Prof. Dr. Armin R. Mikler

Kurs 21811

Fehlertoleranz in Computersystemen und Netzwerken

LESEPROBE

Fakultät für
Mathematik und Informatik

FernUniversität in Hagen

Vorwort

Dieser Kurs besteht aus vier Kurseinheiten, die eine Einführung in das Gebiet der fehlertolerierenden Systeme vermitteln. Die erste Einheit macht den Leser durch eine Reihe von Beispielen zunächst mit dem Problem der Fehlertoleranz bekannt. Darüber hinaus sollen in der ersten Kurseinheit die Unterschiede zwischen Fehlertoleranz und Risikominimierung genauer untersucht werden. Obwohl das Gebiet der fehlertolerierenden Systeme im Allgemeinen im Rahmen der Informatik definiert ist, werden die grundlegenden Prinzipien anhand allgemeiner Beispiele dargestellt. Insbesondere werden hier verschiedene Arten von Fehlern definiert und untersucht, unter welchen Bedingungen ein System trotz Auftretens solcher Fehler weiterarbeiten kann. In der zweiten Kurseinheit werden die theoretischen Grundlagen behandelt, die im Wesentlichen ein Basiswissen der Wahrscheinlichkeitsrechnung und der Statistik umfassen. Hierbei werden die Prinzipien jeweils mittels Beispielen demonstriert, die es ermöglichen die behandelten Methoden auf verschiedene Probleme anzuwenden. In der dritten Kurseinheit werden Ansätze vorgestellt, die zu einer fehlertolerierenden Datenspeicherung und Datenübertragung beitragen. Dabei werden im Speziellen die Unterschiede zwischen Fehlererkennung und Fehlerkorrektur demonstriert. Darüber hinaus wird in der dritten Kurseinheit der Begriff der Redundanz genauer untersucht. Die vierte Kurseinheit fokussiert auf die Entwicklung fehlertoleranter Kommunikationssysteme und verteilte Rechnersysteme mittels protokollbasierter Ansätze. Verschiedene Protokollelemente werden hier auf ihren Beitrag zur Fehlertoleranz hin untersucht. Dabei dient das 7-Schichten OSI Modell dazu die unterschiedlichen Protokollelemente den jeweiligen Systemkomponenten zuzuordenen.

Am Ende jeder Kurseinheit findet der Leser eine Sammlung von Übungsaufgaben, die dazu beitragen einen Bezug zu der oft sehr abstrakten Materie der fehlertoleranten Systeme herzustellen. Zur Lösung dieser Aufgaben muss der Leser die jeweilige Kurseinheit sorgfältig bearbeiten und möglicherweise einen Blick in die verfügbare Literatur werfen.

Die Themen der einzelnen Kurseinheiten basieren auf aktuellen Fachbüchern der Fehlertoleranz, Kommunikationssysteme, Betriebssysteme, verteilte Systeme und der Wahrscheinlichkeitsrechnung und Statistik. Die hier vorgestellte Materie wird im Allgemeinen von allen Fachbüchern dieser Gebiete abgedeckt. Darüber hinaus bietet es sich an, einige spezielle Journalartikel zu konsultieren, die zu einem die historische Entwicklung der fehlertoleranten Systeme darstellen und zum anderen den aktuellen Forschungsaufwand darstellen.

Inhaltsverzeichnis

Kapitel 1

Prinzipien der Fehlertoleranz und Ausfallsicherheit

1.1 Lernziel der Kurseinheit

In dieser Kurseinheit sollen die Begriffe der Fehlertoleranz und der Risikominimierung definiert werden und darüber hinaus die folgenden Konzepte vorgestellt werden:

- Kategorien von Fehlern und Defekten,
- Redundanz,
- Kosten der Fehlertoleranz,
- Ausfallsicherheit und Verfügbarkeit.

Anhand verschiedener Beispiele, die nicht exklusiv aus der Informatik stammen, werden diese Konzepte genauer untersucht. Dabei dienen die am Ende des Kapitels gestellten Übungsaufgaben dazu das Verständnis der vorgestellten Konzepte selbst zu überprüfen. Im Folgenden werden verschiedene Konzepte vorgestellt, die in der zweiten Kurseinheit aufgegriffen und genauer behandelt werden.

1.2 Fehlertoleranz, Risiko und Sicherheit

Täglich treffen wir Entscheidungen, an denen sich unsere persönliche Einstellung zu möglichen Fehlern und unsere Risikobereitschaft widerspiegeln. In diesem Kapitel werden wir versuchen solche Konzepte etwas präziser zu definieren und Entscheidungen oder Verhaltensweisen genauer zu untersuchen. Zwar sind die Begriffe der Fehlertoleranz und der Systemsicherheit meist im Rahmen der Informatik definiert, dennoch ist es nützlich und hilfreich diese Konzepte im Kontext des täglichen Lebens zu untersuchen und zu verstehen. Während die oben aufgeführten Konzepte prinzipiell verschieden sind, ist es oft problematisch sie in expliziten Fällen zu unterscheiden und zu differenzieren. So ist zum Beispiel nicht sofort erkennbar, wie die Entscheidung, sehr früh zum Flughafen aufzubrechen, um das pünktliche Eintreffen zu gewährleisten, einzuordnen ist. Sollte man diese Entscheidung

