$f(x_1, \dots, x_n) = x_i f_1^{(i)}(x_1, \dots, 1, x_{i+1,\dots,} x_n) \vee x'_i f_0^{(i)}(x_1, \dots, 0, x_{i+1,\dots,} x_n).$$

The SC ARBITR uses this theorem (Bool-Shannon decomposition) in two logical variables.

Example B.1:

$$f(a \vee b) = a\, f(1 \vee b) \vee \bar{a} f(0 \vee b) = \text{a} \vee \bar{a} b\ .$$

Bool-Shannon decomposition expansion allows a correct transition from a logical function to a probabilistic function with the replacement of logical variables by the probabilities of their truth.

Example B.2: Logical function for duplicated system has the form $X_1 \vee X_2$.

After Bool-Shannon decomposition we get: $X_1 \text{v}\, X_2 = X_1 \text{v}\, \overline{X_1} X_2$.

On the right side of the equality, we have a logical function for disjoint events. Then probability of sum event is:

$$\Pr(X_1 \text{v}\, X_2 = 1) = \Pr(X_1 = 1) + \Pr(\overline{X_1} X_2 = 1).$$

If $Pr(x) = R(x)$, then

$$Pr(x) = Pr(X1 + X2) = R(X1) + \big(1 - R(X2)\big)R(X2) = R(X1) + Q(X1)R(X2).$$

Theorem 2. The inverse (negation) of the elementary conjunction *abc*… is equivalent to the disjunction $\overline{abc} = \bar{a} \vee a\bar{b} \vee ab\bar{c}$.

Example B.3:

Compile a FIS for analyzing the reliability of a technical system, the structure of which is shown in Figure B.6.

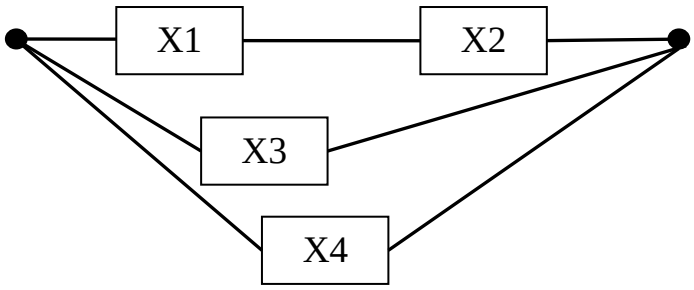


Figure B.6 – Structure of a technical system

Reliability of technical system element:

$R1$=0.9; $R2$=0.85; $R3$=0.7; $R4$=0.65.

Required:

1 Calculate reliability of technical system.

2 Calculate probability of failure of technical system

3 Determine the number of minimum paths for success

4 Determine the size of the logical function

5 Determine the number of minimum cut sets

6 Determine item with maximum and minimum importance

Solution.

FIS for analyzing the reliability of a technical system shows on Figure B.7.

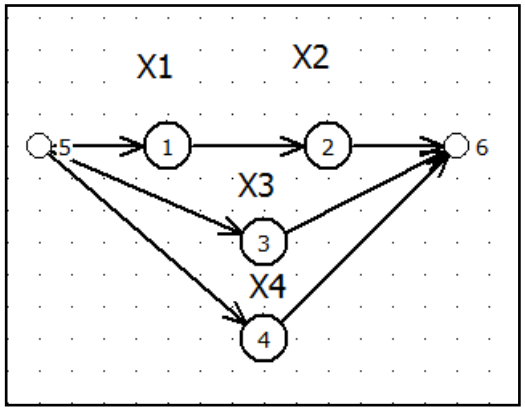


Figure B.7 – FIS for analyzing the reliability of a technical system

The Figure B.8 shows the simulation result.

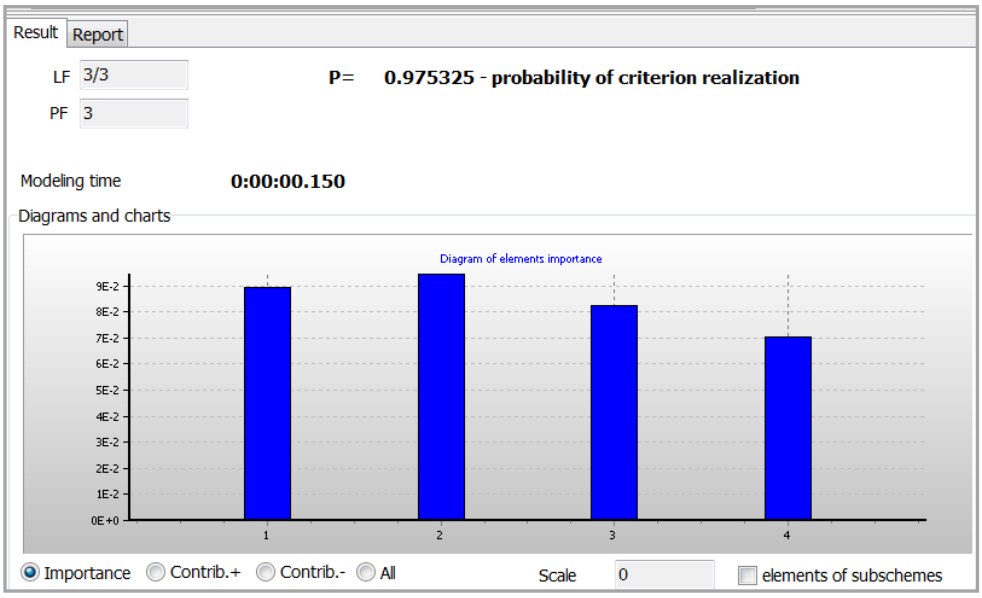


Figure B.8 – Result tab

On the Result tab we see result of modeling and diagram of elements importance:

- reliability of technical system R=0.975325;
- size of the logical function = 3;
- item N2 have maximum importance;
- item N4 have minimum importance.

On the Report tab (Figure B.9), you can see the expression for the logical function (minimal paths), the results of the simulation according to the *y6* criterion, as well as a table of elements characteristics – initial data on the reliability of elements (*Pi*), element importance, positive and negative contribution.

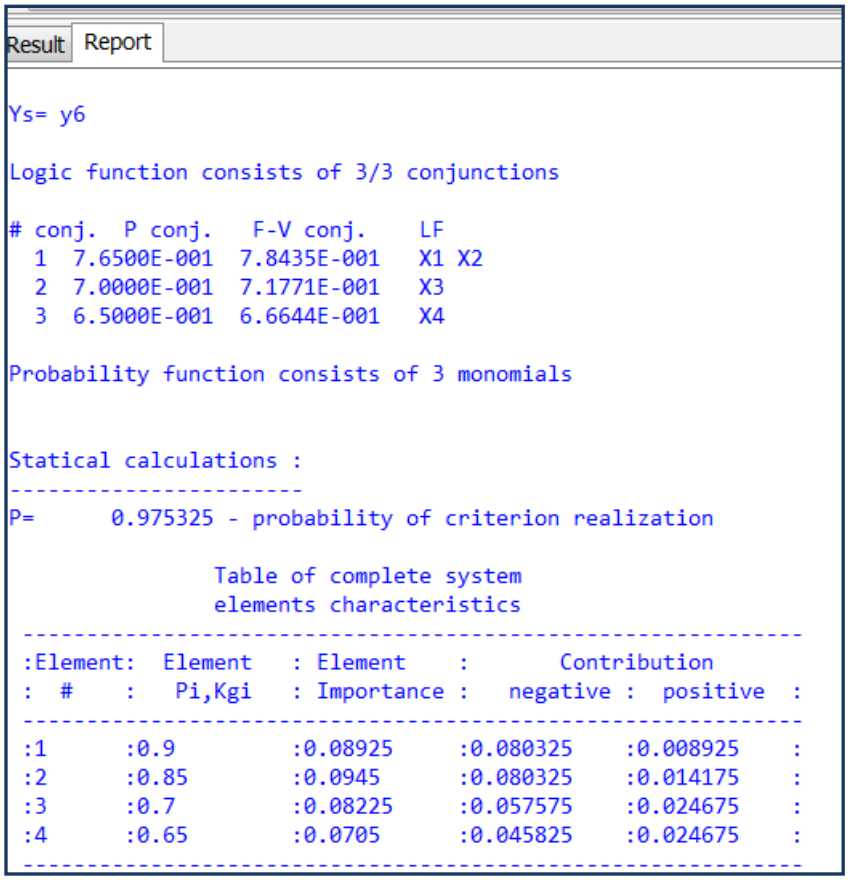


Figure B.9 – Report tab. Reliability calculation

On the Report tab (Figure B.10), you can see the results of the simulation according to the *y"6* criterion – logical function in the form of cut sets and the probability of system failure P=0.024675.

```
Ys= y"6

Logic function consists of 2/2 conjunctions

# conj.  P conj.   F-V conj.    LF
  1  1.5750E-002  6.3830E-001   X"2 X"3 X"4
  2  1.0500E-002  4.2553E-001   X"1 X"3 X"4

Probability function consists of 2 monomials


Statical calculations :
-----------------------
P=      0.024675 - probability of criterion realization
```

Figure B.10 – Report tab. Probability of failure calculation

Appendix C

Fault Tree Analysis method

Fault Tree Analysis (FTA) is a tool for analyzing, visually displaying and evaluating failure paths in a system, thereby providing a mechanism for effective system level risk evaluations.

FTA is a top-down, deductive failure analysis in which an undesired state of a system is analyzed using Boolean logic to combine a series of lower-level events. This analysis method is mainly used in safety engineering and reliability engineering to understand how systems can fail, to identify the best ways to reduce risk and to determine event rates of a safety accident or a particular system level failure.

FTA is one of the most important logic and probabilistic techniques used in probability risk analysis (PRA) and system reliability assessment today.

Methods to perform risk and reliability assessment in the early 1960s originated in US aerospace and missile programs. Fault tree analysis is such an example that was quite popular in the mid sixties. Early in the Apollo project the question was asked about the probability of successfully sending astronauts to the Moon and returning them safely to Earth.

After the Challenger accident, the importance of PRA and FTA in systems risk and reliability analysis was realized and its use at NASA has begun to grow.

There are two types of FTA:

Proactive FTA

- FTA during system design development
- Improve design by mitigating weak links in the design
- Prevent undesired events and mishaps

Reactive FTA

- FTA during system operation
- Find root causes of a mishap/accident

- Modify the design to prevent future similar accidents

FTA Coverage:

Hardware:

- System level
- Subsystem level
- Component level
- Environmental effects

System Events

- Failures Events
- Normal Events
- Environmental Events

Human Interaction

- Human error
- Human performance
- Organizational structures

Major applications of FTA include:

1 Numerical Requirement verification

2 Identification of safety critical components

3 Product certification

4 Product risk assessment

5 Accident/incident analysis

6 Design change evaluation

7 Visual diagrams of cause-consequence events

8 Common cause analysis

The fundamental concept of Fault Tree Analysis is the translation of the failure behavior of a physical system into a visual diagram and logic model (logic function). The diagram segment provides a visual model that very easily portrays system relationships and root cause failure. The logic segment of the model provides a

mechanism for qualitative and quantitative evaluation. FTA is based on Reliability theory, Boolean algebra and probability theory. A very simple set of rules and symbols provides the mechanism for analyzing very complex systems, and complex relationships between hardware, software and humans.

One of the main restrictive assumptions in FTA is that basic events must be assumed to be statistically independent, and their interaction is described by means of Boolean **OR/AND** gates, so that only the combination of events is relevant, and not their sequence.

FTA is a binary analysis. All events are assumed either to occur or not to occur: there are no intermediate options.

The most common logical operations when building fault trees are **OR** (disjunction), **AND** (conjunction), **NOT** (inverse).

Table C.1 shows the logical function for the **OR** operation (disjunction) by events *A* and *B*. Event *C* is a result event. Disjunction is a logical connective and typically notated "∨".

Denote by logical "1" – the event occurred, through "0" – the event did not occur.

Table C.1 – Truth table for logical **OR**

A	*B*	$C=A \vee B$
0	0	0
1	0	1
0	1	1
1	1	1

Description of the **OR** operation for events *A* and *B*:

- Either *A* or *B* is necessary and sufficient to cause *C;*
- Both *A* and *B* can occur together to cause *C*.

Example: Light is off because light bulb fails **OR** power fails.

If *P(A)* is probability of event *A*, and *P(B)* is probability of event *B*, then probability of event *C* is expressed by the formula

$$P(C) = P(A) + P(B) - P(A) \cdot P(B). \quad (1)$$

In classic theory of probability formula (1) named the theorem of probability of sum events.

If A and B are mutually exclusive events, then $P(A) \cdot P(B) = 0$. Consequently

$$P(C) = P(A) + P(B). \quad (2)$$

For a graphic representation of the logical operation **OR**, special, standard symbols can be used (Figure C.1).

