Test und Verlässlichkeit (F1)
Kapitel 1: Grundbegriffe, Wahrscheinlichkeit, Experimente

Prof. G. Kemnitz

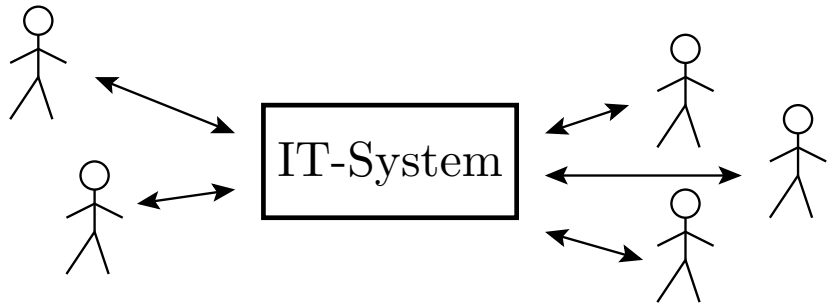
Institut für Informatik, TU Clausthal (TV-F1)

26. April 2016

Vertrauen und Verlässlichkeit

IT-Systeme automatisierten intellektuelle Aufgaben:

- betriebliche Abläufe,
- Steuerung von Prozessen und Maschinen,
- Entwurfsaufgaben, ...



Einsatzvoraussetzung ist Vertrauen, dass

- das System, wenn es gebraucht wird, funktioniert,
- seine Service-Leistungen korrekt und pünktlich ausführt,
- keine unkalkulierbaren Schäden und Kosten verursacht.

Fakt 1

Vertrauen setzt Verlässlichkeit voraus.

Fehlfunktionen in IT-Systemen müssen nicht, aber können erheblichen Schaden verursachen:

- Datenverlust,
- Hintertüren für den Datenmissbrauch,
- Unfälle, Selbstzerstörung, Produktionsausfälle, ...

Am 3. Juni 1980 meldete ein Rechner des nordamerikanischen Luftverteidigungszentrums den Anflug sowjetischer Nuklearraketen. Sofort wurden Vergeltungsmaßnahmen vorbereitet. Eine Überprüfung der Daten von Radarstationen und Satelliten konnte den Angriff nicht bestätigen, so dass der Start amerikanischer Raketen mit Nuklearsprengköpfen in letzter Minute gestoppt wurde[1].

Urheber der nahen Katastrophe war ein defekter Schaltkreis in einem Rechner.

[1] Hartmann, J., Analyse und Verbesserung der probabilistischen Testbarkeit kombinatorischer Schaltungen, Diss. Unversität des Saarlandes, 1992

Umgangssprachlich beschreibt Verlässlichkeit (von Personen, Apparaten, Werkstoffen, ...), dass man ihnen trauen kann. Dabei treffen Wunschvorstellungen und Wirklichkeit aufeinander.

Sprichworte mit tiefem Wahrheitsgehalt:

- *Allen Leuten recht getan, ist eine Kunst, die keiner kann.* Erhebliche Abweichungen in der subjektiven Wahrnehmung.
- *Verlorenes Vertrauen ist schwer wieder herzustellen.* Das subjektiv empfundene Vertrauen passt sich der objektiven Wahrnehmung nur zögerlich an.
- Murphys Law: *Whatever can go wrong will go wrong.*

  Problemüberbewertung nach frustrierenden negativen Erfahrungen.
- *It ist not a bug, it is a feature.*

  Mit geeigneter Rethorik lassen sich offensichtliche Probleme aus dem menschlichem Bewussein verdrängen oder zum gottgegebenen Zustand erklären.

Verlässlichkeit von IT-Systemen

Der aktuelle Verlässlichkeitsbegriff für IT-Systeme beschreibt Verlässlichkeit durch »Attributes«, »Threads« und »Means«.[2].

Die Verlässlichkeit charakterisierende Eigenschaften (Attributes):

- Verfügbarkeit (Availability): Bereitschaft für einen korrekten Service.
- Zuverlässigkeit (Reliability): Stetigkeit korrekter Service-Leistungen.
- Wartbarkeit (Maintainability): Möglichkeiten, Wartungsarbeiten und Reparaturen durchzuführen.
- Integrität (Integrity): Ausschluss missbräuchlicher Systemveränderungen.
- Sicherheit (Safety): Ausschluss katastrophaler Folgen für Benutzer und Umwelt.

[2] J.C. Laprie. "Dependable Computing and Fault Tolerance: Concepts and terminology," 15th IEEE Int. Symp. on Fault-Tolerant Computing, 1985

Problemquellen für die Verlässlichkeit (Threads):

- Fehler (Fault, Bug): Fehlerhafte Realisierung des Systems. Potentielle Ursache für Fehlfunktionen.
- Fehlfunktion (Error): Verhalten im Widerspruch zum spezifizierten Sollverhalten.
- Versagen (Failure): Nicht vom System abgefangene oder korrigierte Fehlfunktion.

Maßnahmen zur Sicherung der Verlässlichkeit (Means):

- Fehlervermeidung (Prevention): Verhinderung Fehlerentstehung.
- Fehlerbeseitigung (Removal): Vor oder während des Einsatzes.
- Fehlfunktionen vorausschauend umgehen (Forcasting): Vorhersage wahrscheinlicher Fehlfunktionen, um diese zu umgehen.
- Toleranz (Tolerance): Mittel, dass ein Service auch mit Fehlern und Fehlfunktionen korrekt ausgeführt wird, gegebenenfalls mit verringerter Leistung.

Der Verlässlichkeitsbegriff, seine Teilaspekte und deren Definitionen befinden sich noch in einem Entwicklungsprozess.

Der Schlüssel zu objektiv verlässlichen Systemen

Kontrollen und das Abstellen der dabei erkannten Mängel auf drei Ebenen:

- Fehlervermeidung während des Entwurfs und der Fertigung,
- Test und Fehlerbeseitigung vor und während des Einsatzes,
- Ergebniskontrolle und Umgehung bzw. Tolerierung erkannter Fehlfunktionen im Betrieb.

Quantitative Bewertung durch Zählen/Messen/Schätzen

- der Anzahl der entstandenen, erkannten, vermiedenen, ... unerwünschten Fehler, Fehlfunktionen, Schadensfälle, ...) und
- der Zeiten zwischen ihrem Eintreten.

Testen – der Schwerpunkt in der Vorlesung – ist der »Filter« für das Erkennen der Fehler als Voraussetzung für ihre Beseitigung.

Inhalt und Lernziele der Vorlesung

- Modelle für die Bewertung der Verlässlichkeit incl. einer darauf angepassten Einführung in die Stochastik
- Verlässlichkeitssichernde Maßnahmen
  - im Betrieb: Kontrollen, Fehlertoleranz, ...
  - vor dem Einsatz: Test, Reparatur, ...
  - während der Entstehung: Fehlervermeidung.
- Schwerpunkt Test, Testauswahl, Zufallstest, ...

Lernziele:

- Bewertung und Abschätzung von Verlässlichkeitsaspekten.
- Verständnis des Zusammenwirkes der unterschiedlichen verlässlichkeitssichernden Maßnahmen.
- Antworten auf die Fragen: Wie, wie viel, wie vollständig wird getestet / müßte getestet werden?

Foliensätze

- F1: Grundbegriffe, Wahrscheinlichkeit, Experimente
  - Ausgewählte Begriffe und Grundlagen der Wahrscheinlichkeitsrechnung,
  - Verlässlichkeitsaspekte als Zufallsexperimente.
- F2: Struktur, Verteilungen, Fehler und Fehlfunktionen
  - Systemstruktur, Fehler und ihr Nachweis,
  - Verteilungen der Anzahl der Fehler, Fehlerfunktionen, ...
  - Statistische Zusammenhänge zwischen Test und Verlässlichkeit.
- F3: Ergebniskontrollen
  - Informationsredundanz, Format- und Wertekontrollen.
- F4: Statische Tests
  - Direkte Kontrollen auf Fehlerabwesenheit.
- F5: Dynamische Tests
  - Funktionskontrolle mit Beispieleingaben.
- F6: Problembeseitigung

Inhalt Foliensatz 1 (TV-F1): Grundbegriffe, Wahrscheinlichkeit, Experimente

Grundbegriffe

1.1 Modell
1.2 Service-Modell
1.3 Fehlfunktionen und Fehler
1.4 Potentielle und Modellfehler

Wahrscheinlichkeit

2.1 Verkettete Ereignisse
2.2 Fehlerbaumanalyse
2.3 Markov-Ketten

Zufallsexperimente

3.1 Service als Zufallsexperiment
3.2 Verfügbarkeit
3.3 Zuverlässigkeit
3.4 Sicherheit
3.5 Kontrollen
3.6 Test
3.7 Zufallstest
3.8 Fehleranteil, Entst.-Proz.
3.9 Reifeprozesse

Grundbegriffe

Modell

Der Begriff »Modell« in der Informatik

Selbst die einfachsten Sachverhalte in der Informatik wie die Abarbeitung eines Befehls werden sehr schnell kompliziert, wenn alle Details berücksichtigt werden.

Definition 2

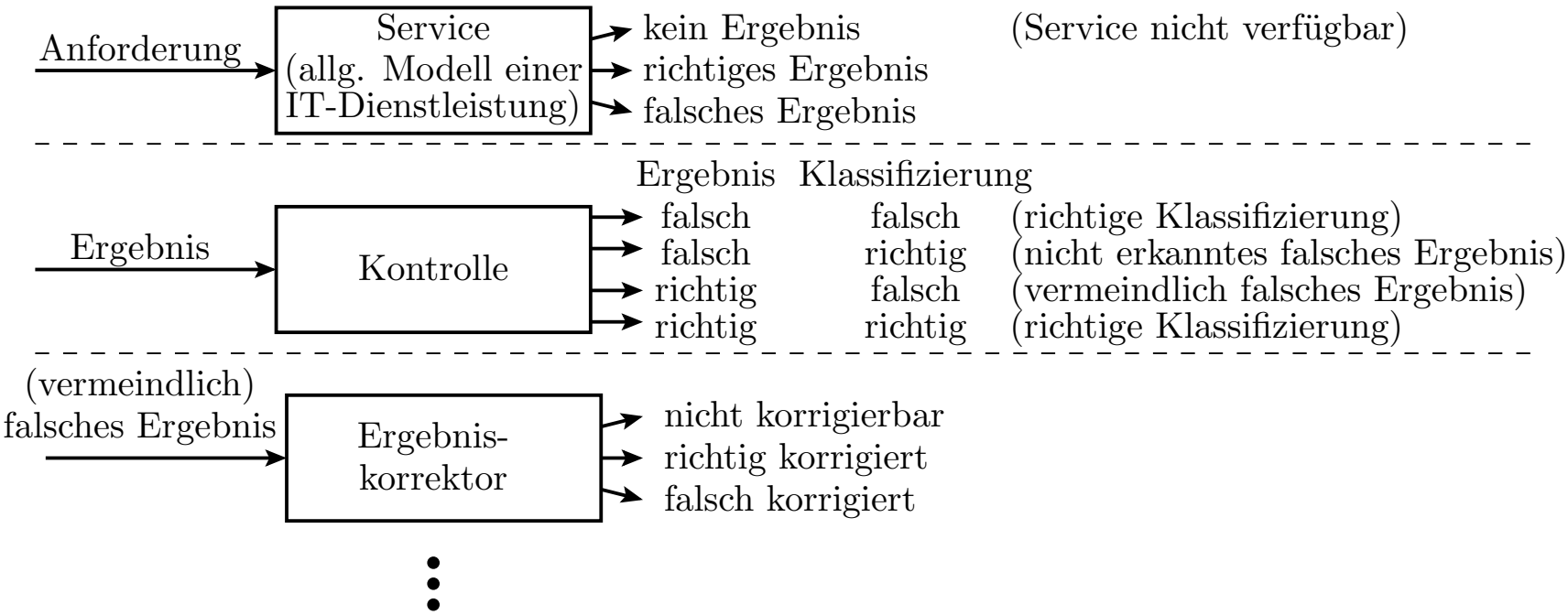
Ein Modell ist ein Mittel, um einen Zusammenhang zu veranschaulichen. Es stellt die wesentlichen Sachverhalte dar und verbirgt unwesentliche Details.

In dieser Vorlesung sind die Modelle Zufallsexperimente für

- das Funktionieren und Versagen von IT-Systemen,
- für Kontrollen, Tests, Reviews, Fehler, Ausfälle, ... ,
- Fehlervermeidung, Fehlerbeseitigung und Schadensbegrenzung.

für Systeme aus Hardware und/oder Software [+ Mechanik].

Modelle zur Beschreibung der Verlässlichkeit



- Unwesentlich sind tatsächliche Funktionen und Ergebnisse.
- Wesentlich ist die Unterscheidung zwischen gewünschten und unerwünschten Ergebnissen, ...

Service-Modell

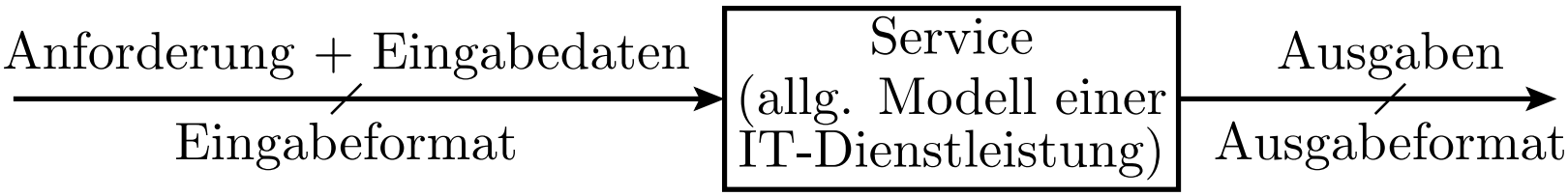
Das Service-Modell

Definition 3

Eine Service ist ein Vorgang, der mit einer Service-Anforderung gestartet wird und aus Eingaben Ergebnisse erzeugt.

Die Eingabe und die Ergebnisse sind Daten, bei Software Programmdaten, bei Hardware auch Werte elektrischer Signale und bei CP-Systemen[3] auch Werte anderer physikalischer Größen. Das Format hängt von der Art des Systems ab und legt die Struktur und Bedeutung der Daten festlegt. Die Berechnung hat idealerweise eine Sollfunktion und optional weitere Vorgaben z.B. eine max. Ausführungszeit.

[3] CPS (Cyber-Physical Systems) sind informationsverarbeitende Systeme mit Fähigkeiten zur Interaktion in der physikalischen Welt (Radhakisan Baheti and Helen Gill: Cyber - physical Systems. http://ieeecss.org/sites/ieeecss.org/files/documents/IoCT-Part3-02 CyberphysicalSystems.pdf)



Das Service-Modell eignet sich für alle Systeme mit gerichtetem Verarbeitungsfluss:

- Digitalschaltungen: Gatter, Rechnerbausteine, Rechner, ...
- Programme: Einzelanweisung, Module, Server-Dienste, ...
- Systeme aus Hard- und Software: programmierte Rechner, ...
- CPS (Cyber-Physical Systems): Smartphones, Motorsteuergeräte, ...

Hervorgehobene wesentliche Aspekte:

- Zählbare Menge von Anforderungen/Leistungen.
- Klassifikation in erbrachte und nicht erbrachte Leitungen.
- Klassifikation erbrachter Leistungen in »gut« oder »schlecht

Programmmodul als Service

Funktion mit EVA-Struktur:

```
int UP(int a, int b){
    return 23*(a+b);
};
```

- Ein- und Ausgabeformat siehe Funktionsdeklaration.
- Service-Anforderung durch Funktionsaufruf.

Die Beispielfunktion

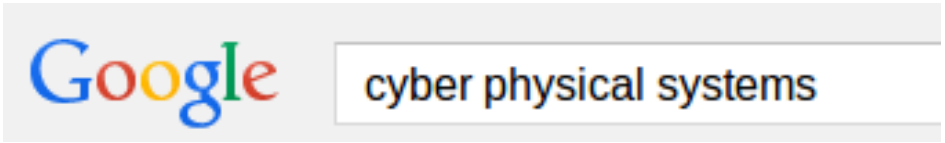
- hat kein Gedächtnis (siehe später),
- sollte bei jeder Anforderung nach wenigen Maschinentakten ein Ergebnis liefern.

Bei Sollfunktion »23*(a+b)« ist das Programm fehlerhaft:

1. Für welche Eingaben wird ein falsches Ergebnis berechnet?
2. Wie lässt sich der Fehler beseitigen?

Server-Dienst als Service

- Beispiel einer Service-Anforderung:



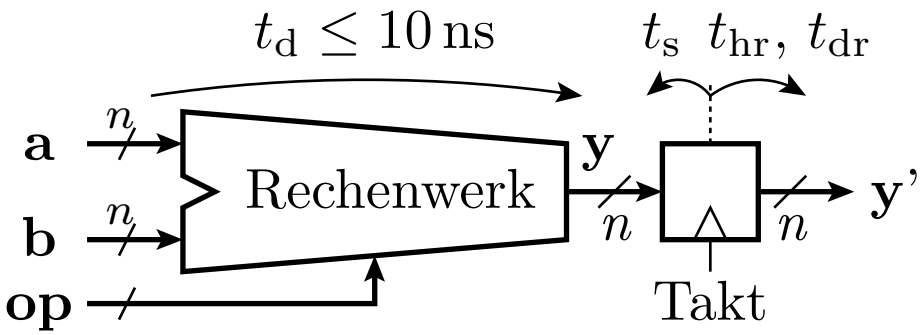
- Ausschnitt aus dem gelieferten Ergebnis

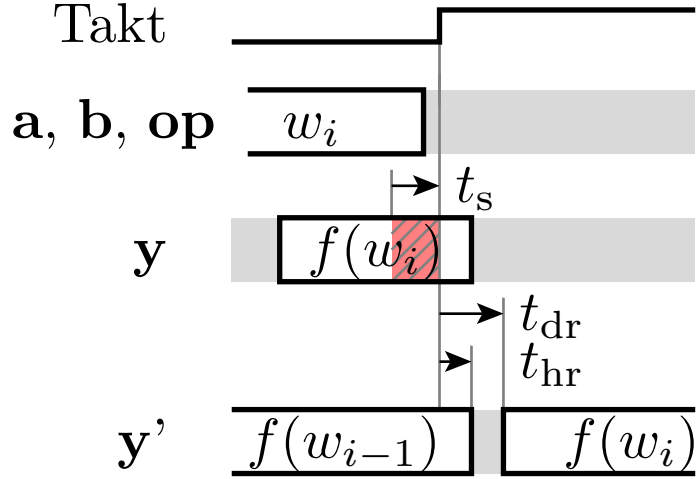
cyber physical systems

Webdefinitionen

Ein cyber-physisches System bezeichnet den Verbund informatischer, softwaretechnischer Komponenten mit mechanischen und elektronischen Teilen, die über eine Dateninfrastruktur, wie z. B. das Internet, kommunizieren. ...
http://de.wikipedia.org/wiki/Cyber_Physical_Systems

Digitalschaltung als Service





a, **b**	Operanden
op	Operationscode
y	Ergebnis
n	Bitbreite
t_d	Verzögerungszeit

t_s	Vorhaltezeit
t_{dr}	Registerverzögerung
t_{hr}	Registerhaltezeit
	Abtastfenster

- Service-Anforderung ist eine Eingabeänderung.
- Eingabeformat: ausreichend lange stabil anliegende Eingabewerte.
- Ergebnisformat: im Abtastfenster stabile Ergebniswerte.

CP-Systeme als Service-Anbieter

Motorsteuergerät:

- Service-Anforderung für jede Schrittzeit.
- Eingabe: Sensordaten vom Motor, Benutzereingaben, Nachrichten anderer Steuergeräte, ...
- Ausgaben: Stellwerte für Aktoren (Schalter, Ventile, ...), Benutzerausgaben, Nachrichten an andere Steuergeräte.

Ein Service kann auch komplette physikalische Vorgänge »kapseln«.
Beispiel »Waschvorgang einer Waschmaschine«:

- Service-Anforderung: Start des Waschvorgangs.
- Eingabe: Programmauswahl, Wäsche bzw. deren Beschreibung durch Daten.
- Ergebnis: Bewertungsdaten der Service-Leistung, z.B. »Sauberkeit« der Wäsche.

Entstehungsprozesse als Service

Fehler als wesentliche Ursache für Fehlfunktionen von Service-Leistungen entstehen mit dem System:

- Entwurfsfehler während der Entwurfsprozesse,
- Fertigungsfehler während der Fertigungsprozesse.

Die Entstehungsprozesse sind auch als Service modellierbar:

- Anforderung: Entwicklungs- oder Produktionsauftrag.
- Eingaben: Entwicklungs- oder Produktionsvorgaben (+ Ressourcen, Material, ... bzw. deren Beschreibungsdaten).
- Ergebnis: Entwurfsbeschreibung oder Produkt (bzw. dessen Bewertungsdaten).

Fehlfunktionen in einem Entstehungs-Service sind auf der nächsten Modellierungsebene Fehler im entstandenen Service.

Klassifizierung von Service-Leistungen

Für die Modellierung der Verlässlichkeit von Service-Leistungen sind wesentlich[4], dass:

- Ergebnisse als gut, schlecht oder nicht erbracht klassifiziert werden können,
- potentielle und tatsächliche Fehler gezählt werden können.
- Abhängigkeiten zwischen Entstehung/Nachweis/... unterschiedlicher Fehler und Fehlfunktionen, ... darstellbar sind.

Weitere wesentliche Eigenschaften für Maßnahmen zur Sicherung der Verlässlichkeit sind:

- Determinismus und
- Gedächtnis.

[4] Unwesentliche Details für die Modellierung der Verlässlichkeit sind die Funktion, die physikalische Realisierung (nur Software, HW+SW, ...), ...