als fehlertolerierendes oder risikominimierendes Verhalten einstufen? Die Antwort ist ein definitives „Es kommt darauf an...“. Um diese Frage zu beantworten, muss man zunächst genauer definieren, was man unter den Begriffen Fehler und Risiko in dieser Situation versteht. Ist als Fehler das Verpassen des Flugs definiert, so wird zweifelsohne ein frühes Aufbrechen zum Flughafen die Toleranz eines solchen Fehlers nicht erhöhen. Wird allerdings ein unvorhergesehener Stau auf der Autobahn oder gar eine Auto-Panne als möglicher Fehler definiert, so kann man argumentieren, dass die Maximierung der verfügbaren Zeit, den Flughafen rechtzeitig zu erreichen, tatsächlich zu einer Erhöhung der Fehlertoleranz führen kann. Das bedeutet, man kann den definierten Fehler bis zu einem bestimmten Grade tolerieren und trotz einer Verzögerung den Flughafen zeitgerecht erreichen. Wenn man ein System auf seine Fehlertoleranz hin untersuchen möchte, muss man zunächst dessen Ziele und Erwartungswerte genau festlegen.
Während das Konzept der Fehlertoleranz versucht die Effekte eines auftretenden Fehlers zu minimieren, so beschäftigt sich das Konzept der Systemsicherheit damit, die Wahrscheinlichkeit des Auftretens eines Fehlers zu minimieren oder sogar komplett auszuschließen. Die Sicherheit eines Systems wird durch den Designprozess bestimmt, in dem das System so konzipiert wird, dass die Mehrzahl der möglichen Fehlerquellen berücksichtigt werden. Diesbezüglich werden verschiedene Parameter definiert, in deren Schranken das System fehlerfrei operieren muss. Bestimmte Grenzwerte für Temperatur, Feuchtigkeit, Strahlung, Druck, usw. sind Beispiele solcher Parameter. In der Informatik konzentrieren sich solche Parameter allerdings häufig auf Datenbereiche, Datenstrukturen, numerische Grenzwerte und Abweichungen, die für ein Programm oder ein System definiert werden. Die Systemsicherheit wird generell durch verschiedene Tests verifiziert. Die Erstellung dieser Tests, insbesondere im Bereich der Softwareentwicklung, ist nicht trivial, da die Anzahl der Testfälle häufig exponentiell mit der Anzahl der zu berücksichtigenden Parameter ansteigt. Auch in anderen Anwendungsbereichen verlangt die Erstellung adäquater Tests detailliertes Wissen über das System und dessen Anwendungsumgebung.
Als Beispiel eines Systems, von dem extreme Ausfall-Sicherheit gefordert wird, dient der bekannte Flugschreiber (oder *Black Box*), die in jedem Flugzeug installiert ist und verschiedene Systemgrößen der Maschine und Kontrollentscheidungen der Piloten aufzeichnet. Sie dient dazu, im Falle eines Absturzes, die Situation zu rekonstruieren, um die Ursache durch detaillierten Einblick in den Status des Flugzeugs zur Zeit des Absturzes zu erlangen. Natürlich darf diese *Black Box* bei einem Absturz nicht zerstört werden und muss daher so konzipiert werden, dass sie gegen auftretende Kräfte und mögliche Umwelteinflüsse geschützt ist.
Die oben aufgeführten Beispiele sollen helfen, die Konzepte der Fehlertoleranz und der Systemsicherheit zu differenzieren und dem Leser einen ersten Eindruck der Vielfältigkeit dieses Themas zu vermitteln. In der Informatik werden Systeme häufig abstrahiert und durch ein zusammen arbeitendes Netz der verschiedenen Funktionskomponenten dargestellt. Dadurch lässt sich ein solches System einfacher analysieren und auf Fehlerhäufigkeit und Fehlerwahrscheinlichkeit untersuchen.

1.2.1 Fehlertoleranz

Definition 1 (Fehlertoleranz)**.** *Ein System ist* **fehlertolerant***, wenn es trotz des Auftretens unvorhergesehener Fehler weiterhin in der Lage ist seine Funktionen korrekt auszuführen.*

Ein fehlertolerantes System hat somit die Eigenschaft, dass es bei Auftreten von Fehlern graziös degeneriert, d.h. möglicherweise mit verringerter Effizienz oder Genauigkeit weiterarbeitet.

1.2.2 Systemsicherheit

Definition 2 (System Sicherheit)**.** *Ein System gilt als* **sicher***, wenn die Wahrscheinlichkeit auftretender Fehler minimiert ist.*

Das Konzept der Systemsicherheit befasst sich mit der Minimierung des Fehlerrisikos. Natürlich kann man das Auftreten von Fehlern nie völlig ausschließen.

1.2.3 Denkaufgaben

In jedem der folgenden Kontexte sollen je zwei Maßnahmen identifiziert werden, die sowohl die Ausfallsicherheit als auch die Fehlertoleranz erhöhen. Dabei soll genau definiert werden, was als Fehler angesehen werden muss.

1. Fahrzeug (z.B. Motorrad, PKW, usw.)
2. Investition im Aktien- oder Geldmarkt
3. Internet
4. Reiseplanung
5. Industrielle Projektplanung

1.3 Klassifikation möglicher Fehler

In der Umgangssprache verwenden wir Begriffe wie *Fehler, Defekt, Ausfall, und Error.* Während der jeweilige Kontext klar bestimmt was diese Ausdrücke bedeuten, muss man sie für die Fehleranalyse eines Systems genauer definieren. So beschreiben die Begriffe *Fehler, und Defekt* generell das fehlerhafte Verhalten von Systemkomponenten, wobei die Bezeichnung *Error* beschreibt, wie sich ein Fehlverhalten eines Systems manifestiert.
So wird häufig das Verhalten eines logischen Schaltkreises, der unabhängig von den jeweiligen Eingangsgrößen immer den Wert 0 aufweist, als Fehler deklariert. Ein Error tritt auf, wenn dieser Wert dann weiterverarbeitet wird, z.B. in einem Addierer, dessen Ergebnis dann demzufolge fehlerhaft (oder Error-behaftet) ist.
Betrachtet man eine Funktion, bei der dem Programmierer ein *Fehler* bei der Implementierung unterlaufen ist und daher nur positive Funktionswerte berechnet werden. Es präsentiert sich dieser Fehler allerdings nur als *Error*, falls die Funktion mit Parametern aufgerufen wird, die zu einem Ergebnis mit negativen Werten führen.

Generell kann man das Konzept eines Fehlers noch weiter differenzieren:

- **Permanente Fehler** sind Fehler, die bei Auftreten die jeweilige Komponente oder das System permanent außer Betrieb setzen.
- **Transiente Fehler** sind solche Fehler, die dazu führen, dass eine Komponente oder ein System für ein bestimmtes Zeitintervall nicht korrekt arbeitet. Nach diesem Zeitintervall ist der Fehler nicht mehr präsent und das korrekte Systemverhalten ist wiederhergestellt.
- **Sporadische Fehler** sind generell immer präsent, zeigen sich allerdings nur in unregelmäßigen Intervallen als Fehlverhalten des Systems.