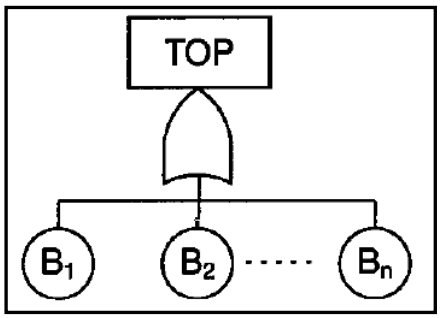


Figure C.1 – Standard fault tree with single **OR**-gate

Here the TOP event (system failure) occurs if at least one of the basic events (element, block failure) occurs. Since the basic events of this fault tree to be independent

$$Q_{TOP} = 1 - \prod_{i=1}^{n}(1 - q_i). \quad (3)$$

In a more general case, formula for adding probabilities (4) can be written as "inclusive-exclusive" formula or the Poincaré-Sylvestre formula:

$$Q_{TOP} = \Pr(\sum_{i=1}^{n} q_i) = \sum_{i} q_i - \sum_{i,j} q_i q_j + \sum_{i,j,k} q_i q_j q_k - \ \ldots + (-1)^{n-1} q_1 \ldots q_n.$$

Table C.2 shows the logical function for the **AND** operation (conjunction) by events *A* and *B*. Event *C* is a result event. Conjunction is a logical product and typically notated "∧"**,** or "⋅", or without label.

Table C.2 – Truth table for logical **AND**

A	*B*	*C=AB*
0	0	0
1	0	0
0	1	0
1	1	1

Description of the **AND** operation for events *A* and *B*:

- Both *A* and B are necessary to cause *C*
- *A* and *B* must occur simultaneously

Example: No power available because Primary power fails AND Secondary power fails.

Consider the fault tree in Figure C.2.

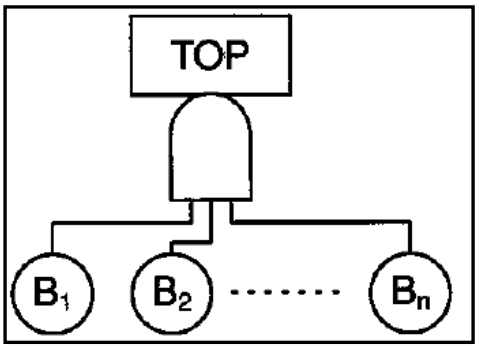


Figure C.2 – Standard fault tree with single **AND**-gate

Here the TOP event (system failure) occurs if and only if all the basic events (element, block failure) occur simultaneously. Since the basic events of this fault tree to be independent

$$Q_{TOP} = q_1.q_{2\ldots}q_n = \prod_{i=1}^{n} q_i, \quad (4)$$

where q_i – probability of failure *i*-element.

If *P(A)* is probability of event A, and *P(B)* is probability of event *B*, and event a and B are *s*-independent, then probability of event *C* is expressed by the formula

$$P(C) = P(A) \cdot P(B). \quad (5)$$

In logic, an **inverse** is a type of logical operations which is an immediate made from another conditional sentence.

Possible designations of inverse logical operation in relation to the event *A*: $\neg A$, $\bar{A}$, A' (Table C.3).

Table C.3 – Truth table for logical inverse (operation "**NOT**")

A	$\bar{A}$
1	0
0	1

There are some basic rules for logical inverse (Morgan' rules):

1 $\bar{\bar{A}} = A$;

2 $\overline{A \vee B} = \bar{A} \wedge \bar{B}$;

3 $\overline{A \wedge B} = \bar{A} \vee \bar{B}$.

FTA is normally carried out in five steps:

II Definition of the problem and the boundary conditions.

III Construction of the fault tree.

IV Identification of minimal cut sets(MCS).

V Qualitative analysis of the fault tree.

VI Quantitative analysis of the fault tree.

I Definition of the problem and the boundary conditions

The first activity of FTA consists of two sub steps:

- Definition of the critical event (the accident) to be analyzed
- Definition of the boundary conditions for the analysis.

The critical event (the accident) to be analyzed is called TOP event. It is very important that TOP event is given a clear and unambiguous definition.

For example, if we need to analyze failure of series system we must to construct fault tree with TOP event "System failure". This fault tree will consist of MCS – failures of system elements.

If we need to analyze system success we must to construct tree with TOP event "System success". This tree will consist of MPS – events of system success.

The description of the TOP event should give answer to the questions *what*, *where*, and *when*.

Example: Fire in the plant

what, – fire;

where, – compressor;

when, – during normal operation.

By boundary conditions we understand the physical boundaries of the system, the initial inner conditions and boundary conditions with respect to external stressed.

And very important to determine the level of resolution – how down in detail should we go to identify potential reasons for TOP event? Is to enough to identified "valve failure", or we need to identified break it further, failures of valve housing, valve stem, actuator and so forth.

II Construction of the fault tree

The fault tree construction always begins with the TOP event.

And we must try to identify all fault states that are result in the TOP event. These causes are connected to the TOP event via a logic gate. It is important that the first level of causes under the TOP event be put in a structured way. This first level is often referred to as the TOP structure of the fault tree.

The TOP structure causes are often to be failures in the prime modules of the system or in the prime functions of the system.

We then proceed, level by level, until all fault events have been developed to the prescribed level of resolution.

Rules for fault tree construction:

1 Describe the fault events.

Each of the basic events should be carefully described (what, where, when) in a "comment rectangle".

2 Evaluate the fault events.

The fault events may be different types, like technical failures, human errors, or environmental stresses.

Primary failures of components are usually classified as basic events, while secondary failures are classified as intermediate events that require a further investigation to identify the prime reasons.

3 Complete the gates.

All inputs to a specific gate should be completely defined and described before proceeding to the next gate. The fault tree should be completed in levels, and each level should be completed before beginning the next level.

III Identification of minimal cut sets (MCS)

A fault tree provides valuable information about possible combinations of fault events that will result in the TOP event. Such combination of fault events is called cut set.

Definition: a cut set is a set if basic events whose occurrence ensures that the TOP event occur (failure of system). A cut set is said to be minimal (MCS) if the set cannot be reduced without losing its status as cut set.

If a FT is developed for a serial elements connection (in the sense of reliability), then the minimum cut sets will be failures of the system individual elements. Then the FT will look like a connection by the OR operator of the system elements failure events.

IV Qualitative Evaluation of the Fault Tree

A qualitative evaluation of the fault tree may be carried out on the basis of the minimal cut sets. The criticality of a set obviously depends on the number of basic events in cut set, i.e., the order of the cut set.

A cut set of order 1 is usually more critical than a cut set of order 2, or more.

Another important factor is the basic events type of a MCS. We can rank the critically of the various cut sets according to the following ranking of basic events:

1) Human error.
2) Active equipment failure
3) Passive equipment failure.

V Quantitative analysis of the fault tree

The purpose of quantitative analysis of the fault tree usually is to determine the probability of the TOP event (system failure).

Example C1. Fault tree with a single AND-gate.

For a graphic representation of the logical operation AND, special, standard symbols can be used (Figure C.3). In SFC, logical symbols are replaced by edges with points.

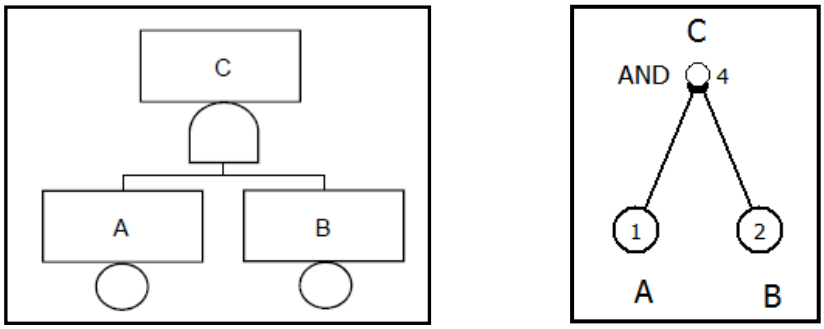


Figure C.3 – Standard gate "**AND**" and FIS with edges "**AND**"

```
Ys= y4

Logic function consists of 1/1 conjunctions

# conj.  P conj.   F-V conj.     LF
  1  5.6000E-001  1.0000E+000   X1 X2

Probability function consists of 1 monomials

Ps=  P1 P2
```

a

```
Ys= y4

Logic function consists of 1/1 conjunctions

# conj.  P conj.   F-V conj.     LF
  1  5.6000E-001  1.0000E+000   A B

Probability function consists of 1 monomials

Ps=  P1 P2
```

b

Figure C.4 – Print screen with LF and PF for logical product (conjunction)

Figure C.4,a shows logical functions in the notation of logical variables through *X1* and *X2*, and Figure C.4,b – with the output of variable names.

Example C2. Fault tree with a single **OR**-gate.

Consider the fault tree in Figure C.6.

For a graphic representation of the logical operation **OR**, special, standard symbols can be used (Figure C.5). In FIS, logical symbols are replaced by edges with arrows.

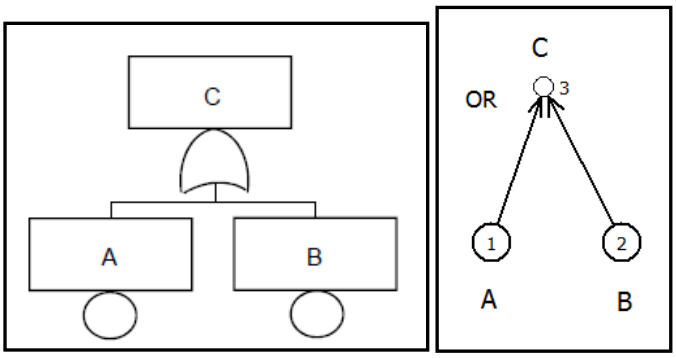


Figure C.5 – Standard gate **OR** and FIS with edges "**OR**"

```
Ys= y3

Logic function consists of 2/2 conjunctions

# conj.  P conj.   F-V conj.    LF
  1  8.0000E-001  8.5106E-001   X1
  2  7.0000E-001  7.4468E-001   X2

Probability function consists of 2 monomials

Ps=  P1
   + Q1 P2
```

a

```
Ys= y3

Logic function consists of 2/2 conjunctions

# conj.  P conj.   F-V conj.    LF
  1  8.0000E-001  8.5106E-001   A
  2  7.0000E-001  7.4468E-001   B

Probability function consists of 2 monomials

Ps=  P1
   + Q1 P2
```

b

Figure C.6 – Print screen with LF and PF for logical sum (disjunction)

Figure C.6, a shows logical function in the notation of logical variables through *X1* and *X2*, and Figure C.6, b with the output of variable names in matrix form

$$Ys=y3= X1 \vee X2 = \begin{matrix} X1 \\ X2 \end{matrix} \quad ; \quad Ys=y3= A \vee B = \begin{matrix} A \\ B \end{matrix}$$

Note that the logical function in the name output mode is displayed in the corresponding names, but the probabilistic function is displayed only in the symbols *P1*, *P2* (in matrix form too).

The form of the probabilistic function output can be different, including the abbreviated one. For example, formula (2) for the probability of the sum of events is transformed as follows:

$$Ps = P1 + P2 - P1 \cdot P2 = P1 + P2(1 - P1) = P1 + P2 \cdot Q1,$$

where $Q1 = 1 - P1$.

Example C3.

Let's look at a simple example of lighting system. Structural diagram of this system shows in Figure C.7.

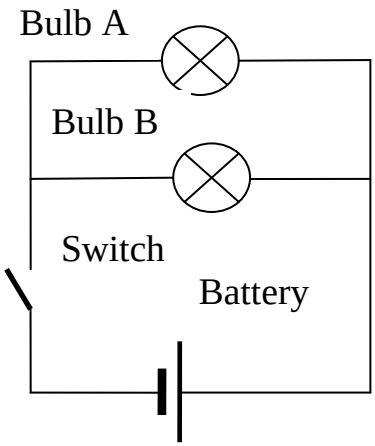


Figure C.7 – Structural diagram of lighting system

The fault tree for the analyzed system can be constructed in various ways. Figure C.8 shows the redundant construction of the fault tree using standard gate notation.

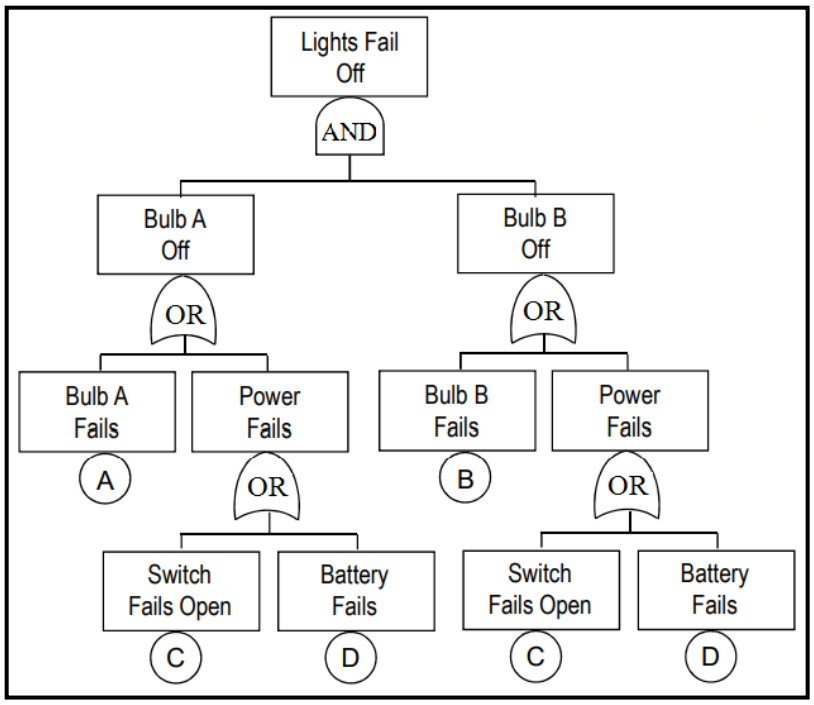


Figure C.8 – Redundant fault tree with standard gate notation

Figure C.9 shows the redundant construction of the fault tree in SC ARBITR.

