Issues and Approaches Regarding Success Terms for Probabilistic Risk Assessment Models

James Knudsen[a], Dr. Curtis Smith[a], and Michael Calley[a]
[a] Idaho National Laboratory, Idaho Falls, ID, USA, James.Knudsen@inl.gov
[a] Idaho National Laboratory, Idaho Falls, ID, USA, Curtis.Smith@inl.gov
[a] Idaho National Laboratory, Idaho Falls, ID, USA, Michael.Calley@inl.gov

Abstract: Solving event tree accident sequences in probabilistic risk assessments (PRAs) involves assumptions about the success of systems, i.e., event tree top events. The primary assumption is that system failure is a rare event; therefore, the system success probability is very close to 1.0. Under most conditions, this assumption is valid. However, when systems have higher failure probabilities, the success probability is not close to 1.0 and this primary assumption causes sequences with success branches to be overestimated. This paper presents an approach to quantify event tree accident sequences when high system failure probabilities are part of the logic. This approach employs two methods to quantify the sequence success system cut sets, which will be converted to a single recovery basic event and then multiplied back into all the cut sets within that specific sequence. This recovery adjustment will be based on the quantified success system cut sets. One approach will quantify the success system cut sets via the minimal cut set upper-bound (MinCut) approximation, and the second will quantify these cut sets using the binary decision diagram (BDD) quantification.

1 INTRODUCTION

This paper discusses the issue of overestimating event tree accident sequences when success top events are not properly accounted for. For most probabilistic risk assessment (PRA) event tree accident sequences, such overestimation is minor, due to small system failure probabilities. However, this overestimation is exacerbated when evaluating external event PRAs, in which the failure probabilities of systems are no longer small. This paper discusses the current state of practice of success term quantification for event tree accident sequences, along with proposed approaches to better estimate the success term quantification of event tree accident sequences.

Sometimes it is necessary in typical PRAs to consider both success and failure basic events within cut sets. Cut sets that contain success terms are classified as non-coherent, while those that contain just failed basic events are termed coherent. Quantification of either coherent or non-coherent cut sets typically employs approximations such as the minimal cut set upper-bound (MinCut). However, these types of quantification approximations can be inaccurate ([1], [2], [3]).

Different approaches for efficiently solving non-coherent logic models have been studied. One approach is presented in NUREG/CR-5242, "A Fast Bottom-Up Algorithm for Computing the Cut Sets of Noncoherent Fault Trees," published in October 1988. Another approach that has received attention in the PRA literature is the binary decision diagram (BDD) methodology ([4], [5], [6], [7]).

In current PRA application, non-coherent logic is encountered in practical applications such as the generation and quantification of event tree accident sequence minimal cut sets. PRA models that include top events with high failure probabilities, e.g., failures due to external events, should explicitly include the probability of system success whenever possible. However, this may be difficult because of the inclusion of random failures along with external event conditional failures. Thus, it may be difficult to properly account for success paths within the event tree accident sequences.

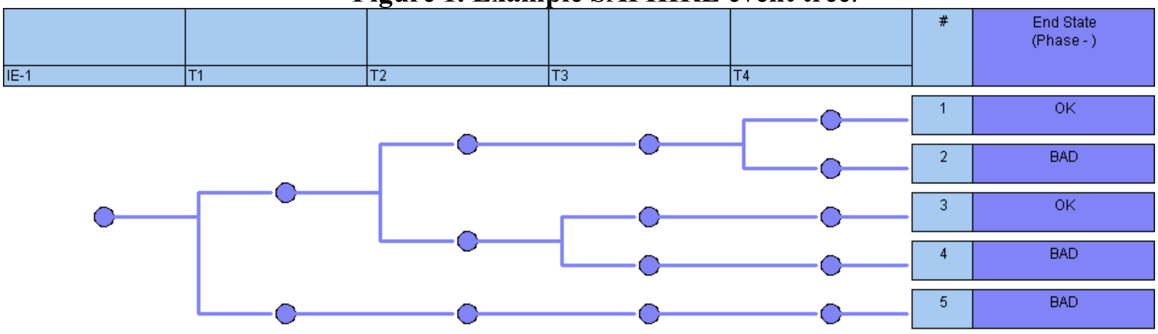
This paper assesses different approaches and the state of practice of solving and quantifying non-coherent logic models. A simple demonstration model, as well as a more realistic PRA model, are used to compare the different approaches. The conclusion pertains to a potential path forward to provide better PRA results when evaluating event tree accident sequences that contain success top events.

2 STATE OF PRACTICE

In PRAs, success terms are primarily determined through event tree accident sequences. Figure 1 shows a simple event tree illustrating the state of practice of quantification as well as the current approaches to success term quantification. Once the event tree has been developed, different software codes will generate the accident sequences. For this example, the software code Systems Analysis Programs for Hands-On Integrated Reliability Evaluations (SAPHIRE) is used. The sequence logic identifies the path from the initiating event to the stated end state, which is identified as BAD in this case. The accident sequence identifies the systems that fail and/or succeed, hence the arrival of success terms.