Transiente und sporadische Fehler lassen sich häufig sehr schwer diagnostizieren, da ihr Auftreten oft keinem vorhersehbaren Muster folgt. So ist es zum Beispiel schwer einen Programmierfehler zu identifizieren, der nur bei bestimmten Konfigurationen der Eingangswerte oder Parameter auftritt. Hier setzt man spezielle Verfahren der Softwaretechnik ein, die sich mit dem Testen von Software und der Identifikation von Fehlern befassen, die daten- oder konfigurationsabhängig sind.
Viele Fehler können in **unkritische** und **kritische** Fehler klassifiziert werden. So ist beispielsweise ein Fehler, der ein System zu einem kompletten Stillstand bringt als *unkritisch* anzusehen, wenn keine falschen Resultate generiert werden. In diesem Kontext umfassen die *kritischen* Fehler solche, die verfälschte Werte an andere Komponenten weitergeben und somit fehlerhafte Ergebnisse erzeugen. Diese Fehler werden auch häufig als byzantinische Fehler bezeichnet, die wir in einer folgenden Kurseinheit genauer untersuchen werden. Exemplarisch für einen kritischen Fehler betrachtet man ein System, das durch das Austauschen von Messwerten mit anderen Systemen oder Komponenten Entscheidungen koordiniert. Wenn das System allen Komponenten die gleichen (korrekten) Messwerte mitteilt, können alle Teilsysteme unabhängig voneinander Entscheidungen treffen, die mit den bekannten Messwerten korrespondieren. Wenn nun das System aber unterschiedliche (falsche) Messwerte an die Teilsysteme verteilt, so werden diese nicht in der Lage sein, ein konsistentes Ergebnis zu errechnen und somit möglicherweise eine inkonsistente Entscheidung treffen.

1.4 Konzept der Redundanz

Prinzipiell basieren die fehlertolerierenden Eigenschaften eines Systems auf dem Management und Nutzen von **Ressourcen**. Die Verfügbarkeit von mehr Ressourcen als minimal notwendig, um das Funktionieren des Systems zu gewährleisten, wird als **Redundanz** bezeichnet. Dabei umfassen diese Ressourcen nicht nur die notwendigen Hardware- und Softwarekomponenten, sondern auch Zeit und Raum. So kann möglicherweise die räumliche Verteilung oder Ausbreitung von Komponenten zur Fehlertoleranz eines Systems beitragen. Dieser Ansatz wird insbesondere im Gebiet der verteilten Rechnersysteme genutzt, indem ein System so konzipiert wird, dass es keinen zentralen Punkt enthält der durch Auftreten eines Fehlers das gesamte System funktionsunfähig macht.

Die wohl am häufigsten angewandte Form der Redundanz ist die Hardwareredundanz. Dabei werden funktionskritische Elemente eines Systems dupliziert (oder multipliziert), oder es wird eine spezielle Funktion auf unabhängige Komponenten verteilt. Es soll dadurch gewährleistet werden, dass bei Auftreten eines Fehlers die Funktion weiterhin ausgeführt werden kann, obwohl es dabei zu einer Reduzierung der Qualität oder Effizienz kommen kann. Als Beispiel dafür betrachte man eine Zweikreisbremse eines PKW, bei dem die Vorderräder und Hinterräder durch zwei separate Bremskreise gesteuert werden. Sollte in einem Bremskreis ein Fehler auftreten (z.B. durch ein Leck in der Bremsleitung), so ermöglicht der andere Bremskreis noch das Fahrzeug abzubremsen. Die Bremsleistung wird durch das Auftreten eines solchen Fehlers jedoch verringert.

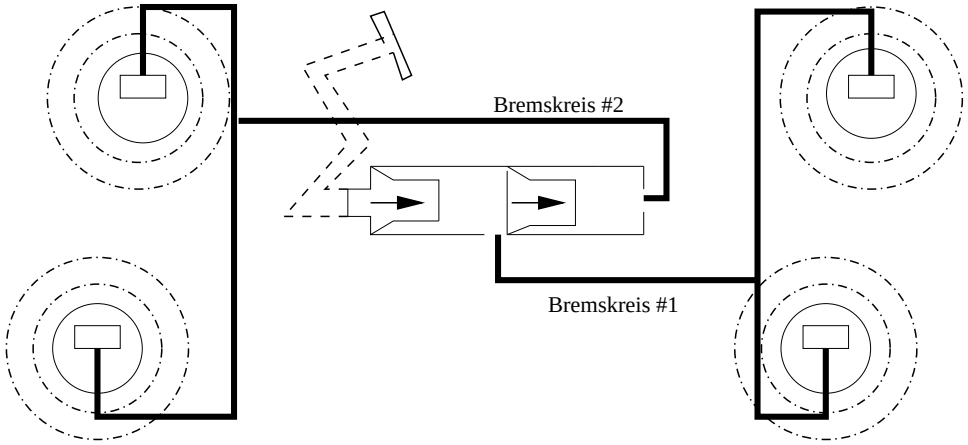

Abbildung 1.1: Zweikreisbremse eines PKW

Weiterhin lässt sich das Konzept der Redundanz in **statische** und **dynamische** Redundanz unterteilen. Während bei statischer Redundanz die redundanten Ressourcen aktiv sind, werden sie bei der dynamischen Redundanz bei Bedarf als *Ersatzkomponenten* aktiviert. Daher wird durch statische Redundanz das Auftreten eines Fehlers maskiert und es ist nicht unmittelbar erkennbar, dass eine Systemkomponente ausgefallen ist. Ein Beispiel für statische Redundanz ist die Verfügbarkeit von Plattenspeichern oder Prozessoren. Das Betriebsystem wird immer versuchen alle verfügbaren Ressourcen zu nutzen, um damit einen Lastausgleich im Rechnersystem zu erreichen. Das Ausfallen dieser Komponenten hat zur Folge, dass diese anfallende zusätzliche Last entweder auf andere Prozessoren verteilt werden muss, oder, im Fall von Speicherfehlern, eine neue Speicherverteilung der Prozesse stattfinden muss.
Ein dynamisches Umschalten auf Ersatzkomponenten erfolgt häufig automatisch, wie zum Beispiel bei einem Netzteil, welches durch einen automatischen Fault-Over im Fehlerfall ein zweites Netzteil aktivieren kann.

1.4.1 Blockdiagramme

Ein System oder ein Prozessablauf wird häufig mittels kanonischer Blockdiagramme abstrahiert. So werden beispielsweise die Komponenten eines Systems und deren Relationen

zueinander als Blockstruktur dargestellt, aus der man die verschiedenen Wege die ein Prozess in dem System nehmen kann, leicht ersehen und analysieren kann. Dabei handelt es sich bei einem solchen Blockdiagramm um einen Graphen G, dessen Knoten die Komponenten des Systems widerspiegeln. Die Kanten von G stellen dar wie die Komponenten voneinander abhängig sind und in welcher Sequenz diese abgearbeitet werden um die Gesamtfunktion des Systems zu realisieren. Während diese Systemdarstellung ihren Ursprung in der Hardware Spezifikation hat, lassen sich aber auch Softwaresysteme durch solche Blockstrukturen analysieren.
Die einfachsten Strukturen bestehen aus seriellen und parallelen Komponenten, aus denen sich die Redundanz des Systems sehr gut darstellen lässt. Ein System, in dem ein Prozess genau vier Komponenten sequentiell durchlaufen muss wird durch das folgende Diagramm dargestellt:

