

# Q6. Zuverlässigkeitsanalyse

## Gliederung

1. Quantitative Bewertung der Systemzuverlässigkeit
2. Verfügbarkeit von Komponenten und einfachen Systemen
3. Markov-Prozesse zur Zuverlässigkeitsmodellierung
4. Nicht-Markovsche-Modelle zur Zuverlässigkeitsanalyse
5. Zuverlässigkeit und Leistung

## Literatur (primär)

- D. A. Menasce, V. A. F. Almeida, . Capacity Planning for Web Services: metrics, models, and methods. Prentice Hall 2001. Kap. 11.
  - J. Pukite, P. Pukite. Modeling for Reliability Analysis. IEEE Press 1998.
  - K. W. Gäde. Zuverlässigkeit. Mathematische Modelle. Hanser 1977.
  - R. A. Sahner, K. S. Trivedi, A. Puliafito. Performance and Reliability Analysis of Computer Systems. Kluwer 1996.
- Software:
- <http://umn.edu/~pub/carms.html>

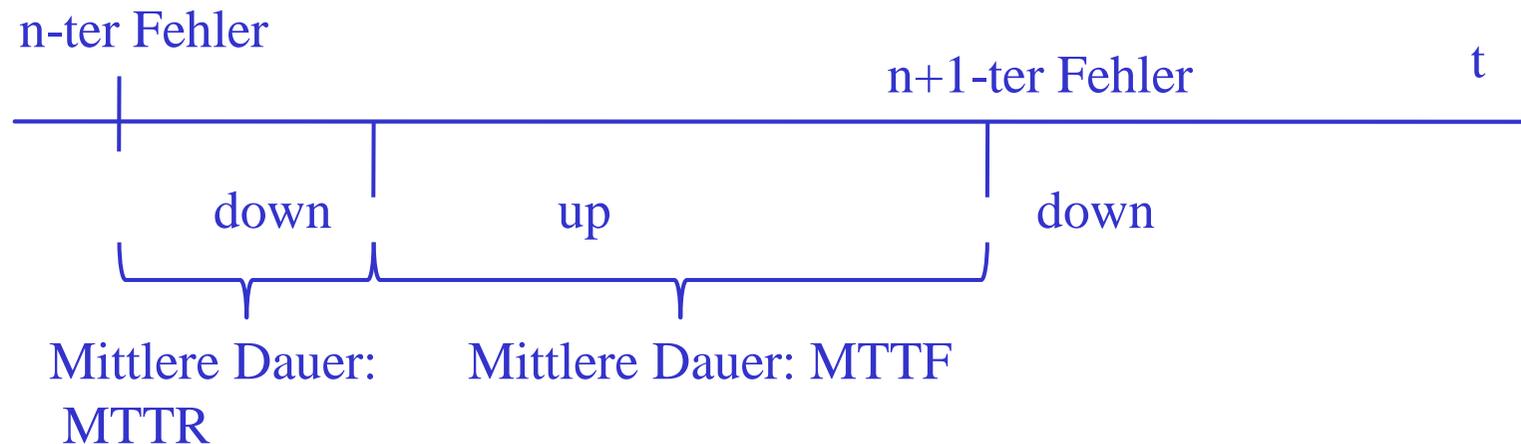
## Ziele:

- Grundlegende Maße der Systemzuverlässigkeit kennen lernen
- Modelle für zuverlässige Hardware- und Software-Komponenten erstellen und einsetzen können
- Effekte von Redundanz erkennen und bewerten können
- Markov-Modelle zur Zuverlässigkeitsanalyse nutzen können
- Ausfallzeiten modellieren können
- Systeme bzgl. Leistung im Fehlerfall bewerten können

## Q6.1 Quantitative Bewertung der Systemzuverlässigkeit

Verschiedene Zuverlässigkeitsgrößen wurden bereits definiert  
nun modellgestützte Analyse dieser Größen

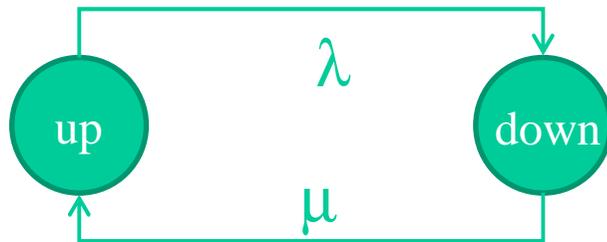
### Typischer Ablauf



Wahrscheinlichkeiten:  $P[\text{down}] = \text{MTTR} / (\text{MTTR} + \text{MTTF})$

$E(V) = P[\text{up}] = \text{MTTF} / (\text{MTTR} + \text{MTTF})$   
(mittlere Verfügbarkeit/Availability)

Einfaches Modell:



Mit  $\lambda = 1 / \text{MTTF}$   
 $\mu = 1 / \text{MTTR}$

Zeiten bis zum Fehler und bis zum Ende der Reparatur sind Zufallsvariablen!

