

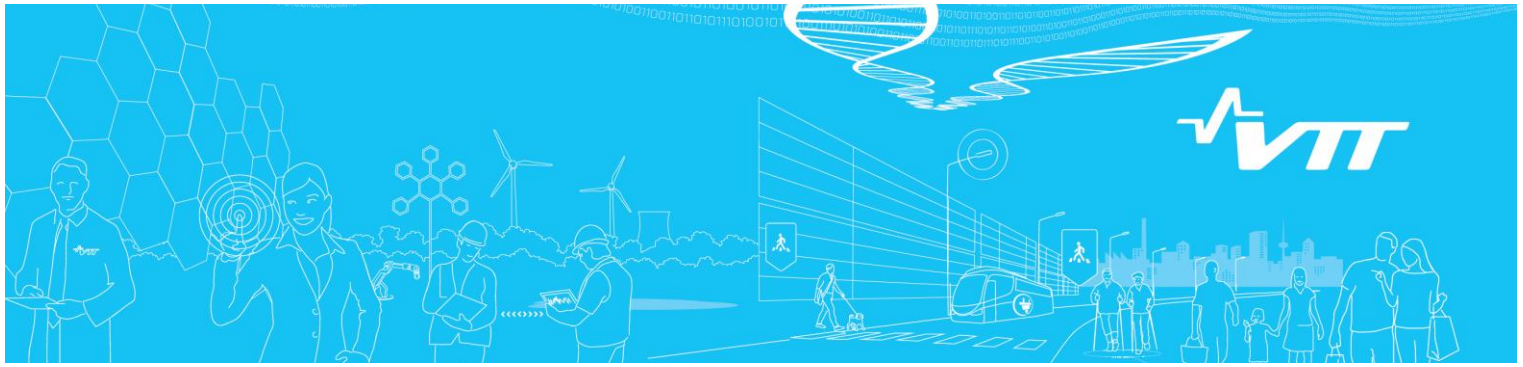
Title	Counterintuitive results from Bayesian belief network software reliability model
Author(s)	Tyrväinen, Tero
Citation	Research Report : VTT-R-04235-14 VTT, 2014
Date	2014
Rights	This report may be downloaded for personal use only.

VTT
<http://www.vtt.fi>
P.O. box 1000
FI-02044 VTT
Finland

By using VTT Digital Open Access Repository you are bound by the following Terms & Conditions.

I have read and I understand the following statement:

This document is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of this document is not permitted, except duplication for research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered for sale.





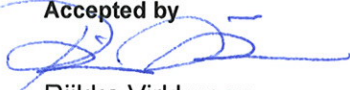
RESEARCH REPORT

VTT-R-04235-14

Counterintuitive results from Bayesian belief network software reliability model

Authors: Tero Tyrväinen

Confidentiality: Public

Report's title Counterintuitive results from Bayesian belief network software reliability model	
Customer, contact person, address VYR	Order reference SAFIR 6/2014
Project name Safety evaluation and reliability analysis of nuclear automation	Project number/Short name 85366/SARANA
Author(s) Tero Tyrväinen	Pages 16/
Keywords Bayesian belief network, reliability, software	Report identification code VTT-R-04235-14
Summary <p>Littlewood and Wright presented a Bayesian belief network model for software reliability analysis in their article The use of multilegged arguments to increase confidence in safety claims for software-based systems: A study based on a BBN analysis of an idealized example. In the model, the confidence on the software's reliability depends on testing and verification results and the prior confidence on the software specification and the "oracle" used in testing. Littlewood and Wright introduced counterintuitive results: testing or verification can reduce the confidence on the software's reliability even if no faults are found. This document provides an explanation why the model produces these counterintuitive results. The results indicate that the counterintuitive results do not completely depend on the calculation formulas and are in theory possible with more comprehensive models as well.</p>	
Confidentiality	Public
Espoo 17.9.2014 Written by  Tero Tyrväinen Research Scientist	
Reviewed by  Kim Björkman Research Scientist	
Accepted by  Riikka Virkkunen Head of Research Area	
VTT's contact address VTT, PL 1000, 02044 VTT	
Distribution (customer and VTT) SAFIR TR2	
<i>The use of the name of the VTT Technical Research Centre of Finland (VTT) in advertising or publication in part of this report is only permissible with written authorisation from the VTT Technical Research Centre of Finland.</i>	