To generate the event tree accident sequence cut sets, two fault trees are created for each accident sequence: one based on the failed top events, and one based on the success top events. Fault tree reduction algorithms then generate the minimal cut sets for that accident sequence.

Figure 1. Example SAPHIRE event tree.



To solve the accident sequence for minimal cut sets, SAPHIRE will, by default, construct a fault tree for those systems designated as failed in the sequence logic by creating an internal AND gate that uses these systems as inputs. In Figure 1, the accident sequence logic for sequence 4 is:

IE-1 AND /T1 AND T2 AND T3

Therefore, SAPHIRE internally creates the following failed systems fault tree:

FAILED AND T2 T3
T2 TRAN
T3 TRAN

where T2 and T3 are example fault trees that illustrate the process and TRAN denotes a transfer to the corresponding system's fault tree. SAPHIRE then solves this internal FAILED fault tree using the user-specified truncation values. This process results in a list of cut sets for the failed systems in the accident sequence.

Next, SAPHIRE uses the "delete term" technique, which is the default process, to *reduce* this list of failed system cut sets to only those that can cause the accident sequence to fail. To perform the delete term technique, the system failure cut sets from the successful system fault trees (i.e., SUCCESS fault trees) are compared to the cut sets generated from the failed system fault trees (i.e., FAILED fault trees). This comparison removes any cut sets in the FAILED list that also show up in the SUCCESS list. Removal of these cut sets is important, since they cannot occur based on the specific sequence logic. The accident sequence minimal cut sets are those that remain after the failure combinations of the successful system cut set terms are deleted.

The successful system fault tree for accident sequence 4 is:

SUCCESS	**OR**	**T1**
T1		**TRAN**

where T1 represents an example fault tree.

In Section 2.1, the delete term process is discussed in relation to the evaluation of the accident sequence logic from the event tree in Figure 1.

2.1 Current Cut Set Generation and Quantification

To illustrate the current state of practice of cut set generation, the event tree in Figure 1 is used, along with the fault trees shown in Figure 2. As part of the illustration, the event tree accident sequences are solved using the default approach (i.e., delete term), along with a modification to enable inclusion of the system success logic (e.g., use of "process flags").

The delete term process is explicitly illustrated for sequence 4 by using the fault trees from Figure 2. For sequence 4, the fault trees from Figure 2 are expanded to the FAILED and SUCCESS logic listed below. Cut sets are developed from the logic resulting in the two cut set lists. The following symbols are used to represent logic operators: * = AND; + = OR.

FAILED = (T2 * T3) = [BE-C + BE-B * BE-E] * [BE-F + BE-D * BE-G]
SUCCESS = T1 = BE-A + BE-C + BE-B * BE-D

FAILED	SUCCESS
1) **BE-C** * BE-F +	1) BE-A +
2) **BE-C** * BE-D * BE-G +	2) **BE-C** +
3) BE-B * BE-E * BE-F +	3) **BE-B * BE-D**
4) **BE-B** * BE-E * **BE-D** * BE-G	

The delete term process compares the cut sets from the SUCCESS listing to those in the FAILED listing. Cut set combinations found in both the FAILED and SUCCESS lists are deleted from the FAILED list. These cut sets cannot be part of the FAILED list since they must be successful for the sequence to occur. For example, cut sets 1 and 2 are deleted from the FAILED list, due to cut set 2 being in the SUCCESS fault tree. If BE-C fails, sequence 4 cannot occur. This process is performed for all cut sets in the FAILED list. As seen, cut set 4 is also deleted. Cut set 3 is all that remains after performing the delete term process. [The cut sets highlighted in **RED** illustrate, which cut sets are removed due to the Delete Term process.]

The delete term process is the typical default method of sequence cut set generation. Software tools such as SAPHIRE can include the success logic in sequence generated cut sets. SAPHIRE uses process flags to make this logic adjustment. The "I" process flag in SAPHIRE is used to allow cut sets to be explicitly generated for SUCCESS top events.

Figure 2. Event tree top events.