Determinismus

Ein Service arbeitet deterministisch, wenn er immer gleich ausgeführt wird und für gleiche Eingaben (und gespeicherte Zustände) gleiche Ergebnisse liefert. Determinismus ist Voraussetzung für:

- die Definition einer Sollfunktion,
- Ergebniskontrolle durch Soll-/Ist-Vergleich,
- die Suche von Tests für den Fehlerausschluss,
- Reparaturkontrolle durch Testwiederholung,
- Fehlerlokalisierung durch Rückverfolgung, ...

- In SW und digitalen Schaltungen hat beobachtbarer Nichtdeterminismus meist Fehler oder Störungen als Ursache.
- Weitere Ursachen für Nichtdeterminismus bei analoger Verarbeitung und CP-Systemen: Quantisierung, Alterung, ...

Service-Leistungen ohne und mit Gedächtnis

Ein deterministischer Service ohne Gedächtnis realisiert im math. Sinne eine Funktion:

$$y = f(x)$$

die jedem zulässigen Eingabewert x genau einen Ausgabewert y zuordnet.

Ein deterministischer Service mit Gedächtnis ist im math. Sinne ein Automat mit einem Zustand s, einer Übergangsfunktion Funktion

$$s = f_{\mathrm{s}}(s, x)$$

und einer Ausgabefunktion:

$$y = f_{\mathrm{y}}(s, x)$$

(x – Eingabe; y – Ausgabe).

Service-Leistungen ohne und mit Gedächtnis gibt es für jede Realisierungsart (SW, HW, CPS, ...):

	ohne Gedächtnis	mit Gedächtnis
Programm-bausteine	Unterprogramme ohne private Daten	OOP-Methoden zur Objektbearbeitung.
Programm	Compiler	Textverarbeitung
Server-Dienst	Ohne Nutzung fremder Daten.	Datenbankanfrage
digitale Schaltung	Rechenwerk	Prozessor
CP-System	Maschine, die aus Vorgaben Werkstücke herstellt	Steuergerät, das sich Daten merkt

Eine Gesamstsystem ohne Gedächtnis kann auch Teilsysteme mit Gedächnis nutzen (z.B. ein Server-Dienst den Server).

Der Preis für das Gedächtnis

Fehler können außer der Ausgabe auch gespeicherte Daten kontaminieren (verfälschen). Das bedeutet für

- die Verlässlichkeit: Eine Zustandskontaminierung erhöht die Häufigkeit von Fehlfunktionen bis zur nächsten Neuinitialisierung.
- den Test: Tests beginnen mit einer Initialisierung und müssen immer insgesamt wiederholt werden.
- die Fehlerlokalisierung: Rückverfolgung verfälschter Daten über Zeitebenen. Erfordert Wiederholbarkeit oder Trace-Aufzeichnung aller Zwischenergebnisse.

Test- und Fehlerlokalisierungprobleme durch ein Gedächtnis lassen sich durch Steuer- und Beobachtbarkeit gespeicherter Daten vermeiden. Test wie »ohne Gedächtnis«.

Service-Hierarchie

IT-Systeme sind hierarchisch aufgebaut:

- Client-Server-Systeme bestehen aus Rechnern und Netzwerkkomponenten.
- Rechner, Netzwerkkomponenten, ... bestehen aus Hard- und Software.
- Software besteht aus Programmbausteinen, diese sind aus Programmieranweisungen zusammengesetzt, die ihrerseits mit Maschinenbefehlen nachgebildet werden.
- Maschinenbefehle sind Service-Leistungen der Hardware. Die Hardware besteht aus Funktionsbausteinen, diese meist aus Gattern und diese wiederum aus Transistoren.

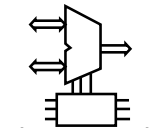
Hierarchie der Hardware

Geräte

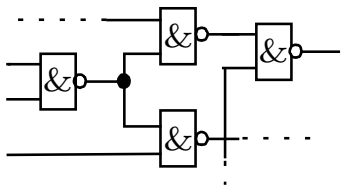
Baugruppen

Schaltkreise

Funktionsblöcke



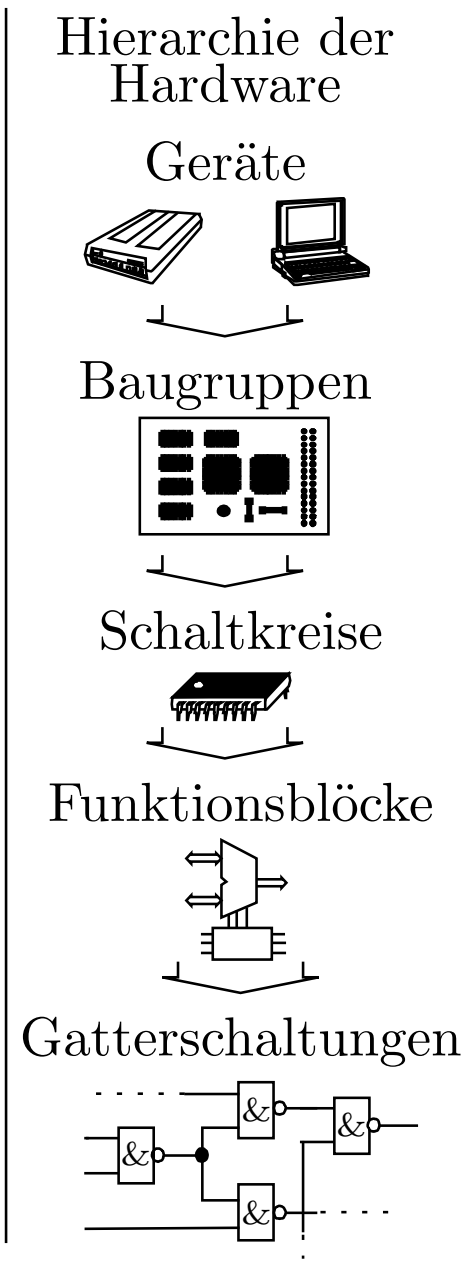
Gatterschaltungen



Im Service-Modell

- stellen die Transistoren elementare Schaltfunktionen bereit (ein/aus) mit denen Gatterfunktionen und Speicherelemente gebildet werden.
- Mit Gattern und Speicherelementen werden komplexere Funktionseinheiten wie Rechenwerke, Register bis hin zu kompletten Rechnern nachgebildet.
- Die Software nutzt Hardware-Funktionen, ...

Ein IT-System funktioniert korrekt, wenn alle Service-Leistungen hierarchisch absteigend korrekt arbeiten und der Informationsfluss dazwischen korrekt abläuft.



Fehlfunktionen und Fehler

Fehlfunktionen, Fehler, Störungen und Ausfälle

Fehlfunktionen (FF) sind erkennbare Abweichungen vom spezifizierten Sollverhalten.

Die Ursachen (root cause) für Fehlfunktionen können sein:

- Fehler:
  - wirken ständig,
  - sind durch Reparatur oder Ersatz beseitigbar,
  - entstehen im Entwurfs- oder Fertigungsprozess,
  - sind durch Beseitigung ihrer Entstehungsursache vermeidbar.
- Störungen:
  - spontane, nicht reproduzierbare Wirkung,
  - vermeidbar durch Verringerung der Störanfälligkeit.
- Ausfälle:
  - während des Betriebs entstehende Fehler.

Fehler in Systemen ohne Gedächtnis

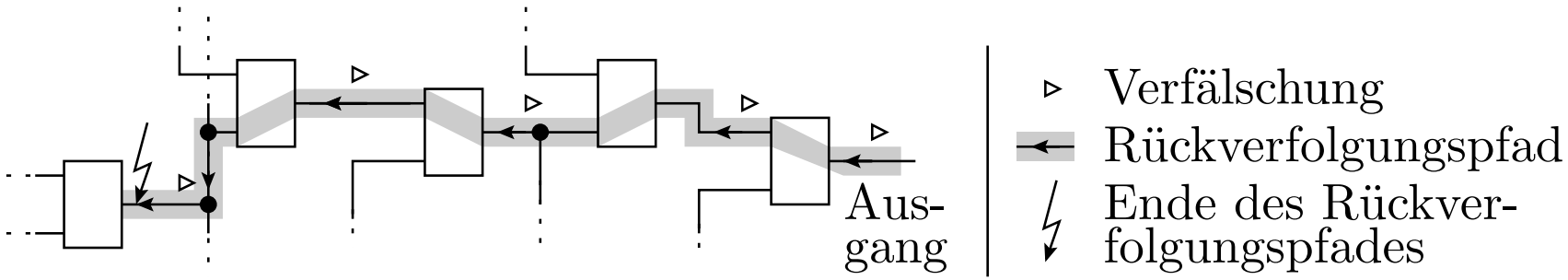
Fehler + beeinträchtigte Service-Anforderung ⇒ Fehlfunktion

Fehlernachweis:

- Service-Anforderung mit fehlernachweisenden Eingaben.
- Kontrolle der Ausgabe.

Fehlerlokalisierung und Reparatur:

- Suche der untersten Teil-Service-Leistung oder Kommunikation, die mit korrekten Daten falsch ausgeführt wird.
- Ersatz, Reparatur, ... der lokalisierten Teil-Service-Leistung.
- Erfolgskontrolle durch wiederholte Service-Anforderung.



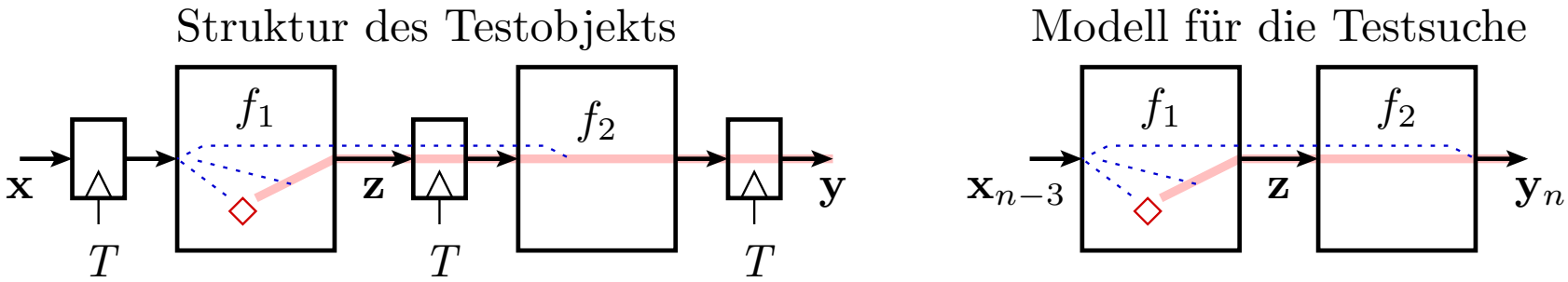
Fehler in Systemen mit Gedächtnis

Fehler + Service-Anforderung ⇒ kontaminierter Zustand
kontaminierter Zustand+ Service-Anforderung ⇒ Fehlfunktion

Testsuche und Fehlerlokalisierung erfolgen »gedanklich« durch »Aufrollen« (pseudo-kombinatorisches Iterationsmodell).

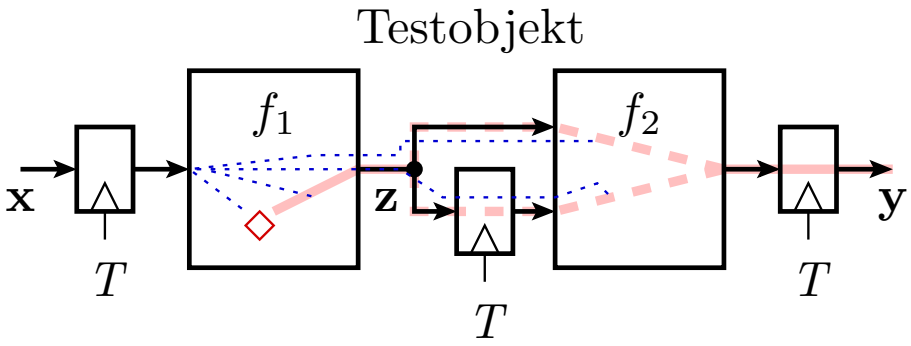
Problemklassen:

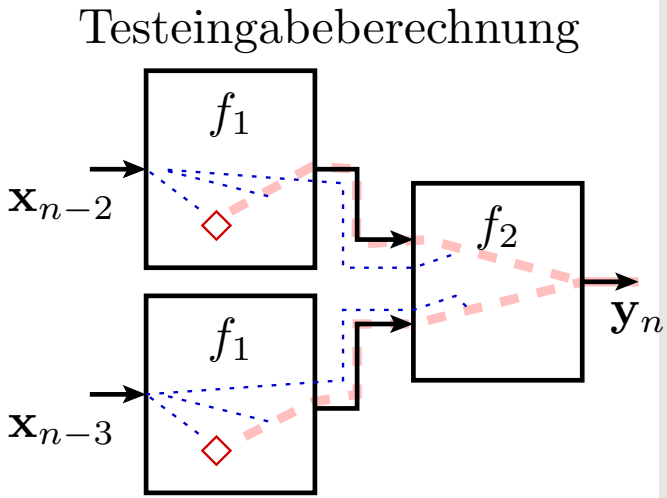
- Datenspeicherung zur Weitergabe an den nächsten Service.



- Testsuche und Fehlerlokalisierung wie ohne Gedächtnis.
- Hinzu kommt ein Zeitversatz zwischen Ein- und Ausgabe.

- Weiterverarbeitung der Ergebnisse mehrerer Zeitebenen.



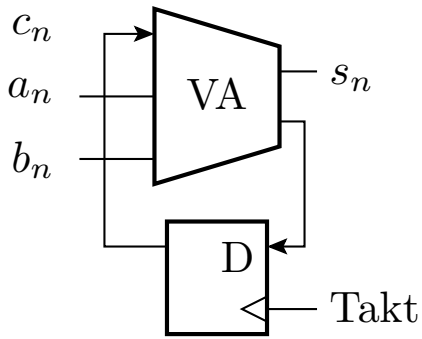


  - Das aufgerollte System enthält mehrere Kopien desselben Service.
  - Die Fehler sind in jeder Kopie.
  - Jedes Testbeispiel besteht aus einer Folge von mehreren Service-Anforderungen.

Im Prinzip jedoch mit denselben Techniken und Mitteln wie für Systeme ohne Gedächtnis lösbar.

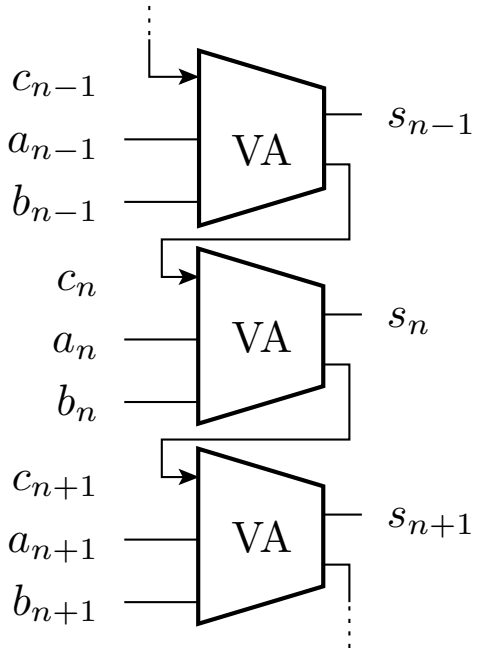
- System mit Rückführung

serieller Addierer



VA - Volladdierer

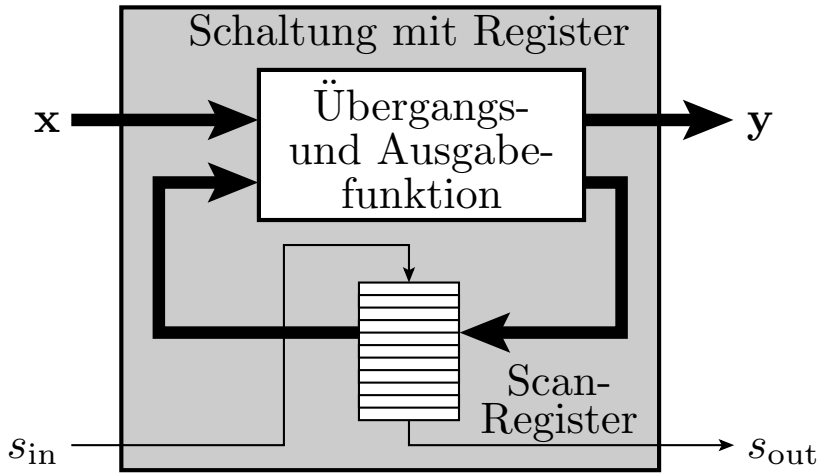
aufgerollter Addierer

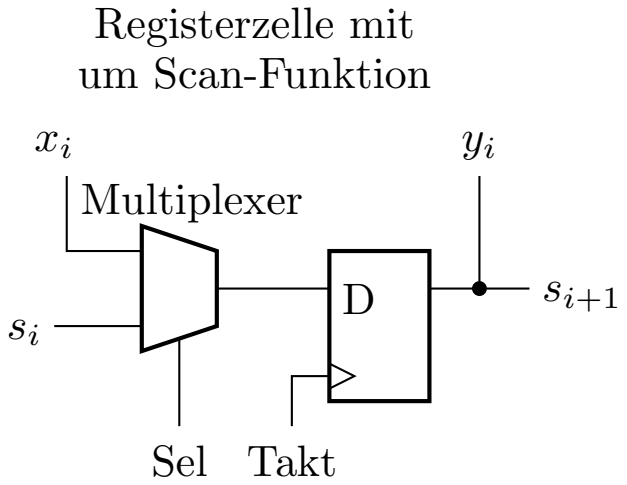


- Aufgerolltes System aus unbegrenzt vielen Kopien.
- Regeln zur Begrenzung der Länge der Steuer- und Beobachtungspfade erforderlich.
- Vermeidung durch Steuerung und Beobachtung der internen Zustände.

Steuerung und Beobachtung interner Zustände

- (Normaler) Debugger: Schrittbetrieb, Unterbrechungspunkte mit Lese- und Schreibmöglichkeit für interne Daten.
- Post Mortem Debugger: Core-Dump-Erzeugung nach System-Crash, Trace-Aufzeichung von Zwischenergebnissen.
- Baugruppen: Test mit Nadeladaptern, ...
- Schaltkreise: (Boundary-) Scan, BIST (Build-In Self-Test), ...





(s_{in} – serieller Eingang, s_{out} – serieller Ausgang, Sel – Auswahl).

Potentielle und Modellfehler

Potentielle Fehler

Für die Modellierung der Verlässlichkeit wird eine abzählbare Menge potentieller Fehler benötigt, von denen jeder

- mit Häufigkeitkeit $p_{i.\text{vorh}}$ vorhanden,
- bei einer Service-Anforderung mit einer Wahrscheinlichkeit $p_{i.\text{nachw}}$ ein Versagen verursacht und
- lokalisier- und beseitigbar ist.

Eine pragmatische, aber nicht perfekte Definition:

Definition 4

Potentielle Fehler seien alle Teil-Service-Leistungen und Kommunikationswege, die falsch ausgeführt werden können, ohne dass innerhalb von ihnen eine genauere Lokalisierung möglich ist.

- Potentielle Fehler sind zähl-, lokalisier- und beseitigbar. $\surd$
- Schätzbare Auftrittshäufigkeitkeit $p_{i.\text{vorh}}$. $\surd$

Die Fehlerhierarchie folgt der Service-Hierarchie

Ebene / Sicht	potenzielle Fehler
Transistorebene	jeder Transistor, jede Verbindung
Gatterebene	jedes Gatter, jede Verbindung
Baugruppe / Hersteller	jedes Bauteil, jede Verbindung
SW / Programmierer	jede Code-Zeile, jede Bibliotheksfunktion
SW / Anwender	jedes austauschbare Programm, im Extremfall nur das Gesamtprogramm
...	...

Schwachstellen des gewählten Modells für potenzielle Fehler:

- Kein simulierbares Fehlverhalten.
- Verhaltensabhängige Wahrscheinlichkeit $p_{i.\mathrm{nachw}}$.

Pareto-Prinzip

Für Fehler und Fehlfunktionen gilt das Pareto-Prinzip[5]:

»20% der Ursachen erzielen 80% der Wirkungen.«

20% und 80% sind Stellvertreterwerte für »kleiner Anteil der Ursachen« und »großer Anteil der Wirkungen«.

- Die meisten Fehler gehen auf wenige Ursachen zurück.
- Ein kleiner Anteil der Fehler verursacht den überwiegenden Anteil der Fehlfunktionen.

Fehlervermeidung während der Entstehung, Fehlerbeseitigung nach der Entstehung, Schadensbegrenzung und Ergebniskorrektur sollten sich vorrangig auf den kleinen Teil der Ursachen konzentrieren, die den großen Teil der Probleme verursachen.

[5] Vilfredo Pareto untersuchte die Verteilung des Bodenbesitzes in Italien. Er fand heraus, dass ca. 20% der Bevölkerung ca. 80% des Bodens besitzen und leitete daraus das Pareto-Prinzip ab.

Modellfehler und Fehlermodell

Für die Bewertung von Testsätzen und die geziele Suche von Testbeispielen wird eine Fehlermenge mit simulierbarem Verhalten und bestimmbarem $p_{i.\text{vorh}}$[6] benötigt.

Definition 5

Ein Modellfehler ist eine Beispielfehler mit exakt simulierbarem Fehlverhalten.

Beispiele für Modellfehler:

- Setze Signal auf ständig null / ständig eins.
- Setze Sprungbedingung auf ständig wahr / ständig falsch.
- Verfälsche Zwischenergebnisse +1 / -1.

Ein Fehlermodell ist ein Algorithmus für die Berechnung einer Menge von Modellfehlern.

[6] $p_{i.\text{vorh}}$ – Wahrscheinlichkeit, dass ein potenzieller Fehler i, wenn er vorhanden ist, ein Service-Versagen verursacht und daran nachweisbar ist.