Verteilungsfunktion der Zeit bis zum Fehler  $F_L(t)$  und bis zur Reparatur  $F_R(t)$

Einfluss von  $E(V)$ ??

Kein Einfluss, nur MTTF und MTTR sind relevant!

Dies gilt nicht für andere Maße!

Z.B.  $\text{Var}(V)$ ,  $P[\text{kein Fehler in } [t, t+\Delta) \mid \text{System im Zustand up zum Zeitpunkt } t]$ ,

...

# Verfügbarkeitsklassen

Klasse	Verfügbarkeit	Unverfügbarkeit (Min./Jahr)	Systemtyp
1	90.0%	52 560	Unmanaged
2	99.0%	5 256	Managed
3	99.9%	526	Well-Managed
4	99.99%	52.6	Fault-Tolerant
5	99.999%	5.3	Highly Available
6	99.9999%	0.53	Very Highly Available
7	99.99999%	0.0053	Ultra Available

- Heutige Systeme (Stromnetze, Telefonnetze, ...) sollen zur Klasse 5 gehören
- Sicherheitskritische Systeme oft in Klasse 7

Problem der Validierung/Analyse:

Ausfälle sind seltene Ereignisse im realen System und im Modell!

Erhöhung der Verfügbarkeit:

1. Durch Einsatz höherwertiger Technologie

bessere Bauteile, höherer Fertigungsaufwand, bessere Kontrollen, ...

oft teuer und durch Technologie begrenzt

aber oft relativ gut analysierbar/validierbar

2. Durch Redundanz bei Nutzung einfacher Komponenten

wenn mehrere Komponenten vorhanden sind, muss nur eine korrekt arbeiten

und es muss klar sein, welche korrekt arbeitet

oft relativ einfach und kostengünstig realisierbar

aber oft schwierig zu analysieren/validieren

Verfügbarkeitsanalyse von Systemen:

➤ Systeme sind aus Komponenten aufgebaut

➤ System funktioniert, wenn bestimmte Menge der Komponenten funktioniert

➤ Bestimmung der Systemverfügbarkeit aus der Komponentenverfügbarkeit

## Q6.2 Verfügbarkeit von Komponenten und einfachen Systemen

Wir betrachten zwei Modelltypen:

- Zuverlässigkeitsblockdiagramme
- Fehlerbäume

Ziel in beiden Fällen,

Berechnung der Systemverfügbarkeit aus der Komponentenverfügbarkeit

Implizite Annahmen:

- Komponenten sind unabhängig (bzgl. Ausfall und Reparatur)
- Systemfehler entstehen nur aus Komponentenfehlern
- Ausfall jeder Komponente ist erkennbar