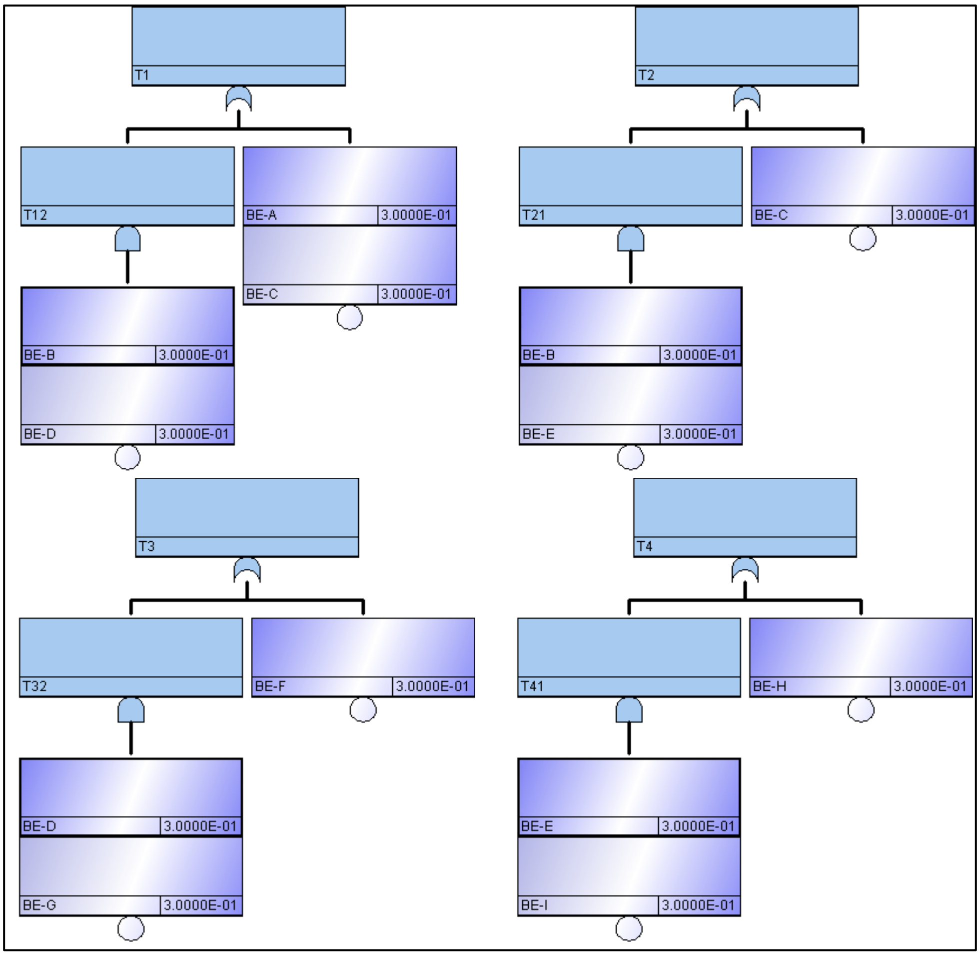
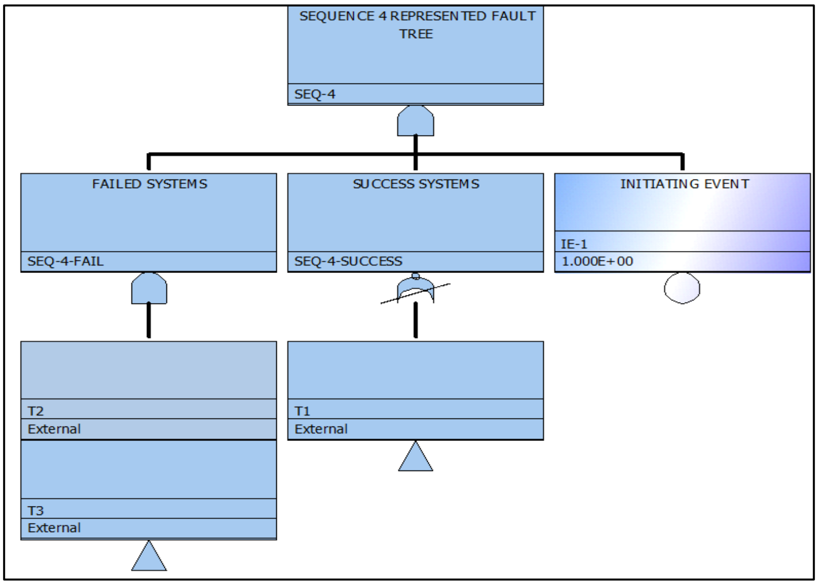


Figure 3 shows the fault tree representation of accident sequence 4 to illustrate the "I" process flag.

Figure 3. Sequence 4 fault tree representation.



Sequence 4 consists of the failure of T2 and T3 and the success of T1. The NOT OR gate in the success path converts all the gates to the opposite logic operator and complements all the inputs. The following illustrates this transformation [the "/" symbol represents success of the basic event]:

SEQ-4-SUCCESS	NOT OR T1
SEQ-4-SUCCESS	NOT OR [BE-A + BE-C + (BE-B * BE-D)]
SEQ-4-SUCCESS	[/BE-A * /BE-C * (/BE-B + /BE-D)]
SEQ-4-SUCCESS	(/BE-A * /BE-C * /BE-B) + (/BE-A * /BE-C * /BE-D)

The sequence 4 cut sets generated are:

1) IE-1 * **BE-C** * BE-F * /BE-A * **/BE-C** * /BE-B +
2) IE-1 * **BE-C** * BE-F * /BE-A * **/BE-C** * /BE-D +
3) IE-1 * **BE-C** * BE-D * BE-G * /BE-A * **/BE-C** * /BE-B +
4) IE-1 * **BE-C** * BE-D * BE-G * /BE-A * **/BE-C** * /BE-D +
5) IE-1 * **BE-B** * BE-E * BE-F * /BE-A * /BE-C * **/BE-B** +
6) IE-1 * BE-B * BE-E * BE-F * /BE-A * /BE-C * /BE-D +
7) IE-1 * **BE-B** * BE-E * BE-D * BE-G * /BE-A * /BE-C * **/BE-B** +
8) IE-1 * BE-B * BE-E * **BE-D** * BE-G * /BE-A * /BE-C * **/BE-D** +

The highlighted basic events in the cut sets above represent both the success AND the failure of the same component in the same cut set. These are removed due to the Boolean algebra complementation law. Thus, cut set 6 is all that remains when generating cut sets with the "I" process flag.