Deterministisches und unbeständiges Fehlverhalten

Fehler in deterministischen Systemen haben meist determinis- tische Wirkung, d.h.

- Ein Service ohne Gedächtnis versagt mit denselben Eingaben immer in derselben Weise.
- Ein Service mit Gedächtnis versagt bei gleicher Initialisierung, derselben Eingabefolge immer in derselben Weise.

Es gibt jedoch auch Ursachen für unbeständiges Fehlverhalten:

- unbeabsichtigte Abhängigkeiten von anderen Daten, Speicherbelegungen, physikalischen Größen, ...
- Störungen, ...

Ein unbeständiges Fehlverhalten verbietet Fehlerlokalisierung durch Rückverfolgung, ... Vor Fehlerlokalisierungen Beständigkeit des Fehlverhaltens kontrollieren.

Störungen und Ausfälle

Wirkung ungewollter physikalischer Einflüsse (Rauschen, Strah- lung, ...). Datenverfälschung oder Zerstörung von Teilsystemen.

Definitionen 6

Störung: zufällige Datenverfälschung ohne lokalisierbare Ursache.
Ausfall: Ein Fehler, der während des Einsatzes entsteht.

- Störungen sind nicht wie Fehler beseitigbar, sondern nur in ihrer Häufigkeit minderbar. Kompliziertere Lokalisierungs- und Vermeidungstechniken als für Fehler.
- Ausfälle sind erst nach dem Ausfallereignis nachweisbar. Sprunghafte Erhöhung der Häufigkeit der Fehlfunktionen[7]. Erfordert Wartungstests und Reparaturen im Einsatz.

[7] Die Zunahme der Fehlfunktionen durch einen Ausfall kann je nach entstandenem Fehler unerheblich bis funktionsbeeinträchtigend sein.

Kontrollfragen

1. Wie lauten die drei Ebenen zur Sicherung der Verlässlichkeit?
2. Was unterscheidet Modellfehler von potenziellen Fehlern?
3. Was besagt das Pareto-Prinzip für Fehler und Fehlfunktionen?
4. Sind die potentiellen Fehler in einem im Rechner eingebauten Schaltkreis eine Teilmenge der potenziellen Fehler des Rechners?
5. Nennen Sie Gründe, warum IT-Systeme vorzugsweise deterministisch arbeiten sollten?
6. Unter welchen Bedingungen verursacht ein Modellfehler ein unbeständiges Fehlverhalten? Nennen Sie Beispiele.
7. Wozu dienen die Einschalttests, die im allg. jeder Rechner nach einem Neustart zuerst ausführt?

Antwort auf die Kontrollfragen 1 und 2

Die drei Ebenen zur Sicherung der Verlässlichkeit:

- Fehlervermeidung während der Entstehungsprozesse,
- Test und Fehlerbeseitigung vor und während des Einsatzes und
- während des Einsatzes Kontrollen mit Schadesbegrenzung und Ergebniskorrektur bei erkannten Fehlfunktionen.

Was unterscheidet Modellfehler von potenziellen Fehlern?

- Modellfehler haben ein exakt simulierbares Verhalten.
- Ein potenzieller Fehler, z.B. ein defektes Gatter kann unterschiedliche Wirkungen haben.

Antwort auf die Kontrollfragen 3 und 4

Was besagt das Pareto-Prinzip für Fehler und Fehlfunktionen?

- Die Mehrheit der Fehlfunktionen werden durch einen kleinen Anteil der Fehler verursacht.
- Die meisten Fehler werden durch einen kleinen Teil der möglichen Ursachen im Entstehungsprozess verursacht.

Sind die potentiellen Fehler in einem im Rechner eingebauten Schaltkreis eine Teilmenge der potenziellen Fehler des Rechners?

- Auf der Betrachtungsebene Rechner ist der Schaltkreis insgesamt ein potenzieller Fehler, innerhalb von dem keine detailliertere Lokalisierung angestrebt wird.
- Auf der Betrachtungsebene Schaltkreis hat der Schaltkreis viele unterschiedliche Bestandteile, die als Ursache eines Fehlverhaltens lokalisiert werden können. Im Rechner bilden diese zusammen den potenziellen Fehler »Schaltkreis defekt«.

Antwort auf die Kontrollfragen 5 bis 7

Nennen Sie Gründe, warum IT-Systeme vorzugsweise deterministisch arbeiten sollten?

- Test: Konstruktion von Eingaben, bei denen immer dasselbe Fehlverhalten beobachtbar ist.
- Reparatur: Erfolgskontrolle durch Wiederholung.
- Lokalisierung: Nachträgliche Berechnung von Zwischenwerten.

Unter welchen Bedingungen verursacht ein Modellfehler ein unbeständiges Fehlverhalten? Nennen Sie Beispiele.

- Verursachung von zusätzlichem Speicherverhalten.
- Fehlerbedingte Abhängigkeit von Daten fremder Programme.

Wozu dienen die Einschalttests, die im allg. jeder Rechner nach einem Neustart zuerst ausführt?

- Kontrolle auf Ausfälle.

Wahrscheinlichkeit

Bewertung der Verlässlichkeit

Die einzelnen Aspekte der Verlässlichkeit

- ist das System verfügbar,
- sind die gelieferten Ergebnisse richtig,
- entsteht kein unkalkulierbarer Schaden, ...

sollen mit Wahrscheinlichkeiten bewertet werden. Die Basis für die Definition von Wahrscheinlichkeiten sind Zufallsexperimente.

Zufallsexperiment

Definition 7

Ein Zufallsexperiment ist ein Experiment mit mehreren möglichen Ergebnissen und zufälligem Ausgang.

Zufallsexperimente zu Test und Verlässlichkeit {Wertebereich[8]}:

- Anforderung einer Service-Leistung {richtig, falsch, ...}.
- Ergebniskontrolle: {richtig, falsch, ...}.
- Korrektur falscher Ergebnisse: {erfolgreich, ...}.
- Zählen der Fehler in einem System {0, 1, 2, ... }.
- Aufdecken eines Fehlers mit einem Test {ja, nein}.
- Messen der Zeit bis zum Ausfall: {Zeit größer null}.
- ...

[8] Wertebereich der möglichen Ergebnisse des Experiments

Bernoulli-Versuch

Das einfachste Zufallsexperiment ist der Bernoulli-Versuch. Er hat zwei mögliche Ergebnisse 0/1 (nein/ja, falsch/wahr, ...) und die Verteilung

$$\begin{aligned} P\{X=0\} &= 1-p \\ P\{X=1\} &= p \end{aligned}$$

(p – Wahrscheinlichkeit, dass das Ergebnis 1, ja oder wahr ist).
Bernoulli-Versuche für Aspekte der Verlässlichkeit:

- Anforderung einer Service-Leistung {richtig, falsch}.
- Aufdecken eines Fehlers mit einem Test {ja, nein}.
- ...

In der Vorlesung werden fast alle statistisch untersuchten Zusammenhänge auf Bernoulli-Versuche zurückgeführt, z.B. die Fehleranzahl als Summe potentieller Fehler, ob vorhanden ...

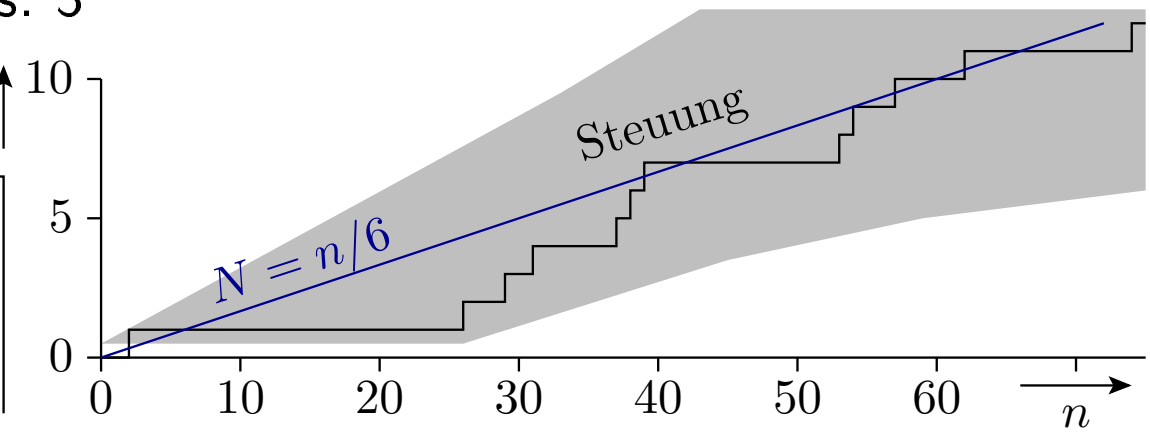
Die Wahrscheinlichkeit von Zufallsexperimenten

Definition 8

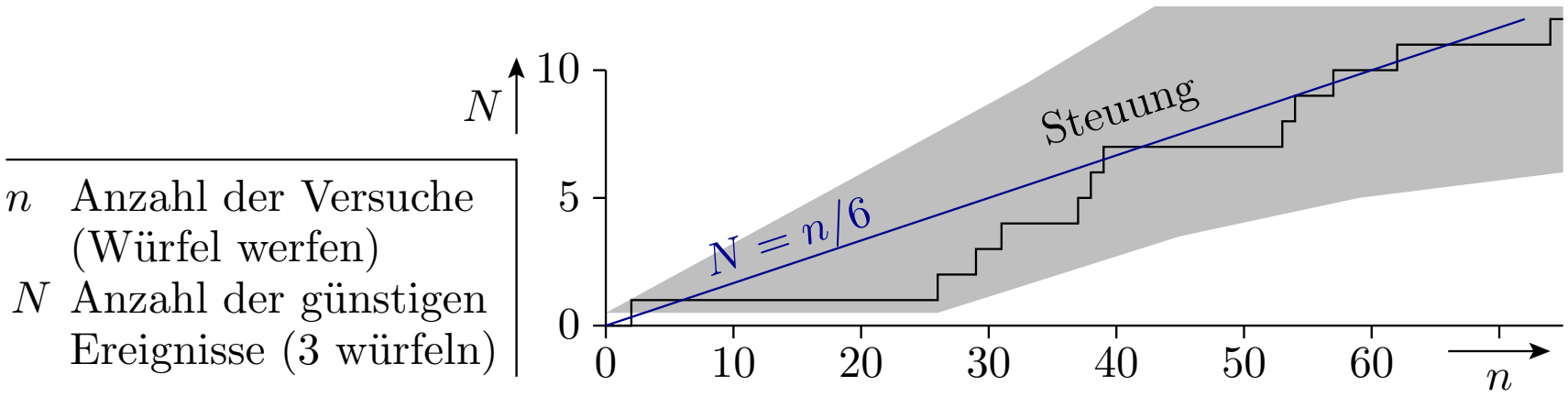
Wahrscheinlichkeit ist das Verhältnis, gegen das bei einem Zufallsexperiment die Anzahl der »günstigen« zur Anzahl aller möglichen Ereignisse mit zunehmender Versuchsanzahl strebt.

Wahrscheinlichkeit, dass eine 3 gewürfelt wird.

- Zufallsexperiment: Würfeln
- Mögliche Ergebnisse: 1, 2, ..., 6
- günstiges Ergebnis: 3



n Anzahl der Versuche (Würfel werfen)
N Anzahl der günstigen Ereignisse (3 würfeln)



Beim Würfeln wird davon ausgegangen, das alle 6 Möglichkeiten gleichwahrscheinlich sind. Mit Versuchsanzahl $n \to \infty$ strebt das Verhältnis aus günstigen Ergebnissen N zur Versuchsanzahl gegen das Verhältnis aus möglichen günstigen und möglichen Ereignissen:

$$p = \lim_{n\to\infty} \left(\frac{N}{n} \right) = \frac{1}{6}$$

Das bedeutet aber keineswegs, dass bei jedem sechsten Versuch eine 3 gewürfelt wird. Es ist durchaus zu beobachten, dass hintereinander mehrere Male die Drei und auch mal lange Zeit keine Drei gewürfelt wird.

Aufteilen und verketten von Experimenten

Zufallsexperimente lassen sich u.U. in mehrere Teilexperimente aufteilen oder mehrere unabhängige Experimente zu einem zusammenfassen. Im nachfolgenden wird bei jedem Experiment zweimal gewürfelt (Ereignisse A und B, Wertebereich jeweils $\{1, 2, \ldots, 6\}$. Daraus werden mit Vergleichsoperatoren die zweiwertigen Ereignisse C und D gebildet und diese einmal UND- und einmal ODER verknüpft und gezählt.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	... 20	... 40
A	6	1	5	4	1	1	2	2	4	6	4	3	1	6	5
B	6	5	6	2	1	3	3	6	4	5	1	3	1	4	3
$C = (A > 3)$	1	0	1	1	0	0	0	0	1	1	1	0	0	1	1
$D = (B < 3)$	0	0	0	1	1	0	0	0	0	0	1	0	1	0	0
$E = (C \wedge D)$	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
$F = (C \vee D)$	1	0	1	1	1	0	0	0	1	1	1	0	1	1	1
$\sum C$	1	1	2	3	3	3	3	3	4	5	6	6	6	11	21
$\sum D$	0	0	0	1	2	2	2	2	2	2	3	3	4	6	9
$\sum E$	0	0	0	1	1	1	1	1	1	1	2	2	2	5	6
$\sum F$	1	1	2	3	4	4	4	4	5	6	7	7	8	13	24

Nach 40 Versuchen betragen die Schätzwerte der Wahrscheinlichkeiten als Verhältnis der günstigen Ergebnisse, dass die Bedingungen C bis F erfüllt sind, zur Versuchsanzahl:

Ereignis	Schätzwert	Wahrscheinlichkeit
$C = (A > 3)$	$21/40 = 53\%$	$3/6 = 50\%$
$D = (B < 3)$	$9/40 = 23\%$	$2/6 = 33\%$
$E = (C \wedge D)$	$6/40 = 15\%$	$6/36 = 17\%$
$F = (C \vee D)$	$24/40 = 60\%$	$24/36 = 67\%$

Die Wahrscheinlichkeit als Grenzwerte für $n \to \infty$ ergibt sich für jeden Versuch aus dem Verhältnis der günstigen zur Anzahl der möglichen Ergebnisse. Die Würfelexperimente haben 6 mögliche Ergebnisse. Davon sind für die Ereignisse C und D 3 bzw. 2 günstig. Die verketten Ereignisse E und F haben $6^2 = 36$ mögliche Ergebnisse, von denen 6 bzw. 24 günstig sind.

Die Schätzung einer Wahrscheinlichkeit mit weniger als 100 günstigen Ereignissen ist recht ungenau.

Bedingte Wahrscheinlichkeiten

Bei einer bedingten Wahrscheinlichkeit werden nur die Versuche und Ereignisse gezählt, die die Bedingung erfüllen. Beispiel sei die ODER-Verknüpfung sich ausschließender Ereignisse:

$$E = C \vee D \text{ unter der Bedingung } C \wedge D = 0.$$

																				$\sum$	$\sum$
n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	20	16
C	1	0	1	1	0	0	0	0	1	1	1	0	0	1	1	0	1	0	1	1	11	7
D	0	0	0	1	1	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	6	2
$C \vee D$	1	0	1	1	1	0	0	0	1	1	1	0	1	1	1	0	1	0	1	1	13	9

nicht mitgezählte Ereignisse bzw. Summe ohne diese Ereignisse

Sowohl die Anzahl der gezählten Versuche als auch die günstigen Ergebnisse verringern sich um die vier nicht mitzuzählenden Ergebnisse mit $C \wedge D = 1$. Das undokumentierte Aussortieren ungewollter Ergebnisse ist ein unauffälliger und beliebter Trick, Statistiken zu fälschen[9].

[9] Traue nie einer Statistik, die du nicht selbst gefälscht hast.

Verkettete Ereignisse

Wahrscheinlichkeit verketteter Ereignisse

- Wahrscheinlichkeit, dass ein Ereignis A nicht eintritt:

$$P\left(\bar{A}\right) = 1 - P\left(A\right) \tag{1}$$

- Wahrscheinlichkeit, dass von zwei unabhängigen Ereignissen A und B beide eintreten:

$$P\left(A \wedge B\right) = P\left(A\right) \cdot P\left(B\right) \tag{2}$$

- Wahrscheinlichkeit, dass von zwei unabhängigen Ereignissen mindestens eines eintritt:

$$\begin{aligned} P\left(A \vee B\right) = P\left(\overline{\bar{A} \wedge \bar{B}}\right) &= 1 - \left(1 - P\left(A\right)\right) \cdot \left(1 - P\left(B\right)\right) \\ &= P\left(A\right) + P\left(B\right) - P\left(A\right) \cdot P\left(B\right) \end{aligned} \tag{3}$$

Beispielaufgabe

In einem System mit drei Fehlern seien diese unabhängig voneinander mit den Nachweiswahrscheinlichkeiten $p_1 = 10\%$, $p_2 = 5\%$ und $p_3 = 20\%$ nachweisbar. Wie groß sind die Wahrscheinlichkeiten der verketten Ereignisse, dass

E_1 : alle Fehler,
E_2 : kein Fehler,
E_3 : mindestens ein Fehler und
E_4 : genau zwei Fehler nachgewiesen werden?

Lösung: Definition von Ereignissen F_i für Fehler i nachweisbar und Beschreibung von E_i durch logische Verknüpfungen:

- Alle Fehler nachweisbar:

$$\begin{aligned} E_1 &= F_1 \wedge F_2 \wedge F_3 \\ P(E_1) &= P(F_1) \cdot P(F_2) \cdot P(F_3) \\ &= p_1 \cdot p_2 \cdot p_3 = 10\% \cdot 5\% \cdot 20\% = 0{,}1\% \end{aligned}$$

- Kein Fehler nachweisbar:

$$\begin{aligned} E_2 &= \overline{F_1 \vee F_2 \vee F_3} \\ P(E_2) &= 1 - (1 - (1 - P(F_1)) \cdot (1 - P(F_2)) \cdot (1 - P(F_2))) \\ &= (1 - p_1) \cdot (1 - p_2) \cdot (1 - p_3) = 90\% \cdot 95\% \cdot 80\% = 68{,}4\% \end{aligned}$$

- Mindestens ein (nicht kein) Fehler nachweisbar:

$$\begin{aligned} E_3 &= \bar{E}_2 \\ P(E_3) &= 1 - P(E_2) = 1 - 68{,}4\% = 31{,}6\% \end{aligned}$$

- Genau 2 Fehler werden nachgewiesen, wenn
  - die ersten beiden und der dritte nicht,
  - die zweiten beiden und der erste nicht oder
  - der erste und der dritte, aber nicht der zweite

  nachgewiesen werden (ausschließendes ODER, nächste Folie):

$$\begin{aligned} E_4 &= \left(F_1 \wedge F_2 \wedge \bar{F}_3\right) \vee \left(\bar{F}_1 \wedge F_2 \wedge F_3\right) \vee \left(F_1 \wedge \bar{F}_2 \wedge F_3\right) \\ P(E_4) &= p_1 \cdot p_2 \cdot (1 - p_3) + (1 - p_1) \cdot p_2 \cdot p_3 + p_1 \cdot (1 - p_2) \cdot p_3 \\ &= 90\% \cdot 5\% \cdot 20\% + 10\% \cdot 95\% \cdot 20\% + 10\% \cdot 5\% \cdot 80\% = 6{,}8\% \end{aligned}$$

Abhängige Ereignisse

Fakt 9

Ein Ereignis B ist von einem Ereignis A abhängig, wenn das Eintreten von A die Eintrittswahrscheinlichkeit von B beeinflusst.

Für sich ausschließende Ereignisse ist die Wahrscheinlichkeit für das gleichzeitige Eintreten

$$P(A \wedge B) = 0 \tag{4}$$

und für das Eintreten des einen oder des anderen Ereignisses:

$$P(A \vee B)|_{P(A \wedge B)=0} = P(A) + P(B) \tag{5}$$

Für abhängige, aber sich nicht ausschließende Ereignisse ist das Experiment so umformulieren, dass die UND oder ODER zu verknüpfenden Teilereignisse danach entweder unabhängig sind oder sich gegenseitig ausschließen.

Beispielaufgabe »abhängiger Fehlernachweis«

Wie groß sind die Wahrscheinlichkeiten, dass von zwei Fehlern im System 0, 1 oder 2 Fehler nachweisbar sind, wenn die Nachweiswahrscheinlichkeit für Fehler 1 unabhängig vom Nachweis von Fehler 2 $p_1 = 10\%$ beträgt und für Fehler 2 bei Nachweis von Fehler 1 $p_2 = 20\%$ und sonst 0 beträgt. (Der Nachweis des zweiten Fehler hängt vom Nachweis des ersten ab.)

Lösung: Definition von Ereignissen F_i für Fehler i nachweisbar und E_i für i Fehler nachweisbar.

- Kein Fehler ist nachweisbar, wenn der erste Fehler nicht nachweisbar ist[10]:

$$\begin{aligned} E_0 &= \bar{F}_1 \\ P(E_0) &= 1 - P(F_1) = 1 - p_1 = 1 - 10\% = 90\% \end{aligned}$$

[10] Der Fall, Nachweis des zweiten ohne den ersten Fehler ist ausgeschlossen.