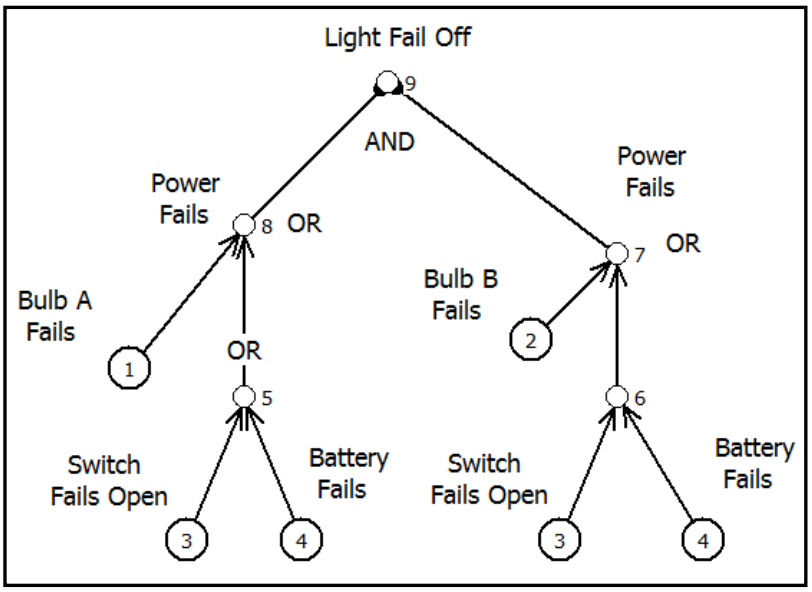


Figure C.9 – Redundant fault tree in SC ARBITR

To optimize the construction of a fault tree, the following technique is often used:

a) a reliability block diagram of the system is being developed

b) an inverse solution is carried out

c) the fault tree is built using the resulting minimized logic function

Figure C.10 shows the RBD of lighting system and result of modeling.

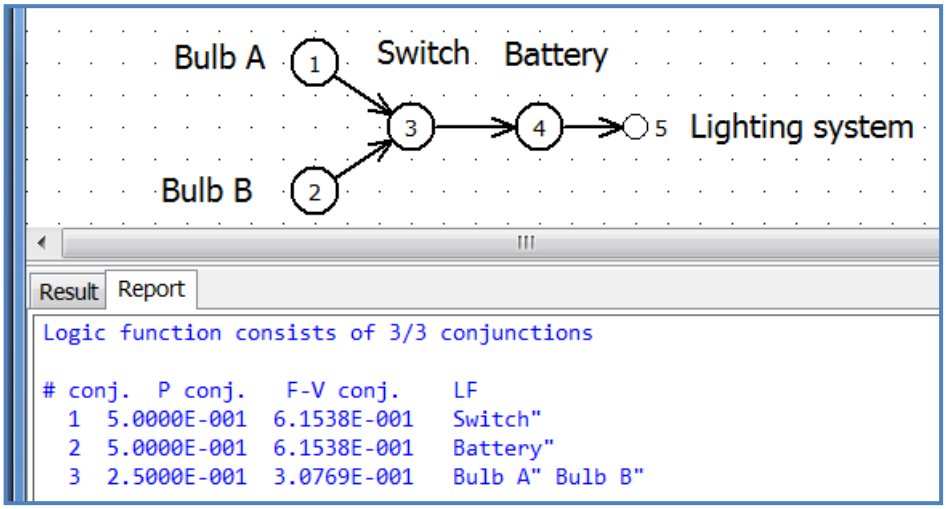
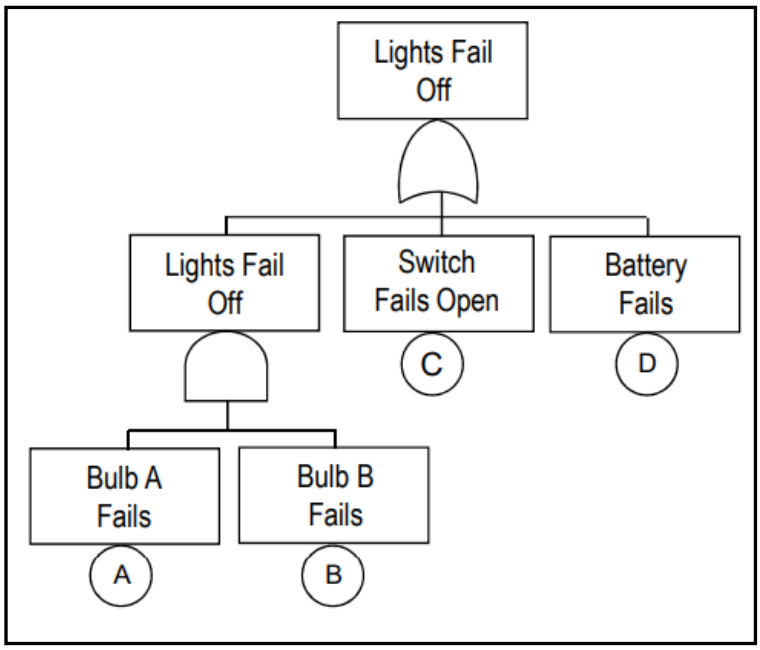
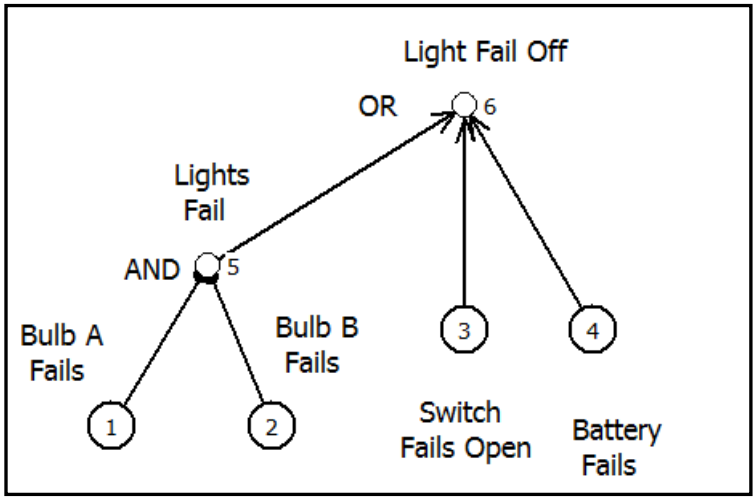


Figure C.10 – RBD of lighting system and result of modeling

On the basis of the resulting minimized logical function, it is possible to construct an optimal fault tree shown in Figure C.11.



a



b

Figure C.11 – Fault tree of lighting system with standard gate notation (a) and in SC ARBITR (b)

Example C4.

Let us consider a simplified version of a fire detector system located in a production room.

The fire detector system is divided into two parts, heat detection and smoke detection. In addition, there is an alarm button that can be operated manually.

Heat detector

In the production room there is a closed, pneumatic pipe circuit with four identical fuse plugs, FP1, FP2, FP3 and FP4. These plugs let air out if they are exposed to temperature higher than 72^{o}C. If one of or more of the plugs are activated, the switch will be activated and give an electrical signal to the start relay (SR) for the alarm and shutdown system. In order to have an electrical signal, the direct current (DC) source must be intact.

Smoke detector

The smoke detector consists of three smoke detector – SD1, SD2 and SD3. These detectors are very sensitive and can give warning of fire alarm early stage. In order to avoid false alarms, the three smoke detectors are connected via logical 2-out-of-3 voting unit. This means that at last two detectors must give the fire alarm are activated.

If at least two of the three detectors are activated, the voting unit will give an electrical signal to the start relay (SR) for the alarm and shutdown system.

Manual Activation

Together with pneumatic pipe circuit with the four fuse plugs, there is also a manual switch (MS) that can be turned to relieve the pressure in the pipe circuit. If operator (OP), who should be continually present, notices a fire, he can activate this switch. When the switch is activated, the pressure in the pipe circuit is relieved and electrical signal to the start relay.

Assume now that a fire starts. The fire detector system should detect and give warning about the fire. Let the TOP event be "No signal from the start relay SR when a fire condition is present" – dummy node.

We consider three signal sources:

1 from smoke detectors operating on 2oo3 architecture

2 from fuses connected according to the 1oo4 architecture

3 from manual switch.

If the signal does not come from at least one source, the system will not work.

Let's represent this logical condition on the FIS using three functional nodes (Figure C.12) – No. 2, 3 and 4.

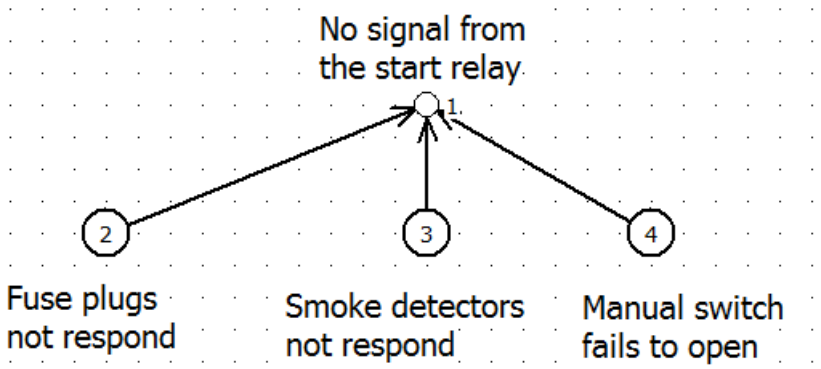


Figure C.12 – First step to develop fault tree

The fire suppression system will also not work if the direct current power supply (DC), manual switch (MS) or start relay (SR) fails – No. 2, 3 and 4.

Let's represent this logical condition on the FIS using another three functional nodes (Figure C.13) – No. 5, 6 and 7.

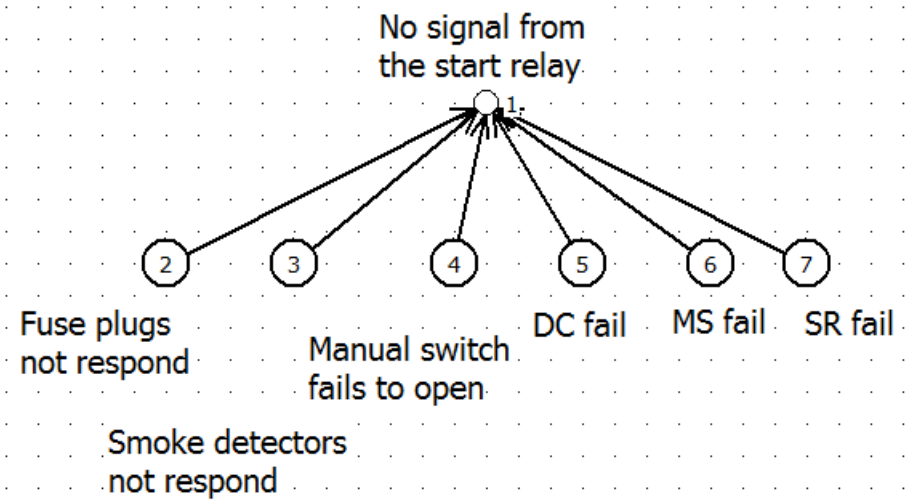


Figure C.13 – Second step to develop fault tree

Since fuses and smoke detectors have architectures other than 1oo1, we will create these architectures inside equivalently nitrated vertices – Figures C.14, C.15.

