Journal of Universal Computer Science, vol. 5, no. 10 (1999), 633-643
submitted: 21/4/99, accepted: 29/9/99, appeared: 28/10/99 © Springer Pub. Co.

Advanced Fault Tree Modeling

Winfrid G. Schneeweiss
(Computer Engg., Fern University, D - 58084 Germany
winfrid.schneeweiss@fernuni-hagen.de)

Abstract: Fault trees show which joint components' faults mean system faults. Fault trees can often be used to determine dependability parameters of systems. Here it is shown that i) binary decision diagrams (BDDs) can also be used to calculate system mean failure frequency, ii) modeling dynamics of fault trees does not always mean Markov modeling, iii) a deeper understanding of interrelations between s-dependent components is supported, rather, by Petri nets than by state transition graphs.

Key Words: Fault tree, Shannon tree, Binary decision diagram, Markov approach, Petri net

Category: F.2.2, G.3

1 Introduction

To this author's surprise quantitative dependability investigations are often "forgotten" in the context of fault tolerant systems design. But any such design should be regarded as incomplete unless values of unavailability/unreliability, MTBF etc. are included.

In these days fault tree modeling reaching back to the early 60s has gained a maturity that allows for little monographs, e.g., [see Limnios (91)],[see Schneeweiss (99,1)] to be published. But fault tree analysis is not yet at its end. Advances are still possible, as will be shown here.

1.1 Motivation and overview

In recent papers on fault tree analysis, typically in [see Pullum, Bechta Dugan (96)],[see Schneeweiss (96)], one finds certain opinions which, even though correct in their main thrust, suffer from certain restrictions some of which are going to be discussed here.

The first point concerns the emphatically appraised binary decision diagrams [see Coudert, Madre (93)],[see Bouissou (96)],[see Drechsler, Becker (98)], whose usefulness for the determination of mean failure frequency is disputed in [see Schneeweiss (96)]. However, here we will show a way to overcome the major problem discussed in [see Schneeweiss (96)] by using subtrees as "condensed" leaves of a binary decision tree. Furthermore, it will be shown that even for determining system unavailability with many small size subsystems or, rather, modules a well designed Shannon decomposition tree, being a binary decision tree, is simpler than a binary decision diagram with reconverging branches. Details are given in section 2.

The second point concerns the all too general statement [see Pullum, Bechta Dugan (96)] that fault tree dynamics must be modeled by the Markov approach. It will be shown – again for smaller modules – that this is not necessarily true.

Rather, under fairly mild restrictions as to the s-dependence of diverse components' state durations, much more general results can be found; see section 3.

In connection with the second main point it will be shown (also in section 3) that often randomly timed Petri nets give a better insight into system dynamics than standard state transition graphs do.

A few general conclusions follow in section 4.

1.2 Notation and acronyms

A availability (with index i for component i)
$E(Z)$ expected value of Z
F_L, f_L probability distribution (Cdf), -density function (pdf) of L
φ Boolean fault tree (structure) function
λ failure rate, $\lambda \equiv 1/(MTTF)$
μ repair rate, $\mu \equiv 1/(MTTR)$
L life (time)
n number of components
ν mean failure frequency; $\nu \equiv 1/(MTBF)$; $(MTBF) = (MTTF) + (MTTR)$
p_i place i of a PN
$Pr\{\alpha\}$ probability of α
S index for system
U unavailability
X_i fault tree input variable; $X_i = 1$ for the faulty component or module i
$\bar{X}_i$ negated X_i; arithmetically: $\bar{X}_i = 1 - X_i$
$X_{\{i,j\}}$ indicator of the connectedness of nodes i and j
$\vee$ disjunction operator (Boolean OR)
BDD, BDT binary decision diagram, - tree
DAG directed acyclic graph
FT fault tree
PN Petri net
ST Shannon decomposition tree = BDT. It is based on the decomposition of the Boolean function φ (here that of the fault tree):

$$\varphi = X_i\, \varphi_{/X_i=1} + \bar{X}_i\, \varphi_{/X_i=0} \quad . \tag{1}$$

In pictures of the above BDDs and BDTs the above "+" (meaning arithmetic addition) is written inside of the nodes, and X_i and $\bar{X}_i$ (sometimes with cofactors extracted from $\varphi_{/X_i=1}$ and $\varphi_{/X_i=0}$, respectively) are the edge marking.