- Ein Fehler ist nachweisbar, wenn der erste Fehler nachweisbar ist und der zweite nicht:

$$\begin{aligned} E_1 &= F_1 \wedge \bar{F}_2 \\ P(E_1) &= p_1 \cdot (1 - p_2) = 10\% \cdot 80\% = 8\% \end{aligned}$$

- Zwei Fehler sind nachweisbar, wenn beide Fehler nachweisbar sind:

$$\begin{aligned} E_2 &= F_1 \wedge F_2 \\ P(E_2) &= p_1 \cdot p_2 = 10\% \cdot 20\% = 2\% \end{aligned}$$

- Probe: Summe der Wahrscheinlichkeiten aller möglichen Ergebnisse muss immer 100% sein:

$$P(E_0) + P(E_1) + P(E_2) = 90\% + 2\% + 8\% = 100\%\surd$$

Beispiel »Bedatungswahrscheinlichkeit«

Wie groß ist die Wahrscheinlichkeit, dass ein 8-Bit-Vektor für eine Service-Anfrage an eine Schaltung mit dem Wert $\mathbf{x} =$ "11111110" bedatet wird, wenn

1. unabhängig voneinander für jedes Bit mit einer Wahrscheinlichkeit[11] von $g = 50\%$ zufällig eine Eins und sonst eine Null gewählt wird.
2. Dasselbe wie im Aufgabenteil zuvor, nur mit $g = 60\%$.
3. Dasselbe wie in den Aufgabenteilen zuvor, nur dass für die höchstwertigen vier Bits immer derselben Zufallswert ausgewählt wird.

[11] Die Wahrscheinlichkeit g wird auch als Wichtung der Bitstelle bezeichnet. Bitweise Wichtung wird beim Test digitaler Schaltungen eingesetzt, um die Nachweiswahrscheinlichkeiten sehr schlecht nachweisbarer Fehler zu erhöhen.

Lösung

Definieren von Ereignissen G_i, dass für Bit i eine Eins ausgewählt wird.

- Für die beiden ersten Aufgabenteile gilt:

$$\begin{aligned} \mathbf{x} = \text{"}11111110\text{"} &= G_7 \wedge G_6 \wedge G_5 \wedge G_4 \wedge G_3 \wedge G_2 \wedge G_1 \wedge \bar{G}_0 \\ P\left(\mathbf{x} = \text{"}11111110\text{"}\right) &= g^7 \cdot (1-g) \end{aligned}$$

- Für den letzten Aufgabenteil folgt aus $G_7 = G_6 = G_5 = G_4$:

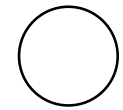
$$\begin{aligned} \mathbf{x} = \text{"}11111110\text{"} &= G_4 \wedge G_3 \wedge G_2 \wedge G_1 \wedge \bar{G}_0 \\ P\left(\mathbf{x} = \text{"}11111110\text{"}\right) &= g^4 \cdot (1-g) \end{aligned}$$

g	50%	60%
G_4 bis G_7 unabhängig	$2^{-8} \approx 0{,}4\%$	$0{,}6^7 \cdot 0{,}4 = 1\%$
$G_7 = G_6 = G_5 = G_4$	$2^{-5} \approx 3\%$	$0{,}6^4 \cdot 0{,}4 = 5\%$

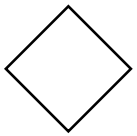
Fehlerbaumanalyse

Fehlerbaumanalyse (FTA – fault tree analysis)

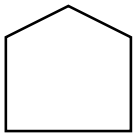
Verfahren zur Abschätzung der Eintrittswahrscheinlichkeit von Problemen in Abhängigkeit vom Eintreten anderer Ereignisse (Gefahrensituationen, Ausfälle, Service-Versagen, ...). Arten von Ereignissen bzw. Problemen:



Problem mit bekannter oder auf anderem Wege abgeschätzter Eintrittswahrscheinlichkeit.



Problem, dessen Eintrittswahrscheinlichkeit nicht untersucht wurde.

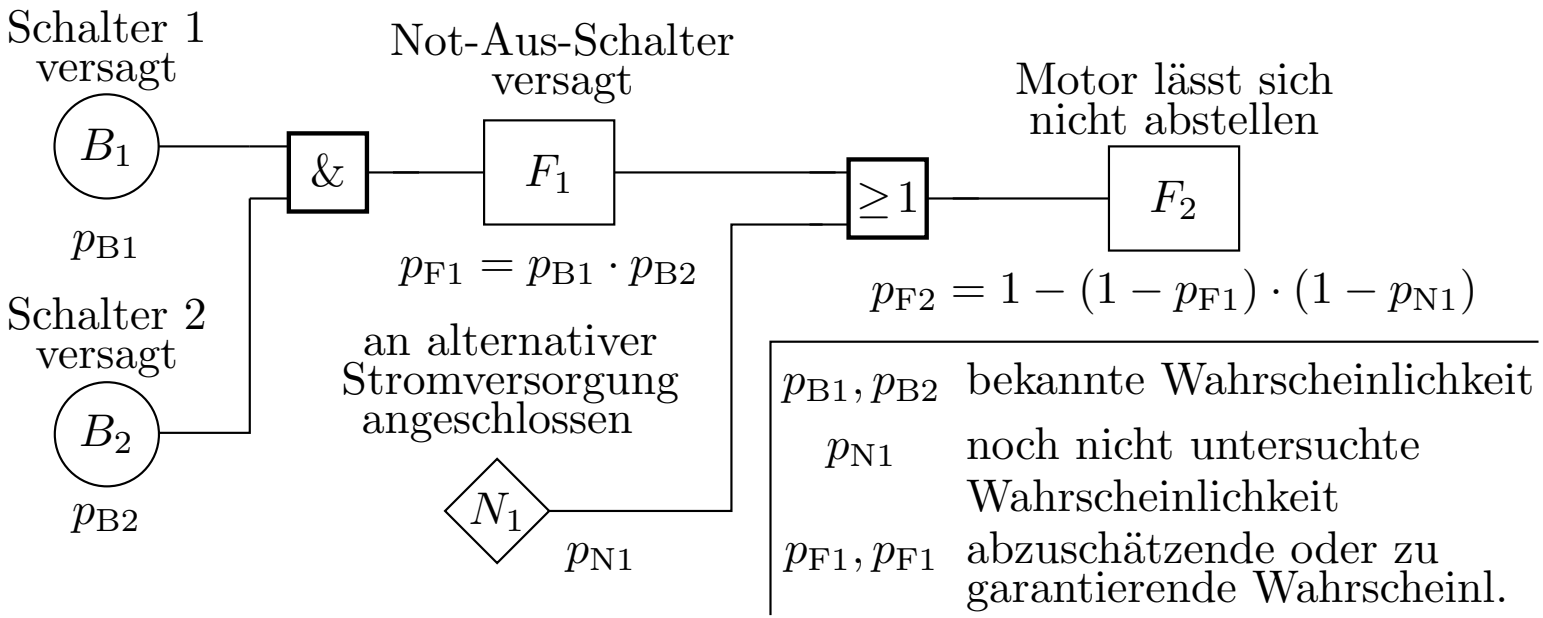


Ereignis im gewöhnlichen Betrieb, das in Kombination mit anderen Probleme verursachen kann.

Problem, dessen Eintrittswahrscheinlichkeit aus denen von ○, ◇ oder ⬠-Ereignissen folgt.

Verknüpfung mit UND, ODER, NICHT.

Beispiel: Motor lässt sich nicht abstellen



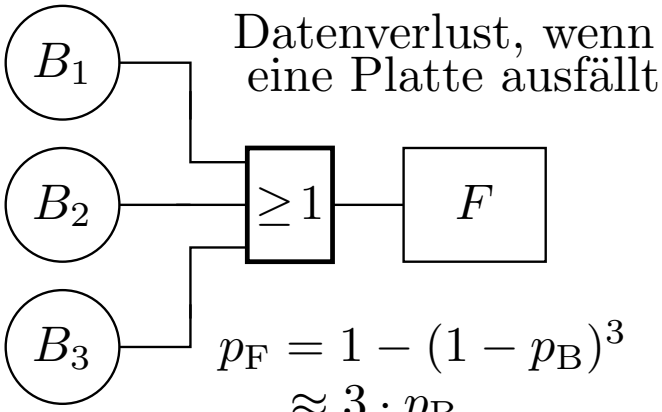
Formulierbare Aufgabe: Wenn $p_{B1} = p_{B2} = 10^{-3}$ ist und $p_{F2} \leq 10^{-6}$ sein darf

- ist dieses Ziel erreichbar?
- Wie groß darf p_{N1} dann maximal sein?

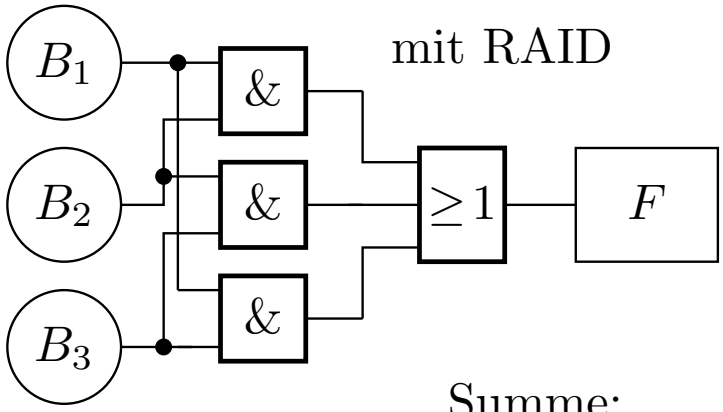
(Ziel hier nur mit $p_{N1} = 0$ erreichbar. Realistisch/andere Lösung?)

Datenverlust mit RAID

Bei einem RAID 3 und RAID 5 tritt nur ein Datenverlust ein, wenn zwei Platten gleichzeitig versagen. Fehlerbaum für $n = 3$ Platten:



$$p_F = 1 - (1 - p_B)^3 \approx 3 \cdot p_B$$



B_3	B_2	B_1	F	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$p_B^2 \cdot (1 - p_B)$
1	0	0	0	
1	0	1	1	$p_B^2 \cdot (1 - p_B)$
1	1	0	1	$p_B^2 \cdot (1 - p_B)$
1	1	1	1	p_B^3

Summe:

$$p_F = 3 \cdot p_B^2 - 2 \cdot p_B^3$$

p_B	Wahrscheinlichkeit Plattenversagen
p_F	Wahrscheinlichkeit Datenverlust

Rekonvergente Auffächerungen

Wenn sich der Bedingungsfluss verzweigt und wieder zusammentrifft, werden zum Teil abhängige Ereignisse verknüpft. Im Beispiel:

$$F = B_1 B_2 \vee B_2 B_3 \vee B_1 B_3$$

haben die ODER-verknüpften UND-Terme jeweils eine gemein- same Variable. Für Wahrscheinlichkeitsabschätzung ungeeignet.

Umstellung in Verknüpfung sich ausschließender Ereignisse:

- disjunktive Normalform:

$$\begin{aligned} F &= B_1 B_2 \bar{B}_3 \vee \bar{B}_1 B_2 B_3 \vee B_1 \bar{B}_2 B_3 \vee B_1 B_2 B_3 \\ p_\mathrm{F} &= p_\mathrm{B}^2 \cdot (1-p_\mathrm{B}) + p_\mathrm{B}^2 \cdot (1-p_\mathrm{B}) + p_\mathrm{B}^2 \cdot (1-p_\mathrm{B}) + p_\mathrm{B}^3 = 3 \cdot p_\mathrm{B}^2 - 2 \cdot p_\mathrm{B}^3 \end{aligned}$$

- Alternative Umstellung:

$$\begin{aligned} F &= B_1 B_2 \vee \bar{B}_1 B_2 B_3 \vee B_1 \bar{B}_2 B_3 \\ p_\mathrm{F} &= p_\mathrm{B}^2 + p_\mathrm{B}^2 \cdot (1-p_\mathrm{B}) + p_\mathrm{B}^2 \cdot (1-p_\mathrm{B}) = 3 \cdot p_\mathrm{B}^2 - 2 \cdot p_\mathrm{B}^3 \end{aligned}$$

Verallgemeinerung auf n Platten

Die Wahrscheinlichkeit, dass mindestens eine von n Platten versagt, ist etwa:

$$p_{\mathrm{F}} \approx n \cdot p_{\mathrm{B}}$$

(p_{B} – Wahrscheinlichkeit, dass eine Platte versagt). Die Wahrscheinlichkeit, dass mindestens zwei Platten versagen, ist eins abzüglich der Wahrscheinlichkeiten, dass null oder eine Platte versagen:

$$p_{\mathrm{F}} \approx 1 - (1 - p_{\mathrm{B}})^n - n \cdot p_{\mathrm{B}} \cdot (1 - p_{\mathrm{B}})^{n-1}$$

Die Anzahl der versagenden Platten ist bei dieser Aufgabenstellung binomialverteilt (siehe später Abschnitt »Verteilungen, Binomialverteilung«).

Zur Geschichte der Fehlerbaumanalyse

- Einführung 1960: Abschlusssicherheitsbewertung von Interkontinantalraketen vom Typ LGM-30 Minuteman.
- Folgejahre: auch für Sicherheitsbewertung kommerzieller Flugzeuge
- ab 70er bis 80er Jahre: Sicherheitsbewertung Atomkraftwerke
- später auch Automobilindustrie und deren Zulieferer.

Beim Einsatz zur Sicherheitsbewertung:

- sind die sicherheitsrelevanten Ereignisse,
- die Basisereignisse und
- deren Wahrscheinlichkeiten

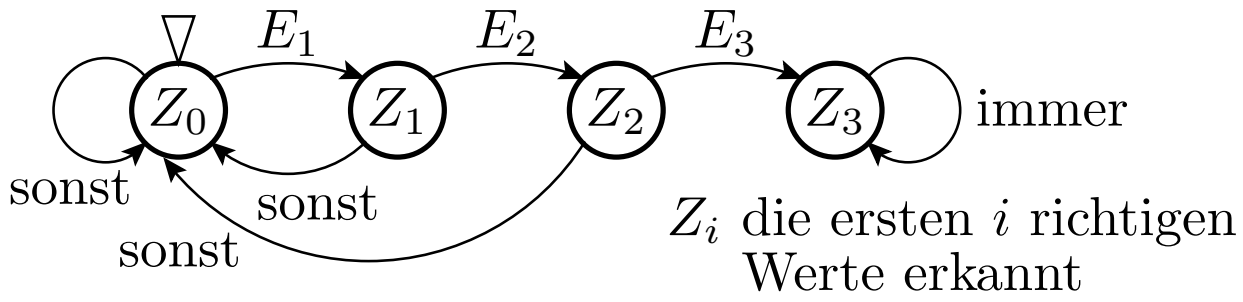
zuvor auf andere Weise abzuschätzen: Vorexperimente, Expertenbefragungen, Ursache-Wirkungs- (Ishikawa-) Diagramm, ...

Markov-Ketten

Markov-Ketten[12]

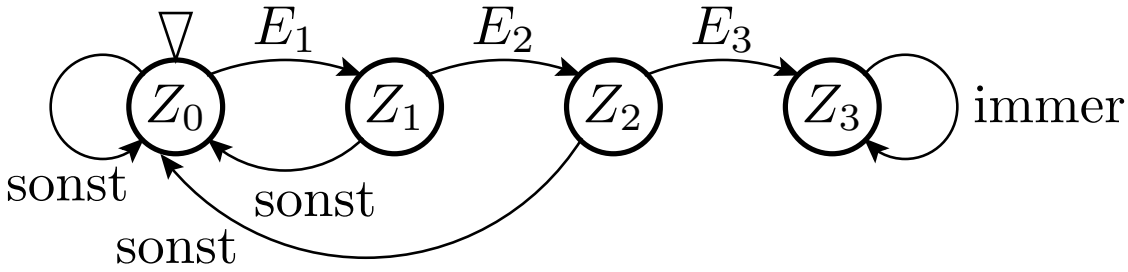
Modellierung eines stochastischen Prozesses durch einen Zustandsautomaten, dessen Kanten mit Übergangswahrscheinlichkeiten beschriftet sind, z.B. zur Bestimmung von Fehlernachweis- und Fehlerbeseitigungswahrscheinlichkeiten.

Fehlernachweis mit einer Eingabefolge $E_1E_2E_3$:

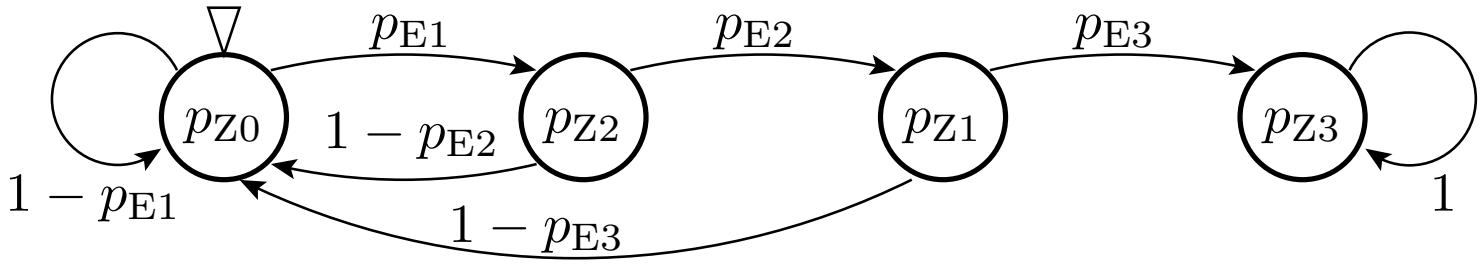


Der Automat startet im Zustand Z_0 »keine richtige Eingabe« und bleibt nach drei richtigen Eingaben im Zustand Z_3 »Fehler nachgewiesen«.

[12] Nach Andrej Andreevič Markov, russischer Mathematiker, 1856-1922.

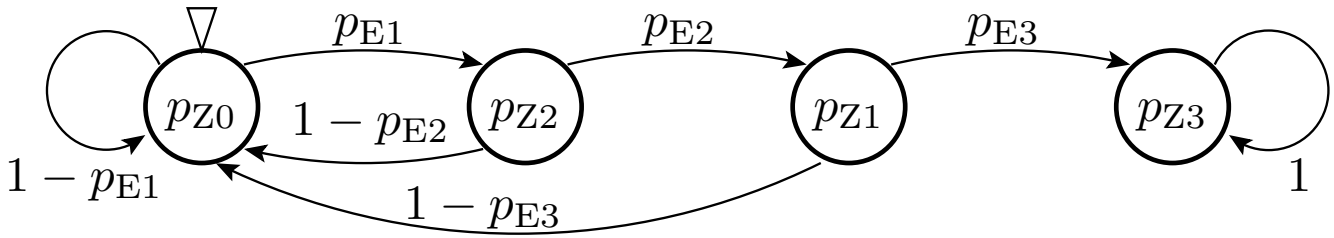


Zur Umwandlung in eine Markov-Kette werden die Übergangsbedingungen durch die Übergangswahrscheinlichkeiten p_{E1} bis p_{E3} und die Zustände durch Zustandswahrscheinlichkeiten $p_{\mathrm{Z}i}$ ersetzt.



Der Anfangszustand hat zu Beginn die Zustandswahrscheinlich- keit $p_{\mathrm{Z0}} = 1$ und die anderen $p_{\mathrm{Z}i} = 0$.

Simulation von Markov-Ketten



Eine Markov-Kette beschreibt ein lineares Gleichungssytem zur Berechnung der Zustandswahrscheinlichkeiten für den Folgeschritt:

$$\begin{pmatrix} p_{Z0} \\ p_{Z1} \\ p_{Z2} \\ p_{Z3} \end{pmatrix}_n = \begin{pmatrix} 1-p_{E1} & 1-p_{E2} & 1-p_{E3} & 0 \\ p_{E1} & 0 & 0 & 0 \\ 0 & p_{E2} & 0 & 0 \\ 0 & 0 & p_{E3} & 1 \end{pmatrix} \cdot \begin{pmatrix} p_{Z0} \\ p_{Z1} \\ p_{Z2} \\ p_{Z3} \end{pmatrix}_{n-1}$$

mit $\begin{pmatrix} p_{Z0} & p_{Z1} & p_{Z2} & p_{Z3} \end{pmatrix}^T = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}$.

$$\begin{pmatrix} p_{\mathrm{Z}0} \\ p_{\mathrm{Z}1} \\ p_{\mathrm{Z}2} \\ p_{\mathrm{Z}3} \end{pmatrix}_n = \begin{pmatrix} 1-p_{\mathrm{E}1} & 1-p_{\mathrm{E}2} & 1-p_{\mathrm{E}3} & 0 \\ p_{\mathrm{E}1} & 0 & 0 & 0 \\ 0 & p_{\mathrm{E}2} & 0 & 0 \\ 0 & 0 & p_{\mathrm{E}3} & 1 \end{pmatrix} \cdot \begin{pmatrix} p_{\mathrm{Z}0} \\ p_{\mathrm{Z}1} \\ p_{\mathrm{Z}2} \\ p_{\mathrm{Z}3} \end{pmatrix}_{n-1}$$

Zur Kontrolle:

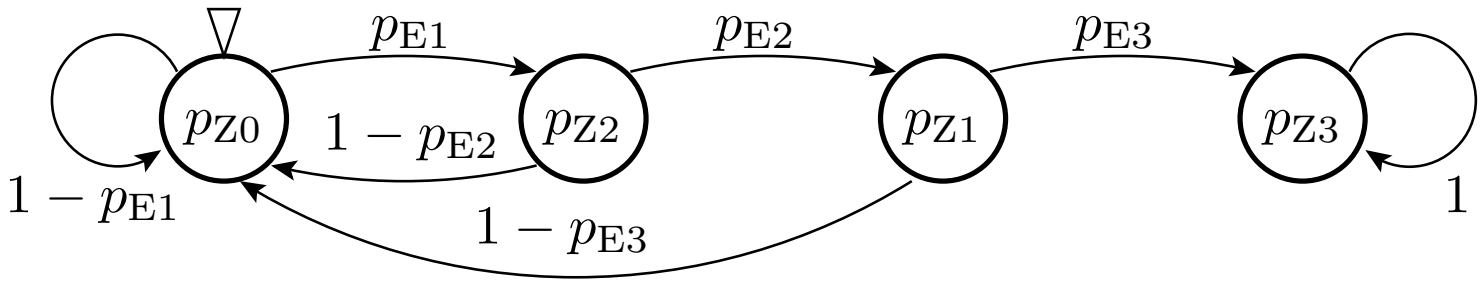
- Die Summe der Wahrscheinlichkeiten in jeder Spalte muss eins sein.
- Die Summe der Zustandswahrscheinlichkeiten $p_{\mathrm{Z}i}$ muss in jedem Schritt eins sein.