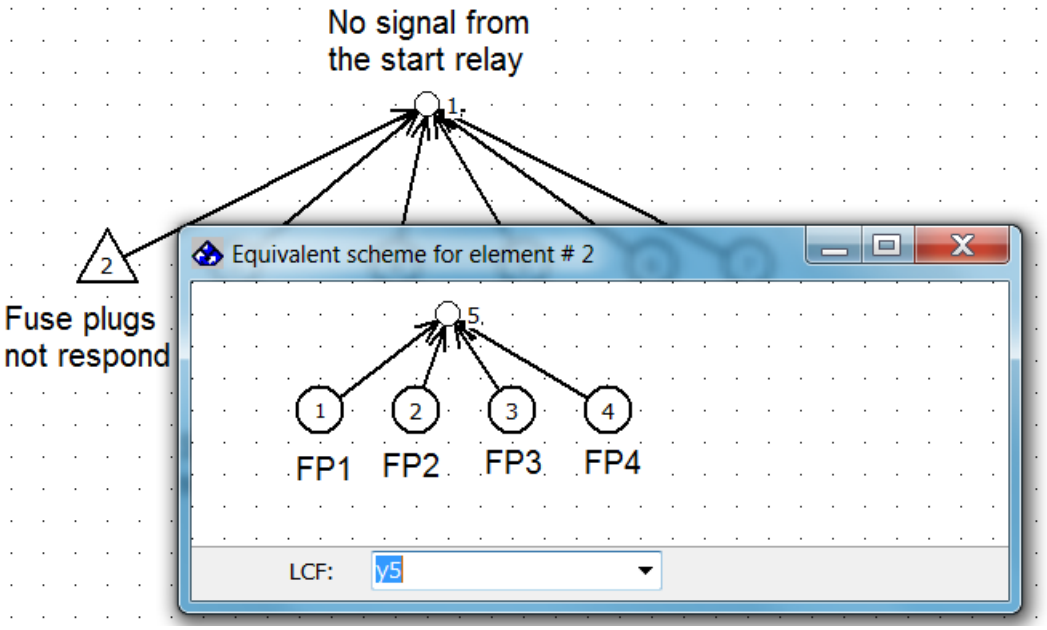


Figure C.14 – Third step to develop fault tree – voting scheme 1oo1

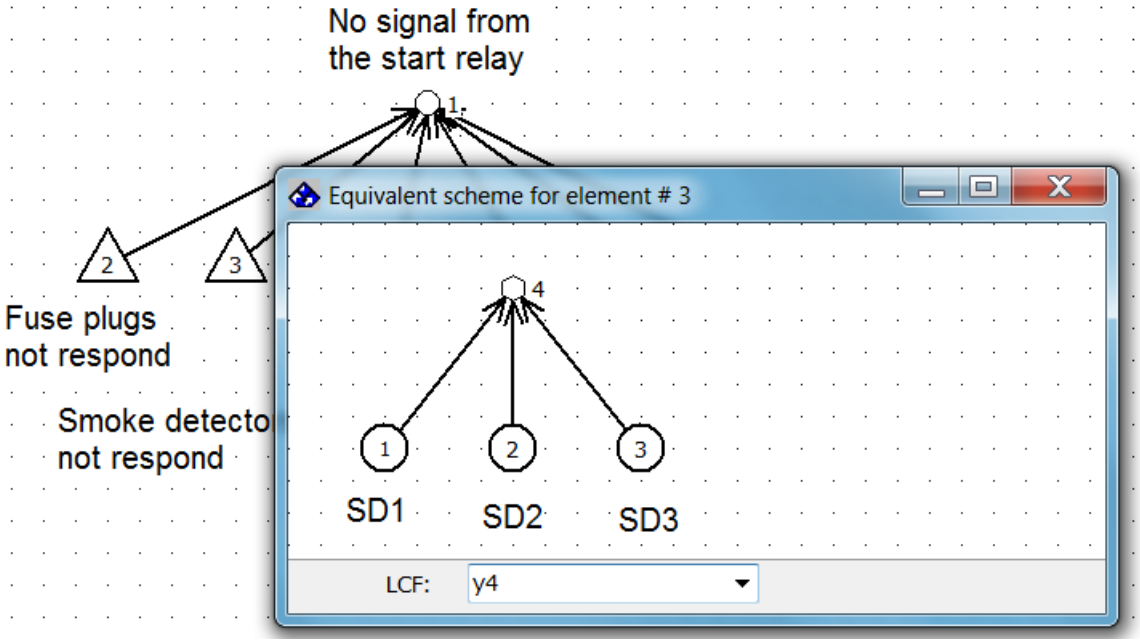


Figure C.15 – Fourth step to develop fault tree – voting scheme 2oo3

The final FIS "Fire Detector System" is shown in Figure C.16.

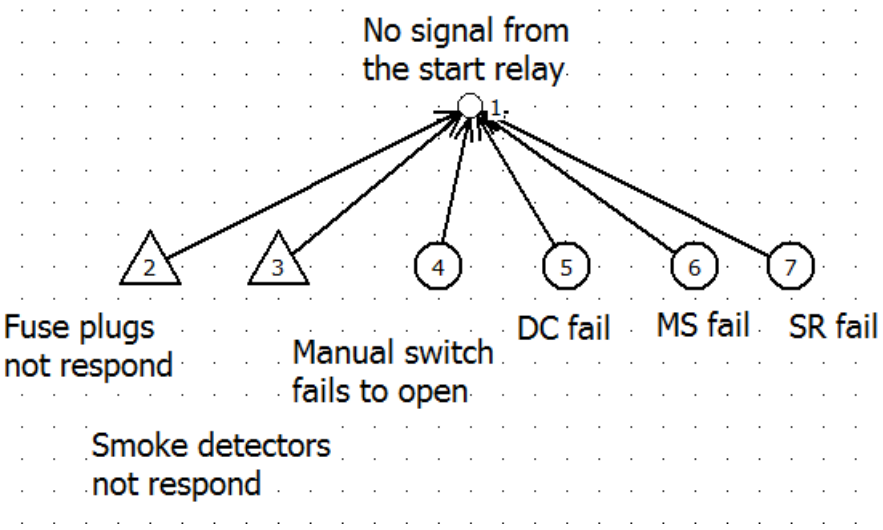


Figure C.16 – Final step to develop fault tree

Appendix D

Event Tree Analysis (ETA)

In many accident scenarios, the initiating event, for example, a ruptured pipeline may have a wide spectrum of possible outcomes, ranging from no consequences to a catastrophe.

An accident development scenario is a sequence of separate logically related events caused by a specific initiating event, leading to the occurrence of damaging factors of the accident and causing damage from the accident to human and/or material resources or components of the natural environment.

In many well-designed systems, number of safety functions, or barriers, are provided to stop or mitigate the consequences of potential accidental (dangerous) events. The safety functions may comprise technical equipment, human interventions, emergency procedure, and combinations of these. Examples of technical safety function are close a valve, emergency shutdown (ESD), fire and gas detection systems (F&G), safety interlock, fire walls, and evacuation systems.

The consequences of the accidental event are determined by how the accident progression is affected by subsequent failure or operation of these safety functions, by human errors made in responding to the accident event, and by various factors like weather conditions and time of the day.

The accident progression is best analyzed by an inductive method. The most used method is the event tree analysis (ETA).

An event tree is a logic tree diagram that starts from a basic initiating event (I_E) and provides a systematic coverage of the time sequence of event propagation to its potential outcomes or consequences. In the development of the event tree, we follow each of the possible sequence of events that result from assuming failure or success of the safety functions affected as the accident propagates.

Each event in the tree will be conditional on the occurrence of the previous events in the event chain. The outcomes of each event are most often assumed to be

binary (*failure* or *success*, *true* or *false*, *yes*, or *no*) but may also include multiple outcomes (e.g., *yes*, *partly*, and *no*).

Event tree analyses have been used in risk and reliability analyses of a wide range of technological systems. The event tree analysis is a natural part of most risk analyses but may be used as a design tool to demonstrate the effectiveness of protective systems in a plant.

An event tree analysis is usually carried out of 6 steps (MR-2004, p.129):

1 Identification of a relevant initiating event that give rise to dangerous consequence.

2 Identification of the safety functions that are design to deal with the initiating event.

3 Construction of the event tree.

4 Description of the resulting accident event sequences.

5 Calculation of probabilities for the identified consequence.

6 Compilation and presentation of the results from analysis.

Basic graphical representation of the event tree according to IEC 62502:2010.

To describe the basic principles of analysis, for clarity, the main graphical representation of the event tree is used, shown in the Figure D.1.

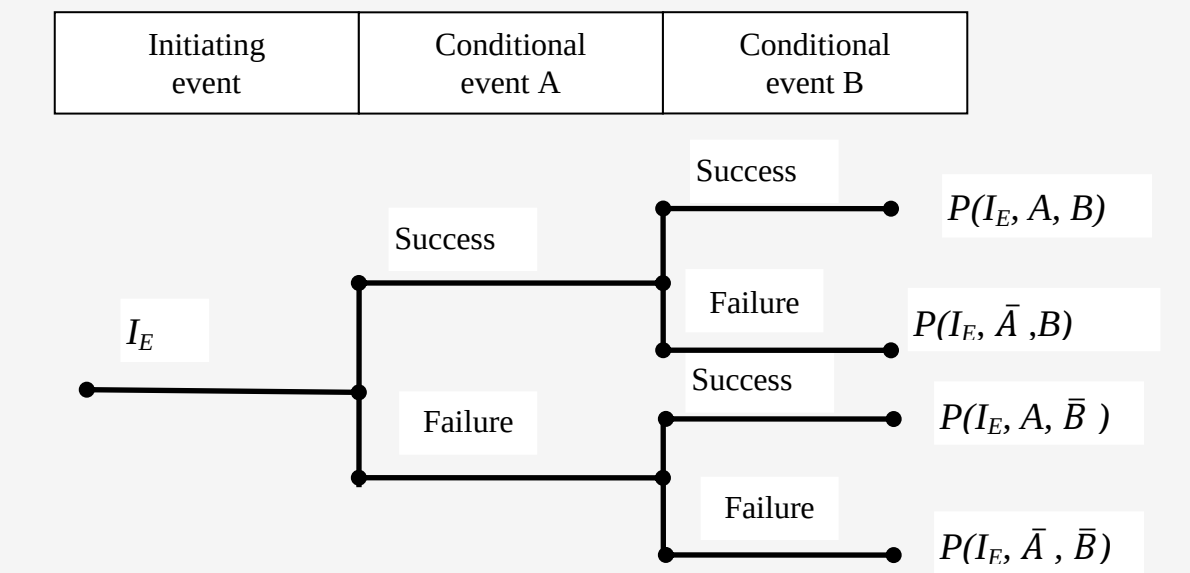


Figure D.1 – Event tree

The following notations are used in the Figure D.1:

- $P(I_E, A, B)$ – the probability of the scenario that initiating event, event A and event B occur;
- $P(I_E, \bar{A}, B)$ – the probability of the scenario that initiating event occur, event A does not occur and event B occur;
- $P(I_E, A, \bar{B})$ – the probability of the scenario that initiating event occur, event A occur, and event B does not occur;
- $P(I_E, \bar{A}, \bar{B})$ – the probability of the scenario that initiating event occur, event A and event B will not occur.

The figure D.2 shows the stages of graphical construction of the event tree by means of the SC ARBITR.

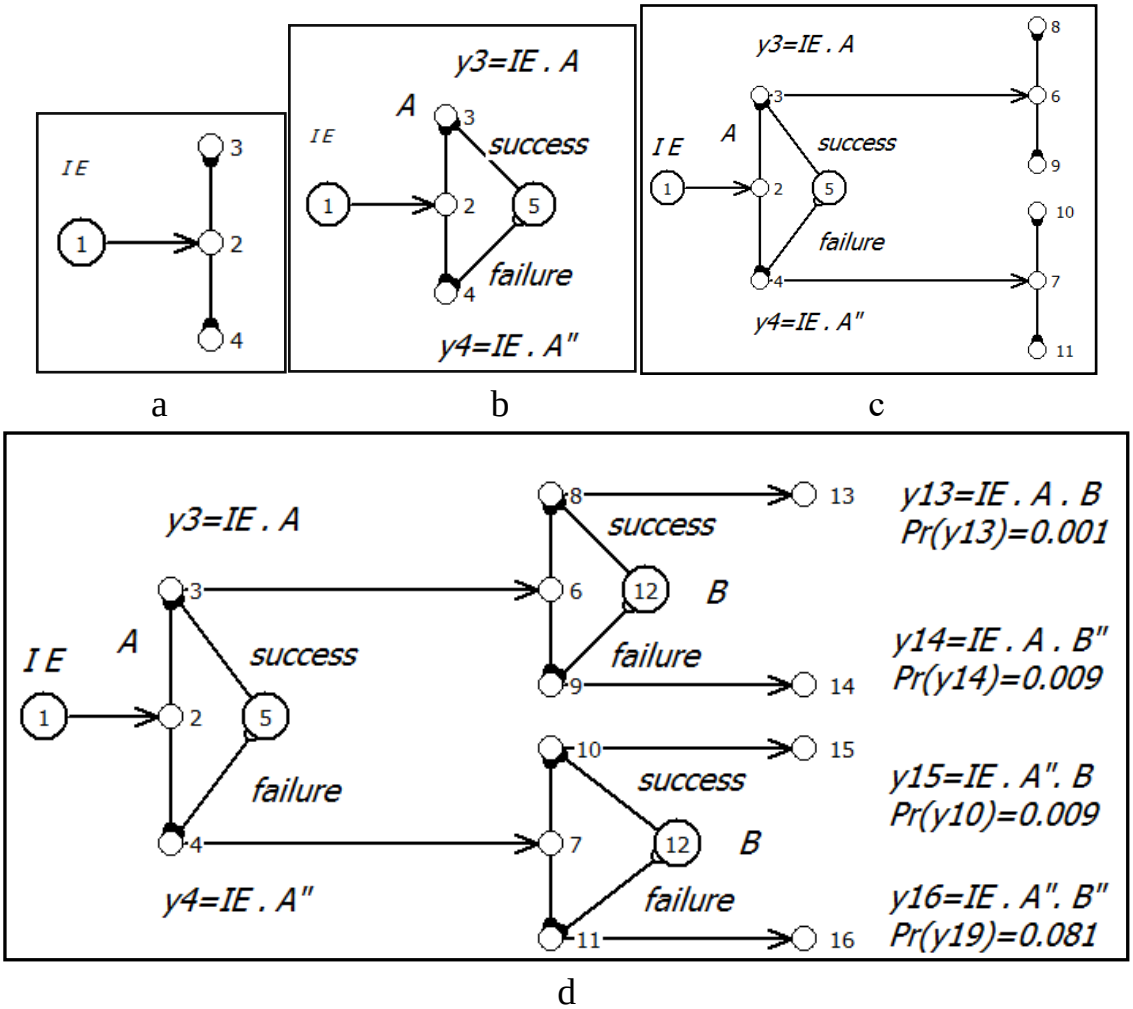


Figure D.2 – Construction an event tree in SC ARBITR

At the beginning on stage Figure D.2, a, the functional node No.1 is displayed. The functional node No.1 simulates the implementation of the initiating event IE. The dummy nodes No.3 and 4 are prepared to simulate the development of the scenario under the influence of the first conditional event (safety function).