1.3 The role of fault trees in dependability modeling

The fault tree is the picture of a Boolean (switching) function describing which combinations of components' faults mean system fault.

As is obvious from Fig. 1, a system's or mission's fault tree is only the peak of an iceberg. The more involved problems of modeling concern the binary processes which are inputs of the fault tree. Hence it should be no surprise that only the case of stationary s-independent binary random processes $\{X_1(t)\}, ..., \{X_n(t)\}$ are easily modeled down to numerical results. Yet even a partial FT analysis is of some value also in the general case. (The usefulness of the FT for gaining a qualitative insight into the redundancy structure of as system is obvious.)

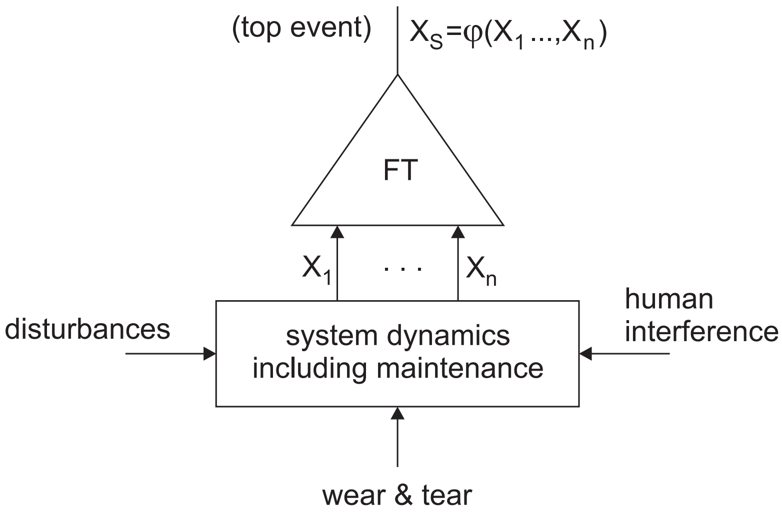

Fig. 1: The fault tree as the simpler part of a dependability model.

2 Limitations of BDDs partially renounced

There is little doubt that, by and large, BDDs are superior to BDTs [see Drechsler, Becker (98)], i.e., trees of the recursive complete Shannon decomposition (short: Shannon trees, STs [see Pullum, Bechta Dugan (96)]). The obvious reason is that certain subtrees have to be decomposed only once. Yet there are exceptions which should be kept in mind in the present euphoria towards BDDs [see Coudert, Madre (93)],[see Bouissou (96)]. A major limitation is concerned with difficulties when trying to calculate mean system failure frequency (or intensity). Details are discussed in [see Schneeweiss (96)], where it is pointed out that in that context BDTs are superior.

However, in addition to [see Schneeweiss (96)] I should like to point out that the criticsm expressed there was a bit too radical. The main point of BDDs as compared to BDTs is not the "loss" of the property of being a tree, but rather the fact that multiple subtrees need to be evaluated only once. In fact, doing the latter, but staying with a BDT with equal subtrees condensed like modules [see Schneeweiss (93)] to single new – even though not independent – variables, drawn as leaves in the condensed tree, retains the more important advantage of BDDs. This is illustrated by the following example.

Example 1: Proper use of a BDD for calculating system mean failure frequency

Have a look at Fig. 2, where the subtree (ψ) is not a module [see Schneeweiss (93)], since it depends also on X_k.