$$\begin{pmatrix} p_{Z0} \\ p_{Z1} \\ p_{Z2} \\ p_{Z3} \end{pmatrix}_n = \begin{pmatrix} 1-p_{E1} & 1-p_{E2} & 1-p_{E3} & 0 \\ p_{E1} & 0 & 0 & 0 \\ 0 & p_{E2} & 0 & 0 \\ 0 & 0 & p_{E3} & 1 \end{pmatrix} \cdot \begin{pmatrix} p_{Z0} \\ p_{Z1} \\ p_{Z2} \\ p_{Z3} \end{pmatrix}_{n-1}$$

Simulation mit Octave bzw. Matlab:

```
pE1=...; pE2=...; pE3=...;
M=[1-pE1 1-pE2 1-pE3 0;
    pE1    0     0   0;
     0    pE2    0   0;
     0     0    pE3  1];
Z=[1; 0; 0; 0];
for idx=1:100
 Z = M * Z;
 printf('%3i %6.2f%% %6.2f%% %6.2f%% %6.2f%%\n', ...
          idx, 100*Z);
end;
```

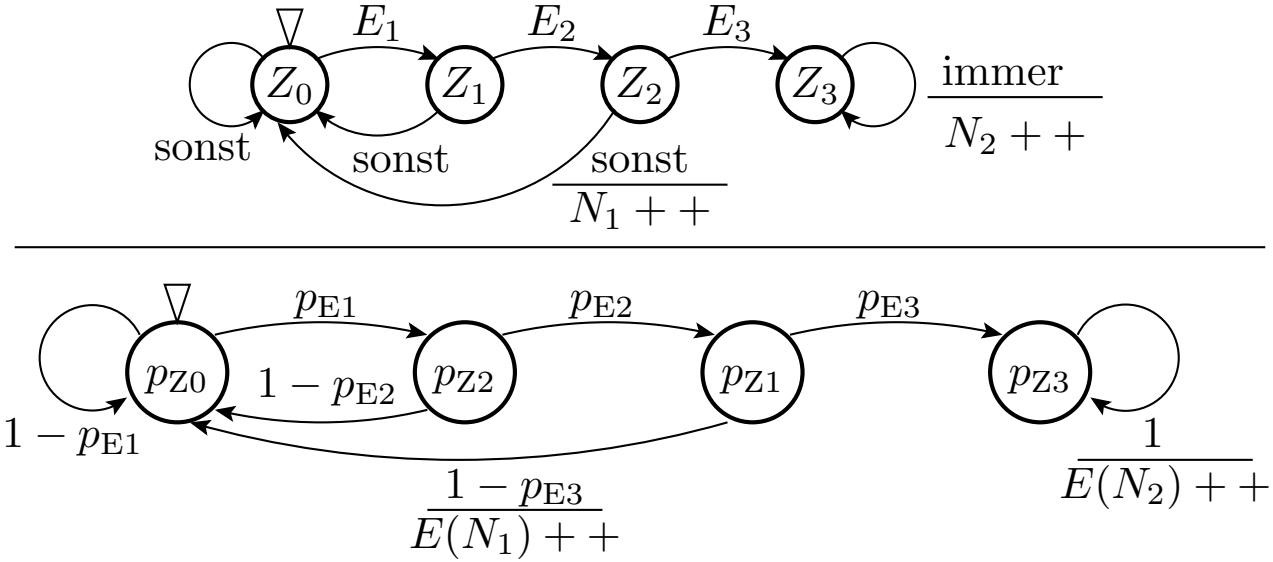
Simulation mit den Beispielwerten $p_{E1} = 30\%$, $p_{E2} = 20\%$ und $p_{E3} = 60\%$:



Schritt	p_{Z0}	p_{Z1}	p_{Z2}	p_{Z3}	Summe
0	100,00	0,00	0,00	0,00	100,00
1	70,00	30,00	0,00	0,00	100,00
2	73,00	21,00	6,00	0,00	100,00
3	70,30	21,90	4,20	3,60	100,00
4	68,41	21,09	4,38	6,12	100,00
...	...	...	...	...	...
10	59,43	18,34	3,77	18,46	100,00
...	...	...	...	...	...
50	19,27	5,95	1,22	73,56	100,00
...	...	...	...	...	...
100	4,73	1,46	0,30	93,53	100,00

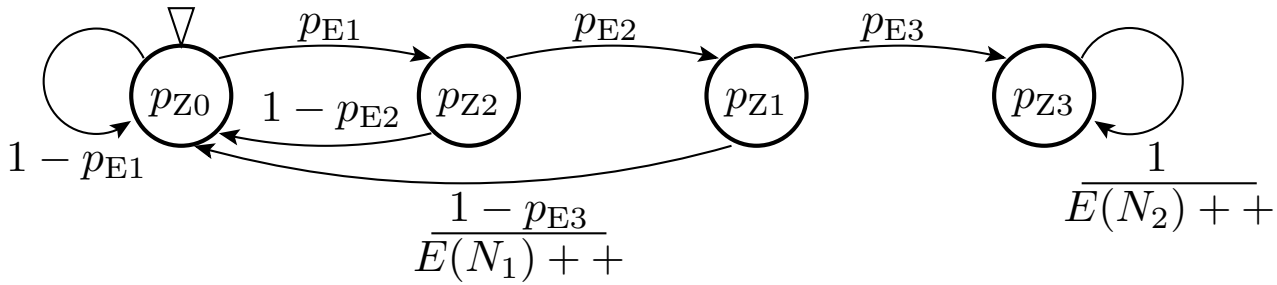
Kantenkosten

Mit Zählern an den Kanten lässt sich zusätzlich die zu erwartende Anzahl der Kantenübergänge bestimmen:



Der Zähler N_1 zählt, wie oft nach zwei richtigen Eingaben eine falsche folgt, der Zähler N_2 die Anzahl der Eingaben im Zustand Z_3 (Fehler nachgewiesen). Die zu erwartende Anzahl der Schritte bis zum Nachweis ist $n - N_2$ (n – Anzahl simulierter Schritte).

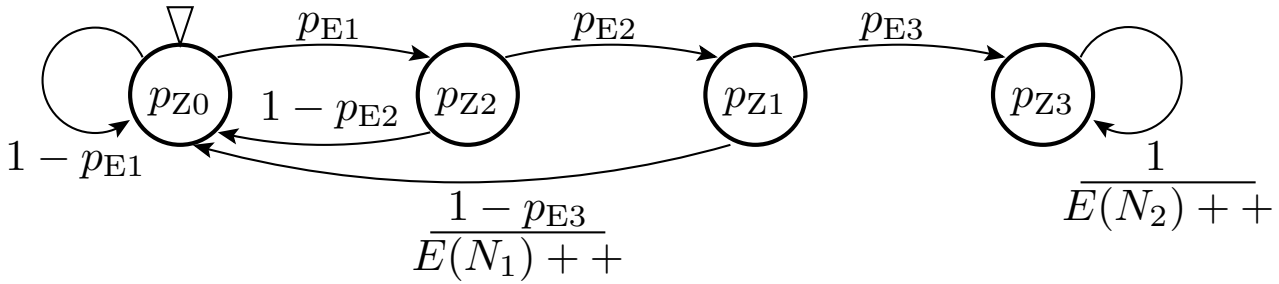
Die korrespondierenden Zähler in der Markov-Kette berechnen die Erwartungswerte der Zählgrößen.



Erweiterung des Simulationsprogramms:

```
...
N1=0; N2=0;
for idx=1:100
 Z = M * Z;
 N1 = N1+Z(3)*(1-pE3);
 N2 = N2+Z(4);
 printf('%3i %6.2f%% %6.2f%% %6.2f%% %6.2f%%', ...
          idx, 100*Z);
 printf(' %6.2f %6.2f\n', N1, N2);
end;
```

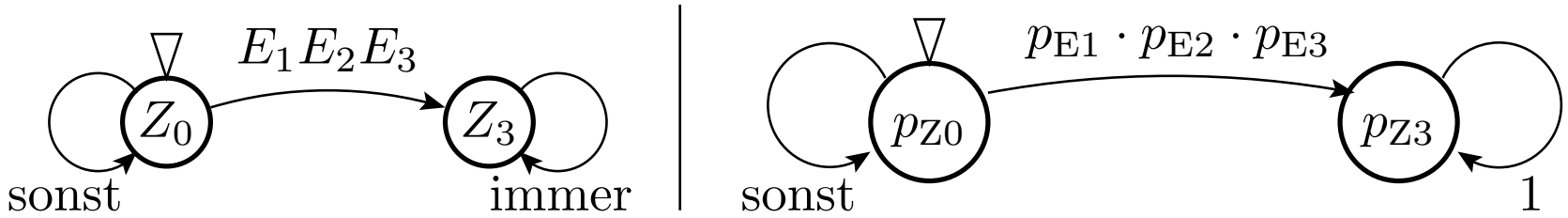
Simulation mit den Beispielwerten $p_{\mathrm{E1}} = 30\%$, $p_{\mathrm{E2}} = 20\%$ und $p_{\mathrm{E3}} = 60\%$:



n	p_{Z0}	p_{Z1}	p_{Z2}	p_{Z3}	$E(N_1)$	$E(N_2)$
1	70,00%	30,00%	0,00%	0,00%	0,00	0,00
2	73,00%	21,00%	6,00%	0,00%	0,02	0,00
3	70,30%	21,90%	4,20%	3,60%	0,04	0,04
4	68,41%	21,09%	4,38%	6,12%	0,06	0,10
...	...	...	...	...	...	...
10	57,78%	17,83%	3,67%	20,73%	0,15	0,99
...	...	...	...	...	...	...
50	18,74%	5,78%	1,19%	74,29%	0,50	22,23
...	...	...	...	...	...	...
100	4,59%	1,42%	0,29%	93,71%	0,63	65,43

- Die mittlere Anzahl $E(N_1)$, dass nach zwei richtigen Eingaben eine falsche folgt, strebt gegen einen Wert kleiner eins.
- Die mittlere Nachweisdauer $n - E(N_2)$ strebt gegen $\approx 100 - 65 = 35$

»Drei richtige Eingaben« als Einzelereignis



Gleichungssystem der modifizierten Markov-Kette:

$$\begin{pmatrix} p_{Z0} \\ p_{Z3} \end{pmatrix}_{n+1} = \begin{pmatrix} 1-p_{E1} \cdot p_{E2} \cdot p_{E3} & 0 \\ p_{E1} \cdot p_{E2} \cdot p_{E3} & 1 \end{pmatrix} \cdot \begin{pmatrix} p_{Z0} \\ p_{Z3} \end{pmatrix}_{n} \text{ mit } \begin{pmatrix} p_{Z0} \\ p_{Z3} \end{pmatrix}_{0} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

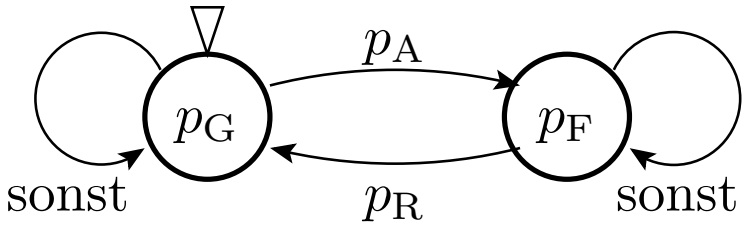
Daraus ablesbar:

$$\begin{aligned} p_{Z0}(n) &= (1 - p_{E1} \cdot p_{E2} \cdot p_{E3}) \cdot p_{Z0}(n-1) = (1 - p_{E1} \cdot p_{E2} \cdot p_{E3})^n \\ p_{Z3}(n) &= 1 - p_{Z0}(n) = 1 - (1 - p_{E1} \cdot p_{E2} \cdot p_{E3})^n \end{aligned}$$

Was ist für $p_{Z0}(n)$ und $p_{Z3}(n)$ im Vergleich zur Markov-Kette mit allen vier Zuständen auf den Folien zuvor zu erwarten?

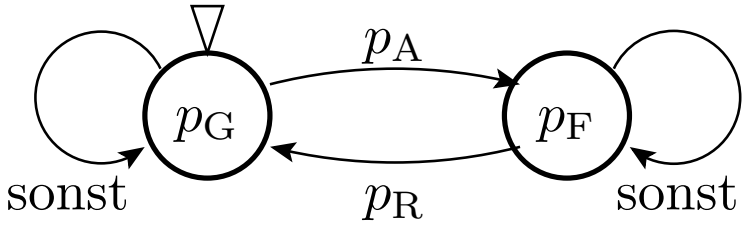
Reparaturprozess als Markov-Kette

Ein System sei zu Beginn funktionsfähig (Zustand G), fällt in jedem Zeitschritt, wenn es ganz ist, mit einer Wahrscheinlichkeit p_A aus (Übergang in Zustand F) und wird, wenn es kaputt ist, mit einer Wahrscheinlichkeit p_R repariert (Übergang in Zustand G):

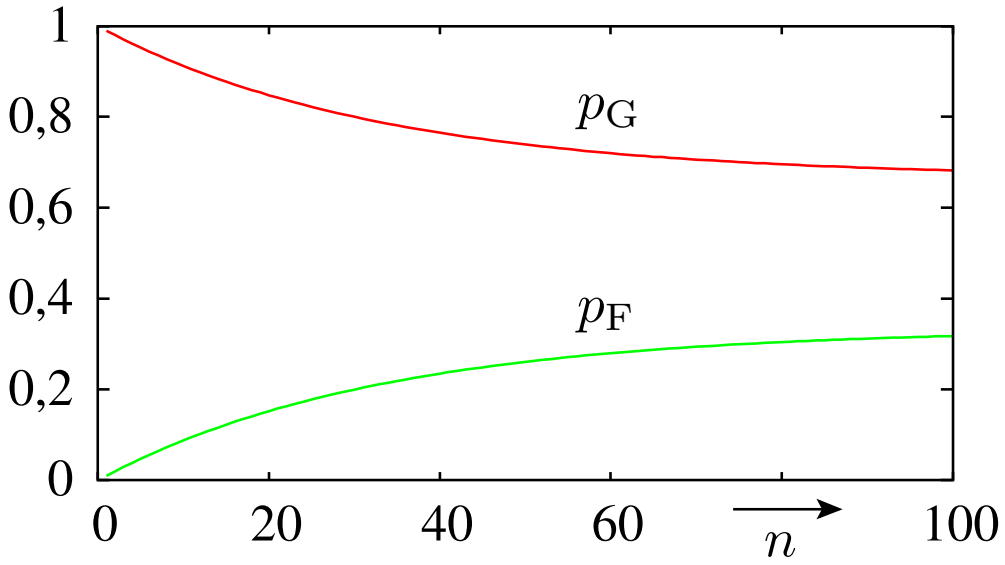


Beschreibung als simulierbares Gleichungssystem:

$$\begin{pmatrix} p_\mathrm{G} \\ p_\mathrm{F} \end{pmatrix}_{n+1} = \begin{pmatrix} 1-p_\mathrm{A} & p_\mathrm{R} \\ p_\mathrm{A} & 1-p_\mathrm{R} \end{pmatrix} \cdot \begin{pmatrix} p_\mathrm{G} \\ p_\mathrm{F} \end{pmatrix}_{n} \text{ mit } \begin{pmatrix} p_\mathrm{G} \\ p_\mathrm{F} \end{pmatrix}_{0} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$



Simulation mit $p_A = 1\%$ und $p_R = 2\%$:

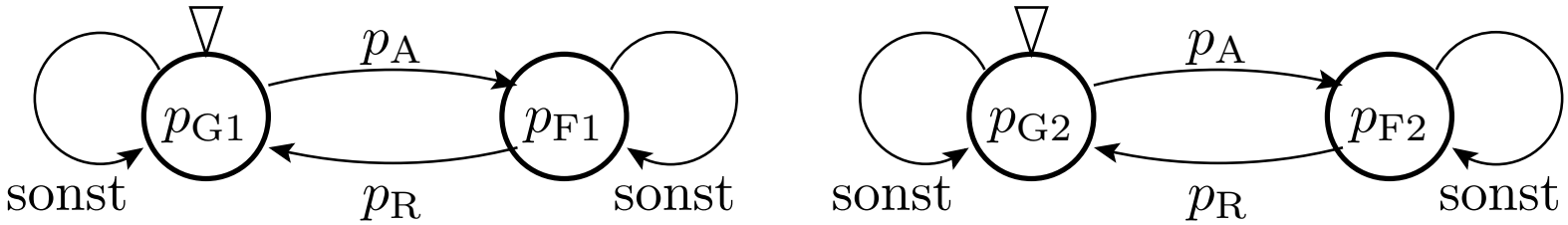


Für große n strebt der Reparaturprozess gegen den stationären Zustand:

$$p_G = \frac{p_R}{p_R + p_A};\ p_F = \frac{p_A}{p_R + p_A}$$

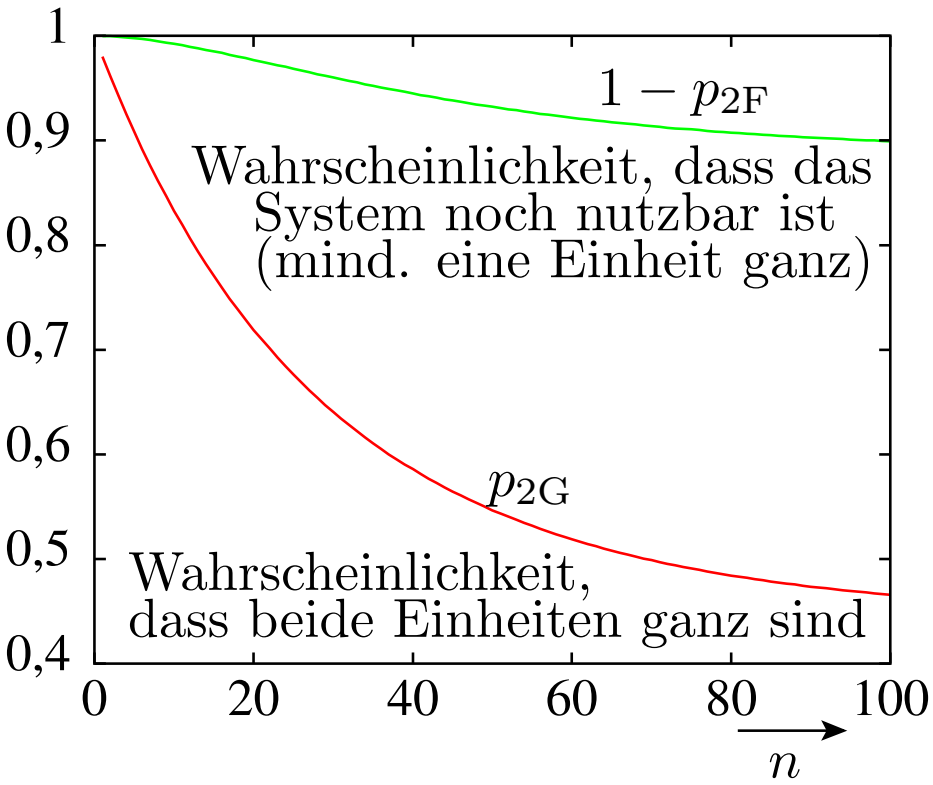
Reparatur mit Redundanz

System aus zwei gleichartigen Teilsystemen, das solange funktioniert, wie ein Teilsystem funktioniert:




```
pA=0.01; pR=0.02;
M=[1-pA  pR; pA  1-pR];
Z=[1; 0];
for idx=1:100
  Z = M * Z;
  p2G(idx)=Z(1)**2; % beide Einheiten ganz
  p2F(idx)=Z(2)**2; % beide Einheiten defekt
end;
plot(1:100, p2G, 1:100, 1-p2F);
```

Simulation mit $p_A = 1\%$ und $p_R = 2\%$:



n Anzahl der Simulationsschritte

Zufallsexperimente

Aspekte und Zählgrößen der Verlässlichkeit

- Verlässlichkeit beschreibt die Abwesenheit von Problemen (Fehler, Fehlfunktionen, Ausfälle, Datenverlust, ...).
- Quantitative Bewertung durch Zeitmessungen und Zählen von Service-Leistungen, Fehlern, Fehlfunktionen, ...

Verlässlichkeitsaspekt	zu zählende Größen
Verfügbarkeit	nicht ausgeführte und ausgeführte SL
Zuverlässigkeit	richtig und falsch ausgeführte SL
Sicherheit	schadensträchtige falsch ausgeführte SL
Kontrolle	erkannte und nicht erkannte FF
Test	erkannte und nicht erkannte Fehler
...	...

(SL – Service-Leistung; FF – Fehlfunktion).

Service als Zufallsexperiment

Service als Zufallsexperiment

Ein Service arbeitet Anfragen ab. Jede Anfrage ist ein Zufallsexperiment mit den möglichen Ergebnissen:

- SK Service korrekt ausgeführt,
- FF Fehlfunktion
- SN Service nicht verfügbar.

und optional einer Ausführungs- bzw. Nichtverfügbarkeitsdauer.

Beispielprotokoll für Service-Ergebnisse:

Nummer	1	2	3	4	5	6	7	...
Ergebnis	SK	SK	FF	SN	SK	SK	FF	...
Dauer	10 ms	25 ms	11 ms	30 ms	15 ms	18 ms	43 ms	...

Aus den Zählwerten und Zeiten lassen sich weitere Kennwerte abschätzen.

Nummer	1	2	3	4	5	6	7	...
Ergebnis	SK	SK	FF	SN	SK	SK	FF	...
Dauer	10 ms	25 ms	11 ms	30 ms	15 ms	18 ms	43 ms	...

Abschätzbare Kennwerte:

- MTBF Mittlere Zeit zwischen zwei Fehlfunktionen (Mean Time between Failures[13]),
- MTTR Mittlere Reparaturzeit (Mean Time to Repair), ...