At stage Figure D.2, b, intermediate events are modeled. The probability of realization intermediate events depends on the probability of realization of the initiating event, and depends on the probability of the realization or non-realization of the conditional event *A*.

Next, we connect dummy node No.3 with an **AND** edge, and dummy node No.4 with an **NOT-AND** edge. Then the logical function in dummy node No.3 will look like

```
Criterion of functioning

Ys= y3

Logic function consists of 1/1 conjunctions

# conj.  P conj.   F-V conj.    LF
  1  1.0000E-002  1.0000E+000   IE A
```

The logical function in dummy node No.4 will look like

```
Ys= y4

Logic function consists of 1/1 conjunctions

# conj.  P conj.   F-V conj.    LF
  1  9.0000E-002  1.0000E+000   IE A"
```

To simulate the further development of scenarios, the creation of dummy nodes No. 6÷11 is required, as shown in the Figure D.2, c.

Multiplied node No.12 (conditional event *B*) provides modeling of scenarios for the development of events after the implementation of the conditional event *A* at nodes No.8 and 9, or after the non-realization of the conditional event *A* at nodes No.10 and 11 (Figure D.2, d).

For the convenience of further risk analysis, we will form dummy nodes No.13÷16 corresponding to four possible scenarios for the development of events. The logical functions of the considered scenarios are shown at Figure D.2,d.

Since, by definition, risk is a combination of the probability of an event and its consequences, let's consider the "Efficiency/Risk Calculation" modeling mode. To do this, on the "Modeling & calculation parameters" tab, select the "Effectiveness/risk calculation" mode (Figure D.3).

In the "Criterion" table, in the lines of the corresponding dummy nodes, we will introduce some amounts of damage in the implementation of the corresponding scenarios for the development of events (Figure D.3).

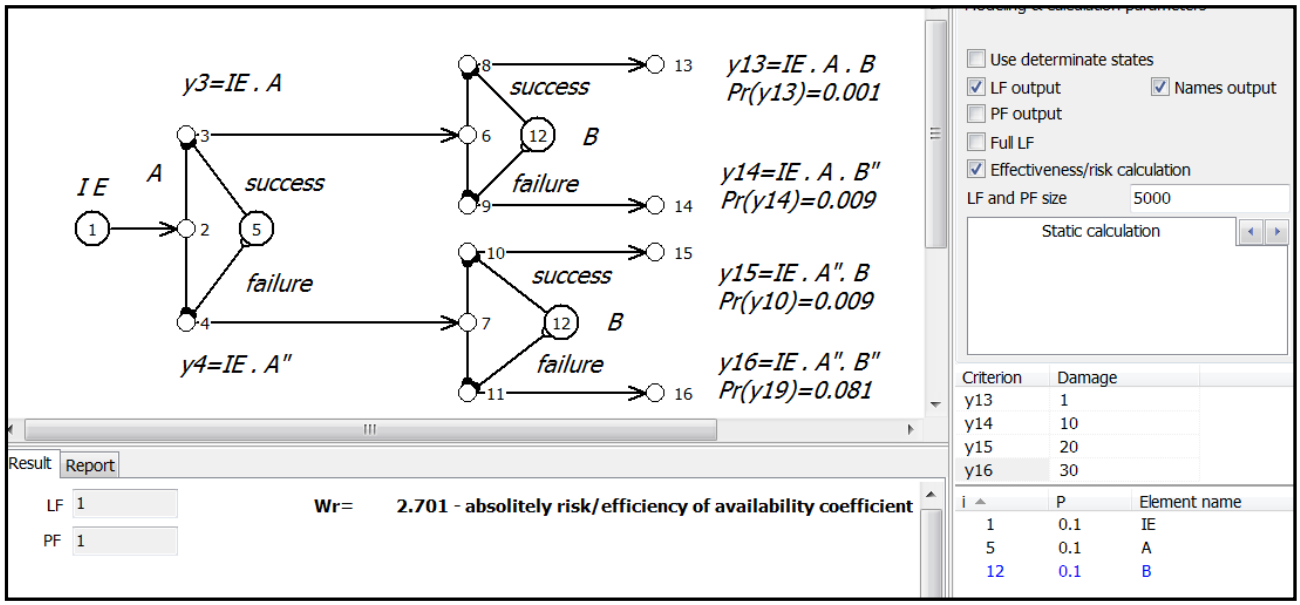


Figure D.3 – Result of Effectiveness/risk calculation

In this mode, the so-called weighted average risk is calculated using the formula

$$W_r = \sum_{i=1}^{n=4} Pr_i \cdot D_i = 0.001 \cdot 1 + 0.009 \cdot 10 + 0.009 \cdot 20 + 0.081 \cdot 3 = 2.701,$$

where Pr_i – probability if realization of i-th scenario;

D_i – damage in the implementation of the i-th scenario.

Conditional events No. 5 and 12 can be equivalent vertices and have an internal structure in the form of RBD or fault trees.

Figure D.4 shows a symbolic representation of the combination of the fault tree and the event tree according to IEC 61078 and IEC 62502.

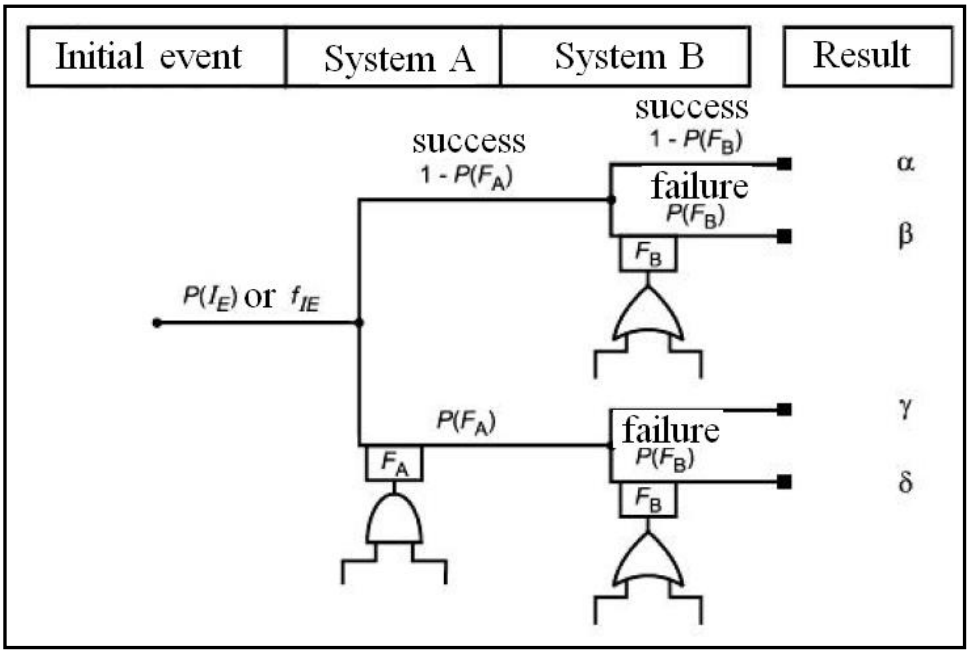


Figure D.4 – Combination of the fault tree and the event tree

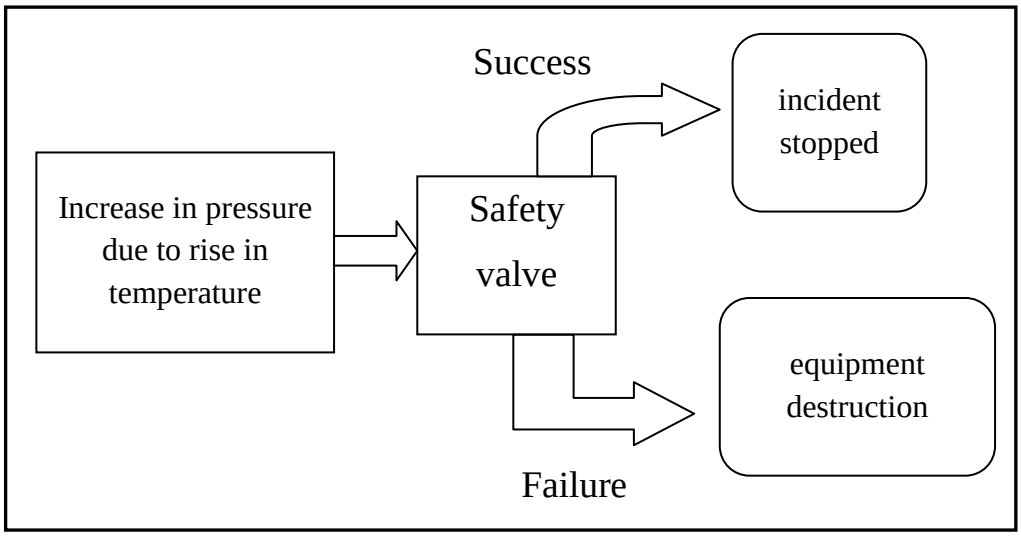
Example D1. **Assessing the effectiveness of safety barriers**

The effectiveness of a barrier is the ability of a technical device to perform a safety function for a certain period of time. Efficiency is expressed either as a percentage or as a probability of performing a certain safety function.

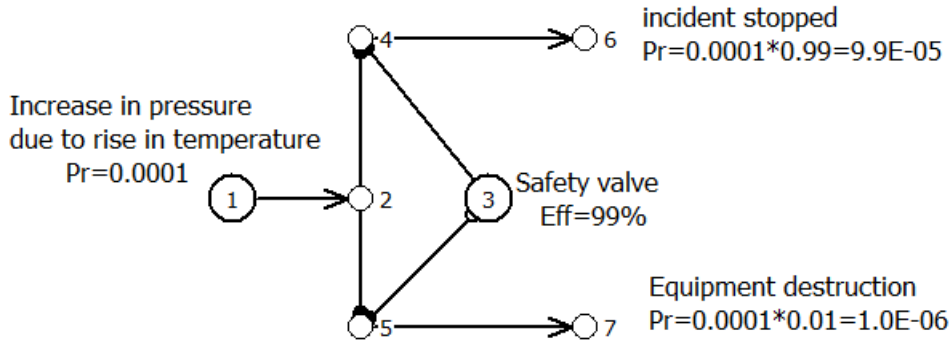
The safety function must reduce the likelihood of a hazardous event occurring.

Let's consider an example of evaluating the effectiveness of the warning valve operation when the pressure in the tank is exceeded with an increase in temperature.

Figure D.5, a shows the operation of the safety valve. Figure D.5, b shows the FIS for evaluating the efficiency of the valve in SC ARBITR.



a



b

Figure D.5 – Evaluating the efficiency of the safety valve

Example D2. **Risk analysis of pedestrian injuries at a pedestrian crossing of the 3rd category**

According to the technical requirements at pedestrian crossings of the 3rd category, if necessary, fencing configurations and light and sound signaling can be installed, as well as accumulation zones can be equipped. These additional measures can increase the likelihood of a correct hazard assessment by a pedestrian.

When building an event tree, consider:

- existing protective measures:

1) warning sign (poster);

2) traffic light automatic signaling;
3) additional technical means of informing pedestrians (signals, speech synthesizers, train direction indicators, etc.);

- human actions (both pedestrian and driver):

1) the pedestrian looks around;
2) the driver sounds the sound signal of the locomotive.

When calculating the probabilities of events, it is assumed that, according to expert data, 5% of pedestrians do not assess the danger of an approaching train, 10% of pedestrians incorrectly assess the danger (they believe that they will have time to cross in front of an approaching train, etc.).

Two types of outcomes are considered: - Incident; - No incident.

The values of the probabilities of outcomes are determined according to the rules for constructing an event tree: the probability of an outcome for each branch is equal to the product of the conditional probabilities for each node.

The overall probability of occurrence of an outcome type is calculated as the sum of all probabilities of occurrence of each outcome of a given type.

Figure D.6 shows an example of an analysis of the risk of pedestrian injury at a pedestrian crossing of the 3rd category based on the construction of an event tree.

The intensity L of the danger of a train colliding with a pedestrian in Figure D.5 is assumed L=1.

As a result of the simulation, the probability of an incident should be 0.0204344; the probability of avoiding an incident will be 0.9795656.