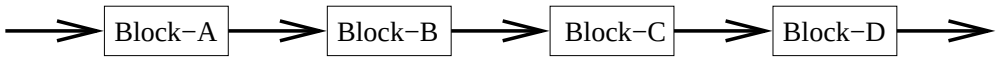

Abbildung 1.2: Diagramm serieller Systemkomponenten

Hier fällt auf, das ein Fehler in irgendeiner der vier Komponenten zu einem nicht funktionierenden System führt. Das heißt, dieses System enthält keine redundante Komponente, die im Fehlerfall die Funktionen der fehlenden Komponente übernehmen kann. Der Ausfall des Gesamtsystems kann aber auch durch einen einfachen logischen Ausdruck definiert werden. Dabei werden die Fehlerstati der Komponenten, $F(U_i)$, durch die logischen Operatoren *UND*, $\wedge$, und *ODER*, $\vee$, miteinander verknüpft.
Es seien $F(U_i)$ ein einfaches Prädikat, dass ausdrücken soll, dass Komponente U_i einen Fehler aufweist. Der logische Zustand des seriellen Gesamtsystems $F(S)$ kann nun durch den folgenden logischen Ausdruck dargestellt werden:

$$F(S) = F(U_1) \vee F(U_2) \vee ... \vee F(U_n)$$

Für das Beispiel mit vier Komponenten in Abbildung 1.2 gilt der Ausdruck:

$$F(S) = F(A) \vee F(B) \vee F(C) \vee F(D)$$

Im Gegensatz zu einem komplett seriellen System kann ein System so konzipiert sein, dass alle Komponenten parallel zueinander angeordnet sind und es somit einen hohen Grad an Redundanz aufweist. Das korrespondierende Blockdiagramm macht dieses deutlich:
Es wird deutlich, das hier nur ein Fehler aller Komponenten zu einem Fehler des Gesamtsystems führen kann. Ähnlich wie bei dem seriellen System, lässt sich das parallele System auch durch einen logischen Ausdruck beschreiben:

$$F(S) = F(U_1) \wedge F(U_2) \wedge ... \wedge F(U_n)$$

Im gegebenen Beispiel (Abbildung 1.3) gilt:

$$F(S) = F(A) \wedge F(B) \wedge F(C) \wedge F(D)$$

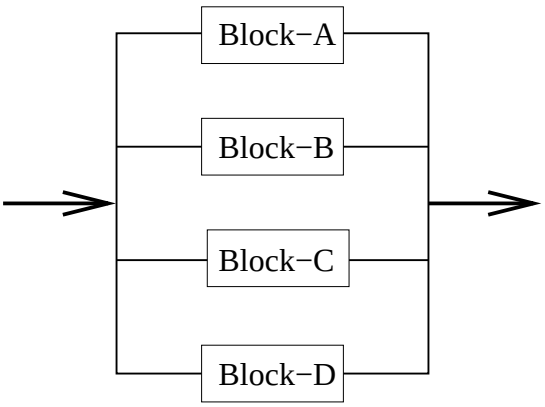

Abbildung 1.3: Diagramm paralleler Systemkomponenten

Wir werden später untersuchen, wie man die Ausfallsicherheit solcher Strukturen berechnen kann wenn die Fehlerwahrscheinlichkeiten der Komponenten bekannt sind. Dazu muss man allerdings Blockstrukturen untersuchen die über die fundamentalen seriellen und parallelen Diagramme hinausgehen. So ist es möglich, serielle und parallele Strukturen zu verbinden um komplexere Systeme darzustellen.

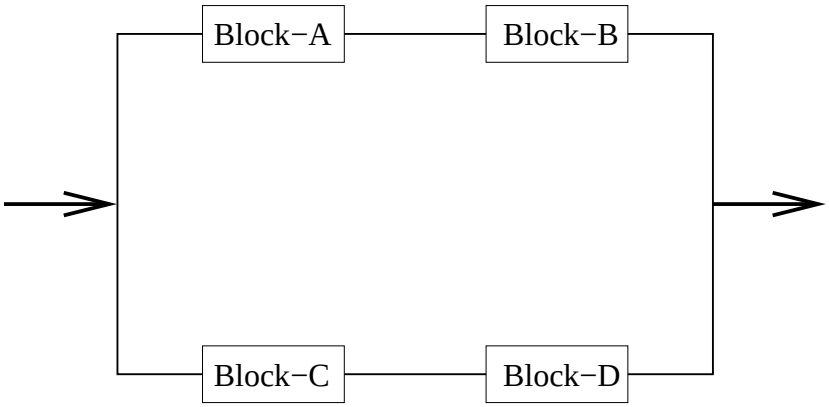

Abbildung 1.4: Serielle-Parallele Systemkomponenten

In dieser Anordnung hängt die Ausfallsicherheit und damit die Fehlertoleranz des Systems davon ab, mit welcher Wahrscheinlichkeit in beiden parallelen Pfaden (also (Block-A, Block-B) oder (Block-C, Block-D)) ein Fehler auftreten kann. Der somit entstehende logische Ausdruck für das Beispiel in Abbildung 1.4, der das Fehlverhalten des Gesamtsystems darstellt, ist somit:

$$F(S) = (F(A) \vee F(B)) \wedge (F(C) \vee F(D))$$

Um diesen Ausdruck zu verallgemeinern definieren wir $\mathcal{P}_{l,r}$ als Menge der parallelen Pfade p_i zwischen zwei Punkten l und r im Blockdiagramm. Zudem definieren wir $\mathcal{S}_{p_i}$ als Menge der seriellen Komponenten, s_j auf einem Pfad p_i. Somit ist der logische Ausdruck, der den Fehlerstatus des Gesamtsystems beschreibt:

$$F(S) = \wedge_{\forall p_i \in \mathcal{P}_{l,r}} [\vee_{\forall s_j \in \mathcal{S}_{p_i}}]$$

Die bis zu diesem Punkt diskutierten Diagramme werden im allgemeinen als Seriell-Parallel oder SP-Diagramme bezeichnet, da sie sich nur aus einfachen seriellen und parallelen Sub-Strukturen zusammensetzen.
Komplexe Systeme lassen sich allerdings nur selten durch einfache SP-Diagramme darstellen und es ist daher notwendig allgemeine, sogenannte Nicht-SP-Strukturen zu untersuchen. Wie später noch genauer gezeigt wird, beeinflusst die Systemstruktur die Methodik mit der die Ausfallsicherheit und die jeweilige Fehlertoleranz eines Systems berechnet und analysiert werden kann.
Im folgenden Beispiel handelt es sich um ein Nicht-SP Blockdiagramm, bei dem der Fehlerstatus des Gesamtsystems durch logische Ausdrücke dargestellt werden kann, die von dem jeweiligen Systempfad abhängig sind.