The sequence cut sets generated via the delete term and the "I" process flag approaches are now ready for quantification. Section 2.2 discusses the two quantification methods used in this paper.

2.2 Current Cut Set Quantification

Multiple cut set quantification processes are utilizable for quantifying minimal cut sets. This paper focuses on the MinCut approximation and BDD. Most PRA software programs use MinCut by default. The MinCut equation is as follows:

$$\Pr(FT) = 1 - \prod_{i=1}^{n}(1 - CS_n) \tag{1}$$

This equation gives an upper-bound approximation to the exact solution. This process provides a good approximation, so long as the cut sets do not contain a lot of shared basic events. For external event PRAs and recovery events, the MinCut equation often overestimates the correct results, making the BDD process a better option for quantification in such cases.

The following is a simple example that illustrates BDD quantification of cut sets. For this example, the following cut set is quantified: A * B. Let us assume that the probability for these basic events is 0.2. The BDD for this cut set is shown in Figure 4.

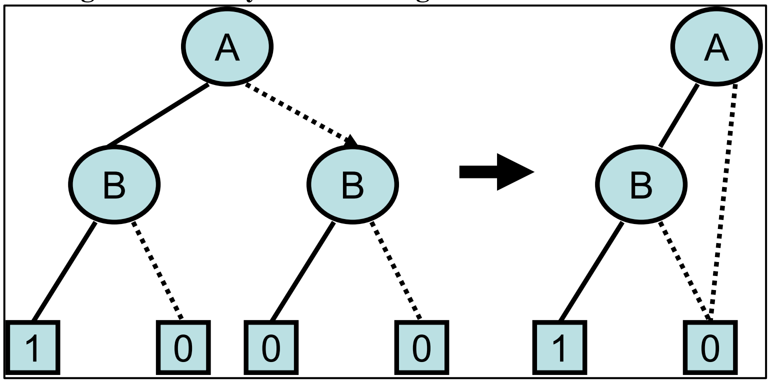
For this example, only one path exists:

A * B

Assuming A and B are independent, the probability of this cut set is:

$$P(CD)_{BDD} = P(A)P(B) = 0.2 * 0.5 = 0.1$$

Figure 4. Binary decision diagram for cut set A * B.



The MinCut approximation for this cut set is:

$$P(CD)_{MinCut} = 1 - [1 - P(A)P(B)] = 1 - [1 - 0.1] = 0.1.$$

2.3 Example Event Tree Cut Set Generation and Quantification

The sections above discussed cut set generation via the delete term process and the "I" process flag for the event tree in Figure 1 and its top event fault trees (see Figure 2). This section of the paper presents how cut sets from all the sequences are generated and then quantified using the MinCut and BDD quantification methods.

Table 1 provides the cut sets generated for each sequence via the delete term process. These accident sequence cut sets are then quantified using the MinCut and BDD quantification methods. The quantified results listed in Table 1 are based on the following basic event values: freq(IE-1) = 1.0/yr and Pr(BE-*) = 0.3 (all basic events are set to a 0.3 probability).

Table 1. Accident Sequence Cut Sets and Quantification

Sequence	Cut Set(s)	MinCut	BDD
2	IE-1*BE-H + IE-1*BE-E * BE-I	3.63E-01	3.63E-01
4	IE-1*BE-B * BE-E * BE-F	2.70E-02	2.70E-02
5	IE-1*BE-A + IE-1*BE-C + IE-1*BE-B * BE-D	5.54E-01	5.54E-01
sum		9.44E-01	9.44E-01

Quantification using both the MinCut approximation and BDD yields the same results, due to there being no shared basic events in the individual sequences. If this were not the case, the two results would be different.

The second option is to generate the accident sequences by using the "I" process flag. These cut sets are shown in Table 2. These accident sequence cut sets are then quantified using the MinCut approximation and BDD quantification. The results of the quantification are also listed in Table 2.

Table 2. Accident Sequence Cuts and Quantification

Sequence	Cut Set(s)	MinCut	BDD
2	IE-1*/BE-A * /BE-B * /BE-C * BE-H + IE-1*/BE-A * /BE-C * /BE-D * /BE-E * BE-H + IE-1*/BE-A * /BE-B * /BE-C * BE-E * BE-I	1.93E-01	1.46E-01
4	IE-1*/BE-A * /BE-B * /BE-C * /BE-D * BE-E * BE-F	9.26E-03	9.26E-03
5	IE-1*BE-A +	5.54E-01	5.54E-01

Sequence	Cut Set(s)	MinCut	BDD
	IE-1*BE-C + IE-1*BE-B * BE-D		
sum		7.57E-01	7.10E-01

The accident sequence quantification in Table 2 reveals a difference between the two methods. BDD quantification provides the exact solution based on the cut sets for each accident sequence. The MinCut approximation overestimates the result for sequence 2, due to (1) shared basic events across the three minimal cut sets, (2) the high failure probability of the basic events, and (3) the fact that the cut sets contain success terms.