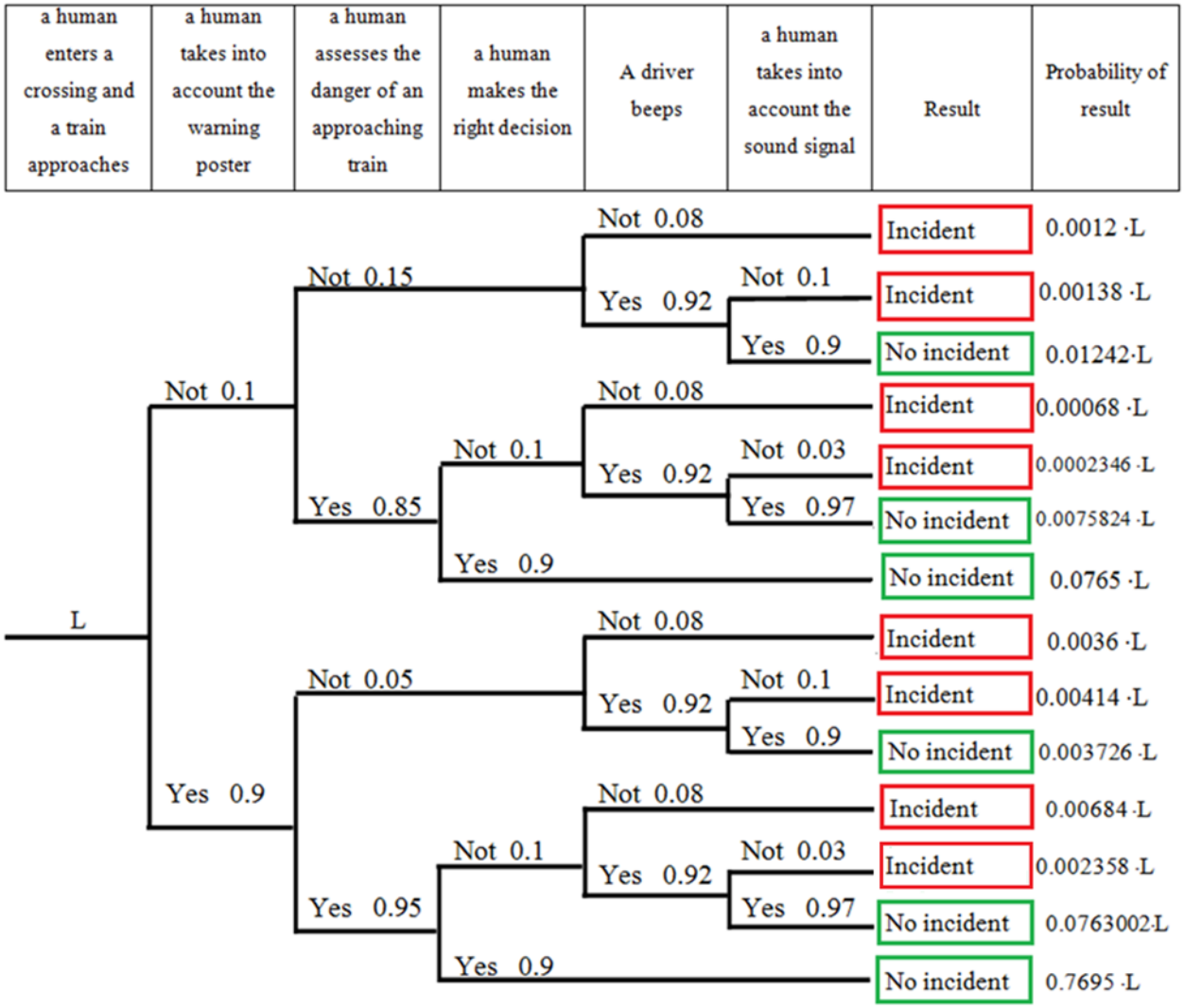


Figure D.6 – Analysis of the risk based on the construction of an event tree

Figure D.7 shows the FIS corresponding to the event tree in Figure D.6

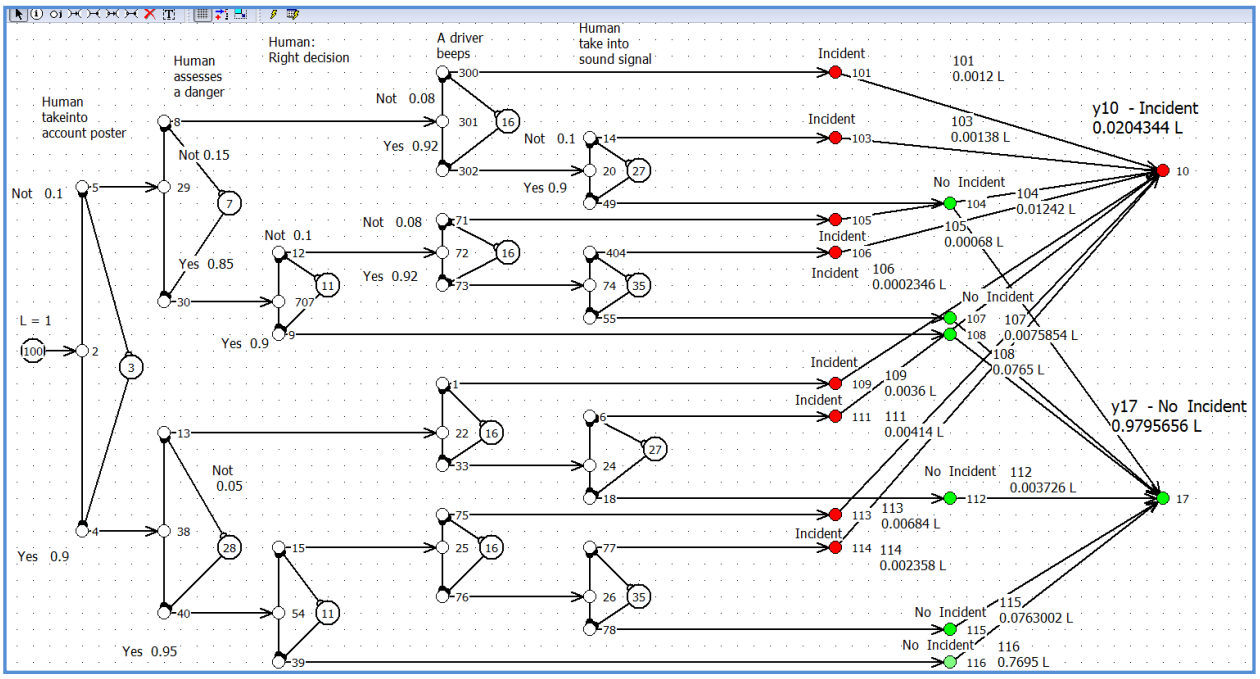


Figure D.7 – Example of construction of fault tree for risk analysis of pedestrian injuries

Example D3

Event Tree Analysis Method in Functional Safety Problems (IEC 61511-3-2016).

The overall objective of the example is to outline a procedure to identify the required safety instrumented functions (SIF) of safety instrumental system (SIS) and establish their safety integrity levels (SILs). The basic steps required to comply are the following:

1 Establish the safety target (tolerable risk) of the process.

2 Perform a hazard and risk analysis to evaluate existing risk.

3 Identify safety function(s) needed.

4 Allocate safety function(s) to protection layers.

NOTE Protection layers are independent from each other.

5 Determine if a SIF is required.

6 Determine required SIL of SIF.

Step 1 establishes the safety target of the process. Step 2 focuses on the risk analysis of the process, and Step 3 derives from the risk analysis what safety

functions are required and what risk reduction they need to meet the safety target. After allocating these safety functions to protection layers in Step 4, it will become clear whether a safety instrumented function is required (Step 5) and what SIL it will need to meet (Step 6).

Consider a process comprised of a pressurized vessel containing volatile flammable liquid with associated instrumentation (see Figure D.8). Control of the process is handled through a Basic Process Control System (BPCS) that monitors the signal from the level transmitter and controls the operation of the valve. The engineered systems available are: a) an independent pressure transmitter to initiate a high pressure alarm and alert the operator to take appropriate action to stop inflow of material; and b) in case the operator fails to respond, a non-instrumented protection layer to address the hazards associated with high vessel pressure. Releases from the protection layer are piped to a knock out tank that relieves the gases to a flare system. It is assumed in this example that the flare system is under proper permit and designed, installed and operating properly; therefore potential failures of the flare system are not considered in this example.

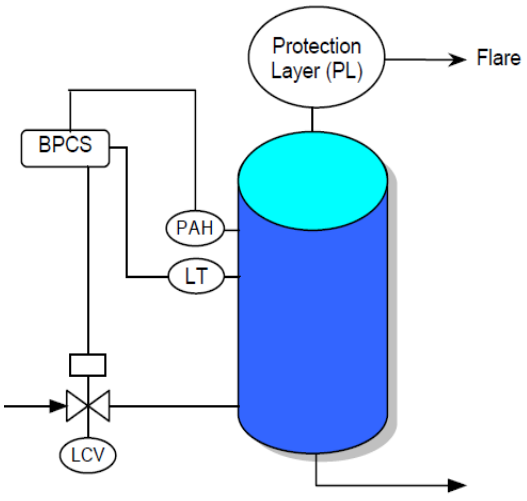


Figure D.8 – Pressurized vessel with existing safety systems

Key:

- PL – Protection Layer for additional mitigation (that is, dikes, pressure relief, restricted areas, holding tank)

- PAH – Pressure Alarm High
- LT – Level Transmitter
- LCV – Level Control Valve
- BPCS – Basic Process Control System

Process safety target level

A fundamental requirement for the successful management of industrial risk is the concise and clear definition of a desired process safety target level (tolerable risk). This may be defined using national and International Standards and regulations, corporate policies, and input from concerned parties such as the community, local jurisdiction and insurance companies supported by good engineering practices. The process safety target level is specific to a process, a corporation or industry. Therefore, it should not be generalized unless existing regulations and standards provide support for such generalizations. For the illustrative example, assume that the process safety target is set as an average release rate of less than 10–4 per year based on the expected consequence of a release to environment.

Hazard analysis

A hazard analysis to identify hazards, potential process deviations and their causes, available engineered systems, initiating events, and potential hazardous events (accidents) that may occur should be performed for the process. This can be accomplished using several qualitative techniques.

One such technique that is widely applied is a Hazard and Operability study (HAZOP). The hazard and operability study identifies and evaluates hazards in a process plant, and non-hazardous operability problems that compromise its ability to achieve design productivity.

The objective of this HAZOP study analysis is to evaluate hazardous events that have the potential to release the material to the environment. An abridged list is shown in Table D.1 to illustrate the HAZOP results.

The results of the HAZOP study identified that an overpressure condition could result in a release of the flammable material to the environment. This is an initiating event that could propagate into a hazardous event scenario depending on the response of the available engineered systems. If a complete HAZOP was conducted for the process, other initiating events that could lead to a release to the environment may include leaks from process equipment, full bore rupture of piping, and external events such as a fire. For this illustrative example, the overpressure condition is examined.

Table D.1 – HAZOP study results

Item	Deviation	Causes	Consequences	Safeguards	Action
Vessel	High level	Failure of BPCS	High pressure	Operator	
	High pressure	1) High level, 2) External fire	Release to environment	1) Alarm, operator, protection layer 2) Deluge system	Evaluate conditions for release to environment
	Low/no flow	Failure of BPCS	No consequence of interest		
	Reverse flow		No consequence of interest		

The assessment of process risk using semi-quantitative techniques can be distinguished in the following major steps. The first four steps can be performed during the HAZOP study.

1 Identify process hazards.

2 Identify safety layer composition.

3 Identify initiating events.

4 Develop hazardous event scenarios for every initiating event.

5 Ascertain the frequency of occurrence of the initiating events and the reliability of existing safety systems using historical data or modeling techniques (Fault Tree Analysis, Markov Modeling).

6 Quantify the frequency of occurrence of significant hazardous events.

7 Evaluate the consequences of all significant hazardous events.

8 Integrate the results (consequence and frequency of an accident) into risk associated with each hazardous event.

In Figure D.9, a simple fault tree is shown that identifies some events that contribute to the development of an overpressure condition in the vessel. The top event, vessel overpressurization, is caused due to the failure of the basic process control system (BPCS), or an external fire (see Table D.1). The fault tree is shown to highlight the impact of the failure of the BPCS on the process. The BPCS does not perform any safety functions. Its failure, however, contributes to the increase in demand for the SIS to operate. Therefore, a reliable BPCS would create a smaller demand on the SIS to operate. The fault tree can be quantified, and for this example the frequency of the overpressure condition is assumed to be in the order of 10–1 in one year.

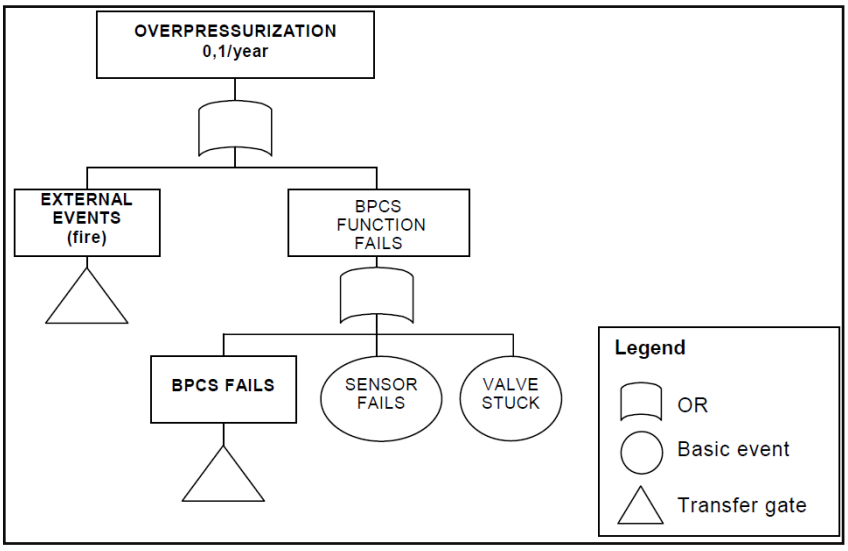


Figure D.9 – Fault tree for overpressure of the vessel

Once the frequency of occurrence of the initiating event has been established, the success or failure of the safety systems to respond to the abnormal condition is modeled using event tree analysis (ETA).