Contents

1	Introduction	2
2	The Bayesian belief network model	4
3	Testing decreases confidence	4
3.1	General analysis using an example case	4
3.2	Assumptions related to testing	7
3.3	Parameter values	8
4	Verification decreases confidence	11
4.1	General analysis using an example case	11
4.2	Assumptions related to verification	12
4.3	Parameter values	14
5	Conclusions	15

1. Introduction

In the future, the safety automation of nuclear power plants will be software-based to a great extent. Therefore, when assessing the risk of a nuclear power plant, software failure have to be taken into account. Assessing software failures is challenging because software failures are usually not caused by random errors like failures of physical components. Instead, they are mainly caused by systematic faults originating, for example, from faulty design or mistakes in programming.

One approach that has been studied in the context of software reliability analysis is Bayesian Belief Networks (BBN) [1]. Littlewood and Wright presented a BBN model for software reliability analysis in their article The use of multilegged arguments to increase confidence in safety claims for software-based systems: A study based on a BBN analysis of an idealized example [2]. In the model, a claim that the probability that the software fails on demand (p.f.d.) is smaller than a specified value is analysed and the confidence to that claim depends on testing and verification results and the prior confidence on the software specification and the oracle used in testing. The node structure of the model is presented in Figure 1.

Littlewood and Wright introduced counterintuitive results: testing or verification can reduce the confidence on the software's reliability even if no faults are found. In the article, it is explained that testing decreases the confidence because the doubt on the oracle increases when a large number of tests is performed without finding any faults. Similarly, it is explained that a positive verification result can increase the doubt on the software specification so that the confidence on the software decreases. Littlewood and Wright

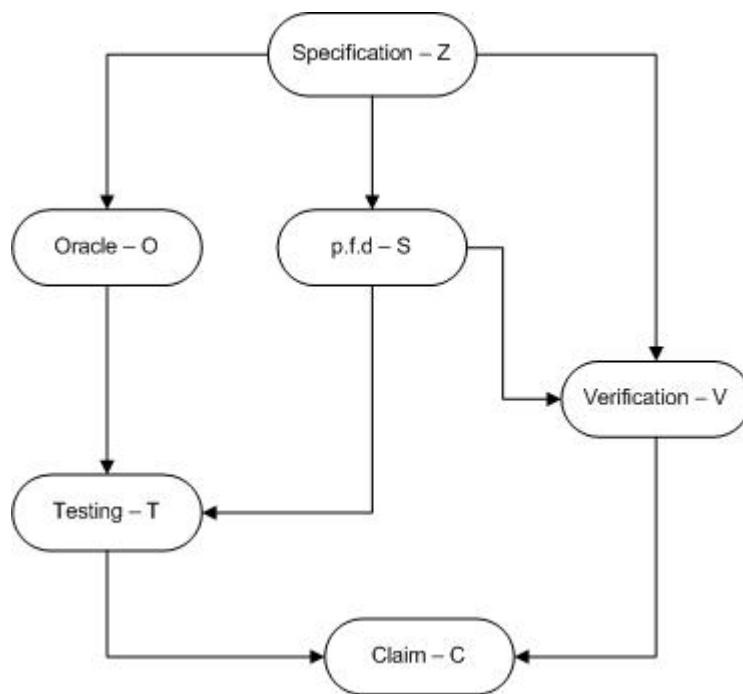


Figure 1: A BBN model for software reliability analysis.

do not know if this is only a property of the model or a generally valid result. This document explains why the model produces these counterintuitive results.

