A fault-classification scheme based on safety and liveness

Felix Gärtner

Darmstadt University of Technology
Germany

Motivation

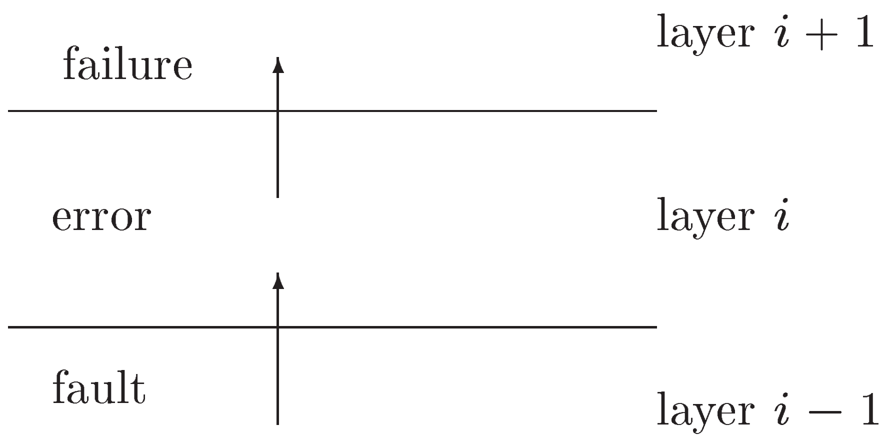
- There are many fault classifcation schemes:
  - hierarchic [Hadzilacos and Toueg 1993]
  - time/value distinction [Powell 1992]
- Unifying approaches [Echtle and Masum 1999]
- Are they too close to reality?

Overview

- Recap: fault, error, failure
- Transition systems, safety, liveness
- Abstract classification

Recap: fault, error, failure

- Observation: faults are bound to one level of abstraction
- Terminology [Laprie 1992]:



- fault model: description of which components may fail in what way

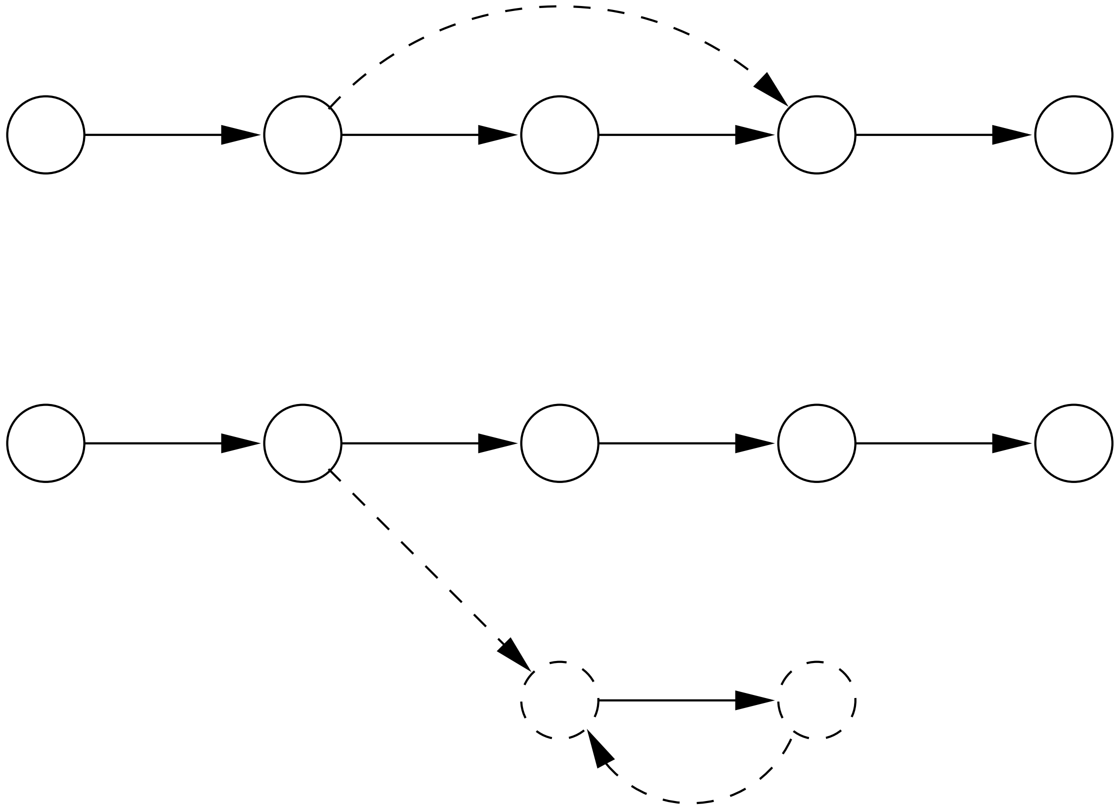
Transition Systems, Safety, Liveness

- Transition system $A = (S, I, T, L)$ — a "generator of traces"
- Set of traces is a *property*
- Property P holds for A: $A \subseteq P$
- Every property is the intersection of *safety* and *liveness* [Alpern and Schneider 1985]
- Safety = "always", liveness = "(always) eventually"

Fault Models as Transformations

- discrete faulty behavior can be "programmed" [Cristian 1985]
- Fault model can be viewed as a transformation F
- F maps A to $A' = F(A)$ where $A \subset F(A)$
- F can add (or remove?) states or transitions

Adding Transitions and States



Effects on Satisfaction of Safety

add	effect on safety	tolerable
transitions	none	yes
transitions	direct violation	no
transitions	indirect violation	yes
states	none	yes

Effects on Satisfaction of Liveness

add	effect on liveness	tolerable
states	none	yes
transitions	none*	yes
transitions	fault/program loop violation**	yes
transitions and states	direct stopping violation	no
transitions and states	pure fault loop violation	no
transitions and states	fault/program loop violation	yes

* Liveness properties of the form $\Diamond\phi$ (ϕ is a "final" state) or $\Box\Diamond\varphi$.

** only possible with certain kinds of fairness.

Summary

- Formal treatment yields well-separable classes of faults
- "Theorems":
  - Not possible to cause violation of safety by only adding states
  - Not possible to cause violation of liveness by only adding transitions (depends on type of liveness and fairness)
- Too far from reality?

Back Matter

References

ALPERN, B. AND SCHNEIDER, F. B. 1985. Defining liveness. *Information Processing Letters 21*, 181–185.

CRISTIAN, F. 1985. A rigorous approach to fault-tolerant programming. *IEEE Transactions on Software Engineering 11*, 1 (Jan.), 23–31.

ECHTLE, K. AND MASUM, A. 1999. Understanding cooperative byzantine failures: A novel failure classification to enable efficient fault-tolerant protocols. In *Proceedings of the Annual IEEE Workshop on Fault-Tolerant Parallel and Distributed Systems (FTPDS'99)* (San Juan, Puerto Rico, USA, April 1999). Kluwer.

HADZILACOS, V. AND TOUEG, S. 1993. Fault-tolerant broadcasts and related problems. In S. MULLENDER Ed., *Distributed Systems* (Second ed.)., Chapter 5, pp. 97–145. Addison-Wesley.

LAPRIE, J.-C. Ed. 1992. *Dependability: Basic concepts and Terminology*, Volume 5 of *Dependable Computing and Fault-Tolerant Systems*. Springer-Verlag.

POWELL, D. 1992. Failure mode assumptions and assumption coverage. In D. K. PRADHAN Ed., *Proceedings of the 22nd Annual International Symposium on Fault-Tolerant Computing (FTCS '92)* (Boston, MA, July 1992), pp. 386–395. IEEE Computer Society Press.