With U_ψ, λ_ψ, and μ_ψ as unavailability, failure rate and repair rate of the subsystem with the FT function ψ, respectively, system unavailability is (by Fig. 2b):

$$U_S = U_i(U_jA_k + A_jU_\psi) + A_iU_jU_\psi \ , \ U_\psi = E(\psi) \ . \tag{2}$$

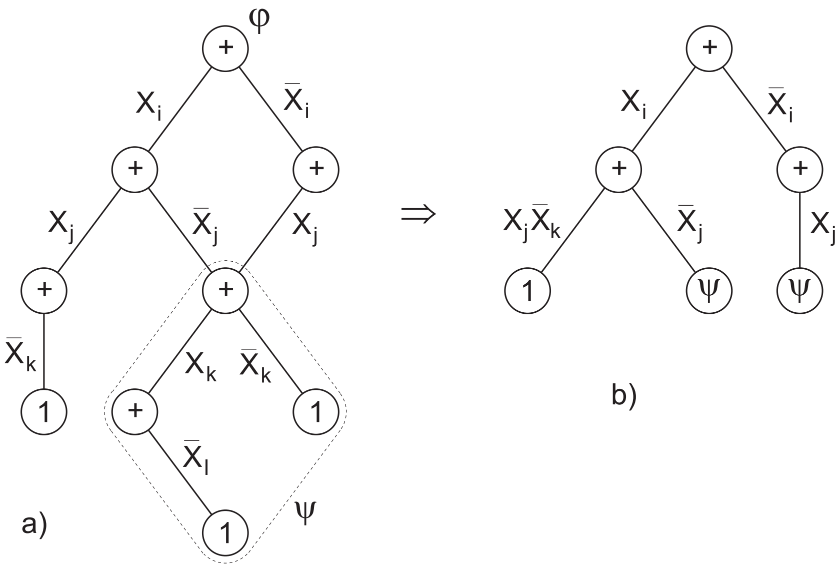

Fig. 2: BDD transformed to a BDT with equal pseudo-leaves of "value" ψ.

This follows readily from the fault tree's arithmetical form of fig. 2b:

$$X_S = \varphi = X_i(X_j\bar{X}_k + \bar{X}_jX_\psi) + \bar{X}_iX_jX_\psi \ ;$$

on taking expectations. (By definition, $U = Pr\{X = 1\} = E(X)$, [see Henley, Kumamoto (92)].)

Furthermore, system mean failure frequency is [see Schneeweiss (99,1)],[see Schneeweiss (93)]

$$\begin{aligned}\nu_S = U_i[U_jA_k(\mu_i + \mu_j - \lambda_k) + A_jU_\psi(\mu_i - \lambda_j + \mu_\psi)]\\ +A_iU_jU_\psi(-\lambda_i + \mu_j + \mu_\psi) \ ,\end{aligned} \tag{3}$$

where

$$U_\psi = U_kA_l + A_k \ , \ \nu_\psi = U_kA_l(\mu_k - \lambda_l) + A_k(-\lambda_k) \ , \tag{4}$$

and λ_ψ and μ_ψ follow from the well known defining equations [see Schneeweiss (99,1)]

$$\nu_\psi = U_\psi\mu_\psi = A_\psi\lambda_\psi \ . \tag{5}$$

Notice that in (3) an explicit use of λ_ψ and μ_ψ can be avoided, since in (3) i) λ_ψ doesn't show up, ii) μ_ψ can be multiplied with U_ψ to yield ν_ψ of (4).

Now it is shown that also in other (small-size) cases BDTs can be better than BDDs.

This "problem" with BDDs was stimulated by an indepth study of [see Pullum, Bechta Dugan (96)]. In the context of analyzing modules of FTs, i.e., sub-FTs with "private" sets of variables [see Pullum, Bechta Dugan (96)],[see Schneeweiss (93)] the sub-FTs are often so small that in the ST development single variables or groups of them can be factored out in order to abbreviate the decomposition substantially. In [see Pullum, Bechta Dugan (96)] the following example is discussed.