2. The Bayesian belief network model

The details of the Littlewood and Wright model can be found in [2]. This document presents the formulas in the extent that is needed to understand the analyses. The case that is analysed is how probable it is that the software's failure on demand probability is smaller than specified value s if no faults were found in testing or verification. The calculation formula that was used in the article is

$$\begin{aligned}
 P(S < s|IO) & \tag{1} \\
 = 1 - & \frac{\xi(1 - p_{0|c})[\pi_{cc}\mu' I_{1-s}(b' + n, a') + \pi_{ci}I_{1-s}(b', a')]}{(1 - \alpha)p_{0|c}\pi_{c*} + p_{0|i}\pi_{i*} + \xi(1 - p_{0|c})[\pi_{cc}\mu' + \pi_{ci}] + (1 - p_{0|i})[\pi_{ic}\mu + \pi_{ii}]} \\
 - & \frac{(1 - p_{0|i})[\pi_{ic}\mu I_{1-s}(b + n, a) + \pi_{ii}I_{1-s}(b, a)]}{(1 - \alpha)p_{0|c}\pi_{c*} + p_{0|i}\pi_{i*} + \xi(1 - p_{0|c})[\pi_{cc}\mu' + \pi_{ci}] + (1 - p_{0|i})[\pi_{ic}\mu + \pi_{ii}]}
 \end{aligned}$$

The variables and parameters of the model are presented in Table 1.

3. Testing decreases confidence

3.1. General analysis using an example case

The article presented a case where the confidence that p.f.d. is smaller than 0.001 decreased from 0.99583 to 0.66803 when the number of tests

Table 1: The variables and parameters of the model.

Notation	Meaning
S	stochastic variable representing failure on demand probability
IO	ideal observations, i.e. no faults found in testing or verification
α	probability that verification fails if $S = 0$ and spec. is correct
ξ	probability that software passes verification if $S > 0$ and spec. is correct
$p_{0 c}$	parameter of failure on demand probability distribution if spec. is correct
$p_{0 i}$	parameter of failure on demand probability distribution if spec. is incorrect
π_{c*}	prior probability that spec. is correct
π_{i*}	prior probability that spec. is incorrect
π_{cc}	prior probability that spec. and oracle are correct
π_{ci}	prior probability that spec. is correct and oracle is incorrect
π_{ic}	prior probability that spec. is incorrect and oracle is correct
π_{ii}	prior probability that spec. and oracle are incorrect
μ	$\frac{\beta(a,b+n)}{\beta(a,b)}$
μ'	$\frac{\beta(a',b'+n)}{\beta(a',b')}$
β	beta function
a	parameter of failure on demand probability distribution if spec. is incorrect
b	parameter of failure on demand probability distribution if spec. is incorrect
a'	parameter of failure on demand probability distribution if spec. is correct
b'	parameter of failure on demand probability distribution if spec. is correct
n	the number of tests
$I_{1-s}(b, a)$	regularized incomplete beta function
spec.	specification

was increased from 0 to 17,921. The parameter values were $s = 0.001$, $a = 2.58276$, $b = 4.77020$, $a' = 16.68483$, $b' = 41,133.7$, $p_{0|i} = 2.00200 \cdot 10^{-3}$, $p_{0|c} = 4.21724 \cdot 10^{-3}$, $\pi_{cc} = 0.994192$, $\pi_{ci} = 1.63910 \cdot 10^{-3}$, $\pi_{ic} = 7.81537 \cdot 10^{-5}$, $\pi_{ii} = 4.09042 \cdot 10^{-3}$, $\alpha = 0$ and $\xi = 1$. With these parameters, the dominating term of the numerator of (1) is $(1 - p_{0|i})\pi_{ii}I_{1-s}(b, a)$. It has value 0.0041 while the values of other terms are below 0.0001 in the prior case and much smaller in the posterior case. Term $(1 - p_{0|i})\pi_{ii}I_{1-s}(b, a)$ does not depend on the number of tests. Hence, the numerator does not change much when the number of tests is increased. In the prior case, the dominating term of the denominator is $(1 - p_{0|c})\pi_{cc}\mu'$. It has value 0.9900. The sum of other terms is approximately 0.01. When the number of tests is increased from 0 to 17,921, the value term $(1 - p_{0|c})\pi_{cc}\mu'$ decreases to 0.0024. Because of this, the value of the denominator decreases significantly and as the numerator value remains approximately the same, confidence that p.f.d. is smaller than 0.001 decreases.

