Fault Tree Analysis with Temporal Gates and Model Checking Technique for Qualitative System Safety Analysis

Kwang Yong Koh[*] and Poong Hyun Seong
Department of Nuclear and Quantum Engineering, Korea Advanced Institute of Science and Technology,
371-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea
[*]Corresponding author: goeric1@kaist.ac.kr

1. Introduction

Fault tree analysis (FTA) has suffered from several drawbacks such that it uses only static gates and hence can not capture dynamic behaviors of the complex system precisely, and it is in lack of rigorous semantics, and reasoning process which is to check whether basic events really cause top events is done manually and hence very labor-intensive and time-consuming for the complex systems [1] while it has been one of the most widely used safety analysis technique in nuclear industry. Although several attempts have been made to overcome this problem, they can not still do absolute or actual time modeling because they adapt relative time concept and can capture only sequential behaviors of the system. [2-4].

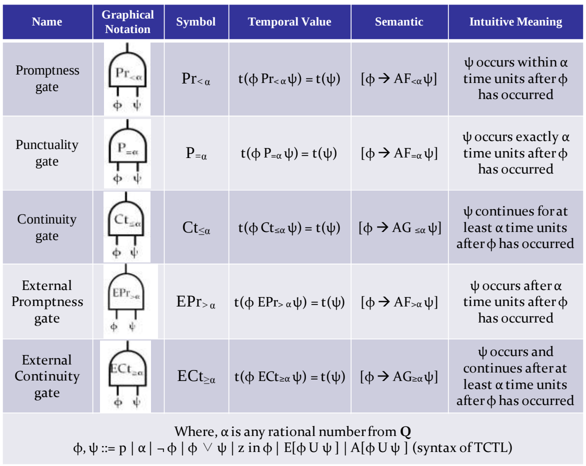
In this work, to resolve the problems, FTA and model checking are integrated to provide formal, automated and qualitative assistance to informal and/or quantitative safety analysis. Our approach proposes to build a formal model of the system together with fault trees. We introduce several temporal gates based on timed computational tree logic (TCTL) to capture absolute time behaviors of the system and to give concrete semantics to fault tree gates to reduce errors during the analysis, and use model checking technique to automate the reasoning process of FTA.

2. Temporal Fault Tree

In this section we introduce new temporal gates based on TCTL to describe dynamic behaviors of system which changes its states as time goes on, and define the temporal gates in terms of 'name', 'graphical notation', 'symbol', 'semantic' and 'intuitive meaning'.

2.1 Temporal gates

Since a fault tree cannot represent conditions that change over time, the labels (or descriptions of events) often include text fragment like 'too late', 'eventually', 'before', 'after', and in more detail expression, 'seven seconds after', 'three minutes before', 'during four seconds' and so on which need to be understood formally. In order to handle this problem, we introduce several temporal gates and define their notations. With these gates we easily specify the temporal dependence between events and preserve the simple, qualitative and visual nature of the fault trees. Each temporal gate has it is own usage. For example, the 'continuity gate' is useful in the description of the situation where an event should continue for at least particular time after the other event has occurred and the corresponding expression in the form of TCTL is [φ → AG $_{\leq\alpha}$ ψ] (ψ continues for at least α time units after φ has occurred). Users could easily understand the usages of other temporal gates from the intuitive meaning in the definition of Figure1.



Name	Graphical Notation	Symbol	Temporal Value	Semantic	Intuitive Meaning
Promptness gate	$Pr_{<\alpha}$ ϕ ψ	$Pr_{<\alpha}$	$t(\phi\ Pr_{<\alpha}\ \psi) = t(\psi)$	$[\phi \rightarrow AF_{<\alpha}\psi]$	ψ occurs within α time units after ϕ has occurred
Punctuality gate	$P_{=\alpha}$ ϕ ψ	$P_{=\alpha}$	$t(\phi\ P_{=\alpha}\ \psi) = t(\psi)$	$[\phi \rightarrow AF_{=\alpha}\psi]$	ψ occurs exactly α time units after ϕ has occurred
Continuity gate	$Ct_{\leq\alpha}$ ϕ ψ	$Ct_{\leq\alpha}$	$t(\phi\ Ct_{\leq\alpha}\ \psi) = t(\psi)$	$[\phi \rightarrow AG_{\leq\alpha}\psi]$	ψ continues for at least α time units after ϕ has occurred
External Promptness gate	$EPr_{>\alpha}$ ϕ ψ	$EPr_{>\alpha}$	$t(\phi\ EPr_{>\alpha}\ \psi) = t(\psi)$	$[\phi \rightarrow AF_{>\alpha}\psi]$	ψ occurs after α time units after ϕ has occurred
External Continuity gate	$ECt_{\geq\alpha}$ ϕ ψ	$ECt_{\geq\alpha}$	$t(\phi\ ECt_{\geq\alpha}\ \psi) = t(\psi)$	$[\phi \rightarrow AG_{\geq\alpha}\psi]$	ψ occurs and continues after at least α time units after ϕ has occurred

Where, α is any rational number from **Q**
ϕ, ψ ::= p | α | ¬ ϕ | ϕ ∨ ψ | z in ϕ | E[ϕ U ψ] | A[ϕ U ψ] (syntax of TCTL)

Fig. 1. Definitions of several temporal gates

2.2 Comprehensiveness of temporal gates

We developed five temporal gates in total for describing dynamic behaviors of system in fault trees and those are comprehensive enough to explain most of system behaviors even needs of more temporal gates if necessary.

