# Generating Synthetic Event Datasets for Root Cause Analysis Algorithm Tuning

D. Gueorguiev 7/19/23

## Notation

- directed graph

- the vertex set of the directed graph

– the arc set of the directed graph

– event type for which we would like to do root cause analysis

– set of event types associated with the event subject to analysis

– set of all event types

- the set of all instances of associated events of types in

– the set of all constraints applied to event types in and their instances

Let us denote by the set of the event types which are relevant in root cause analysis of specific event type .

That is, the event types and the event type will form causal pairs for which we want to calculate causal significance factor and construct Directed Causal Graph (DCG).

We will assume that we can have multiple instances of each event type . Here denote different sets of arguments for the same event type .

We will consider the following constraint types which can be imposed on the event types. We may impose *directly follows* () constraint and *reachable from* () constraint to a subset of event types. Generalization of the *reachable from* constraint is *reachable from in at least steps* () and *reachable from in at most steps* ().

Another relevant constraint is the *multiplicity type* constraint with possible multiplicity types: *max children count* , *min children count* , *max total count*  and *minimum total count* for an event type within the dataset of *associated events*[[1]](#footnote-1)[[2]](#footnote-2)[[3]](#footnote-3)[[4]](#footnote-4)[[5]](#footnote-5). The most general set of constraints which deal with non-deterministic conditions can be expressed using *Probabilistic Temporal Logic* (PTL) (for details see Appendix). Let us denote the set of all constraints imposed on even types in

We would like to construct a sequence of events from the specified types set obeying the set of constraints .

## Appendix: Probabilistic Temporal Logic

A detailed and thorough survey by Clarke et al on the most relevant logic systems for model verification can be found here: [Clarke, E. M., Schlingloff, B. H. (2001). Model Checking](https://github.com/dimitarpg13/root_cause_analysis_and_model_checking/blob/main/literature/ModelChecking/ModelChecking_ClarkeSchlingloff1999.pdf)

As quick intro into Temporal Logic can serve Clarke, Emerson, and Sistla's paper: [Clarke, E.M., Emerson, E.A., Sistla, A.P. (1983). Automatic Verification Finite State Concurrent Systems Using Temporal Logic Specifications: A Practical Approach](https://github.com/dimitarpg13/root_cause_analysis_and_model_checking/blob/main/literature/ModelChecking/AutomaticVerifictionOfFiniteStateConcurrentSystemUsingTemporalLogicSpecification.pdf)

An excellent tutorial for Probabilistic Temporal Logic (PTL) is the Hanssen and Jonsson's paper: [Hansson, H., Jonsson, B. (1994). A Logic about Reasoning about Time and Reliability](https://github.com/dimitarpg13/root_cause_analysis_and_model_checking/blob/main/literature/ModelChecking/ALogicforReasoningaboutTimeandReliability_Hansson_Johnson_1994.pdf)

A refresher on first order logic which is the fundament of the logic systems for model verification can be found here: [First-Order Logic, Open Logic Project](https://github.com/dimitarpg13/root_cause_analysis_and_model_checking/blob/main/literature/ModelChecking/first-order-logic-OpenLogicProject.pdf)

1. is from **[μ](https://www.wordhippo.com/what-is/the-meaning-of/greek-word-125d36fa78073a7c4d390e61ab9efaf50ccb1340.html)**[έγιστο](https://www.wordhippo.com/what-is/the-meaning-of/greek-word-125d36fa78073a7c4d390e61ab9efaf50ccb1340.html) (Greek for *maximum*) [↑](#footnote-ref-1)
2. is from **ε**λάχιστο (Greek for *minimum*) [↑](#footnote-ref-2)
3. The subscript is from **π**αιδί (Greek for *child*) [↑](#footnote-ref-3)
4. The subscript is from **ο**λικός (Greek for *overall*) [↑](#footnote-ref-4)
5. The subscript is from **σ**υνεταιρισμός (Greek for *association*) [↑](#footnote-ref-5)