Term $(1 - p_{0|i})\pi_{ii}I_{1-s}(b, a)$ represents the probability that the failure on demand probability is over s and that no faults are found in testing when the specification and oracle are incorrect. The value of the term does not depend on the number of tests because it is assumed that no faults are found when the oracle is incorrect regardless of the number of tests. Term $(1 - p_{0|c})\pi_{cc}\mu'$ represents the probability that no faults are found in testing when the specification and oracle are correct. The value of the term decreases when the number of tests decreases because it is assumed that no faults are found in testing with probability $(1 - s)^n$ if the oracle is correct ($(1 - s)^n$ is included in μ').

In the example case, there are two main factors that cause the confidence to decrease when the testing is added. First, parameter values are such that term $(1 - p_{0|i})\pi_{ii}I_{1-s}(b, a)$ dominates the numerator and term $(1 - p_{0|c})\pi_{cc}\mu'$ dominates the denominator in the prior case. Second, the probability that no faults are found in tests is assumed to be much smaller if the oracle is correct than if the oracle is incorrect.

3.2. Assumptions related to testing

Littlewood and Wright stated in their article that the conservative assumption that no faults are found when the oracle is incorrect has a large role in these counterintuitive results but it is still not essential to them. It seems quite evident that the conservative assumption is not realistic in most cases. However, it is a difficult problem to determine a better formula. In this study, alternative formula

$$P(T = \text{no failures} \mid S = s, O = \text{incorrect}) = g + (1 - g) \times (1 - s)^{\frac{n}{e}}, \quad (2)$$

where $0 \leq g \leq 1$ and $e \geq 1$, is used to demonstrate how phenomena of the above example becomes rarer when the assumption is less conservative. This document does not speak out which is the most realistic formula. In formula (2), no faults are found with probability g even if the number of tests reaches infinity. In the case of correct oracle, it is assumed that a fault is found in a test with probability s . In the case of formula (2), a fault can be found in a test but the probability for that is smaller than s if $g > 0$ or $e > 1$.

Table 2 and Figure 2 present how confidence $P(S < 0.001|IO)$ depends

Table 2: Confidence with alternative formulas

g	e	$P(T = \text{no failures} \mid S = 0.001, O = \text{incorrect})$	$P(S < 0.001 \mid IO)$
0.9	2	0.900	0.687
0.9	10	0.917	0.689
0.7	2	0.700	0.730
0.7	10	0.750	0.736
0.5	2	0.500	0.785
0.5	10	0.583	0.793
0.3	2	0.300	0.853
0.3	10	0.417	0.862
0.1	2	0.100	0.943
0.1	10	0.250	0.948
0	2	0.000128	1.00
0	10	0.167	1.00

on parameters g and e in the example case. In this case, the probability that faults are found in testing when the oracle is incorrect has to be very high so that the confidence increases due to 17,921 tests. It should be noted that confidence $P(S < 0.001 \mid IO)$ does not only depend on $P(T = \text{no failures} \mid S = 0.001, O = \text{incorrect})$ but also the whole distribution of $P(T = \text{no failures} \mid S = s, O = \text{incorrect})$. Because of this, confidence is higher when $g = 0$ and $e = 10$ than when $g = 0.1$ and $e = 2$ even though probability $P(T = \text{no failures} \mid S = 0.001, O = \text{incorrect})$ is larger. It can be seen that parameter g has more significant role in increasing the confidence than e .

3.3. Parameter values

Parameter values determine which terms dominate the numerator and denominator in (1). The prior confidence on the specification and the oracle is