This simple example reveals the issue with not including success terms with high probability values in the cut sets for sequences with successful top events, especially when there are shared basic events across cut sets and the cut sets are quantified using the MinCut approximation. As this simple example shows, the current state of practice overestimates the exact solution: 0.944 versus 0.71. This issue of overestimating accident sequence frequencies is found in external event PRAs, due to the nature of the values of those top events that are successful but not included in the cut sets. As external event PRAs become more widely used in risk decisions, a better method of cut set generation that includes the success terms must be determined, along with corresponding quantification methods. Overestimation can lead to incorrect decisions. The next section discusses methods for better estimating the accident sequence results by including success terms in the cut sets and performing quantification in a manner superior to the MinCut approximation. For real-world PRAs, accounting for all success terms would be impossible.

3 SUCCESS TERM RECOVERY APPROACH

The example presented above demonstrates that the current state of practice overestimates the exact frequency (probability) when success terms are not included in sequence cut sets, especially when the success term is not close to 1.0 or is in shared cut sets for a particular sequence. The example shows that attempting to properly account for success terms only works for simple PRA models and would be made impossible in large PRAs. Therefore, a process must be developed for solving the model in a reasonable timeframe and getting the closest approximation to the exact answer.

This paper presents a process for including a success term based on sequence-specific recovery basic events. These sequence-specific recovery events are themselves based on the quantified cut sets that represent only the success top events of an event tree accident sequence path. The sequence-specific recovery events represent the success probability based on all top events that succeed. The process is similar to allowing the individual success events to be generated with the sequence cut sets. The advantage of this process is that the standard default application, the delete term process, can be used to generate the sequence cut sets. Then all the cut sets within a sequence are ANDed with a recovery basic event that represents the probability of the success top events. This process will enable better estimates of the exact frequencies of the accident sequences and ultimately the overall core damage frequency (CDF).

This paper presents two different methods of generating the sequence-specific recovery event. These two methods are applied to the simple model presented in this paper, followed by a discussion of the pros and cons of each.

3.1 Method 1: Success "NOR" Gate

Method 1 creates a sequence-specific success fault tree by using a NOR gate as the top event gate. This fault tree is solved for its minimal cut sets, then quantified via the MinCut or BDD quantification process. The quantified probability becomes the sequence-specific recovery basic event, which will then be ANDed to each cut set for that specific sequence. The resultant sequence cut sets can then be quantified using the MinCut or BDD quantification process.

Based on the example event tree above, sequences 2 and 4 contain success top events. The sequence-specific success fault trees for these sequences are shown in Figure 5.

Figure 5. Sequence 2 and 4 success-term fault tree NOR gate.

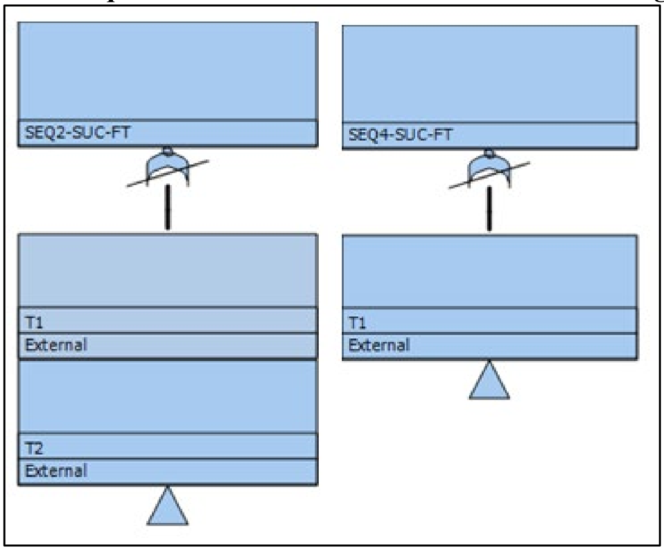


Table 3 provides the minimal cut sets for these two fault trees, along with quantification of these cut sets via MinCut and BDD.

Table 3. Success Fault Trees Cut Sets and Quantification

Fault Tree	Cut Set(s)	MinCut	BDD
SEQ2-SUC-FT	/BE-A * /BE-B * /BE-C + /BE-A * /BE-C * /BE-D * /BE-E	5.007E-01	4.150E-01
SEQ4-SUC-FT	/BE-A * /BE-B * /BE-C + /BE-A * /BE-C * /BE-D	5.684E-01	4.459E-01

The sequence-specific fault tree success result will now be applied to each cut set within their respective sequence. Applying this sequence-specific recovery event enables better estimation of the sequence quantification by accounting for the success terms in the respective sequences. The sequence cut sets are generated using the default application of the delete term process, then the sequence-specific recovery basic event is ANDed to each cut set. Table 4 shows the three sequences for our simplified event tree example, with the cut sets obtained via the delete term process and then quantified via the MinCut approximation. The table includes the cut sets for each sequence but also the sequence-specific recovery basic event ANDed to each cut set and then quantified via the MinCut approximation. The sequence-specific recovery basic event probability was quantified using the BDD method.