Example 2: (See Fig. 2 of [see Pullum, Bechta Dugan (96)])
Let a fault tree's Boolean function be

$$X_S = (X_1 \vee X_2 X_5)(X_3 \vee X_4 X_5) \ . \tag{6}$$

(In [see Pullum, Bechta Dugan (96)]

$X_1 \equiv A_1, X_2 \equiv A_2, X_3 \equiv B_1, X_4 \equiv B_2, X_5 \equiv A_B$.)

The sequential, i.e., repeated Shannon decomposition of (6) with respect to X_1, X_3, etc. results in

$$X_S = X_1(X_3 + \bar{X}_3 X_4 X_5) + \bar{X}_1(X_3 X_2 X_5 + \bar{X}_3 X_2 X_4 X_5) \ ; \tag{7}$$

see the syntax diagram, i.e., the BDT of Fig. 3 and the BDD of Fig. 4.

The algebraic form corresponding to Fig. 3 is found by starting each path bottom up with the value 1 of its leaf and by multiplying intermediate values by the edge marking. (If an inner node has only one input edge, then there is a trivial addition of the type $A + 0 = A$.)

In this context it is a bit awkward to expand $X_i X_j X_k \cdots$ according to (1) as

$$X_i X_j X_k \cdots = X_i(X_j X_k \cdots) + \bar{X}_i 0 = X_i(X_j X_k \cdots)$$

which is trivial.

However, extracting common factors, an ST of smaller depth (height) is readily found. Instead of (7), (6) yields as a decomposition with respect to X_1:

$$X_S = X_1(X_3 \vee X_4 X_5) + \bar{X}_1 X_2 X_5 (X_3 \vee X_4 X_5) \ . \tag{8}$$

Using

$$X_5(X_3 \vee X_4 X_5) = X_5(X_3 \vee X_4) \ , \tag{9}$$

a single further decomposition gives the final result; see Fig. 5:

$$X_S = X_1(X_3 + \bar{X}_3 X_4 X_5) + \bar{X}_1 X_2 X_5 (X_3 + \bar{X}_3 X_4) \ . \tag{10}$$

This ST is simpler (smaller in size and of a more regular form) than the BDD of Fig. 4.

Similar results are obtained for other small-scale examples. One such example is the following.

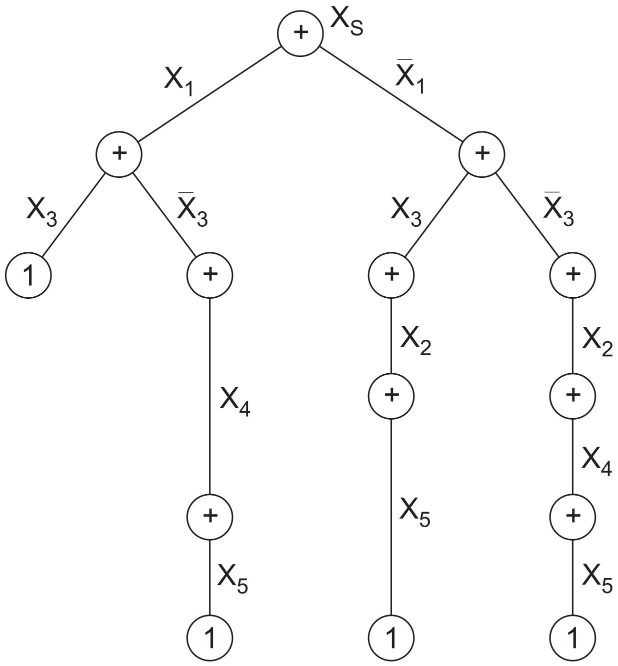

Fig. 3: ST of (6).