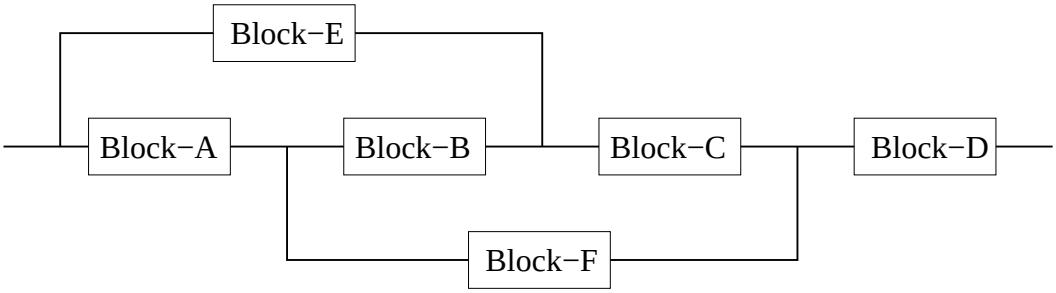

Abbildung 1.5: Nicht-S-P Diagramm

Eine genauere Betrachtung identifiziert *Block D* als eine kritische Systemkomponente, da das Versagen von *Block D* einen Totalausfall des Gesamtsystems hervorruft. Es ist allerdings nicht ganz so offensichtlich, wie sich ein Ausfall der verbleibenden Komponenten auf den Fehlerstaus des Gesamtsystems auswirkt. Um das Fehlerverhalten des Gesamtsystems durch einen logischen Ausdruck zu beschreiben müssen alle möglichen Fehlersituationen berücksichtigt werden, die zu einem Ausfall des Gesamtsystems führen. Man kann dieses Fehlerverhalten einfach durch das Erstellen einer Wertetabelle darstellen, wobei T und F den Fehlerstatus jeder der Komponenten repräsentiert. Wir verwenden hier T für eine fehlerfreie Komponente und F für einen Komponentenfehler. Da *Block D* schon als kritische Komponente identifiziert wurde, wird die folgende Wertetabelle nur den Fehlerstatus der verbleibenden Komponenten darstellen.

Die Ausfallanalyse mittels Wertetabelle ist offensichtlich nur für relativ kleine Subsysteme möglich. Für größere Systeme wird diese Methode schnell unübersichtlich und verlangt somit die Entwicklung rechnergestützter Werkzeuge zur System- und Fehleranalyse.
Generell muss man zwischen *notwendigen* und *hinreichenden* Fehlerbedingungen unterscheiden. So ist das Versagen aller möglichen Pfade zwischen zwei Punkten X_i, X_j im System eine notwendige Fehlerbedingung. Mathematisch wird dies wie folgt beschrieben:

A	B	C	E	F	SYSTEM
T	T	T	T	T	T
T	T	T	T	F	T
T	T	T	F	T	T
T	T	T	F	F	T
T	T	F	T	T	T
T	T	F	T	F	F_2
T	T	F	F	T	T
T	T	F	F	F	F_3
T	F	T	T	T	T
T	F	T	T	F	T
T	F	T	F	T	T
T	F	T	F	F	F_3
T	F	F	T	T	T
T	F	F	T	F	F_3
T	F	F	F	T	T
T	F	F	F	F	F_4
F	T	T	T	T	T
F	T	T	T	F	T
F	T	T	F	T	F_2
F	T	T	F	F	F_3
F	T	F	T	T	F_2
F	T	F	T	F	F_3
F	T	F	F	T	F_3
F	T	F	F	F	F_4
F	F	T	T	T	T
F	F	T	T	F	T
F	F	T	F	T	F_3
F	F	T	F	F	F_4
F	F	F	T	T	F_3
F	F	F	T	F	F_4
F	F	F	F	T	F_4
F	F	F	F	F	F_5

Tabelle 1.1: Tabelle aller möglichen Systemzustände zur Fehleranalyse

$F(S) = 1 \implies \neg\exists(Pfad(X_i, X_j))$, wobei X_i und X_j in unserem System den Eingang und Ausgang zum Gesamtsystem darstellen. Diese Implikation sagt allerdings nichts über die Fehlerbedingungen individueller Komponenten aus, die ausreichen, um alle Pfade zwischen X_i und X_j zu unterbrechen. Die hinreichenden Fehlerbedingungen unseres Beispielsystems sind in der folgenden Tabelle zusammengefasst. Diese Minimalbedingungen (oder hinreichenden Bedingungen) können durch Inspektion der Wertetabelle und des Systemdiagramms hergeleitet werden. Dabei wird eine Bedingung der Form $F(U_1) \wedge F(U_2) \wedge ...F(U_k)$ nur dann als minimal bezeichnet wenn ein Fehler aller k Komponenten für den Ausfall des Gesamtsystems notwendig ist

1 Komponentenfehler	keine (nur Block D im Gesamtsystem)
2 Komponentenfehler	$(A, E), (A, C), (C, F)$
3 Komponentenfehler	(B, E, F)
4 Komponentenfehler	keine

Das bedeutet, dass nur ein Versagen aller möglichen Pfade im System zu einem totalen Systemfehler führt. So ist das gleichzeitige Versagen der Komponenten A und C eine hinreichende minimale Bedingung für das Versagen des Systems. Wie wir später sehen werden, muss die Fehlerwahrscheinlichkeit für jeden dieser Pfade berechnet werden. In Kapitel zwei werden dazu die nötigen theoretischen Grundlagen diskutiert.
Auch ohne die Anwendung theoretischer Grundlagen lassen sich einige Merkmale des Systems in Abbildung 1.5 erkennen. So kann man die Systemkomponente D klar als kritischsten Punkt des Systems identifizieren, da alle möglichen Pfade sie einschließen. Dabei ist Komponente B am unkritischsten, denn nur der Pfad $A - B - C - D$ enthält sie.
Die Ausfallanalyse eines seriell-parallelen (SP) Systems ist generell einfacher als die eines nicht-SP Systems, wie im bearbeiteten Beispiel. Der Grund dafür besteht darin, dass ein SP System hierarchisch zerlegt werden kann und alle seine Komponenten individuell analysiert werden können.