Table 4. Method 1 Sequence Cut Sets without/with the Recovery Basic Event

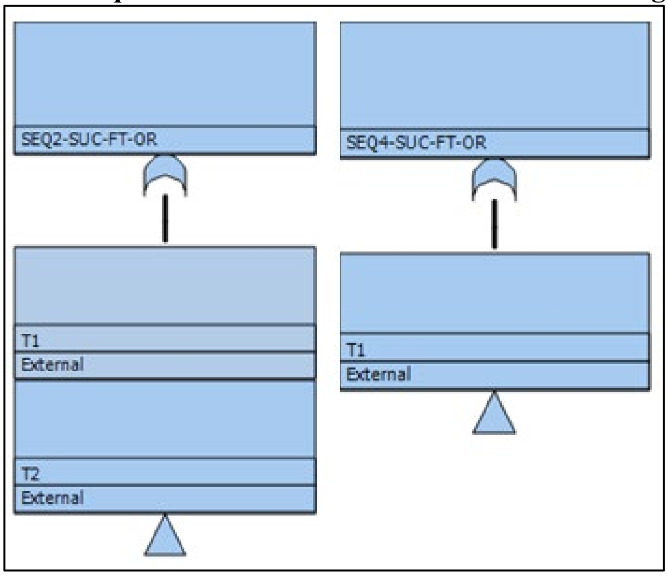
Sequence	Cut Set(s)	MinCut	Cut Set(s) with success term recovery	MinCut
2	BE-H + BE-E * BE-I	3.63E-01	BE-H * SEQ2-SUC-FT + BE-E * BE-I * SEQ2-SUC-FT	1.57E-01
4	BE-B * BE-E * BE-F	2.70E-02	BE-B * BE-E * BE-F * SEQ4-SUC-FT	1.20E-02
5	BE-A + BE-C + BE-B * BE-D	5.54E-01	BE-A + BE-C + BE-B*BE-D	5.54E-01
sum		9.44E-01		7.23E-01

Table 4 shows that when the sequence-specific recovery basic event is applied to each sequence cut set and quantified via BDD, the results provide a good approximation of the exact answer. Table 2 presents the results when all success terms were carried through all the cut sets and quantified via BDD. That approach yielded a 7.10E-1 frequency, whereas the Table 4 approach yielded a 7.23E-1 frequency. The Table 2 approach definitely fosters a better approximation than the standard delete term approach without success terms included, which yielded a 9.44E-1 frequency.

3.2 Method 2: Success OR Gate

Method 2 creates a sequence-specific success fault tree using an OR gate as the top event gate, as shown in Figure 6. This fault tree is solved to obtain the minimal cut sets. These minimal cut sets are then quantified using either the MinCut approximation or BDD.

Figure 6. Sequence 2 and 4 success-term fault trees OR gate.



The minimal cut sets and quantified results are shown in Table 5.

Table 5. Fault Tree Failure Cut Sets for Success Top Events and Quantification

Fault Tree	Cut Set(s)	MinCut	BDD
SEQ2-SUC-FT	BE-A + BE-C + BE-B * BE-E + BE-B * BE-D	5.942E-01	5.850E-01
SEQ4-SUC-FT	BE-A + BE-C + BE-B * BE-D	5.541E-01	5.541E-01

The results shown in Table 5 represent the failure of any top event that is listed as a success in any given sequence, i.e., each fault tree is ORed for the failure cut sets. Once the failure probability is determined, this result is subtracted from 1.0 to then provide the sequence-specific success recovery basic event, which is then ANDed into the respective sequence cut sets.

Table 6 provides the three sequences for our simplified event tree for the case without top event success factored in followed by the results from factoring in the sequence-specific recovery basic event obtained from this second method. The sequence-specific recovery basic event probability was obtained using the BDD quantification method.

Table 6. Method 2 Sequence Cut Sets without/with Recovery Basic Event

Sequence	Cut Set(s)	MinCut	Cut Set(s) with success term recovery	MinCut
2	BE-H + BE-E * BE-I	3.63E-01	BE-H * SEQ2-SUC-FT + BE-E * BE-I * SEQ2-SUC-FT	1.57E-01
4	BE-B * BE-E * BE-F	2.70E-02	BE-B * BE-E * BE-F *	1.20E-02

Sequence	Cut Set(s)	MinCut	Cut Set(s) with success term recovery	MinCut
			SEQ4-SUC-FT	
5	BE-A + BE-C + BE-B * BE-D	5.54E-01	BE-A + BE-C + BE-B*BE-D	5.54E-01
sum		9.44E-01		7.23E-01

The results shown in Table 6 are identical to those in Table 4. For this simple example, using either method generates the same result. However, if the model is more complex, these two methods would provide different results assuming Method 1 could be solved.