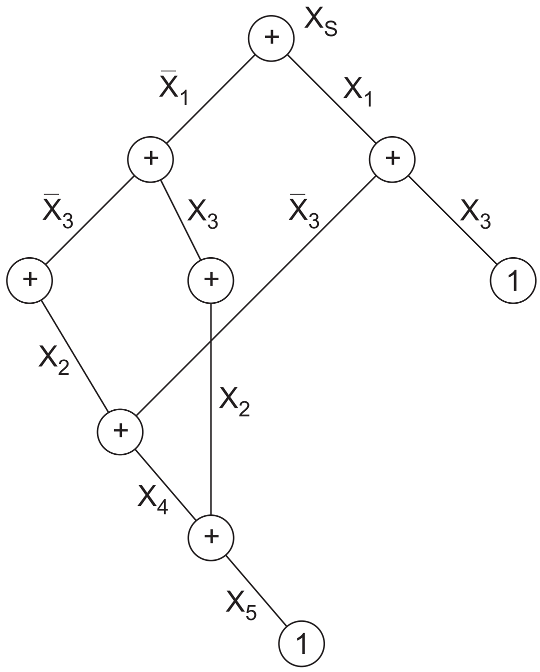

Fig. 4: BDD found from Fig. 3.

Example 3: Simplified ARPA net
Fig. 6 shows the ubiquitous strongly simplified ARPA net, where there should be a path from the source (node 1) to the target (node 4) via fallible links (and ideal nodes). From the 6 obvious mincuts we find the FT function [see Schneeweiss (99)]

$$X_{\{1,4\}} = X_1X_6 \vee X_2X_4X_6 \vee X_3X_4X_5X_6$$
$$\vee X_1X_4X_5X_7 \vee X_2X_5X_7 \vee X_3X_7 \ . \tag{11}$$

This function shows qualitatively the degree of fault tolerance of the network: A minimum of 2 edges (communication links) have to be down for the communication from source to target to fail.

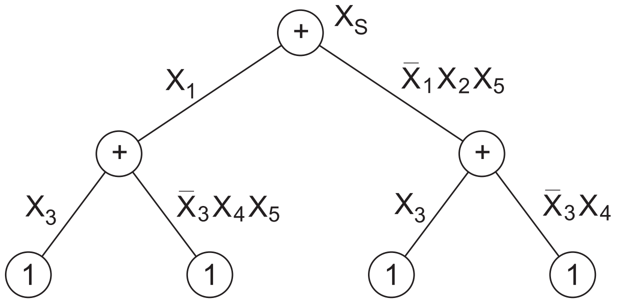

Fig. 5: An extremely small ST for (6).

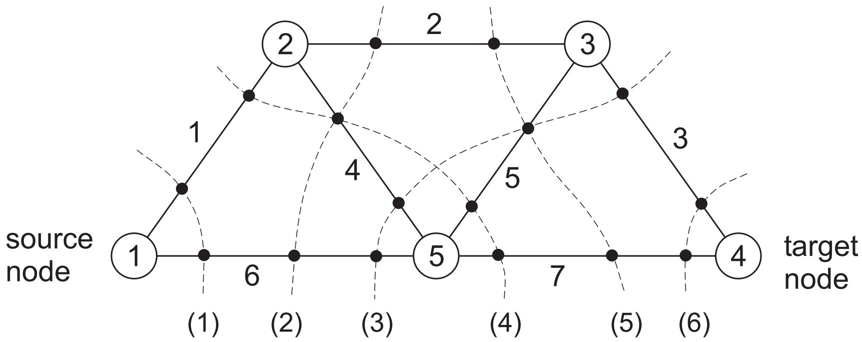

Fig. 6: An s,t-problem in the simplified ARPA net; $s = 1, t = 4$.