In der Tabelle werden:

- ingesamt für 10 ms + 25 ms + 11 ms +15 ms +18 ms + 43 ms Service-Leistungen ausgeführt und 2 FFs beobachtet.
  Schätzwert der MTBF:
  $$MTBF \approx \frac{122\,\mathrm{ms}}{2} = 61\,\mathrm{ms}$$
- Aus der einen Zeit für Service nicht verfügbar von $30\,\mathrm{ms}$ ist abschätzbar:
  $$MTTR \approx 30\,\mathrm{ms}$$

[13] Mittlere Betriebsdauer zwischen zwei Fehlfunktionen ohne Einrechnung der Zeiten für Reparaturen, Neuinitialisierung, ... bis wieder nutzbar.

Kenngrößen der Verlässlichkeit

Nach Folie 5 umfasst Verlässlichkeit u.a. die Teilaspekte (attributes):

- Verfügbarkeit (Availability): Bereitschaft für einen korrekten Service.
- Zuverlässigkeit (Reliability): Stetigkeit korrekter Service-Leistung.
- Sicherheit (Safety): Ausschluss katastrophaler Folgen für Benutzer und Umwelt.

Für jeden dieser Aspekte lassen sich unterschiedliche Kenngrößen definieren.

Im folgenden werden Kenngrößen definiert, die diese drei Eigenschaften quantitativ beschreiben und sich aus den Zählergebnissen und gemessenen Zeiten des »Service-Experiments« abschätzen lassen.

Verfügbarkeit

Verfügbarkeit

Die Kenngröße »Verfügbarkkeit« sei die anteilmäßige Zeit, die der Service verfügbar ist:

$$V = \frac{\text{Summe SK- und FF-Zeiten)}}{\text{Summe SK-, FF- und SN-Zeiten}}$$

Für eine große Anzahl von Zählwerten strebt die Zufallsgröße »Verfügbarkeit« gegen die Wahrscheinlichkeit p_{V}, dass der Service verfügbar ist

$$E(V) = p_{\mathrm{V}}$$

Diese ist gleich dem Verhältnis aus $MTBF$ zur Summe $MTBF + MTTR$:

$$E(V) = p_{\mathrm{V}} = \frac{MTBF}{MTBF + MTTR}$$

Die Verfügbarkeit kann entweder durch Verkürzung der Reparaturzeiten oder durch Verlängerung der mittleren Zeit zwischen den Fehlfunktionen erhöht werden.

Beispielaufgabe

Ein System soll mit einer Wahrscheinlichkeit $p_\mathrm{V} \geq 99,9\%$ verfügbar sein. Die mittlere Reparaturzeit beträgt $MTTR = 1\,\mathrm{h}$. Wie groß muss die mittlere Zeit zwischen den Fehlfunktionen mindestens sein?

Lösung

$$\begin{aligned} 99{,}9\% \leq p_\mathrm{V} &= \frac{MTBF}{MTBF + 1\,\mathrm{h}} \\ MTBF &\geq 1\,\mathrm{h} \cdot \frac{99{,}9\%}{1 - 99{,}9\%} \approx 10^3\,\mathrm{h} \end{aligned}$$

Zuverlässigkeit

Zuverlässigkeit

Zuverlässigkeit: Stetigkeit korrekter Service-Leistung.

Kenngrößen:

- $MTBF$ (Reparaturzeiten herausgerechnet).
- Parameter »Zuverlässigkeit« (Service-Leistungen je Fehlfunktion):

$$Z = \frac{N_{\mathrm{SK}} + \zeta}{\zeta};\ E(Z) = \frac{MTBF}{E(t_{\mathrm{S}})}$$

(N_{SK} – Anzahl korrekt ausgeführter Service-Leistungen; ζ – Anzahl der Fehlfunktionen; $E(t_{\mathrm{S}})$ – mittlere Service-Dauer).

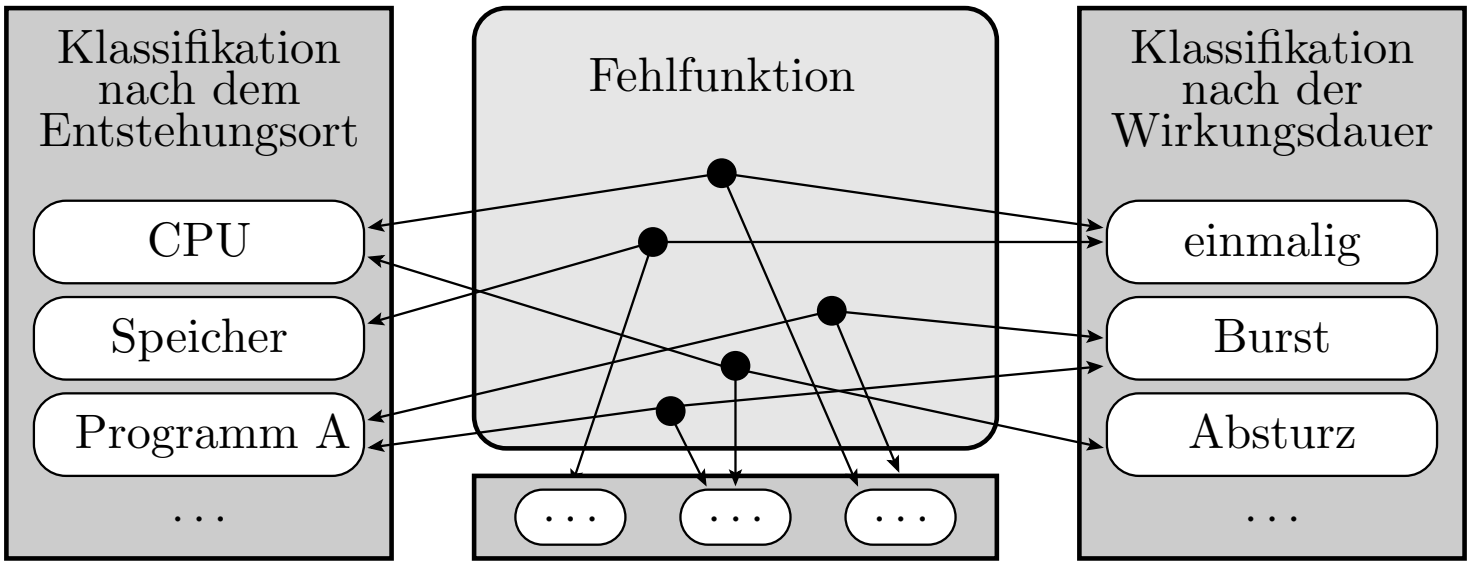
- Der Kehrwert des Erwartungswertes der so definierten Zuverlässigkeit ist die Wahrscheinlichkeit einer Fehlfunktion:

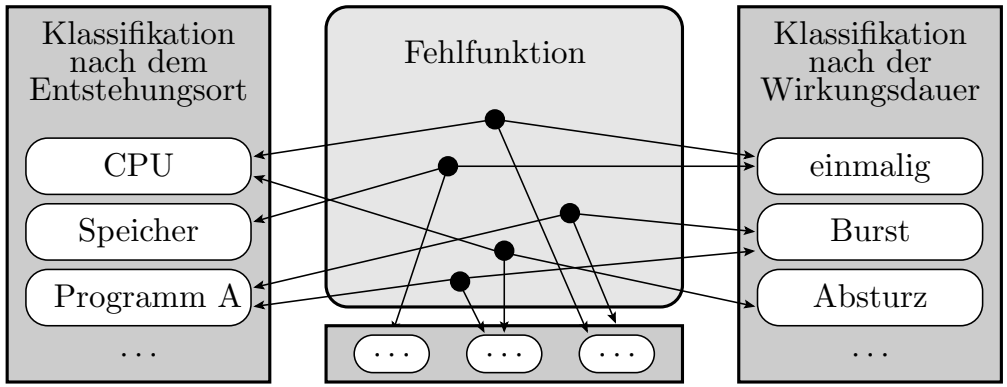
$$p_{\mathrm{FF}} = \frac{1}{E(Z)} = \frac{E(t_{\mathrm{S}})}{MTBF}$$

Teilverlässigkeiten

Die Fehlfunktionen (FF) eines Systems lassen sich nach Ort, Ursache und Schaden unterschiedlichen Klassen zuordnen:

- nur FFs eines bestimmten Teilsystems,
- nur durch HW, nur durch SW verursachte FFs,
- nur FF, die die Betriebssicherheit / die Datensicherheit / die Zugangssicherheit gefährden:





Bei einer eindeutigen Zuordnung jeder Fehlfunktion zu genau einer Klasse i ist die Gesamtanzahl ξ der Fehlfunktionen die Summe ξ_i der Fehlfunktionen jeder Klasse i:

$$\xi = \sum_{i=1}^{N_{\mathrm{FFKl}}} \xi_i$$

(N_{FFKl} – Anzahl der Fehlfunktionsklassen). Der Kehrwert der Gesamtzuverlässigkeit ist die Summe der Kehrwerte der Teilzuverlässigkeiten alle Klassen:

$$\frac{1}{Z} = \frac{\zeta}{N_{\mathrm{SK}} + \zeta} = \sum_{i=1}^{N_{\mathrm{FFKl}}} \frac{1}{Z_i} = \sum_{i=1}^{N_{\mathrm{FFKl}}} \frac{\xi_i}{N_{\mathrm{SK}} + \zeta}$$

Beispielaufgaben

Die Fehlfunktionen seien entweder vom Speicher, vom Prozessor, von der Software oder vom Rest verursacht. Es liegen folgende $MTBF$-Werte vor:

Teilsystem	Speicher	Prozessor	Software	Rest
$MTBF_i$	10.000 h	3.000 h	1000 h	7.000 h

Mittlere Service-Dauer $E(t_S) = 10\,\mathrm{s}$.

1. Wie groß sind die vier aus den $MTBF$-Werten ableitbaren Teilzuverlässigkeiten.
2. Wie groß sind die Zuverlässigkeit und die $MTBF$ des Gesamtsystems?
3. Wie groß ist die Wahrscheinlichkeit einer Fehlfunktion des Gesamtsystems?

Lösungen

1. Wie groß sind die vier aus den $MTBF$-Werten ableitbaren Teilzuverlässigkeiten.

Teilsystem	Speicher	Prozessor	Software	Rest
$E(Z_i) = \frac{MTBF}{10\,\mathrm{s}}$	$3{,}6 \cdot 10^6$	$1{,}08 \cdot 10^6$	$3{,}6 \cdot 10^5$	$2{,}52 \cdot 10^6$

(Maßeinheit: Service-Leistungen je Fehlfunktion)

2. Wie groß sind die Zuverlässigkeit und die $MTBF$ des Gesamtsystems?

$$E(Z) = \frac{1}{\frac{1}{60.000} + \frac{1}{18.000} + \frac{1}{6.000} + \frac{1}{42.000}} = 3808{,}6$$

$$MTBF = E(Z) \cdot 10\,\mathrm{s} = 634{,}44\,\mathrm{h}$$

3. Wie groß ist die Wahrscheinlichkeit einer Fehlfunktion des Gesamtsystems?

$$p_{\mathrm{FF}} = \frac{1}{E(Z)} = 2{,}63 \cdot 10^{-4}$$

Sicherheit

Sicherheit

Sicherheit: Ausschluss katastrophaler Folgen für Benutzer und Umwelt.

Definition 10

Eine Sicherheit sei eine Teilzuverlässigkeit in Bezug auf Fehlfunktionen mit erheblichem Schaden.

Damit lässt sich eine Sicherheit durch gleiche Parameter wie eine Zuverlässigkeit beschreiben:

- $MTBF_{\mathrm{S}}$ – mittlere Zeit zwischen sicherheitskritischen FFs.
- Z_{S} – Sicherheit, Service-Leistungen je sicherheitskritische Fehlfunktion.
- p_{FFS} – Wahrscheinlichkeit einer sicherheitskritischen FF.

Je nach den Fehlfunktionen, die als sicherheitkritisch gelten, wird zwischen unterschiedlichen Arten von Sicherheiten unterschieden.

Arten von Sicherheiten

Sicherheit	sicherheitskritische FFs
Betriebssicherheit (Safty)	Personen- und Umweltschäden
Datensicherheit (security)	Datendiebstahl
Sicherheit Datenerhalt	Datenverlust
...	...

Eine zusätzliche wichtige Größe zur Beschreibung von Sicher- heiten ist der Anteil der sicherheitskritischen FFs an allen auftretenden Fehlfunktionen:

$$\eta_S = \frac{\zeta_S}{\zeta} = \frac{Z}{Z_S}$$

$$E(\eta_S) = \frac{MTBF}{MTBF_S} = \frac{p_{FFS}}{p_{FF}}$$

Beispielaufgaben

Eine Fahrzeug habe eine $MTBF = 1000\,\mathrm{h}$. Der zu erwartende Anteil der für die Betriebssicherheit kritischen FFs sei $E(\eta_\mathrm{S}) = 1\%$ und die mittlere Service-Dauer (mittlere Fahrtdauer) $E(t_\mathrm{S}) = 1\,\mathrm{h}$.

1. Wie hoch sind die $MTBF_\mathrm{S}$ für betriebssicherheitskritische FFs, die Betriebssicherheit und die Wahrscheinlichkeit p_FFS für eine betriebssicherheitskritische FF?
2. Ein zusätzliches elektronisches Steuergerät soll den Anteil der betriebssicherheitskritischen FFs auf ein Zehntel absenken. Wie groß muss die $MTBF_\mathrm{SG}$ des Steuergeräts mindestens sein, damit sich die Betriebssicherheit des Fahrzeugs verfünffacht?

Lösung Aufgabenteil 1

Wie hoch sind die $MTBF_{\mathrm{S}}$ für betriebssicherheitskritische FFs, die Betriebssicherheit und die Wahrscheinlichkeit p_{FFS} für eine betriebssicherheitskritische FF?f

$$
\begin{aligned}
MTBF_{\mathrm{S}} &= \frac{MTBF}{E\left(\eta_{\mathrm{S}}\right)} = \frac{1000\,\mathrm{h}}{1\%} = 10^5\mathrm{h} \\
Z_{\mathrm{S}} &= \frac{MTBF_{\mathrm{S}}}{E\left(t_{\mathrm{S}}\right)} = \frac{10^5\mathrm{h}}{1\mathrm{h}} = 10^5 \\
p_{\mathrm{FFS}} &= \frac{1}{Z_{\mathrm{S}}} = 10^{-5}
\end{aligned}
$$

Lösung Aufgabenteil 2

Ein zusätzliches elektronisches Steuergerät soll den Anteil der betriebssicherheitskritischen FFs auf ein Zehntel absenken. Wie groß muss die Zuverlässigkeit Z_{SG} und die $MTBF_{\mathrm{SG}}$ des Steuergeräts mindestens sein, damit sich die Betriebssicherheit des Fahrzeugs verfünffacht?

$$
\begin{aligned}
Z_{\mathrm{S.MSG}} &= 10 \cdot \frac{Z_{\mathrm{MSG}}}{E\left(\eta_{\mathrm{S}}\right)} \geq 5 \cdot \frac{Z}{E\left(\eta_{\mathrm{S}}\right)} \\
Z_{\mathrm{MSG}} &\geq \frac{Z}{2} \\
Z_{\mathrm{MSG}} &= \frac{1}{\frac{1}{Z} + \frac{1}{Z_{\mathrm{SG}}}} \\
Z_{\mathrm{SG}} &\geq Z = 10^3
\end{aligned}
$$

Das Steuergerät muss mindestens so zuverlässig wie das Fahrzeug sein und auch mindestens dessen $MTBF$ von $1000\,\mathrm{h}$ haben.

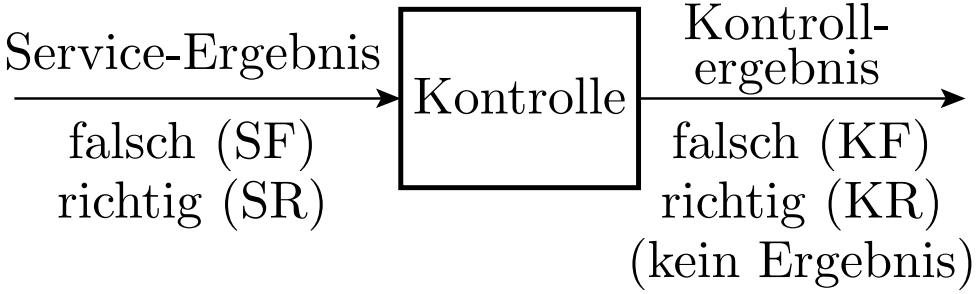
Kontrollen

Kontrolle

Eine Kontrolle klassifiziert Merkmale

- Ergebnisse,
- Formate,
- Invarianten, ...

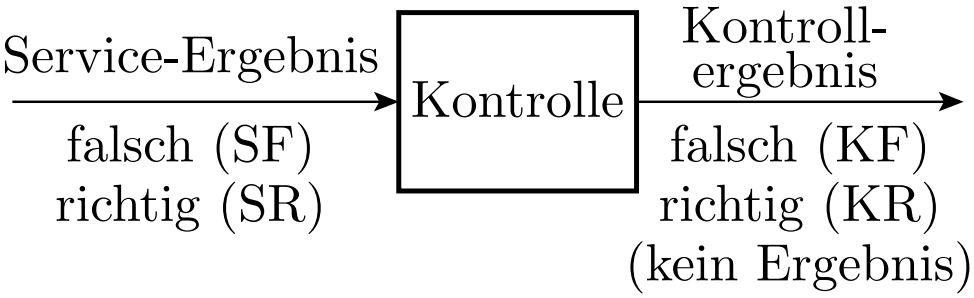
als richtig oder falsch und ist selbst ein Service mit potentiellen Fehlfunktionen.



Ergebnis	Kontroll-ergebnis	Kontroll-aussage
falsch	falsch	KFF
falsch	richtig	KFR
richtig	falsch	KRF
richtig	richtig	KRR

Mögliche Fehlklassifizierungen:

- falsch als richtig (KFR): Maskierung einer Fehlfunktion.
- richtig als falsch (KRF): Phantomfehler.



Ergebnis	Kontroll-ergebnis	Kontroll-aussage
falsch	falsch	KFF
falsch	richtig	KFR
richtig	falsch	KRF
richtig	richtig	KRR

Experimentell schätzbare Größen:

- Anteil der erkannten Fehlfunktionen (Error Coverage):

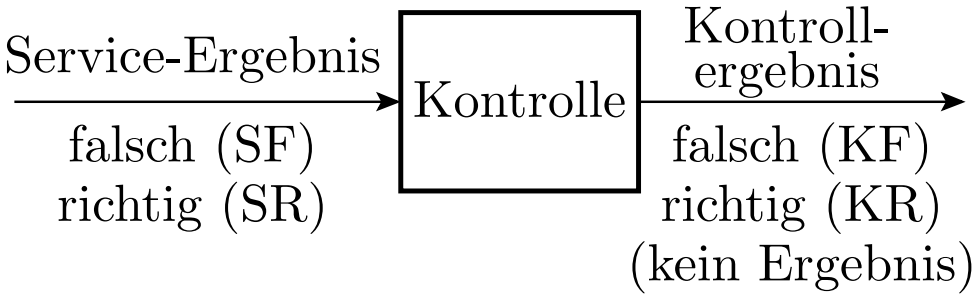
$$EC = \frac{N_{\mathrm{KFF}}}{N_{\mathrm{KFF}} + N_{\mathrm{KFR}}}$$

- Erkennungswahrscheinlichkeit, Erwartungswert des Anteils der erkannten Fehlfunktionen:

$$p_{\mathrm{E}} = E\left(EC\right) = \lim_{N_{\mathrm{KFF}}+N_{\mathrm{KFR}}\to\infty} \left(\frac{N_{\mathrm{KFF}}}{N_{\mathrm{KFF}} + N_{\mathrm{KFR}}} \right)$$

- Maskierungswahrscheinlichkeit, Gegenwahrscheinlichkeit zu p_{E}:

$$p_{\mathrm{M}} = 1 - p_{\mathrm{E}}$$



Ergebnis	Kontroll-ergebnis	Kontroll-aussage
falsch	falsch	KFF
falsch	richtig	KFR
richtig	falsch	KRF
richtig	richtig	KRR

- Phantomfehlerwahrscheinlichkeit, zu erwartender Anteil der Phantomfehler an den richtigen Service-Ergebnissen:

$$p_{\mathrm{Phan}} = E\left(\frac{N_{\mathrm{KRF}}}{N_{\mathrm{KRR}} + N_{\mathrm{KRF}}}\right)$$

($N_{\ldots}$ – Anzahl von ...).

Wenn nur sehr selten wirkliche Fehler oder Gefahrensituationen auftreten, sind die meisten diagnostizierten Fehlfunktionen Phantomfehler.

Bisher war jeder im Gebäude des Instituts für Mathematik ausgelöste Feueralarm ein Fehlalarm.

Beispielaufgabe

Die $MTBF$ eines Services sei 1000 h und die mittlere Service-Dauer $E(t_S) = 1\,\mathrm{h}$. Eine nachgeschaltete Kontrolle erkennt $EC = 99\%$ der Fehlfunktionen.

1. Wie groß ist die $MTBF_K$ (MTBF mit Kontrolle), wenn die als fehlerhaft erkannten Service-Leistungen nicht verwendet (aussortiert) werden?
2. Wie groß ist die zu erwartende Zuverlässigkeit mit und ohne Aussortieren?
3. Wie ändert sich die Zuverlässigkeit mit Aussortieren, wenn die Kontrolle zusätzlich eine Phantomfehlerwahrschein- lichkeit von $p_{Phan} = 10\%$ hat?

Lösung Aufgabenteil 1

Die $MTBF$ eines Services sei 1000 h und die mittlere Service-Dauer $E(t_S) = 1\,\mathrm{h}$. Eine nachgeschaltete Kontrolle erkennt $EC = 99\%$ der Fehlfunktionen. Wie groß ist die $MTBF_K$, wenn die als fehlerhaft erkannten Service-Leistungen nicht verwendet (aussortiert) werden?