The above information was compiled using a very simple model to provide the background on creating a success recovery event. This simple example allows for hand calculations to verify that the process can be developed and applied to obtain final results. The next section discusses the application of these methods to a more realistic PRA model.

3.3 Pros and Cons of Success Term Recovery Events

This paper presents two different methods of handling success terms for situations in which the success term value is not close to 1.0. The pros and cons of each of the two methods are presented below.

Pros of the two methods:
The pros of both methods are that they enable better estimates than those arrived at via the current state of practice, which does not factor in those success terms whose value is not close to 1.0. Method 2 has the advantage over Method 1 in that it facilitates faster cut set generation and quantification. Both methods benefit from using the BDD quantification method for the sequence-specific recovery basic event, which is needed due to shared basic events across cut sets.

Cons of the two methods:
There are cons to both methods. Method 1 appears to possibly have an issue in its ability to generate the success cut sets in a real PRA model, as the success fault tree could become too large to properly handle cut set generation. The cut set generation and quantification time appears to perhaps also be an issue. Method 2 may have an issue when solving the fault trees in order to obtain the sequence-specific recovery basic event value. If the generated cut set quantification is very close to 1.0, there may be an issue in terms of numeric round-off within software packages (i.e., 0.99999 = 1.0 and therefore 1 – Pr[0.99999] = 0, when it actually is 1.0E-5). In Section 4, a more detailed PRA model is evaluated, and the round-off issue within SAPHIRE was addressed by using Excel to quantify the cut sets by allowing additional significant figures. Method 2 appears to have fewer cons than Method 1.

4 DETAILED PRA MODEL

A more detailed PRA model was developed to illustrate the potential benefit of employing Method 2, as discussed in Section 3. This model evaluated four different loss-of-offsite-power (LOOP) events: grid related, plant-centered, switchyard-centered, and weather related. These four LOOP events were chosen due to the failure probabilities of the modelled power recovery events, turbine-driven pump failure modes, and other power-related components. The power recovery events, and turbine-driven pump have high failure probabilities, and if the success probability is not properly handled, the overall results will be overestimated. Along with the four different LOOP events, six seismic bin events were evaluated. The seismic events have lots of high failure probability events, especially for the higher ground acceleration seismic events. Evaluation of external events was the motivation for developing a success term recovery basic event to provide better quantification than the current state of practice.

4.1 Loss-of-Offsite-Power Events

The four LOOP event trees are solved using the current state of practice process. In single-input top events, this process enables the success basic event to be part of the final cut sets. Fault trees containing one or two basic events should be allowed to carry their success events through the cut sets; however, fault trees with more than two basic events can dramatically slow down the cut set generation and potentially make cut set generation impossible to solve. Also, the increased number of success events causes issues when quantifying using the MinCut approximation, as there will be more shared basic events across cut sets.

Table 7 lists the results obtained from the four LOOP event trees, using the state of practice. These cut sets were quantified using both the MinCut approximation and BDD method.

Table 7. LOOP State of Practice Results

Event Tree	Description	MinCut	BDD
LOOPGR	LOOP Grid Related	1.73E-07/yr	1.66E-07/yr
LOOPPC	LOOP Plant Centered	1.61E-08/yr	1.56E-08/yr
LOOPSC	LOOP Switchyard Centered	1.75E-07/yr	1.69E-07/yr
LOOPWR	LOOP Weather Related	2.69E-07/yr	2.55E-07/yr
Total		6.33E-07/yr	6.06E-07/yr

The four LOOP event trees are now solved by first turning off any success fault tree top events. This eliminates the possibility of double counting the success basic events. The next step is to quantify the success fault tree via BDD. The BDD quantification method provides the best estimate, due to the shared basic events within the cut sets. This quantified result is then ANDed to each cut set within the specific sequence. The results of this operation are shown in Table 8.

Table 8. LOOP Success Term Recovery Event Results

Event Tree	Description	MinCut	BDD
LOOPGR	LOOP Grid Related	1.72E-07/yr	1.65E-07/yr
LOOPPC	LOOP Plant Centered	1.61E-08/yr	1.55E-08/yr
LOOPSC	LOOP Switchyard Centered	1.74E-07/yr	1.68E-07/yr
LOOPWR	LOOP Weather Related	2.64E-07/yr	2.51E-07/yr
Total		6.27E-07/yr	6.00E-07/yr

4.2 Seismic Events

The six seismic events are solved using the state of practice process. This process allows for single-input top events to have their successes be included in the final cut sets. The results obtained from this application are listed in Table 9. The resultant cut sets were quantified using both the MinCut approximation and BDD.

Table 9. Seismic State of Practice Results

Event Tree	Description	MinCut	BDD
EQK-BIN1	Seismic Bin 1 (0.17g)	1.13E-09/yr	1.11E-09/yr
EQK-BIN2	Seismic Bin 2 (0.39g)	2.21E-07/yr	2.12E-07/yr
EQK-BIN3	Seismic Bin 3 (0.61g)	2.29E-06/yr	1.83E-06/yr
EQK-BIN4	Seismic Bin 4 (0.87g)	2.83E-06/yr	1.71E-06/yr
EQK-BIN5	Seismic Bin 5 (1.22g)	2.34E-06/yr	1.25E-06/yr
EQK-BIN6	Seismic Bin 6 (2.12g)	5.99E-07/yr	2.28E-07/yr
Total		8.28E-06/yr	5.23E-06/yr

The six seismic event trees are now solved by turning off any success fault tree top events. The next step is to quantify the success fault tree by using the BDD method. This quantified result is then ANDed to each cut set within the specific sequence. The results of this operation are shown in Table 10.