Theorem 1. *Ein Gesamtsystem, bestehend aus Komponenten k_i, kann genau dann durch ein SP Diagramm dargestellt werden, wenn der korrespondierende logische Ausdruck $F(S)$, der den Fehlerstatus des Gesamtsystems ausdrückt, jedes k_i genau einmal enthält.*

Beweis: Der Beweis wird dem Leser als Übungsaufgabe überlassen (siehe Aufgaben am Ende der Kurseinheit).

1.5 Bewertung der Fehlertoleranz

Traditionell hat man Systeme auf ihre Ausfallsicherheit und Verfügbarkeit untersucht, indem man durch Testen die mittleren Zeitintervalle bis zum Auftreten eines Fehlers bestimmt. Dazu wird getestet wie schnell eine fehlerbehaftete Komponente repariert werden kann, da dies die Ausfalldauer direkt beeinflusst. Die Nomenklatur kommt aus dem Englischen und ist in der folgenden Tabelle kurz zusammengefasst:

Englisch	Deutsch	Abkürzung
Reliability	Ausfallsicherheit	$R(t)$
Availability	Verfügbarkeit	$A(t)$
Mean Time to Failure	mittlere ausfallfreie Zeit	MTTF
Mean Time to Repair	mittlere Ausfalldauer	MTTR
Mean Time Between Failures	mittlerer Ausfallabstand	MTBF

Anhand dieser Fehlergrößen kann man nun verschiedene Systemwerte definieren.

Definition 3 (Fehlerrate)**.** *Die zu erwartende Anzahl von auftretenden Systemfehlern, also die Fehlerrate λ, wird wie folgt berechnet:*

$$\lambda = \frac{1}{MTTF}$$

Definition 4 (Reparaturrate)**.** *Die Anzahl der möglichen Reparaturen, die in einer Zeiteinheit durchgeführt werden können wird durch die Reparaturrate μ beschrieben. Da jede Reparatur im Durchschnitt $MTTR$ Zeiteinheiten benötigt, wird die Reparaturrate mit Hilfe der mittleren Ausfalldauer wie folgt berechnet:*

$$\mu = \frac{1}{MTTR}$$

Definition 5 (mittlere Fehlerhäufigkeit)**.** *Die Häufigkeit, mit der Systemfehler auftreten können, werden von den Größen $MTTF$ und $MTTR$ bestimmt. Bei genauerer Betrachtung der Fehlergrößen wird klar, dass die Zeit zwischen zwei Fehlern nicht kleiner als $MTTF + MTTR$ sein kann, denn während der Reparaturzeit ist das System nicht funktionsfähig. Daher wird die mittlere Fehlerhäufigkeit ν wie folgt berechnet:*

$$\nu = \frac{1}{MTTF + MTTR}$$

Mit diesen Definitionen, lässt sich nun die Verfügbarkeit, bzw. die Unverfügbarkeit des Systems ausdrücken. Die *stationäre* Unverfügbarkeit U ist somit

$$U = \frac{MTTR}{MTTF + MTTR} \tag{1.1}$$

Mit $MTBF = MTTF + MTTR$, U kann als

$$U = \frac{MTTR}{MTBF} \tag{1.2}$$

berechnet werden.

Die Unverfügbarkeit ist somit das Verhältnis der durchschnittlichen Reparaturzeit und dem durchschnittlichen Zeitintervall, in dem kein Fehler auftritt. Damit lässt sich die (stationäre) Verfügbarkeit V direkt ausdrücken:

$$V = 1 - U \tag{1.3}$$

Es lässt sich durch algebraische Umformung zeigen das:

$$V = 1 - U = \frac{MTTF}{MTBF} = \frac{MTTF}{MTTF + MTTR} \tag{1.4}$$

Damit beschreibt die Verfügbarkeit das Verhältnis der durchschnittlichen Zeit bis zum Auftreten eines Fehlers und dem durchschnittlichen Zeitintervall, in dem kein Fehler auftritt. Nach einer kurzen Einführung in die Wahrscheinlichkeitsrechnung in der Kurseinheit *Theoretische Grundlagen* wird klar, wie die oben berechneten Fehler und Ausfallraten als Wahrscheinlichkeiten interpretiert werden und somit die Basis für einen stochastischen Ansatz zur Systemanalyse führen.

1.6 Anforderungen an moderne Systeme

Die Anzahl der Operationen und Transaktionen, die moderne Rechner- oder Kommunikationssysteme heute durchführen sollen, verlangen einen hohen Grad an Verfügbarkeit. Schon ein relativ kurzzeitiger Systemausfall kann zu exorbitanten Verlusten führen. Systemingenieure müssen daher ein solches System so konzipieren, dass es den Ansprüchen der jeweiligen Anwendungen gerecht wird. Prinzipiell sind Fehler niemals vermeidbar und somit müssen die einzelnen Systemkomponenten eine Ausfallsicherheit aufweisen, die die geforderte Verfügbarkeit des Gesamtsystems gewährleistet.
Damit lassen sich Ausfallsicherheit und Verfügbarkeit folgendermaßen unterscheiden:

- *Ausfallsicherheit* ist ein (analytisches) vom System festgelegtes Leistungsmerkmal
- *Verfügbarkeit* ist ein vom Benutzer wahrgenommenes Leistungsmerkmal

Es ist somit das Ziel eines fehlertolerierenden Systems die Verfügbarkeit zu maximieren. Eine häufig angewandte Charakterisierung eines Systems ist die **5 x 9** Eigenschaft, die beschreiben soll, dass das System eine Verfügbarkeit von 0.99999 aufweist. Um einen Bezug zu der 5 x 9 Eigenschaft zu entwickeln, betrachte man das folgende Beispiel:

%-Verfügbarkeit V	maximale Ausfallzeit
99%	3 Tage, 15 Stunden, 36 Minuten
99.9%	8 Stunden, 45 Minuten
99.99%	52 Minuten, 36 Sekunden
99.999%	5 Minuten, 15 Sekunden
99.9999%	32 Sekunden

Tabelle 1.2: %-Verfügbarkeit vs. akzeptable Ausfallzeit pro Jahr

Beispiel 1 (Prozentuale Verfügbarkeit)**.** *Ein Jahr hat ca. 525960 Minuten. Damit ergeben sich die folgenden Werte für die Verfügbarkeit und die nicht-planmäßige Ausfallzeit*