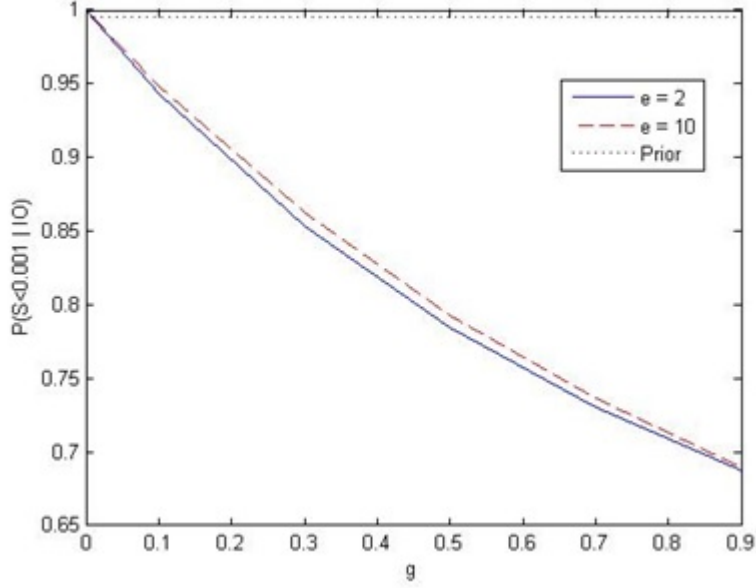


Figure 2: Confidence as the function of g .

very large which causes term $(1 - p_{0|c})\pi_{cc}\mu'$ to dominate the denominator. By decreasing π_{cc} , the prior confidence on the software $P(S < 0.001)$ decreases as can be seen in Table 3, except if only π_{ci} is increased because the prior confidence on the specification does not change. Also, when only π_{ci} and π_{ic} are increased, the posterior confidence increases.

Parameters a' and b' affect $\mu' = \frac{\beta(a', b' + n)}{\beta(a', b')}$ which is the factor that decreases $(1 - p_{0|c})\pi_{cc}\mu'$ from 0.9900 to 0.0024 when n increases. Table 4 shows how varying the parameters affects the prior and posterior confidence. Decreasing a' increases the posterior confidence. Decreasing b' decreases the prior confidence significantly. However, the effects are smaller when both parameters are decreased at the same time.

Table 3: Results when the prior confidence to the specification and oracle is varied

π_{cc}	π_{ci}	π_{ic}	π_{ii}	$P(S < 0.001)$	$P(S < 0.001 IO)$
0.8942	0.04164	0.0100782	0.05409	0.9360	0.4690
0.4942	0.20164	0.0500782	0.25409	0.6964	0.4476
0.0442	0.38164	0.0950782	0.47909	0.4270	0.4448
0.0142	0.98164	0.0000782	0.00409	0.9958	0.9959
0.4942	0.00164	0.5000782	0.00409	0.4968	0.5916
0.8942	0.05164	0.0500782	0.00409	0.9459	0.9339
0.7942	0.10164	0.1000782	0.00409	0.8960	0.9633
0.4942	0.25164	0.2550782	0.00409	0.7463	0.9843

Table 4: Results when parameters a' and b' are varied

a'	b'	$P(S < 0.001)$	$P(S < 0.001 IO)$
10	41, 133.7	0.9958	0.8883
5	41, 133.7	0.9958	0.9763
0.05	41, 133.7	0.9958	0.9958
16.68483	20, 000	0.7992	0.5570
16.68483	10, 000	0.0370	0.4295
16.68483	1, 000	0.0042	0.4241
10	20, 000	0.9909	0.6466
5	10, 000	0.9670	0.7380

4. Verification decreases confidence

4.1. General analysis using an example case

The article presented a case where the confidence that p.f.d. is smaller than 0.001 decreased from 0.99972 to 0.77064 when the software was verified to be correct. This verification is taken into account in (1) so that α changes from 0 to 0.3950 and ξ changes from 1 to $1.2006 \cdot 10^{-4}$. The parameter values were $s = 0.001$, $a = 1.2742$, $b = 0.2106$, $a' = 3.2095$, $b' = 27,095$, $p_{0|i} = 1.5547 \cdot 10^{-4}$, $p_{0|c} = 1.3812 \cdot 10^{-3}$, $\pi_{c*} = 0.9997156$, $\pi_{i*} = 2.844 \cdot 10^{-4}$ and $n = 0$. With these parameters, the dominating term of the numerator of (1) is $(1 - p_{0|i})[\pi_{ic}\mu I_{1-s}(b+n, a) + \pi_{ii}I_{1-s}(b, a)] = (1 - p_{0|i})\pi_{i*}I_{1-s}(b, a)$. It has value $2.8435 \cdot 10^{-4}$ while the other part has prior value $1.0940 \cdot 10^{-9}$ and much smaller posterior value. Term $(1 - p_{0|i})\pi_{i*}I_{1-s}(b, a)$ does not depend on the verification. Hence, the numerator does not change much due to verification. In the prior case, the dominating term of the denominator is $\xi(1 - p_{0|c})[\pi_{cc}\mu' + \pi_{ci}] = \xi(1 - p_{0|c})\pi_{c*}$. It has value 0.9983. The sum of other terms is approximately 0.0017. When the positive verification is performed, the value of term $\xi(1 - p_{0|c})\pi_{c*}$ decreases to $1.1986 \cdot 10^{-4}$. Because of this, the value of the denominator decreases significantly and as the numerator value remains approximately the same, confidence that p.f.d. is smaller than 0.001 decreases.