Figure D.9 shows the potential release scenarios that could be developed given an overpressure condition.

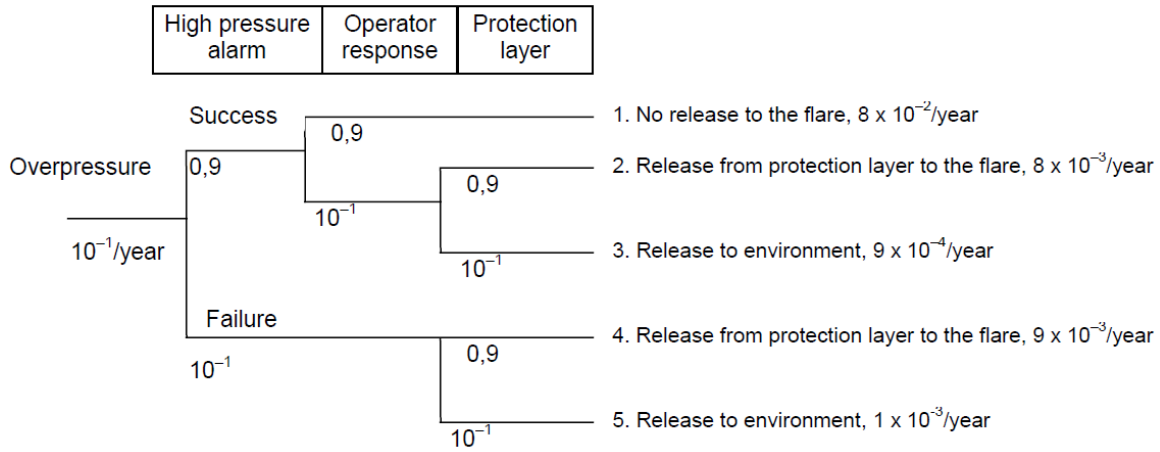


Figure D.10 – Hazardous events with existing safety systems

In Figure D.10, five hazardous events are identified, each with a frequency of occurrence and a consequence in terms of potential releases. Accident scenario 1, no release, is the designed condition of the process. Furthermore, hazardous events 2 and 4 release material to the flare and are also considered as designed conditions of the process. The remainder scenarios, that is, 3 and 5, range from a frequency of occurrence in the order of $9{\cdot}10^{-4}$ to about $1{\cdot}10^{-3}$ per year and will release material to the environment.

As was stated earlier, plant specific guidelines establish the safety target level as: no release of material to the environment with a frequency of occurrence greater than 10^{-4} in one year. Given the frequency of occurrence of the hazardous events and consequence data in Figure D.10, risk reduction is necessary in order for accidents 3 and 5 to be below the safety target level.

Risk reduction using other protection layers

Assume that an additional completely independent, protection layer is introduced to augment the existing safety systems. Figure D.11 shows the process with the new protection layer. Event tree analysis is employed to develop all the

potential hazardous events. From Figure D.11, it can be seen that seven release accidents may occur, given the same overpressure condition.

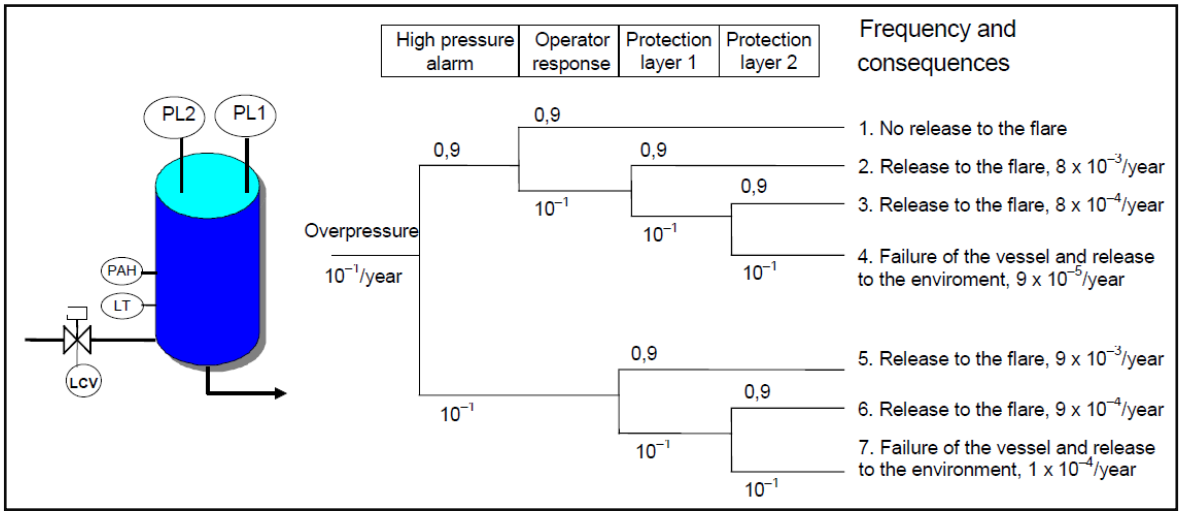


Figure D.11 – Hazardous events with redundant protection layer

Examination of the frequency of occurrence of the modeled hazardous events in Figure D.11 shows that the safety target level for the vessel has not been met because hazardous events 4 and 7 release material to the environment and are still at or above the safety target. In fact, the total frequency of a release to the environment is 1.9×10^{-4} per year. At this point the feasibility of using external risk reduction facilities should be evaluated. To protect against an overpressure and the release of the flammable material SIS is required.

Risk reduction using a safety instrumented function

In order to reduce the overall frequency of releases to the atmosphere, a new SIL2 safety instrumented function implemented in a SIS is required to meet the safety target level. The new safety instrumented function is shown in Figure D.12.

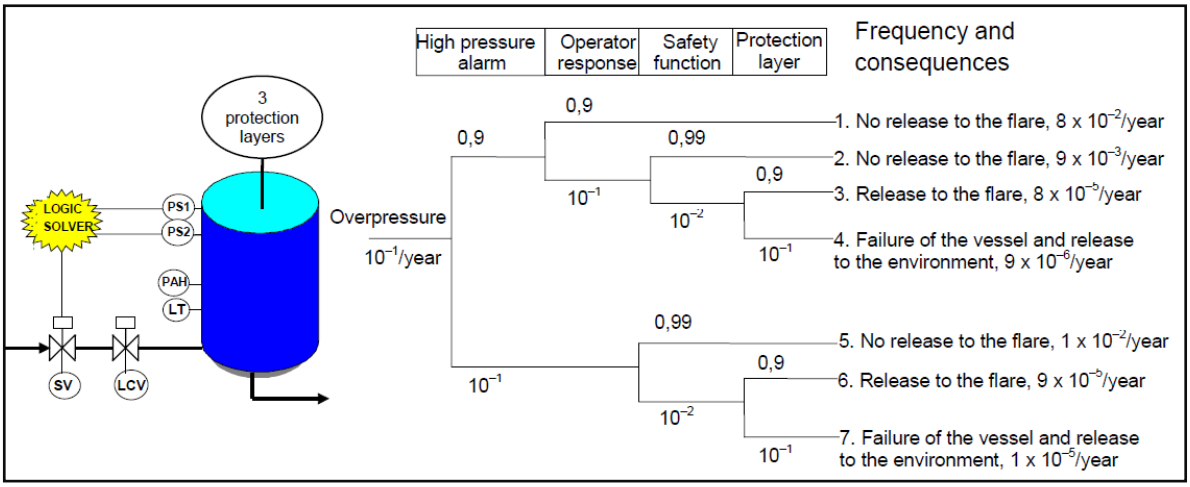


Figure D.12 – Hazardous events with SIL 2 SIS safety function

The goal in this step is to determine if a SIL2 SIF will provide the required risk reduction and allow the achievement of the safety target level. For example, the new safety instrumented function can use dual, safety dedicated, pressure sensors in a 1oo2 configuration sending signals to a logic solver. The output of the logic solver controls one additional shutdown valve.

The new SIF with SIL2 is used to minimize the frequency of a release from the pressurized vessel due to an overpressure. Figure D.12 presents the new safety layer and provides all the potential accident scenarios. As can be seen from this figure, the frequency of any release from this vessel can be reduced to 10^{-4} per year or lower and the safety target level can be met provided the SIF can be evaluated to be consistent with SIL2 requirements. The total frequency or releases to the environment (sum of frequencies of scenarios 4 and 7) has been reduced to 1.9×10^{-5} per year, below the safety target of 10^{-4} per year.

Appendix E

Importance Analysis

The contribution of an element or combination of elements to the reliability of the system or the probability of occurrence of an emergency condition is called importance.

The program calculates the following importance measures during an analysis:

- Birnbaum Importance for component
- Fussell-Vesely Importance for component
- Fussell-Vesely Importance for conjunctions
- Risk Reduction Ratio
- Risk Increase Ratio

Importance is a function of the reliability characteristics of the elements and the structure of the system. Importance analysis is similar to sensitivity analysis and is sometimes referred to as partial sensitivity analysis. Importance analysis can be useful in the design of technical systems and the organization of operation based on a risk oriented approach.

Activities to improve the reliability or optimize the repair and maintenance system can be carried out taking into account the importance of the elements, starting with the most the importance elements.

Birnbaum Importance for component

Birnbaum importance of the element is a partial derivative of the system reliability indicator *Rs* (*Qs*) with respect to the element reliability parameters *ri* (*qi*).

Let the positive contribution to be a change in the system reliability indicator when the element reliability indicator changes from the current value to 1. A negative contribution is a change in the system reliability index when the element reliability index changes from the current value to 0. The geometric interpretation of the positive and negative contributions on the example of calculating the reliability of the system is shown in Figure E.1.

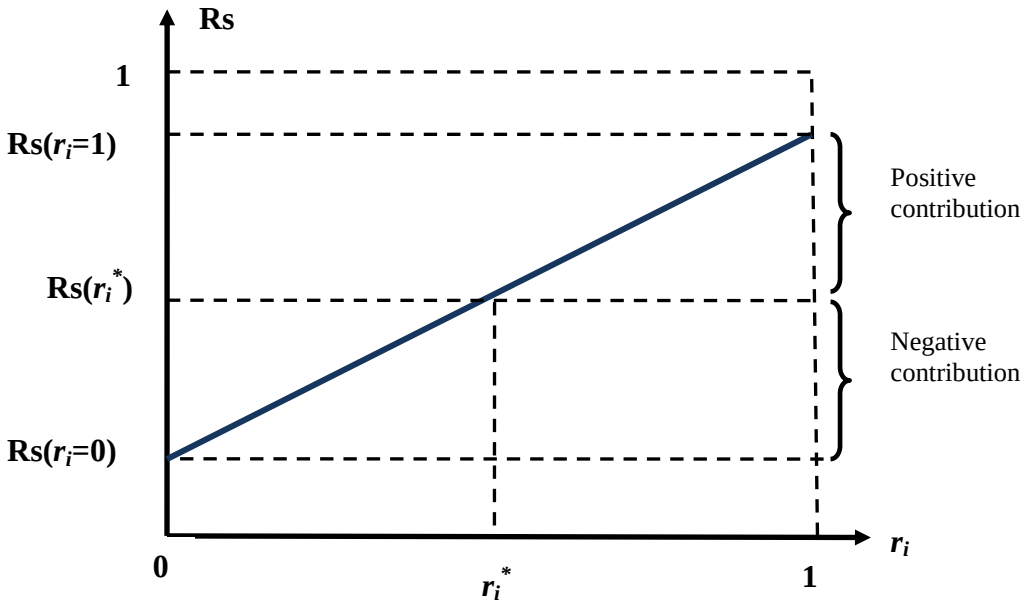


Figure E.1 – Geometric int erpretation of the positive and negative contributions of an element

For coherent technical systems, the reliability of the system *Rs* increases with the increase in the reliability of the *i*-th element. Let at the current value of the reliability of the element r_i, the reliability of the system is equal to $Rs(r_i)$. With an increase in the reliability of the element to r_i =1, the value of the reliability of the system increases to $Rs(r_i$ =1). The difference $Rs(r_i)$ – $Rs(r_i$ =1) is called the positive contribution of the *i*-element – PCon.

When the reliability of the element decreases to r_i =0, the value of the reliability of the system decreases to $Rs(r_i$ =0). The difference $Rs(r_i$ *) – $Rs(r_i$ =0) is called the negative contribution of the i-element – NCon. Importance ζ_i is calculated as the sum of the positive and negative contributions, i.e.

$$\zeta_i = PCon + NCon.$$

Mathematically, the significance of the element according to Birnbaum is the partial derivative of the probabilistic polynomial with respect to the probability of failure-free operation of the i-element**, that is**

$$\zeta_{\text{i}} = \frac{\partial Rs}{\partial r_i} = Rs(r_i = 1) - Rs(r_i = 0)$$

The importance of an element according to Birnbaum is calculated in a static mode both for reliability block diagrams and for fault trees.

Fussell-Vesely Importance for conjunctions

Fussell-Vesely Importance for conjunctions *FVcon* shows the contribution of a conjunction to a systemic result. Most often used to analyze the importance of emergencies (minimal cut sets). The importance *j*-th conjunction *FVconj* is calculated as the ratio of the probability of conjunction to the value of the system indicator.

$$FVconj = \frac{\Pr(con\ j)}{I_{SYS}}$$

where *Pr(conj*) – probability of *j*-th conjunction realization;

Isys – system indicator.