Die Werte in Beispiel 1 wurden wie folgt berechnet:

$$1\ Jahr = 525960\ Minuten$$

$$Betriebszeit\ pro\ Jahr = V \times 525960$$

$$U = 525960 - Betriebszeit$$

$$Akzeptable\ Ausfallzeit = (1 - V) * Zeit$$

1.6.1 Beispiele zur Analyse der Verfügbarkeit

Die prozentuale Verfügbarkeit einer Komponente lässt sich leicht anhand des folgenden Beispiels darstellen:

Beispiel 2. *Angenommen die mittlere ausfallfreie Zeit (MTTF) des Netzteils eines Routers ist 18 Jahre. Um einen Fehler zu beheben kann das Netzteil entweder repariert oder ausgetauscht werden. Die Fehlersuche und Reparatur nehmen 24 Stunden in Anspruch ($MTTR_1 = 24h$), während ein Austausch innerhalb von 30 Minuten ausgeführt werden kann ($MTTR_2 = 0.5h$). Wie beeinflusst die Reparatur bzw. der Austausch des Netzteils die Verfügbarkeit?*

Zunächst berechnet man die MTTF in Stunden:

$$MTTF = 18Jahre = 18 \times 365.25 \times 24 = 157788h$$

Die Verfügbarkeit für $MTTF_1$ und $MTTF_2$ ergeben sich dann wie folgt:

$$V_1 = \frac{157788}{157788 + 24} = 0.99984792 \approx 99.985\%$$

$$V_2 = \frac{157788}{157788 + 0.5} = 0.999996831 \approx 99.99968\%$$

1.6.2 Verfügbarkeit eines Systems

Wie schon im Kapitel 1.4.1 beschrieben, lassen sich viele Systeme durch serielle, parallele und seriell-parallele Diagramme beschreiben. Mittels der Verfügbarkeit der einzelnen Komponenten, lässt sich die Verfügbarkeit des Gesamtsystems berechnen. Die Herleitung der jeweiligen Formeln zur Berechnung der Verfügbarkeit werden in Kapitel 2 genauer beschrieben.

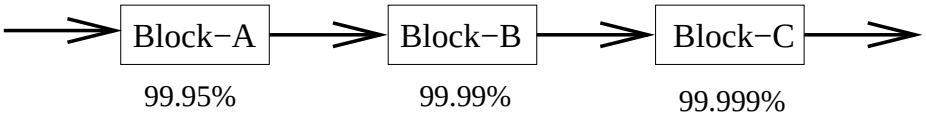

Abbildung 1.6: Diagramm serieller Systemkomponenten

Die Verfügbarkeit eines seriellen Systems ist das Produkt der Verfügbarkeiten der einzelnen Komponenten. Dementsprechend wird die Verfügbarkeit des System im Abbildung 1.6 wie folgt berechnet:

$$V_{sys} = V_A \times V_B \times V_C = 99.95\% \times 99.99\% \times 99.999\% = 99.939\%$$

In einem seriellen System müssen alle Komponenten verfügbar (also funktionsfähig) sein, damit die Funktion des Gesamtsystems gewährleistet ist. Man stellt fest, das V_{sys} immer kleiner ist als die kleinste Verfügbarkeit jeder der Komponenten: $V_{sys} \leq min(V_i) \forall i$

Denkaufgabe: Wie würde man die Unverfügbarkeit U dieses Systems berechnen? (siehe Aufgaben am Ende der Kurseinheit)

Die Produktform zur Berechnung der seriellen Verfügbarkeit muss allerdings modifiziert werden um die Verfügbarkeit eines parallelen Systems zu berechnen.

Für das System in Abbildung 1.7 ergibt sich somit die folgende Formel:

$$V_{sys} = 1 - ((1 - V_A) \times (1 - V_B) \times (1 - V_C)) =$$

$$1 - ((1 - .9995) \times (1 - .9999) \times (1 - .99999)) = .9999999 \approx 99.99999\%$$

In dieser Formel stellen die Terme $(1 - V_i)$ die Unverfügbarkeit der Komponenten dar. Das Produkt der Terme ist daher die Unverfügbarkeit des parallelen Systems, bei dem alle Komponenten ausfallen müssen um einen Fehler des Gesamtsystems zu erzwingen.
Das Berechnen der Verfügbarkeit in einem gemischten seriell-parallel System wird als Übungsaufgabe am Ende dieser Kurseinheit betrachtet.

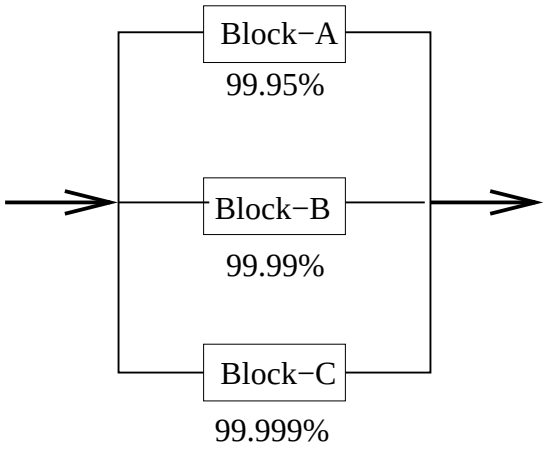

Abbildung 1.7: Diagramm paralleler Systemkomponenten

1.7 Kosten der Fehlertoleranz und Systemsicherheit

Nachdem nun die einführenden Konzepte der Fehlertoleranz vorgestellt wurden, sollte man ein Gefühl dafür entwickeln welche Kosten durch die Fehlertoleranz und Ausfallsicherheit entstehen. Sicherlich sind erhöhte Verfügbarkeit und Ausfallsicherheit mit höheren Entwicklungskosten verbunden, denn eine sorgfältige Systemanalyse im Bezug auf Fehlertoleranz ist recht aufwendig.
Es ist allerdings ungenügend nur die Entwicklungskosten zu betrachten. Die notwendige Redundanz erfordert zusätzliche Komponenten die für das System bereitgestellt werden müssen. Somit schlägt sich die Bereitstellung eines Ersatzrades an einem PKW sicher in seinen Herstellungskosten und Verkaufspreis nieder. Fehlertoleranz und Ausfallsicherheit werden somit wirtschaftstechnische Probleme, die eine Preis-Leistungs- oder Preis-Risiko-Analyse erfordern. Für Systeme deren Ausfall katastrophale Konsequenzen mit sich bringen, sind daher sehr hohe Kosten für die Bereitstellung redundanter Komponenten akzeptabel.
Zusätzlich zu den Kosten der Entwicklung und der Bereitstellung redundanter Komponenten, müssen aber auch Kosten für Zeit, Platz (Raum), und Personal mit in die Kostenrechnung einbezogen werden. So muss zum Beispiel garantiert sein, dass ein Techniker bereitsteht, um das Netzteil, dessen Verfügbarkeit im Beispiel 2 berechnet wurde, austauschen zu können. Nur so kann die erwartete Verfügbarkeit garantiert werden. Da in den meisten Fällen die Fehlertoleranz rund um die Uhr, 365 Tage im Jahr, gewährleistet sein muss, kann der Personalbedarf einen bedeutenden Kostenfaktor darstellen.
Für automatisierte Fehlersysteme werden häufig die redundanten Komponenten in das System integriert um im Fehlerfall automatisch zugeschaltet werden zu können. Das verlangt allerdings eine Vergrößerung der Dimensionen, in denen das System konzipiert ist. Parallele Systeme stellen größere Betriebskapazitäten bereit, verwenden aber nur einen Bruchteil der Kapazität und halten eine Kapazitätsreserve für den Fehlerfall vor. Da dieser Ansatz keine Komponenten ungenutzt lässt, darf dieses System im Normalfall nie 100% ausgelastet sein, um weiterhin auf einen Systemfehler reagieren zu können. Somit muss ungenutzte