The ST, i.e., a BDT rather than a general BDD with common subtrees, is found via the simple heuristic "shortest terms first and in them most frequently appearing variables first" as follows. Even in the first expansion step, i.e., in

$$X_{\{1,4\}} = X_6(X_1 \vee X_2X_4 \vee X_3X_4X_5 \vee X_2X_5X_7 \vee X_3X_7)$$
$$+\bar{X}_6X_7(X_1X_4X_5 \vee X_2X_5 \vee X_3) \ , \tag{12}$$

the extraction of a common factor, is possible; see the last addend of (12). Furthermore, expanding the first addend of (12) sequentially with respect to X_1, X_3, X_4; the algebraic form of the ST is

$$X_{\{1,4\}} = X_6(X_1 + \bar{X}_1\{X_2[X_4 + \bar{X}_4 X_7(X_3 + \bar{X}_3 X_5)] + \bar{X}_2 X_3 \cdot (X_7 + \bar{X}_7 X_4 X_5)\}) + \bar{X}_6 X_7[X_3 + \bar{X}_3 X_5(X_2 + \bar{X}_2 X_1 X_4)] \ , \quad (13)$$

and it would make little sense to construct a BDD.

It is easy to check the correctness of (12) using the following special form of (1).

$$\varphi = X_i \vee \varphi'(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n) = X_i + \bar{X}_i \varphi' \ . \quad (14)$$

Again "system unavailability", i.e., here the probability of not finding a viable path from source to target, is found via (13) on replacing X by U, and $\bar{X}$ by $\bar{U} = 1 - U = A$ for any index.

The determination of system mean failure frequency (intensity) is also possible directly from (13) without further calculations; see [see Schneeweiss (99,1)] for details.

3 Solving selected dynamic fault tree problems at a generality far beyond that of conventional Markov modeling

In [see Pullum, Bechta Dugan (96)] the first paragraph of subsection 2.1 ends with the following surprising statement. "Dynamic fault trees can not be solved with standard combinatorial approaches, but rather depend on Markov solution techniques."

The first thing that comes to one's mind in this context is the trivial FT of the 1-out-of-2:G system (with cold standby of component 2 for component 1): $X_S = X_1 X_2$. The general solution for the case of no repairs is well known [see Henley, Kumamoto (92)] to be possible by convolution, i.e., the system life's Cdf (or strict-sense unreliability)

$$F_{L_S}(t) = f_{L_1} \otimes F_{L_2}(t) \equiv \int_0^t f_{L_1}(\tau) F_{L_2}(t-\tau) d\tau \ . \quad (15)$$

Why think of Markov modeling here?

Furthermore, the case of a fallible switch for activating the spare, which is often modeled via the concept of coverage [see Sahner, Trivedi, Puliafito (96)], i.e., the probability of recovering from the first fault, is easily modeled (see (18),(19)) by

$$F_{L_S}(t) = \int_0^t f_{L_1}(\tau) F_{L_2}(t-\tau) \bar{F}_{L_3}(\tau)\, d\tau \ , \ L_3 \equiv \text{life of the switch} \ , \quad (16)$$

where $\bar{F}_{L_3}(\tau)$ is the probability that the switch is o.k. when the first component fails (at $L_1 = \tau$).

Also the slightly more involved (sub-)system of Fig. 1 of [see Pullum, Bechta Dugan (96)] poses no real problem as to a general non-Markov-type solution. From Fig. 1 of [see Pullum, Bechta Dugan (96)] we have extracted Fig. 7, where P_i and S_i mean changes of system state due to failure of processor i and (cold) spare i, respectively of a system consisting of a pair of processors with a dedicated spare each. (The unorthodox notation P_i and S_i is that of [see Pullum, Bechta Dugan (96)].)

The DAG of Fig. 7 contains 3 paths corresponding to 3 (disjoint) random events. It is a peculiarity of the system that S_2 cannot fail prior to a failure of P_1 and that the case of S_1 failing last is of no concern. (For details see [see Pullum, Bechta Dugan (96)].)