Figure 2 illustrates time aspects of five temporal gates based on their own temporal values. Temporal value of a gate is the time at which it occurs (and so become true). For gates, this is the value of time that is used when one gate is an input to another gate. For example, 'continuity gate' covers the situation that after first event occurrence, second event continues till a specific or interesting time point while 'external continuity gate' covers the situation that after a specific or interesting time point just after first event occurrence, second event occurs and continues. With these two gates, we can describe all behaviors of system in terms that second event should continue. Similarly, 'promptness gate' covers the situation that after first event occurrence, second event occurs once at any time point within a specific or interesting time point with not continuing while 'external promptness gate' covers the situation that after a specific or interesting time point just after first event occurrence, second event occurs once at any time point. With these two gates, we can describe all behaviors of system in terms that second event should occur once at any time point.

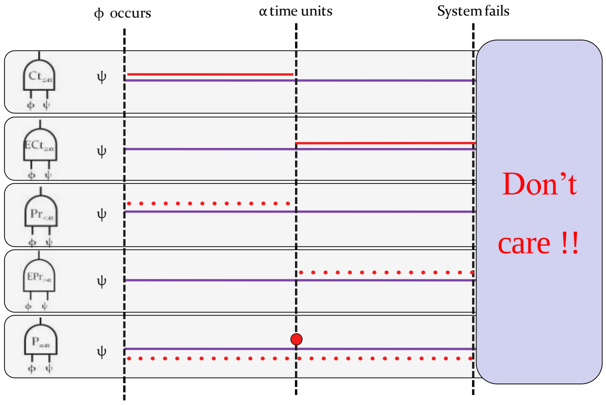


Fig. 2. Time aspects of temporal gates based on their temporal values

3. Model Checking and UPPAAL

Model checking is the most usual formal verification technique and a proven-effective and automated technique in verifying complex behavior of concurrent systems and we selected a real time model checker UPPAAL [5] to support our approach.

We made a fault tree of DFWCS based on the system description and FMEA results in [4]. All the information of a fault tree is translated into UPPAAL query language for automatic verification. First, fault tree gates are translated to corresponding UPPAAL query language based on transition rules between TCTL and CTL, and UPPAAL query language. After completion of the translation process, the translated fault tree information which is now system property leading to unintended system state (hazardous state) is verified by UPPAAL model checker against UPPAAL system model implemented previously: we made fifteen separated models (which called 'template' in UPPAAL) for describing DFWCS behavior. Some of the templates are for describing corresponding components behaviors of DFWCS, and others are additional templates for describing special dependency between components.

From the verification results of the properties, we can conclude more easily that the fault tree has flaws, and hence the analysis result is also erroneous.

4. Conclusions

This paper demonstrated that the new temporal gates are useful to capture dynamic behaviors of system precisely and that model checking technique is helpful when we validate the correctness of informal safety analysis such as FTA. Our approach not only formalizes the semantics of fault trees, but it also extends the expressive power of FTA to model temporal ordering of events. But the concept will need further improvements and validation by larger scaled case studies. The gates proposed here are not yet sufficient to model all situations that arise in digitalized systems. Thus we intend to add some new gates to our framework, if necessary. In other for our method to have any strength over other method reviewed in this paper, the supporting tool to automate the proposed method should be developed. So far, the proposed method was applied to small parts of systems because the approach was not automated. Although the tool to automate our method is being under development, we expect that the method become promising to even large scale complex system with tool support.

REFERENCES

[1] K. Y. Koh, P. H. Seong, SA**CS**^{2}: A Dynamic and Formal Approach to Safety Analysis for Complex Safety Critical System, Proceedings of Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC&HMIT-2009), April 5-9, 2009, Knoxville, TN.

[2] J. B. Dugan, S. J. Bavuso, and M. A. Boyd, Dynamic fault-tree models for fault-tolerant computer systems, IEEE Transactions on Reliability, Vol.41(3), p.363, 1992.

[3] M. Walker, and Y. Papadopoulos, Synthesis and analysis of temporal fault trees with PANDORA: The time of Priority AND gates, Nonlinear Analysis: Hybrid Systems, Vol.2(2), p368, 2008.

[4] F. Ortmeier, W. Reif, G. Schellhorn, A. Thums, B. Hering, and H. Trappschuh, Safety analysis of the height control system for the Elbtunnel, Reliability Engineering and System Safety, Vol.81(3), p259, 2003.

[5] P. Pettersson, and K. G. Larsen, Uppaal2k, Bulletin of the European Association for Theoretical Computer Science, Vol.70, p40, 2000.

[6] T. Aldemir, M.P. and et al., Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments, NUREG/CR-6942, U.S. NRC, Washington, D.C., 2007.