Table 10. Seismic Success Term Recovery Event Results

Event Tree	Description	MinCut	BDD
EQK-BIN1	Seismic Bin 1 (0.17g)	1.12E-09/yr	1.10E-09/yr
EQK-BIN2	Seismic Bin 2 (0.39g)	2.19E-07/yr	2.09E-07/yr

Event Tree	Description	MinCut	BDD
EQK-BIN3	Seismic Bin 3 (0.61g)	2.23E-06/yr	1.78E-06/yr
EQK-BIN4	Seismic Bin 4 (0.87g)	1.08E-06/yr	5.86E-07/yr
EQK-BIN5	Seismic Bin 5 (1.22g)	2.32E-06/yr	1.23E-06/yr
EQK-BIN6	Seismic Bin 6 (2.12g)	5.97E-07/yr	2.28E-07/yr
Total		6.44E-06/yr	4.03E-06/yr

4.3 PRA Results

As per the analyses of the LOOP and seismic events, applying the success term recovery event has different impacts. For the LOOP events, there was minimal difference in the current state of practice's final result of 6.33E-07/yr versus the 6.27E-07/yr result of the Method 2 approach with the success term recovery basic event quantified via BDD. This minimal difference between the final results is the main reason why the state of practice has served so well to date. Obtaining a 1% reduction in the final result does not warrant the extra effort needed to include the success terms.

The two seismic evaluation results do show a much larger difference. Based on the state of practice using BDD quantification versus application of the success term recovery basic event, a reduction of 23% was achieved. This is a big difference when making risk-informed decisions. This difference is also seen in the development of Level 2 PRA. Level 2 PRA models are tied directly to Level 1 accident sequences, and overestimation of the Level 1 accident sequences can lead to overestimation of the Level 2 results.

5 CONCLUSION

This paper illustrated the issues that arise when the state of practice is used for accident sequence cut set generation and quantification. It is important to incorporate the success terms into the analysis in situations when failure basic event values are high and thus the success term reflects a low value (i.e., not close to 1.0, which is the value assumed for success terms in the current state of practice). This issue becomes even more important in the quantification process, since the success terms are shared across cut sets and their values are higher than those of other failed basic events. These characteristics go against the assumptions employed in the MinCut approximation method during cut set quantification. MinCut approximation assumes no shared basic events across cut sets, and any such events would be assigned a low probability value. The paper showed that, in these situations, the BDD quantification method is preferable to the MinCut approximation. This paper illustrated the application of sequence-specific success term recovery basic events to provide better estimates of the accident sequences and thus better estimates of the final results.

This paper showed that the current state of practice for internal events provides a good approximation of the quantified results. There would be little benefit in developing and applying a sequence-specific recovery basic event for situations when failure probabilities are low and thus success probabilities are high. However, for external events, the development and application of sequence-specific recovery basic events provides a large benefit. As risk insights and decisions tend to be based on the inclusion of external events, the methods presented in this paper (or some other method) must be employed to better quantify the PRA model. Using overly conservative results can lead to skewed insights and decisions, causing potential negative impacts. Therefore, proper application of success terms within the PRA must be applied so that proper insights and decisions can be made.

6 REFERENCES

[1] Dutuit, Y. and A. Rauzy, 2001, "Efficient algorithms to assess component and gate importance in fault tree analysis," Reliability Engineering and System Safety, Vol. 72, pp. 213-222.

[2] Remenyte-Prescott, R. and J. Andrews, 2008, "Analysis of non-coherent fault trees using ternary decision diagrams," *Proceedings of International Mechanical Engineering*," Vol. 222, pp 127-138.

[3] Jung, W. S., 2015, “A method to improve cutset probability calculation in probabilistic safety assessment of nuclear power plants,” *Reliability Engineering and System Safety*, Vol. 134, pp. 134-142.
[4] Akers, S., 1978, “Binary Decision Diagrams,” *IEEE Transactions on Computers*, Vol. C-27, No. 6, June.
[5] Rauzy, A. and Y. Dutuit, 1997, “Exact and truncated computations of prime implicants of coherent and non-coherent fault trees within Aralia,” *Reliability Engineering and System Safety*, Vol. 58, pp. 127-144.
[6] Sinnamon, R. M. and J. D. Andrews, 1997 “Improved Accuracy in Quantitative Fault Tree Analysis,” *Quality and Reliability Engineering International*, Vol. 13, pp. 285-292.
[7] Andrews, J. D. and S. J. Dunnett, 2000, “Event-Tree Analysis Using Binary Decision Diagrams,” *IEEE Transaction on Reliability*, Vol. 49, No. 2, June.