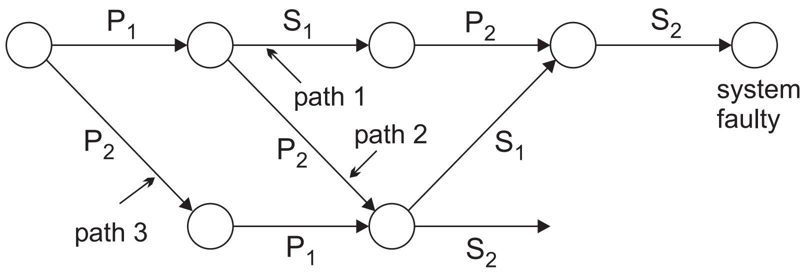

Fig.7: State diagram of a special 4 components system. Movements along edges marked by $P_i(S_i)$ occur, once processor i (spare i) fails.

For s-independent lives of all 4 components the 3 paths of Fig. 7 result in 3 non too complex 3-fold integrals whose sum is system unreliability:

$$\begin{aligned} F_{L_S}(t) &= \int_0^t f_{L_{P1}}(\tau) \int_0^{t-\tau} f_{L_{S1}}(\tau') \int_{\tau+\tau'}^{t} f_{L_{P2}}(\tau'') F_{L_{S2}}(t-\tau'') d\tau'' d\tau' d\tau \\ &+ \int_0^t f_{L_{P1}}(\tau) \int_\tau^t f_{L_{P2}}(\tau') \int_{\tau'-\tau}^{t-\tau} f_{L_{S1}}(\tau'') F_{L_{S2}}(t-\tau'-\tau'') d\tau'' d\tau' d\tau \\ &+ \int_0^t f_{L_{P2}}(\tau) \int_\tau^t f_{L_{P1}}(\tau') \int_0^{t-\tau'} f_{L_{S1}}(\tau'') F_{L_{S2}}(t-\tau-\tau'') d\tau'' d\tau' d\tau\,. \end{aligned} \tag{17}$$

(This lengthy formula follows directly from Fig. 7 by inspection based on (15).)

A high-precision Markov modeling of this system would be quite complex [see Sahner, Trivedi, Puliafito (96)].

Since the state diagram's modeling power for system dynamics is, usually, far inferior to that of a Petri net graph [see Schneeweiss (99,2)], I have depicted system dynamics for the above example via the timed PN of Fig. 8, where the initial marking is given. The right hand part of Fig. 8 contains a priority AND, "working" only when place 1 is marked prior to place 2.

From Fig. 8, rather than from Fig. 7, there follows that the system will fail, if place 1, p_1, should be marked prior to p_2. So, given a token is reaching p_1 at

$\tau < t$, the system will be down at t, if p_2 receives a token between τ and t. This can be formulated according to the well known law of total probability, viz.

$$Pr\{a\} = \sum_{i=1}^{m} Pr\{a|b_i\}Pr\{b_i\} \;\; ; \;\; a \subseteq \bigcup_{i=1}^{m} b_i \;\; ; \;\; b_j \cap b_k = \emptyset \tag{18}$$

in an infinitesimal form, viz.

$$Pr\{a(t)\} = \int_0^t Pr\{a(t)|\tau < L \le \tau + d\tau\} f_L(\tau)\, d\tau \tag{19}$$

where $a(t)$ is the random event connected with the parameter (time) t.

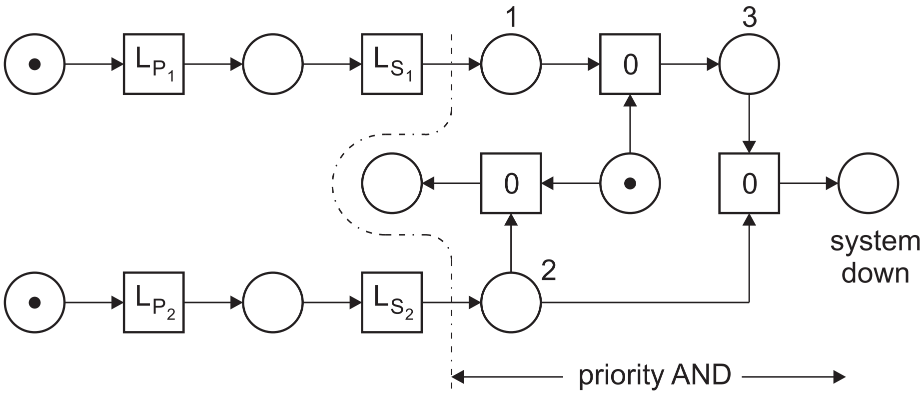

Fig. 8: PN model for (sub-)system with the state diagram of Fig. 7.