- $EC = 99\%$ sei der Schätzwert für den Erwartungswert:

$$p_E = E(EC) = 99\%$$

- Maskierungswahrscheinlichkeit:

$$p_M = 1 - p_E = 1\%$$

- Wenn nur $p_M = 1\%$ der Fehlfunktion nach Kontrolle und Aussortieren übrig bleiben, erhöhen Kontrolle und Aussortieren die MTBF auf das hundertfache:

$$MTBF_K = \frac{MTBF}{1\%} = 10^5\mathrm{h}$$

Lösung Aufgabenteil 2

$MTBF = 1000\,\mathrm{h}$; $E(t_\mathrm{S}) = 1\,\mathrm{h}$, $p_\mathrm{M} = 1\%$. Gesucht: Zu erwartende Zuverlässigkeit mit und ohne Aussortieren.

Für eine sehr große Service-Anzahl N_S folgt aus:

$$E(Z) = E\left(\frac{N_\mathrm{SK} + \zeta}{\zeta}\right) = \frac{MTBF}{E(t_\mathrm{S})}$$

ohne Aussortieren	mit Aussortieren
$N_\mathrm{SK} = N_\mathrm{S} - \frac{N_\mathrm{S} \cdot E(t_\mathrm{S})}{MTBF}$	$N_\mathrm{SKA} = N_\mathrm{S} - \frac{N_\mathrm{S} \cdot E(t_\mathrm{S})}{MTBF}$
$\zeta = \frac{N_\mathrm{S} \cdot E(t_\mathrm{S})}{MTBF}$	$\zeta_\mathrm{A} = \frac{p_\mathrm{M} \cdot N_\mathrm{S} \cdot E(t_\mathrm{S})}{MTBF}$
$E(Z) = \frac{MTBF}{E(t_\mathrm{S})} = 10^3$	$E(Z_\mathrm{A}) = \frac{N_\mathrm{S} - \frac{N_\mathrm{S} \cdot E(t_\mathrm{S})}{MTBF} + \frac{p_\mathrm{M} \cdot N_\mathrm{S} \cdot E(t_\mathrm{S})}{MTBF}}{\frac{p_\mathrm{M} \cdot N_\mathrm{S} \cdot E(t_\mathrm{S})}{MTBF}}$

$$E(Z_\mathrm{A}) = \frac{1 - \frac{1}{E(Z)} + \frac{p_\mathrm{M}}{E(Z)}}{\frac{p_\mathrm{M}}{E(Z)}} = \frac{E(Z) - 1 + p_\mathrm{M}}{p_\mathrm{M}} \approx 99.900$$

Lösung Aufgabenteil 3

$MTBF = 1000\,\mathrm{h}$; $E(t_\mathrm{S}) = 1\,\mathrm{h}$, Anteil der nicht aussortierten Fehlfunktionen $p_\mathrm{M} = 1\%$. Zu erwartende Zuverlässigkeit, wenn zusätzlich $p_\mathrm{Phan} = 10\%$ guter Service-Leistungen aussortiert werden.

mit Aussortieren	zusätzlich mit $p_\mathrm{Phan} = 10\%$
$N_\mathrm{SKA} = N_\mathrm{S} - \frac{N_\mathrm{S} \cdot E(t_\mathrm{S})}{MTBF}$	$N_\mathrm{SKAP} = 0{,}9 \cdot N_\mathrm{SKA}$
$\zeta_\mathrm{A} = \frac{p_\mathrm{M} \cdot N_\mathrm{S} \cdot E(t_\mathrm{S})}{MTBF}$	$\zeta_\mathrm{AP} = \zeta_\mathrm{A}$
$E(Z_\mathrm{A}) = \frac{N_\mathrm{SKA} + \zeta_\mathrm{A}}{\zeta_\mathrm{A}}$	$E(Z_\mathrm{AP}) = \frac{0{,}9 \cdot N_\mathrm{SKA} + \zeta_\mathrm{A}}{\zeta_\mathrm{A}}$

$$E(Z_\mathrm{A}) = \frac{0{,}9 \cdot (E(Z) - 1) + p_\mathrm{M}}{p_\mathrm{M}} \approx 90.000$$

Test

Test

Ein Test ist eine Kontrolle auf Abwesenheit von Fehlern:

- Statischer Test: direkte Kontrolle von Eigenschaften (Widerstandsmessungen für den Verbindungstest auf Baugruppen, Syntaxtest für Programme, ...)
- Dynamischer Test: beispielhaftes Ausprobieren.

S	wiederhole für eine Menge direkt überprüfbarer Merkmale
	Kontrolle, dass sie erfüllt sind

D	Wiederhole für eine Menge von Testbeispielen
	Service mit den Beispielwerten anfordern
	Ergebnisse kontrollieren (meist Soll/Ist-Vergleich)

Ergebnisse für den einmaligen Test eines einzelnen Objekts:

- gut-Aussage für alle erfolgreichen Tests.
- Protokoll der Soll-/Ist-Abweichungen für alle Tests, die fehlgeschlagen sind.

Fehlerüberdeckung und -nachweiswahrscheinlichkeit

- Fehlerüberdeckung: Anteil der nachweisbaren Fehler:

$$FC = \frac{\varphi_{\mathrm{Erk}}}{\varphi}$$

(φ_{Erk} – Anzahl der nachweisbaren, φ – Anzahl der vorhandenen Fehler).

- Fehlernachweiswahrscheinlichkeit: Erwartungswert der Fehlerüberdeckung.

Da die Anzahl der nicht nachgewiesen Fehler fast[14] immer unbekannt ist, wird die Fehlerüberdeckung mit Modellfehlermengen abgeschätzt (vergl. Folie 41):

- Injizierung einer Menge von Modellfehlern in das System.
- Zählen der davon nachweisbaren Fehler.

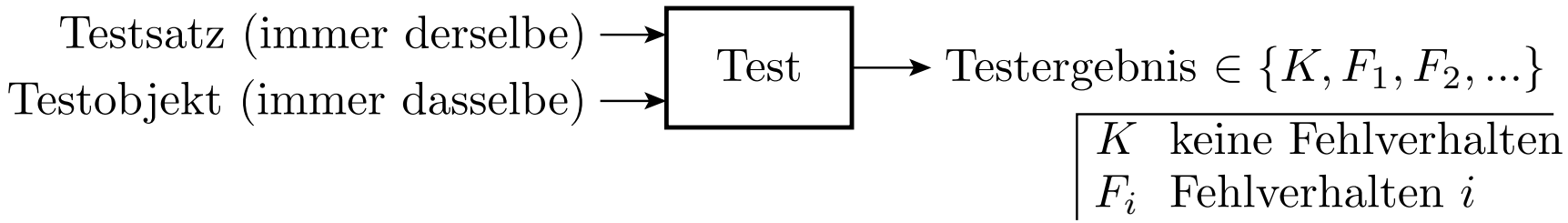
[14] Ausgenommen vorsätzliche oder mit anderen Tests gefundene Fehler.

Reproduzierbarkeit

Ursachen von Fehlfunktionen können nach Folie 31 sein:

- Fehlern (wirken ständig, können aber nach Folie 42 unbeständige FFs verursachen),
- Störungen (spontane, nicht reproduzierbare Wirkung) und
- Ausfälle (Während des Betriebs entstehende Fehler).

Zur Kontrolle von Testergebnissen auf Reproduzierbarkeit wird derselbe Test mit demselben Testobjekt mehrfach wiederholt.



- einmaliges Fehlverhalten $\Rightarrow$ Störung,
- seltenes Fehlverhalten $\Rightarrow$ Schwachstelle im System, die Störungen begünstigt, ...

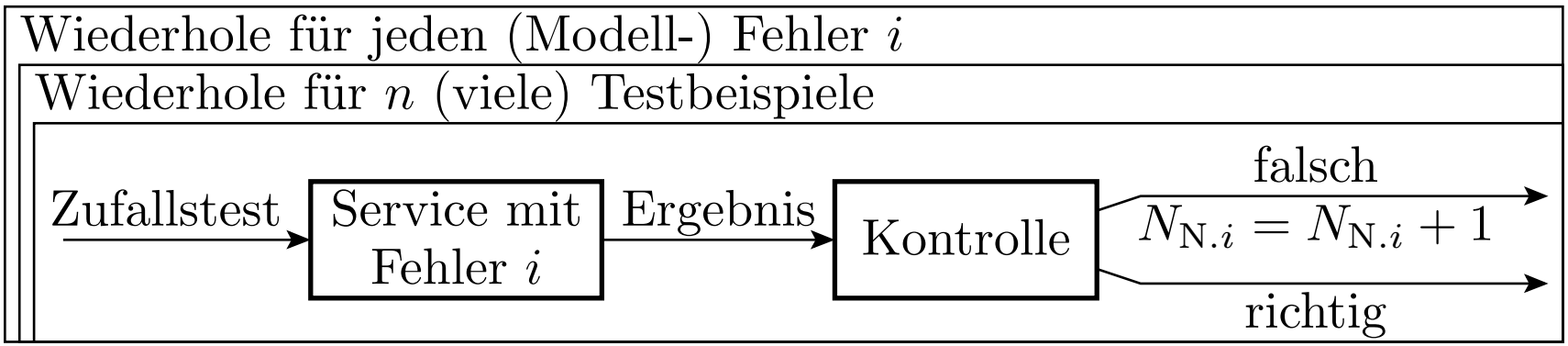
Zufallstest

Zufallstest

Dynamischer Test mit zufälligen Eingaben. Jeder Fehler hat eine Nachweiswahrscheinlichkeit, abschätzbar durch Test mit einer langen Folge von Testbeispielen und Zählen der Testbeispiele, die den Fehler nachweisen:

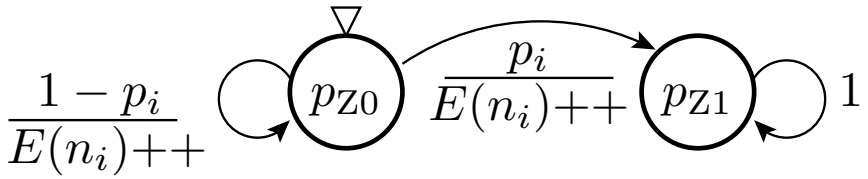
$$p_i \approx \frac{N_{\mathrm{N}.i}}{n};$$

($N_{\mathrm{N}.i}$ – Testbeispiele, die den Fehler nachweisen, n – Anzahl der ausprobierten Testbeispiele). Eine vertrauenswürdige Schätzung verlangt $N_{\mathrm{N}.i} \gg 1$.



Nachweiswahrscheinlichkeit $p_i(n)$ und $E(n_i)$

Der Nachweis eines Fehlers mit n Service-Aufrufen bildet einen Markov-Prozess mit zwei Zuständen und einem Zähler zur Bestimmung der zu erwartenden Anzahl der Testschritten $E(n_i)$ bis zum Fehlernachweis:



Z_0	Fehler i nicht nachgewiesen
Z_1	Fehler i nachgewiesen
$E(n_i)$	Testschrittzähler

- Nachweiswahrscheinlichkeit:

$$p_i(n) = 1 - (1-p_i)^n = 1 - e^{\ln(1-p_i)\cdot n} = \left(1 - e^{-n\cdot p_i}\right)^* \qquad (6)$$

(* Näherung für $p_i \ll 1$).

- Mittlere Anzahl der Testschritte bis zum Fehlernachweis:

$$E(n_i) = \sum_{j=0}^{\infty} (1-p_i)^j = \frac{1}{p_i}$$

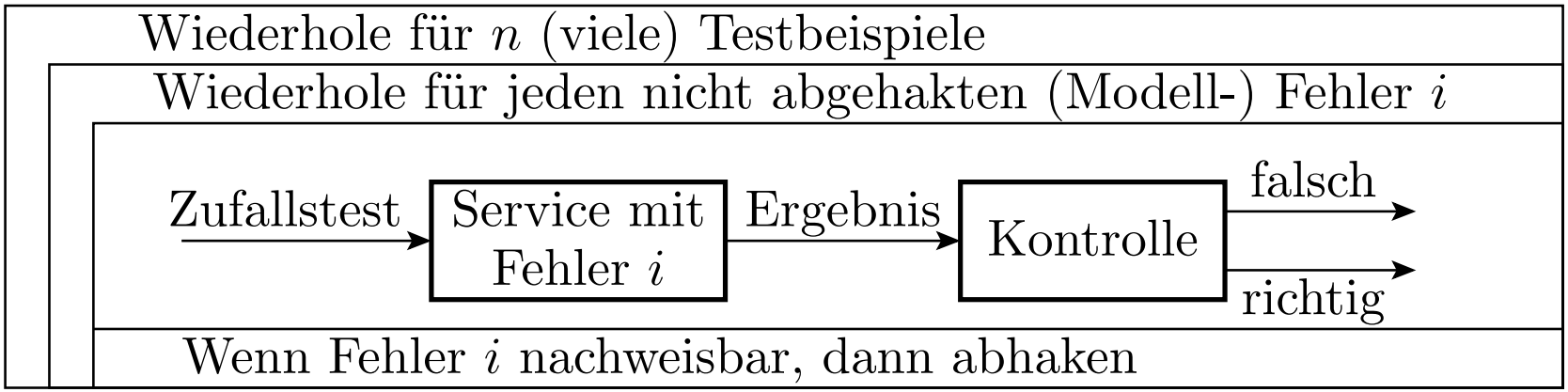
Ausftrittshäufigkeit vs. Anzahl der Nachweisschritte

Ordnet man die experimentell abgeschätzten $E(n_i)$-Werte der potenziellen Fehler eines Systems in Quantile gleicher Anzahl, nimmt der Bereiche der $E(n_i)$-Werte in der Regel mehr als exponentiell mit der Quantilnummer zu. Beispiel:

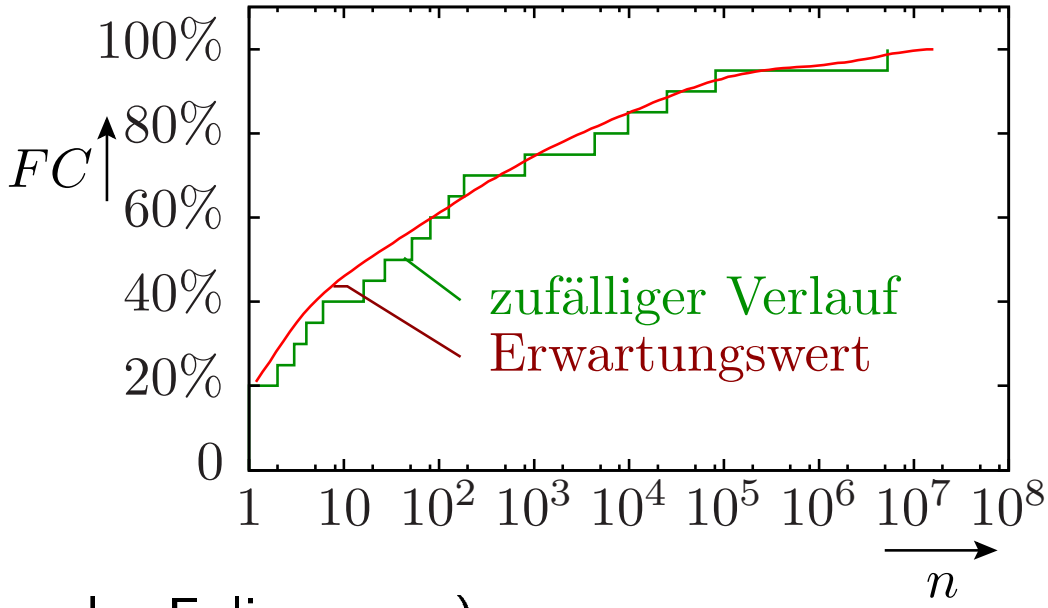
Quantil	$E(n_i)$-Werte des Quantils				Bereich
1	1,1	1,1	1,2	1,3	1 bis 1,5
2	1,9	2	3	5	1,5 bis 7
3	10	20	50	100	7 bis 150
4	200	500	1.000	3.000	150 bis 5.000
5	10.000	40.000	100.000	1.000.000	größer 5.000

Die Mehrheit der Fehler ist mit wenigen Zufallswerten nachweis- bar, aber einige wenige Fehler benötigen sehr lange Testsätze.

Fehlerüberdeckung und Testsatzlänge



Die Fehlerüberdeckung als der Anteil der nachweisbaren Fehler nimmt zuerst stärker und dann immer weniger mit der Testanzahl zu. (Beispiel simuliert mit den Nachweiswahrscheinlichkeiten von der Folie zuvor.)



Beispielaufgabe

Für drei Fehler betrage die mittlere Testschrittanzahl bis zum Fehlernachweis 2, 5 und 10. Wie groß sind die Wahrscheinlichkeiten dass

1. keiner der drei Fehler
2. alle drei Fehler

mit $n = 2$, $n = 10$ und $n = 20$ Testsschritten nachgewiesen werden?

Lösung

- Die Nachweiswahrscheinlichkeiten sind die Kehrwerte der mittleren Testschrittanzahl bis zum Fehlernachweis.
- Für $n > 1$ gilt Gl. 6:

$$p_i(n) = 1 - (1 - p_i)^n$$

Fehler	$E(n_i)$	p_i	$p_i(2)$	$p_i(10)$	$p_i(20)$
1	2	0,5	75%	99,9%	100%
2	5	0,2	36%	89,62%	98,58%
3	10	0,1	19%	65,13%	87,84%
alle Fehler: $\prod_{i=1}^{3} p_i(n)$			5,13%	58,08%	86,83%
kein Fehler: $\prod_{i=1}^{3} (1 - p_i(n))$			12,7%	$3{,}6 \cdot 10^{-5}$	$1{,}3 \cdot 10^{-9}$

Zusatzanmerkungen zum Zufallstest

Die Fehler in einem System sind immer unbekannt. Auch bei einer gezielten Testauswahl ist der Nachweis der vorhandenen Fehler Zufall.

Für Zufallstests lässt sich der Anteil der nachweisbaren Fehler besser als bei gezielter Testauswahl vorhersagen.

Die begonnen Betrachtungen über den Zusammenhang zwischen der Fehlerausftrittshäufigkeit und ihrem Nachweisaufwand dienen später zur Abschätzung der Anzahl der nicht gefundenen Fehler in Systemen und der Häufigkeit, mit der diese Fehler im Einsatz Service-Leistungen versagen lassen.

Die erforderliche große Anzahl von Testschritten eines Zufallstests lässt sich mit Mitteln des »prüfgerechten Entwurfs« bezahlbar halten.

Fehleranteil, Entst.-Proz.

Fehleranteil

Der Fehleranteil DL (Defekt Level) ist der Anteil der defekten Systeme in einer Menge gleichartiger Systeme und ein Schätzer für die Wahrscheinlichkeit, dass ein System defekt ist:

$$DL = \frac{N_{\mathrm{DS}}}{N_{\mathrm{Sys}}} \tag{7}$$

(N_{DS} – Anzahl der defekten Systeme; N_{Sys} – Anzahl aller Systeme). Richtwerte für ungetestete Systeme:

Typ	DL
NLOC	0,01...0,03 dpu
Schaltkreise	200 dpm
diskrete Bauteile	10 dpm
Lötstellen	1 dpm

(dpu – Defects per Unit; dpm – Defects per Million; NLOC – Netto Lines of Code, Codezeilen ohne Kommentar- und Leerzeilen).

Bestimmung des Fehleranteils durch Zählen

Zählbar sind nur die erkannten defekten Systeme. Zur Abschätzung der Anzahl der aufgetretenen Defekte inkl. der nicht erkannten ist die Fehlerüberdeckung einzubeziehen:

$$FC = \frac{N^*_{\mathrm{DS}}}{N_{\mathrm{DS}}}$$

(N^*_{DS} – Anzahl der vom Test erkannten defekten Systeme). Eingesetzt in Gl. 7 ist der tatsächliche Fehleranteil gleich dem beobachtbaren Fehleranteil geteilt durch die Fehlerüberdeckung:

$$DL = \frac{N_{\mathrm{DS}}}{N_{\mathrm{Sys}}} = \frac{N^*_{\mathrm{DS}}}{FC \cdot N_{\mathrm{Sys}}}$$

Beispielaufgabe

Von 10.000 Schaltkreisen wurden 4.000 als fehlerhaft erkannt. Die Fehlerüberdeckung betrage mindestens 80%. Wie groß ist der Fehleranteil maximal?

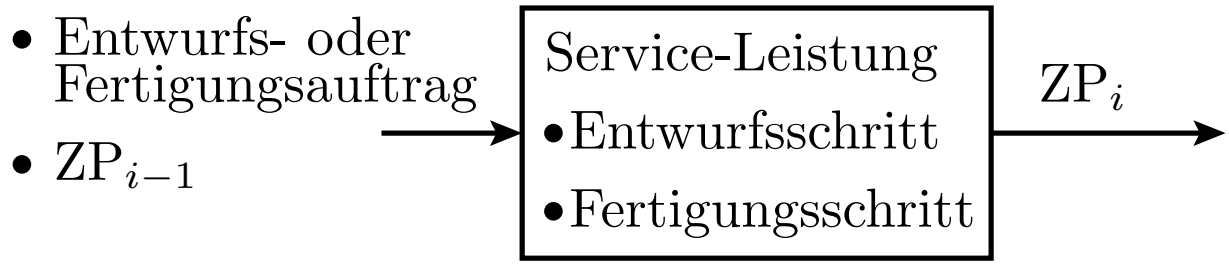
- Anzahl aller Systeme: $N_{\text{Sys}} = 10.000$,
- Anzahl der vom Test erkannten defekten Systeme: $N^*_{\text{DS}} = 4.000$,
- Fehlerüberdeckung: $FC \geq 80\%$:

$$DL \leq \frac{N^*_{\text{DS}}}{FC_{\min} \cdot N_{\text{Sys}}} = \frac{4.000}{0{,}8 \cdot 10.000} = 50\%$$

Der beobachtbare Fehleranteil ist 40% und der tatsächliche 40%...50%.