Term $(1 - p_{0|i})\pi_{i*}I_{1-s}(b, a)$ represents the probability that the failure on demand probability is over s and that no faults are found in verification when the specification is incorrect. The value of the term does not depend on whether the verification is performed because it is assumed that verification

result is always positive if the specification is incorrect. Term $\xi(1 - p_{0|c})\pi_{c^*}$ represents the probability that no faults are found in verification when the specification is correct. The value of the term decreases due to verification because it is assumed that no faults are found in verification with probability ξ if the specification is correct (in the prior case, $\xi = 1$ because no verification has been performed).

Again, the confidence decreases when the verification is added due to two factors. First, parameter values are such that term $(1 - p_{0|i})\pi_{i^*}I_{1-s}(b, a)$ dominates the numerator and term $\xi(1 - p_{0|c})\pi_{c^*}$ dominates the denominator in the prior case. Second, the probability that no faults are found in verification is assumed to be much smaller if the specification is correct than if the specification is incorrect.

4.2. Assumptions related to verification

Littlewood and Wright wrote in their article that the conservative assumption that no faults are found in verification when the specification is incorrect could be replaced by a value obtained from experts' judgement. This would simply be a probability that no faults are found in verification when the specification is incorrect and $S > 0$. Let this probability be denoted by γ . The case where $S = 0$ could be handled similarly. Table 5 and Figure 3 present how the confidence to the software depends on γ . In this example, the probability that faults are found in verification when the specification is incorrect has to be very large so that the confidence increases.

Table 5: Results when parameter γ is varied

γ	$P(S < 0.001 IO)$
0.8	0.8077
0.6	0.8485
0.4	0.8936
0.2	0.9438
0.0005	0.9999

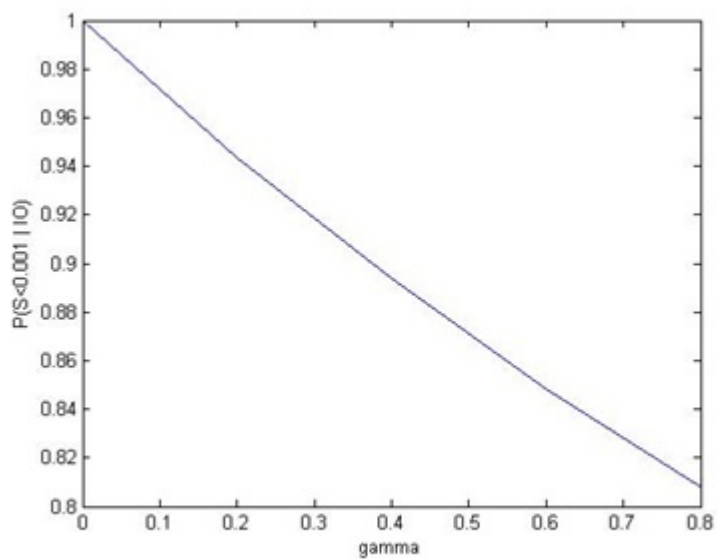


Figure 3: Confidence as the function of γ .