The specific result is here system (strict-sense) unreliability:

$$F_{L_S}(t) = \int_0^t f_{L_{P1}+L_{S1}}(\tau)\bar{F}_{L_{P2}+L_{S2}}(\tau)\, d\tau \;\; , \tag{20}$$

where the convolution

$$f_{L_{Pi}+L_{Si}}(\tau) = \int_0^\tau f_{L_{Pi}}(\tau')f_{L_{Si}}(\tau-\tau')d\tau' \tag{21}$$

is the pdf of $L_{Pi} + L_{Si}$ and the convolution

$$F_{L_{Pi}+L_{Si}}(\tau) = \int_0^\tau f_{L_{Pi}}(\tau')F_{L_{Si}}(\tau-\tau')d\tau' \tag{22}$$

is the Cdf of $L_{Pi} + L_{Si}; i = 1, 2$.

Note the compactness and obvious plausibility of (20) (including (21) & (22)) as compared to (17).

4 Conclusions

Based on several conclusive examples, it could be shown that certain modern dependency-related paradigms like BDD- or Markov model-based approaches should be regarded with due skepticism. Sometimes, though not for really large systems, binary decision trees are better than BDDs, and Petri nets are a better basis for rather compact general results (for non-repairable systems) than the state graphs and the differential equations of typical Markov modeling.

Hence the selection of the most appropriate method of dependability analysis is not a trivial task. If possible, more than a single solution method should be tried in order to find possible errors (of analysis) and to gain a deeper understanding of the problem at hand.

References

[Limnios (91)] N. Limnios: "Arbres de Dèfaillance". Hermes, Paris (1991).

[Schneeweiss (99,1)] W. Schneeweiss: "The Fault Tree Method". LiLoLe, Hagen (1999).

[Sahner, Trivedi, Puliafito (96)] R. Sahner, K. Trivedi, A. Puliafito: "Performance and Reliability Analysis of Computer Systems". Kluwer, Dordrecht (1996).

[Pullum, Bechta Dugan (96)] L. Pullum, J. Bechta Dugan: "Fault tree models for the analysis of complex computer-based systems". Proc. Ann. Reliab. & Maintainab. Symp. IEEE (1996), 200-207.

[Coudert, Madre (93)] O. Coudert, J. Madre: "Fault tree analysis: 10^{20} prime implicants and beyond". Proc. Ann. Reliab. & Maintainab. Symp. IEEE (1993), 240-245.

[Bouissou (96)] M. Bouissou: "An ordering heuristic for building binary decision diagrams from fault-trees". Proc. Ann. Reliab. & Maintainab. Symp. IEEE (1996), 208-214.

[Drechsler, Becker (98)] R. Drechsler, B. Becker: "Graphenbasierte Funktionsdarstellung: Boolesche und Pseudo-Boolesche Funktionen". Stuttgat: Teubner 1998.

[Schneeweiss (96)] W. Schneeweiss: "Limited usefulness of BDDs for mean failure frequency calculation". J. Automatic Contr. Production Syst. (1996), 1131-1136.

[Schneeweiss (93)] W. Schneeweiss: "Calculating MTBF for modularized fault trees". Proc. Ann. Reliab. & Maintainab. Symp. IEEE (1993), 206-213.

[Henley, Kumamoto (92)] E. Henley, H. Kumamoto: "Probabilistic Risk Assessment". New York: IEEE Press (1992) (reprint from 1982).

[Schneeweiss (99,2)] W. Schneeweiss: "Petri Nets for Reliability Modeling". LiLoLe, Hagen (1999).