Entstehungsprozesse als Service-Leistungen

Ein Entstehungsprozess besteht aus vielen Schritten:

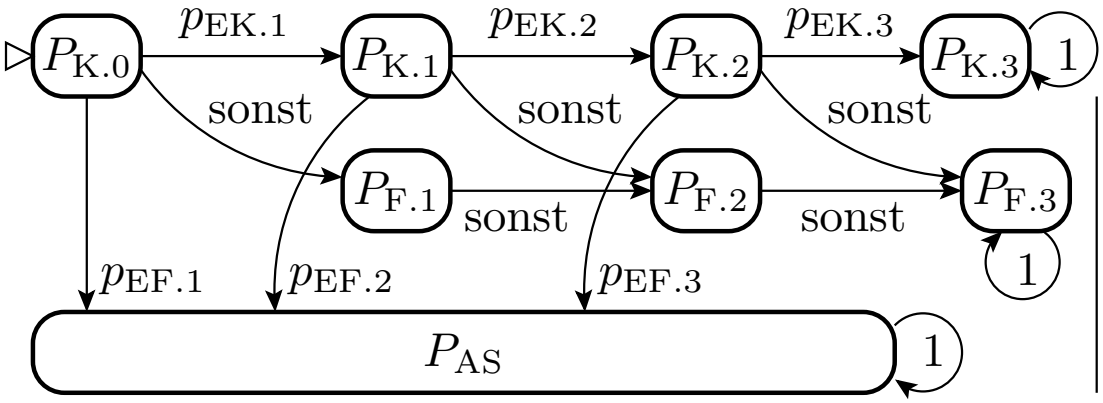


Jeder Schritt

- startet mit einen Entwurfs- oder Fertigungsauftrag
- hat als Eingabe Zwischenprodukte ZP_{i-1} (Material, Teilprodukte, Teilentwürfe) mit potentiellen Fehlern,
- liefert als Ergebnis ein End- oder Zwischenprodukt für den Folgeschritt oder Endprodukt ZP_i, das die Fehler der Zwischenprodukte und die neu entstandenen Fehler enthält.

Erkannte fehlerhafte Produkte werden aussortiert oder repariert.

Entstehungsprozess als Markov-Kette



Zustände:

K.i	i Schritte abgearbeitet, kein Fehler entstanden
F.i	i Schritte abgearbeitet, mindestens ein Fehler, kein Fehler erkannt
AS	Produkt aussortiert

Der Beispielprozess hat $i = 1$ bis 3 Schritte, in denen

- mit einer Wahrscheinlichkeit $p_{EK.i}$ kein Fehler und
- mit $p_{EF.i}$ ein erkennbarer Fehler ensteht.

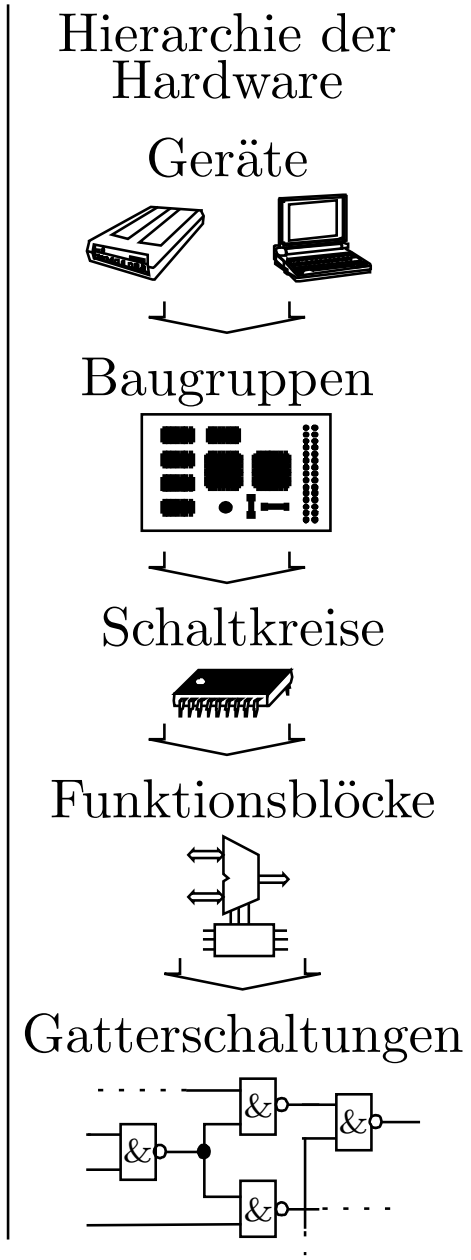
Zwischenprodukte mit erkennbarem Fehler werden aussortiert. Die Markov-Kette liefert die Wahrscheinlichkeiten:

- $P_{K.3}$ es entsteht ein fehlerfreies Produkt.
- $P_{F.3}$ es entsteht ein fehlerhaftes Produkt
- P_{AS} es entsteht kein Produkt.
- en

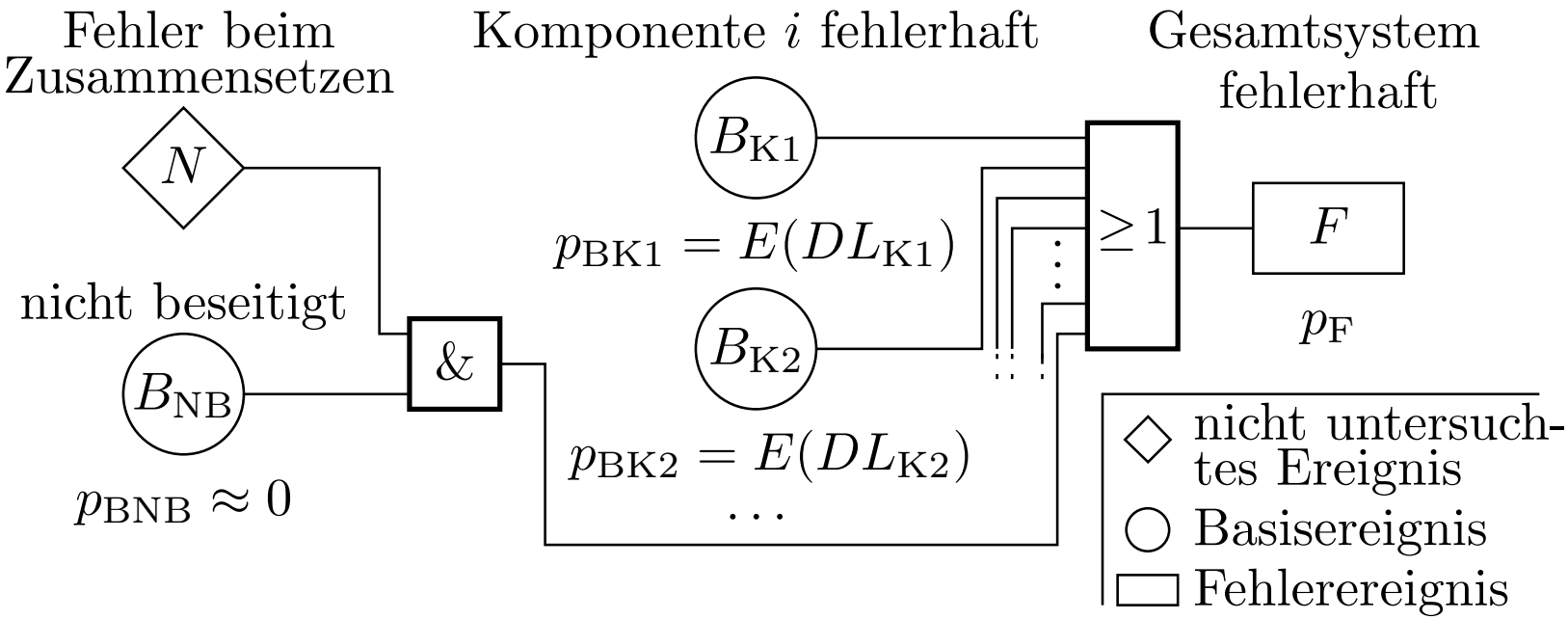
Entstehungsprozess als Fehlerbaum

IT-Systeme sind hierarchisch aufgebaut:

- Client-Server-Systeme bestehen aus Rechnern und Netzwerkkomponenten.
- Rechner, Netzwerkkomponenten, ... bestehen aus Hard- und Software.
- Software besteht aus Programmbausteinen, diese sind aus Programmieranweisungen zusammengesetzt, die ihrerseits mit Maschinenbefehlen nachgebildet werden.
- Maschinenbefehle sind Service-Leistungen der Hardware. Die Hardware besteht aus Funktionsbausteinen, diese meist aus Gattern und diese wiederum aus Transistoren.



Fehleranteil beim Zusammensetzen eines Systems aus N_K Komponenten als Fehlerbaum:



Fehleranteil des zusammengesetzten Systems für $p_\mathrm{BNB} = 0$:

$$p_\mathrm{F} = E\left(DL_\mathrm{ges}\right) = 1 - \prod_{i=1}^{N_\mathrm{K}} \left(1 - E\left(DL_{\mathrm{K}.i}\right)\right)$$

($DL_{\mathrm{K}.i}$ – Fehleranteil Komponente i).

Fehleranteil einer Baugruppe

Eine Baugruppe soll aus nachfolgenden Komponenten mit gegebenen Fehleranteilen bestehen:

Typ	Anzahl	DL_{BT}
Leiterplatte	1	10 dpm
Schaltkreise	20	200 dpm
diskrete Bauteile	35	10 dpm
Lötstellen	560	1 dpm

Welcher Fehleranteil ist für die Baugruppe zu erwarten, wenn die bei der Baugruppenfertigung zusätzlich entstehenden Fehler alle beseitigt werden:

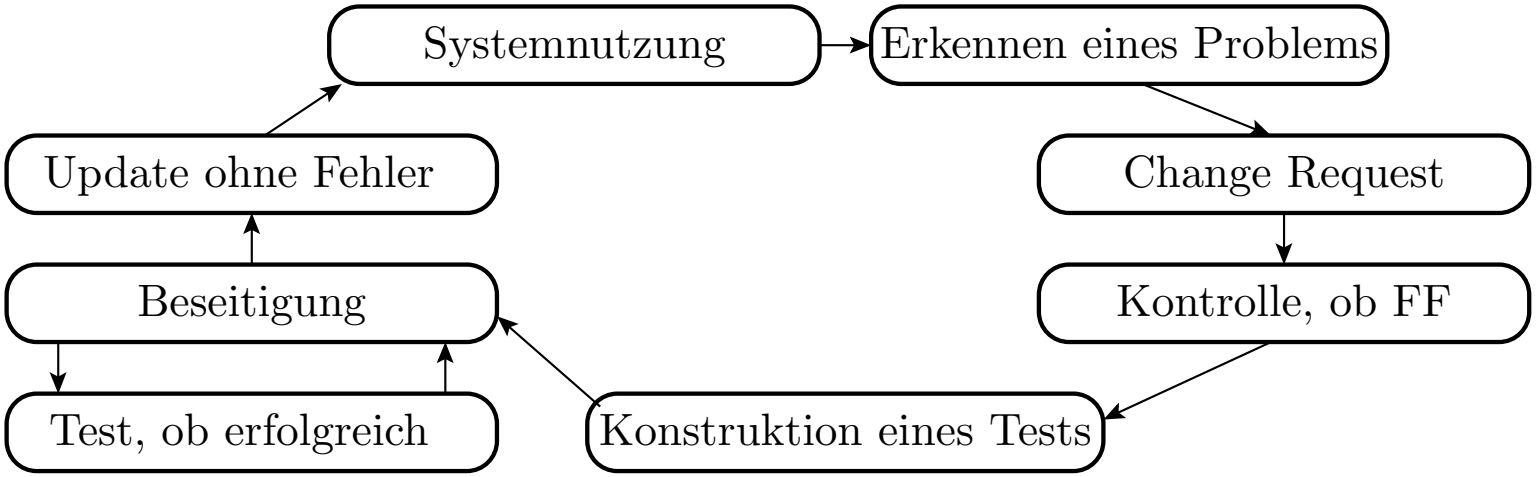
$$\begin{aligned} DL_{\mathrm{Sys}} &\approx 1-\left(1-10^{-5}\right)\cdot\left(1-2\cdot10^{-4}\right)^{20}\cdot\left(1-10^{-5}\right)^{35}\cdot\left(1-10^{-6}\right)^{560} \\ &\approx 10^{-5}+20\cdot2\cdot10^{-4}+35\cdot10^{-5}+560\cdot10^{-6} \\ &\approx 5000\,\mathrm{dpm}=0{,}005\,\mathrm{dpu} \end{aligned}$$

Reifeprozesse

Reifeprozess

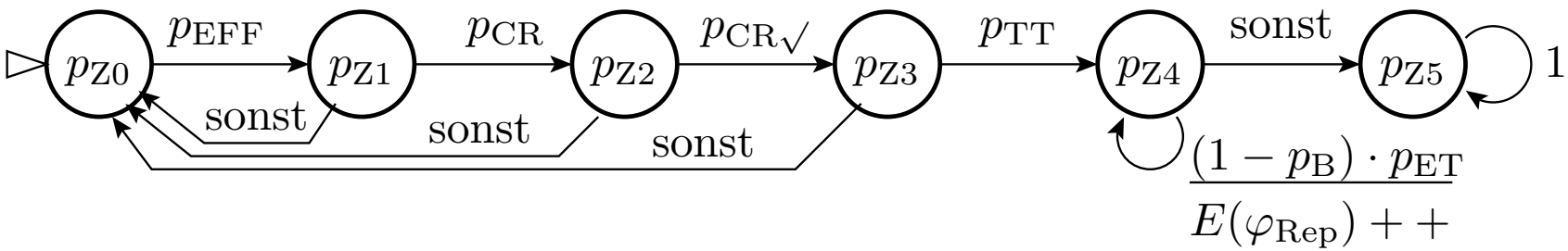
Ein Reifeprozess ist die Fortsetzung der Iteration aus Test und Fehlerbeseitigung für Entwurfsfehler großer Systeme im Einsatz.

- Die Anzahl der Entwurfsfehler in einem System nimmt mit der Systemgröße zu.
- Tests wirken wie Filter, die einen Anteil, aber nicht alle Fehler erkennen, so dass mehr gefundene auch mehr nicht gefundene Fehler erwarten lassen.
- Die Zunahme der Fehleranzahl lässt sich nicht ausreichend durch mehr / bessere Herstellertests kompensieren.
- Einbeziehung der Nutzer als Tester.



- Bei einer vermuteten Fehlfunktionen stellt der Anwender einen Änderungsanforderung (Change Request).
- Der Hersteller prüft diese, selektiert daraus FFs und versucht, für jede FF reproduzierbare Testbeispiele zu finden.
- Die Testbeispiele dienen zur Fehlerlokalisierung und zur Erfolgskontrolle nach jedem Beseitigungsversuch.
- Fehlerbeseitigung beim Nutzer erfolgt über Einspielen von Updates, in seltenen Ausnahmen über eine Rückrufaktion für Hardware oder komplette Geräte.

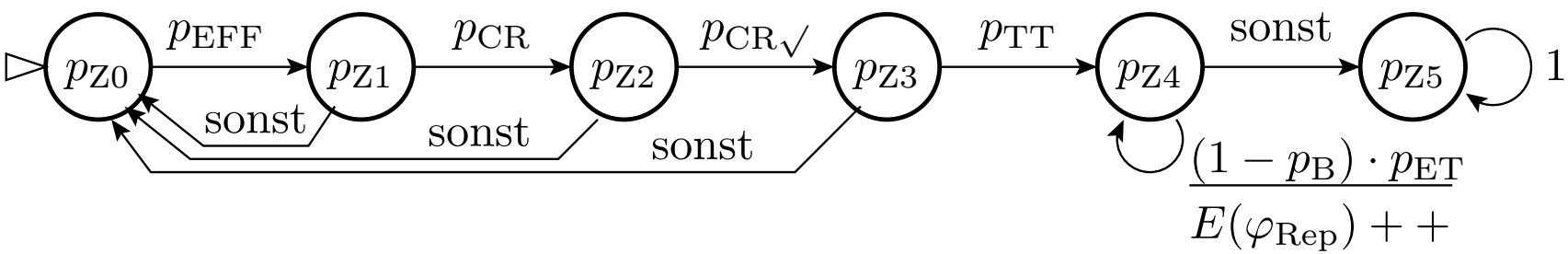
Reifeprozess für einen Fehler als Markov-Kette



p_{EFF}	Fehlererkennungswahrscheinlichkeit	Z_0	System mit Fehler i
p_{CR}	Wahrsch. Change Request gestellt	Z_1	Change Request gestellt
$p_{CR\surd}$	Wahrsch., dass als FF eingestuft	Z_2	Fehlfunktion bestätigt
p_{TT}	Wahrsch. Test konstruierbar	Z_3	Test gefunden
p_{ET}	Erkennungswahrscheinlichkeit des Tests	Z_4	Beseitigungsversuch
p_B	Beseitigungswahrscheinlichkeit	Z_5	Fehler i beseitigt
φ_{Rep}	Anzahl der bei einem Reperaturversuch entstehenden neuen Fehler		

Die Wahrscheinlichkeit, dass ein Fehler beseitigt wird, ist das Produkt der Wahrscheinlichkeiten, dass

- der Fehler bei irgendeinem Anwender eine FF verursacht,
- ein Änderungsantrag (Change Request) gestellt, ...



- ...
- der Hersteller die FF bestätigt,
- einen Test für ihren Nachweis findet,
- die Beseitigungsiteraton erfolgreich ist.

- Bei den Fehlerbeseitigungsversuchen und anderen Verbesserungsversuchen entstehen neue Fehler.
- Wenn mehr Fehler beseitigt werden als neue entstehen, reift das System.
- Reifen ist erkennbar an einer mit der Nutzungsdauer abnehmenden Häufigkeit der beobachtbaren Fehlfunktionen (Bedienprobleme, Abstürze, falsche Ergebnisse, ...).

Anmerkungen

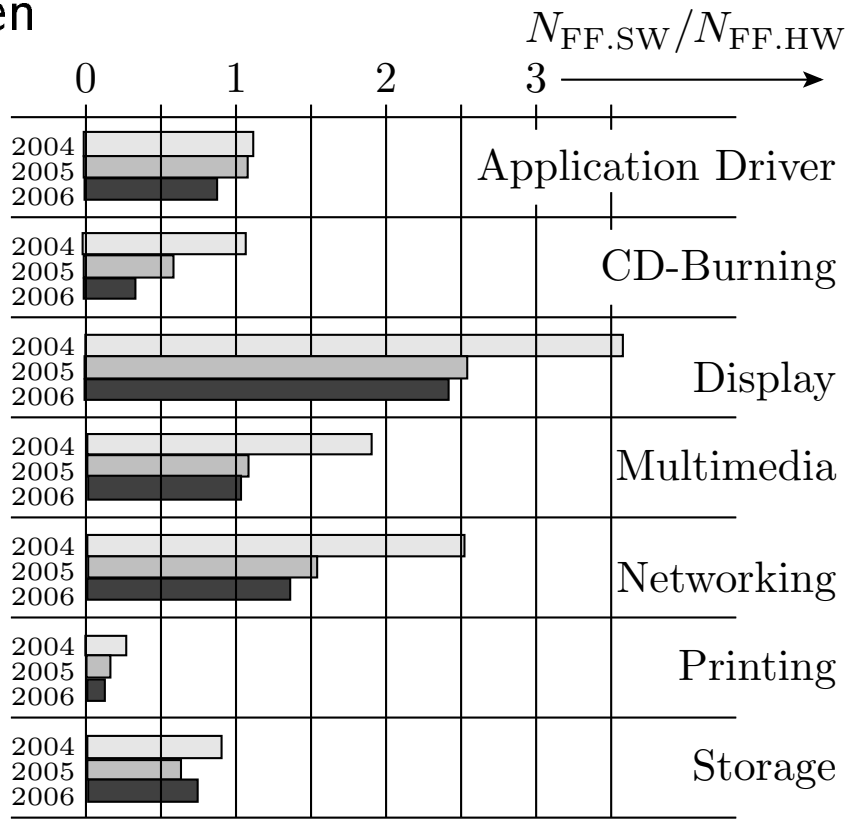
- Bei einem Reifeprozess nimmt die Zuverlässigkeit mit der akkumulierten Nutzungsdauer zu.
- Wichtig für die Geschwindigkeit eines Reifeprozesses sind der Informationsfluss über bemerkte FFs von den Anwendern zum Hersteller und ausreichend Bearbeitungskapazität des Herstellers für die Fehlerbeseitigung.
- Systeme, die lange gereift sind, haben hohe, auf anderem Wege schwer zu erreichende Zuverlässigkeiten. Schwer ersetzbar durch neue Systeme. (siehe Y2K[15] -Problem).
- Neue (innovative) Systeme sind in den ersten Nutzungsjahren vielfach unzuverlässer als die zuvor genutzten Systeme. Wenn das die Akzeptanz beeinträchtigt, reifen sie auch nicht ...

[15] Year 2000.

Zuverlässigkeitswachstum von Windows-Betriebssystemen[16]:

- Windows 98: $MTBF \approx 1$ Woche
- NT 4.0: $MTBF \approx 5{,}5$ Wochen
- Windows 2000 Professional: $MTBF \approx 4$ Monate.

Durch Treiber verursachten Abstürze unter Windows im Verhältnis zur Anzahl der durch Hardware-Fehler verursachen Abstürze[17].



[16] NSTL Test Report, Microsoft Windows 2000 Professional – Comparison of the Reliability of Desktop Operating Systems. Die Quelle sagt nicht genau, was gezählt wurde.

[17] Glerum, K., Debugging in the (Very) Large: Ten Years of Implementation and Experience (2009), S. 11-14, Fig. 15