Kapazität in die Kostenrechnung mit einbezogen werden.
Es ist häufig schwer die Kosten für Fehlertoleranz und Ausfallsicherheit zu quantifizieren. So wird das zusätzliche Gewicht des Ersatzrads in einem PKW sicher zu einem erhöhtem Kraftstoffverbrauch beitragen. Die extra Zeit, die wir in unserem ersten Beispiel einplanen um rechtzeitig einen Flug zu erreichen, falls ein Unfall oder Stau auf dem Weg zum Flughafen eintritt, erhöht die Anfahrzeit drastisch. Der Aufwand für Zeit und Raum, notwendig um die Ausfallsicherheit und Verfügbarkeit zu erhöhen oder das Fehler Risiko zu senken, lassen sich oft nur sehr schwer quantifizieren.
Die akzeptablen Kosten für die Fehlertoleranz werden nicht nur quantitativ analysiert, sondern unterliegen auch qualitativen Betrachtungen. Fehlerhäufigkeit, Risiko, Daten- Verlust und -Sicherheit werden meist statistisch erfasst und verlangen daher eine Analyse, die auf den Prinzipien der Wahrscheinlichkeitsrechnung basiert. In der folgenden Kurseinheit werden die Axiome der Wahrscheinlichkeitsrechnung wiederholt an Beispielen der Fehlertoleranz erläutert.

1.8 KE-1 Übungsaufgaben

Aufgabe-1:
Für jede der folgenden Fehlerarten sind je zwei Beispiele zu entwickeln:

a. Permanente Fehler;

b. Transiente Fehler;

c. Sporadische Fehler;

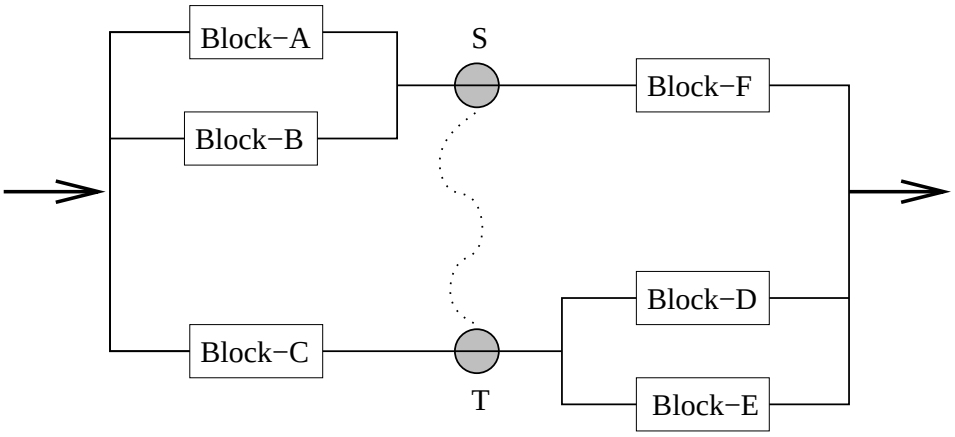

Abbildung 1.8: Blockdiagramm für die folgenden Aufgaben

Aufgabe-2:
Für das Blockdiagramm in Abbildung 1.8 ohne die Verbindung der Punkte S und T ist ein logischer Ausdruck zu entwickeln, der die Fehlerbedingungen des Systems beschreibt.

Wie ändert sich der logische Ausdruck wenn man S und T verbindet? Dabei muss die Durchlaufrichtung berücksichtigt werden:

- Wenn der Systemfluss auf $S \longrightarrow T$ beschränkt ist;
- Wenn der Systemfluss auf $T \longrightarrow S$ beschränkt ist;
- Wenn die Verbindung (S, T) bidirektional durchlaufen werden darf;

Aufgabe-3:
Für das System in Abbildung 1.8 sind die *hinreichenden* Fehlerbedingungen für das Auftreten von $1, 2, \ldots, 6$ Komponentenfehlern zu ermitteln.

Wie ändern sich diese Bedingungen wenn die Verbindung (S, T) berücksichtigt wird? Ändern sich die hinreichenden Bedingungen für einen Systemfehler mit Restriktionen des Systemflusses?

Aufgabe-4:
Es ist formal zu beweisen das:

$$V = 1 - U = \frac{MTTF}{MTBF} = \frac{MTTF}{MTTF + MTTR}$$

Aufgabe-5:
Theorem 1 ist formal zu beweisen.

Aufgabe-6:
Man konstruiere ein Blockdiagramm, welches das fehlertolerante Verhalten eines PKWs mit vier Rädern und einem Ersatzrad darstellt. (Hinweis: Mögliche Kodierung der Elementegruppen)

Aufgabe-7:
Man berechne die Verfügbarkeit des Systems in Abbildung 1.8 wenn die Komponenten $A - F$ eine Verfügbarkeit von je 95% besitzen. Wie verändert sich die Systemverfügbarkeit mit dem Einfügen der (S, T) Verbindung?

Aufgabe-8:
Angenommen das System in Abbildung 1.8 besteht aus Komponenten die innerhalb von 24 Stunden ersetzt oder repariert werden können. Man berechne die Verfügbarkeit des Systems wenn die MTTF für die Komponenten wie folgt vorliegen:

- $Block - A = 12\ Jahre$
- $Block - B = 12\ Jahre$
- $Block - C = 18\ Jahre$
- $Block - D = 15\ Jahre$
- $Block - E = 15\ Jahre$
- $Block - F = 24\ Jahre$