Table 6: Results when parameters a' and b' are varied

a'	b'	$I_{1-s}(b', a')$	$P(S < 0.001)$	$P(S < 0.001 IO)$
10	27,095	0.0000535	0.9997	0.7706
50	27,095	0.9999	0.0014	0.6739
3.2095	10,000	0.0037	0.9960	0.7703
3.2095	1,000	0.9393	0.0620	0.6798

4.3. Parameter values

Term $(1 - p_{0|i})\pi_{i*}I_{1-s}(b, a)$ dominates the numerator because $I_{1-s}(b', a')$ is very small due to the values of parameters a' and b' . Table 6 shows that the prior confidence decreases significantly if a' is increased enough or b' is decreased enough. This example shows that $I_{1-s}(b', a')$ is very sensitive to the parameter values. With the original values, probability S is very likely to be less than 0.001 if the specification is correct and very likely to be larger than 0.001 if the specification is incorrect. If probability S is also likely to be larger than 0.001 if the specification is correct, the prior confidence is very low and the posterior confidence is significantly higher than the prior confidence.

Term $\xi(1 - p_{0|c})\pi_{c*}$ dominates the denominator because parameter π_{c*} is very large compared to π_{i*} . Reducing the prior confidence to the specification does however not make posterior confidence $P(S < 0.001|IO)$ to be larger than prior confidence $P(S < 0.001)$ in this case. It just decreases both the prior and posterior confidence. Also, increasing parameter ξ does increase the posterior confidence, but the posterior confidence only approached the prior confidence asymptotically. The prior confidence is higher than the posterior confidence for all ξ values that are smaller than 1.

5. Conclusions

The calculation formula for confidence $P(S < s|IO)$ is, in more general form,

$$P(S < s|IO) \tag{3}$$

$$= 1 - \frac{P_{cc}\pi_{cc} + P_{ci}\pi_{ci} + P_{ic}\pi_{ic} + P_{ii}\pi_{ii}}{Q_{0c}P_{0|c}\pi_{c*} + Q_{0i}P_{0|i}\pi_{i*} + Q_{cc}\pi_{cc} + Q_{ci}\pi_{ci} + Q_{ic}\pi_{ic} + Q_{ii}\pi_{ii}},$$

where P_{xy} is the probability that $S > s$ and that no faults are found in verification or testing if the specification is correct/incorrect ($x = c/x = i$) and the oracle is correct/incorrect ($y = c/y = i$), Q_{0y} is the probability that no faults are found in verification or testing if $S = 0$ and the specification is correct/incorrect ($y = c/y = i$), $P_{0|y}$ is the probability that $S = 0$ if the specification is correct/incorrect ($y = c/y = i$) and Q_{xy} is the probability that no faults are found in verification or testing if $S > 0$ and the specification is correct/incorrect ($x = c/x = i$) and the oracle is correct/incorrect ($y = c/y = i$).

In formula (3), at least P_{cc} , P_{ic} , Q_{cc} and Q_{ic} decrease due to additional testing. If incorrect oracle can detect some failures, also P_{ci} , P_{ii} , Q_{ci} and Q_{ii} decrease due to additional testing but less than P_{cc} , P_{ic} , Q_{cc} and Q_{ic} . Depending on which terms are dominant before testing, either the numerator or denominator can decrease more. If the denominator decreases more, the confidence to the software decreases. This can happen with calculation formulas that differ from [2]. In this light, it seems theoretically possible that testing where no faults are found could decrease the confidence to the software.

Similarly, at least P_{cc} , P_{ci} , Q_{0c} , Q_{cc} and Q_{ci} decrease due to additional

verification. If failures can be found in verification even if the specification is incorrect, also P_{ic} , P_{ii} , Q_{0i} , Q_{ic} and Q_{ii} decrease due to verification but less than P_{cc} , P_{ci} , Q_{0c} , Q_{cc} and Q_{ci} . Again, it is possible that the denominator decreases more due to this and the confidence to the software decreases. Hence, it seems theoretically possible that additional verification where no faults are found could decrease the confidence to the software.

References

- [1] Roventa E, Spircu T. Bayesian (belief) networks. In: Roventa E, Spircu T. Management of knowledge imperfection in building imperfect systems. Berlin: Springer; 2009. p. 133-52. ISBN 3540774629.
- [2] Littlewood B, Wright D. The use of multilegged arguments to increase confidence in safety claims for software-based systems: A study based on a BBN analysis of an idealized example. IEEE Transactions on Software Reliability. 2007; 33:347-65.