The Fussel-Vesely importance of the element

The Fussel-Vesely importance of the *i*-th element is calculated in the fault tree analysis for the approximate calculation mode as the probability that the failure of the *i*-th element contributes to the failure of the system. The SC ARBITR implements an algorithm for calculating the approximate value of Fussel-Vesely importance IiFV, calculated by the formula

$$I_i^{FV} = 1 - \frac{1 - \prod_{i \notin K_i}(1 - \alpha_i)}{Q_{sys}^{+}}$$

where $\prod_{i \notin K_i}(1 - \alpha_i)$ – multiplication of the probabilities of cut set α_i is carried out over those cut sets that do not include element *i*;

α_i – probability of realization of the *i*-th minimum cut set;

Q_{sys}^{+} – upper bound on the probability of system failure.

Risk Reduction Ratio

The risk reduction importance measure is an indication of how many the results would be reduced if the specific event probability equaled zero, normally corresponding to a totally reliable piece of equipment.

The risk reduction ratio (RRR) is determined by evaluating the fault tree minimal cut set upper bound (or the sequence frequency) with the basic event probability set to its true value and dividing it by the minimal cut set upper bound (sequence frequency) calculated with the basic event probability set to zero.

In equation form, the risk reduction ratio RRR is

$$RRR = \frac{Q_{sys}^{+}}{Q_{sys}^{+}(q_i = 0)}$$

Risk Increase Ratio

Risk Increase Ratio (RIR) is an indication of how much the top event probability would go up if the specific event had probability equal to 1.0, normally corresponding to totally unreliable equipment.

The risk increase ratio is determined by evaluating the minimal cut set upper bound with the basic event probability set to 1.0 and dividing it by the minimal cut set upper bound evaluated with the basic event probability set to its true value. In equation form, the risk increase ratio.

RIR is

$$RIR = \frac{Q_{sys}^{+}(q_i = 1)}{Q_{sys}^{+}}$$

Example E1: Bridge Structure (RBD)

FIS in the form of a reliability block diagram (RBD) and a report on the results of modeling the reliability of the structure are shown in Figure 47 and 49, respectively. Table E.1 shows the initial data and the results of calculating the importance of the elements when using the FIS in the form of RBD and in the form of a fault tree.

Table E.1 – Birnbaum Importance by elements of bridge structure

i	*pi*	*R(i=1)*	*R(i=0)*	*Birn R i*	*Q(i=1)*	*Q(i=0)*	*Birn Q i*
1	0.9	0.99655	0.8892	0.10735	0.1108	0.00345	0.10735
2	0.9	0.99655	0.8892	0.10735	0.1108	0.00345	0.10735
3	0.95	0.9891	0.9234	0.0657	0.0766	0.0109	0.0657
4	0.95	0.9891	0.9234	0.0657	0.0766	0.0109	0.0657
5	0.8	0.987525	0.978975	0.00855	0.021025	0.012475	0.00855

The following notations are used in the table:

- *i*, *pi* – the number of the circuit element and its probability of failure-free operation;
- *R*(*i*=1), *R*(*i*=0) – the *probability* of failure-free operation of the bridge circuit with the probability of failure-free operation of the *i*-th element equal to 1 and 0, respectively;
- *Birn R i* – importance *according* to Birnbaum of the *i*-th element for the structural diagram of reliability;
- *Q*(*i*=1), *Q*(*i*=0) – the probability of *failure* of the bridge circuit when the probability of failure of the *i*-th element is equal to 1 and 0, respectively;
- *Birn Q i* – Birnbaum *importance* of the *i*-th element for the fault tree.

The table shows that the Birnbaum importance of elements values do not depend on the calculation model (RBD or fault tree).

Example E2: Bridge Structure (FT)

The figure E.2 shows the FIS of the bridge circuit in the form of a fault tree, corresponding to the reliability block diagram in Figure 49.

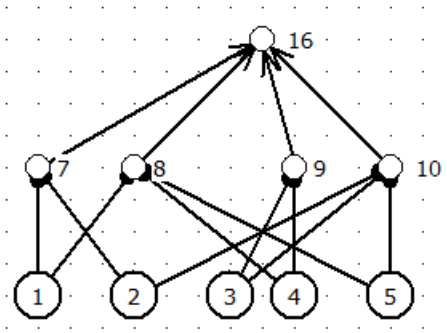


Figure E.2 – Bridge circuit Fault Tree

Table E.2 shows the initial data and the results of calculating the Fussel-Veseli importance for minimal paths (FV MP) in the case of CCH modeling and for minimum failure cross sections (MCS) for the case of fault tree modeling.

Table E.2 – Initial data and results of calculating the Fussel-Veseli importance for minimum paths and minimum sections

i	***pi***	***R(i=1)***	***R(i=0)***	***Birn R i***	***Q(i=1)***	***Q(i=0)***	***Birn Q i***
1	0.9	0.99655	0.8892	0.10735	0.1108	0.00345	0.10735
2	0.9	0.99655	0.8892	0.10735	0.1108	0.00345	0.10735
3	0.95	0.9891	0.9234	0.0657	0.0766	0.0109	0.0657
4	0.95	0.9891	0.9234	0.0657	0.0766	0.0109	0.0657
5	0.8	0.987525	0.978975	0.00855	0.021025	0.012475	0.00855

The importance indicators are calculated as the ratio of the probabilities of implementing the minimum paths and minimum sections to the estimate of the probability of failure-free operation of the bridge circuit or to an approximate estimate of the probability of failure of the structure, i.e.

$$FV\ MP_i = \frac{\Pr(MP_i)}{Rb} \quad ; \quad FV\ MCS_j = \frac{\Pr(MCS_j)}{Q_b^*},$$

where $FV\ MP_i$, $FV\ MCS_i$ – are Fussel-Veseli importance indicators for the *i*-th minimum path and the *j*-th minimum section, respectively;

$\Pr(MP_i)$, $\Pr(MCS_j)$ – are the probabilities of realizing the *i*-th minimum path and the *j*-th minimum section, respectively;

Rb, Q_b^* – the probability of no-failure operation and the upper limit of the estimate of the probability of failure of the bridge circuit (given in the table in the bottom line).

Appendix F

Common-cause failures

The term common-cause failures (CCF) has for a long time been discussed in relation to both risk and reliability analysis. Still, there is no generally accepted definition of a CCF that applies for all types of systems.

In the nuclear power industry, a CCF is defined as Common-cause failure.

Common-cause failures: Dependent failures in which two or more component fault states exist simultaneously, or within a short time interval, and are a direct result of a shared cause.

According to this definition, the components do not have to fail at the same time, but the components must be in a fault state at the same time, or nearly the same time.

On this basis, IEC 61508-4 gives the following definition of a CCF:

Common-cause failures: Failures, that are the result of one or more events, causing concurrent failures of two or more separate channels in a multiple channel system, leading to system failure.

There are two issues related to this definition. The first is that the common-cause is specified to be "one or more events." A condition, such as "higher humidity than specified," is not an event, and can therefore not be a common-cause according to the IEC 61508 definition. The other issue is related to the term "concurrent failures." A failure is an event that takes place at a specific time. The term used in the definition therefore implies that the failure events must occur concurrently, which can be interpreted as rather close in time. This is in opposition to the CCF definition used in the nuclear power industry.

New Definition. By combining the two definitions above, the author proposes a new definition of a CCF of a SIF:

Common-cause failures: Failures, that are the direct result of a shared cause, in which two or more separate channels in a multiple channel system are in fault state simultaneously, leading to system fault.

Useful to split CCF causes into root causes and coupling factors.

A root cause of a failure is the most basic cause that, if corrected, would prevent recurrence of this and similar failures. There is often a series of causes that can be identified, one leading to another. This series of causes should be pursued until the fundamental, correctable cause has been identified.

The concept of root cause is tied to that of defense, because there are, in many cases, several possible corrective actions (i.e., defenses) that can be taken to prevent recurrence. Knowledge about root causes allows system designers to incorporate countermeasures for reducing the susceptibility to both single failures and CCFs.

A coupling factor is a property that makes multiple items susceptible to failure from a single shared cause.

Such properties include:

- Same design
- Same hardware
- Same software
- Same installation staff
- Same maintenance or operation staff
- Same procedures
- Same environment
- Same location.

There are three overall measures that can be used to reduce the probability of dangerous CCFs. These are: (a) Reduce the overall number of random hardware and systematic failures. (b) Maximize the independence of the channels. (c) Reveal

nonsimultaneous CCFs while only one, and before a second, channel has been affected.

Modeling and analysis of CCF as part of a risk or reliability study should, in general, comprise at least the following steps:

1 Development of system logic models. This activity comprises system familiarization, system functional failure analysis, and establishment of system logic models (e.g., fault trees, reliability block diagrams, and event trees).

2 Identification of common-cause component groups. The groups of components which the independence assumption is suspected not to be correct are identified.

3 Identification of root causes and coupling factors. The root causes and coupling factors are identified and described for each common-cause component group. Suitable tools are checklists and root cause analysis.

4 Assessment of component defenses. The common-cause component groups are evaluated with respect to their defenses against the root causes that were identified in the previous step.

5 Explicit modeling. Explicit CCF causes are identified for each common-cause component group and included into the system logic model.

6 Implicit modeling. Residual CCF causes that were not covered in the previous step are included in an implicit model as discussed later in this section. The parameters of this model have to be estimated based on checklists or from available data.

7 Quantification and interpretation of results. The results from the previous steps are merged into an overall assessment of the system. The step also covers importance, uncertainty, and sensitivity analyses and reporting of results. In most cases, we are not able to find high-quality input data for the explicitly modeled CCF causes. However, even with low-quality input data, or guesstimates, the result is

usually more accurate than by including the explicit causes into a general (implicit) CCF model.

The idea of the beta-factor model is to split the failure rate, λ, for a channel into two parts, one part, $\lambda^{(i)}$, covering the individual failures of the channel, and another part, $\lambda^{(c)}$, covering CCFs.

$$\lambda = \lambda^{(i)} + \lambda^{(c)} \quad \text{(F-1)}$$

The beta-factor, β, is introduced as

$$\beta = \frac{\lambda^{(c)}}{\lambda} \quad \text{(F-2)}$$

and is the fraction of all the failures of a channel that are common-cause failures.

True rate λ and the factor β as

$$\lambda^{(c)} = \beta\lambda$$

$$\lambda^{(i)} = (1 - \beta)\lambda$$

A consequence of the beta-factor model is that when a CCF occurs, it affects all the items of the system, such that we either have individual failures or a total failure affecting all items.

EXAMPLE F1. loo2 voted group of identical channels

Consider a group of two identical channels voted loo2 with DU (dangerous undetected) failure rate λ_{DU}. An external event may occur that causes DU failure in both channels of the voted group. This external event can be represented as a "hypothetical" component (CCF) that is in series with the rest of the voted group. The voted group is illustrated by a reliability block diagram in Figure F.1. The group is proof-tested with proof test interval r and we assume that the proof tests are perfect. When using the beta-factor model, the failure rate of component CCF is $\lambda^{(c)} = \beta\lambda$, while the two channels in the parallel structure in Figure F1 may be considered as independent with individual failure rate $\lambda^{(i)} = (1 - \beta)\,\lambda$.

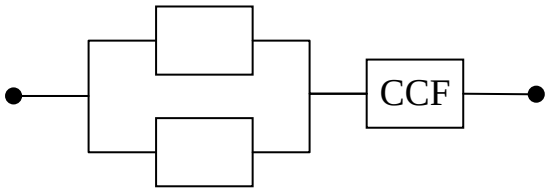


Figure F.1 Group of two identical channels voted loo2 and a CCF component

(Example F1)

The PFDavg of the voted group is

$$PFD_{avg} \approx \underbrace{\frac{[(1-\beta)\lambda_{DU}TI]^2}{3}}_{individual} + \underbrace{\beta\frac{\lambda_{DU}TI}{2}}_{CCF}$$

where the part "Individual" is the PFDavg of a loo2 structure with individual failure rate $\lambda_{DU}^{(i)} = (1-\beta)\lambda_{DU}$ and the part "CCF" is the PFDavg of the single CCF component with failure rate $\lambda_{DU}^{(c)} = \beta\lambda_{DSU}$.

BIBLIOGRAPHY

1. Mozhaev A.S. General logic probabilistic method of complex system reliability analysis. Leningrad, 1988, 68 p.
2. Mozhaev A.S., Gromov, V.N. Theoretical base of logic probabilistic method of automated system simulating. St.Petersburg, 2000, 145 p.
3. Mozhaev A.S. Theory and practice of automated logic structural system simulating. International Conference on Informatics and Control (ICI&C'97). Vol. 3. St.Petersburg: SPIIRAS, 1997, P.1109–1118.
4. Mozhaev A.S. Software for automated logic structural simulating of complex systems (SC ASLS 2001) // Proceedings of International Scientific School "Simulating and analysis of complex systems' reliability, risk, and quality" (MA RRQ - 2001). St.Petersburg, 2001, P.56–61.
5. Ernest J. Henley, Hiromitsu Kumamoto. Reliability engineering and risk assessment. Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1981.
6. Ryabinin I. Reliability of engineering systems. Principles and analysis. English translation, Mir Publishers, 1976.