## Zuverlässigkeitsblockdiagramme:

### Serienschaltung



Sei

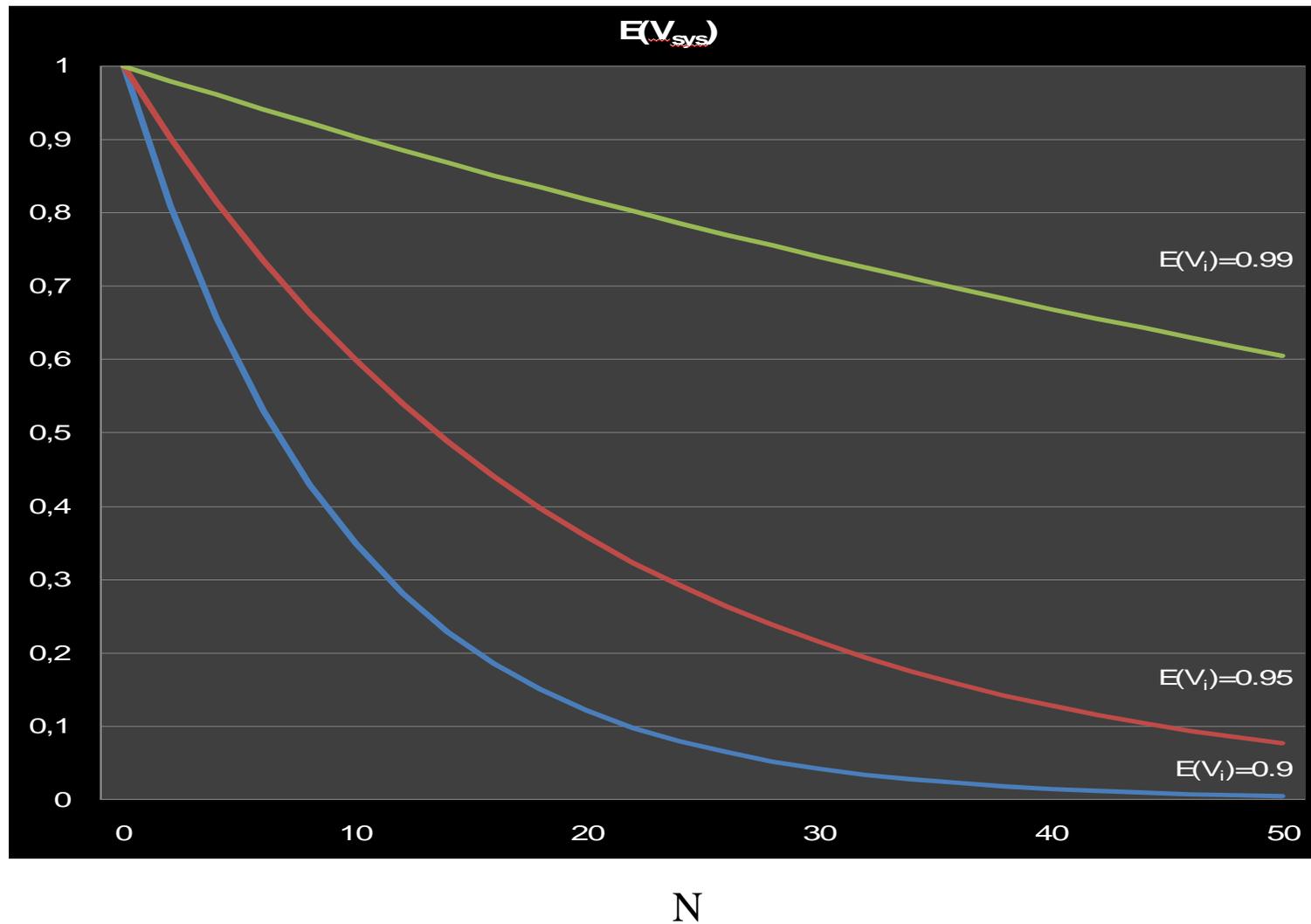
$F_i(t)$  ( $i=1, \dots, N$ ) die Verteilungsfunktion der Zeit bis zum Fehler in Komponente  $i$

$F_{sys}(t)$  die Verteilungsfunktion der Zeit bis zum Systemfehler

Es gilt dann  $F_{sys}(t) = \prod_{i=1}^N F_i(t)$ , wenn Komponenten nicht(!) repariert werden

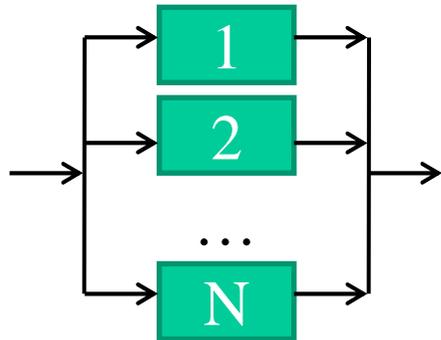
Falls alle Komponenten unabhängig repariert werden gilt  $E(V_{sys}) = \prod_{i=1}^N E(V_i)$

Verfügbarkeit von Serienschaltungen identischer Komponenten:  
 $E(V_i)$  Komponentenverfügbarkeit,  $E(V_{sys})$  Systemverfügbarkeit



## Zuverlässigkeitsblockdiagramme:

Parallelschaltung



System arbeitet, wenn mindestens eine Komponente arbeitet!

Es gilt dann  $F_{sys}(t) = 1 - \prod_{i=1}^N (1 - F_i(t))$ , wenn Komponenten nicht(!) repariert werden

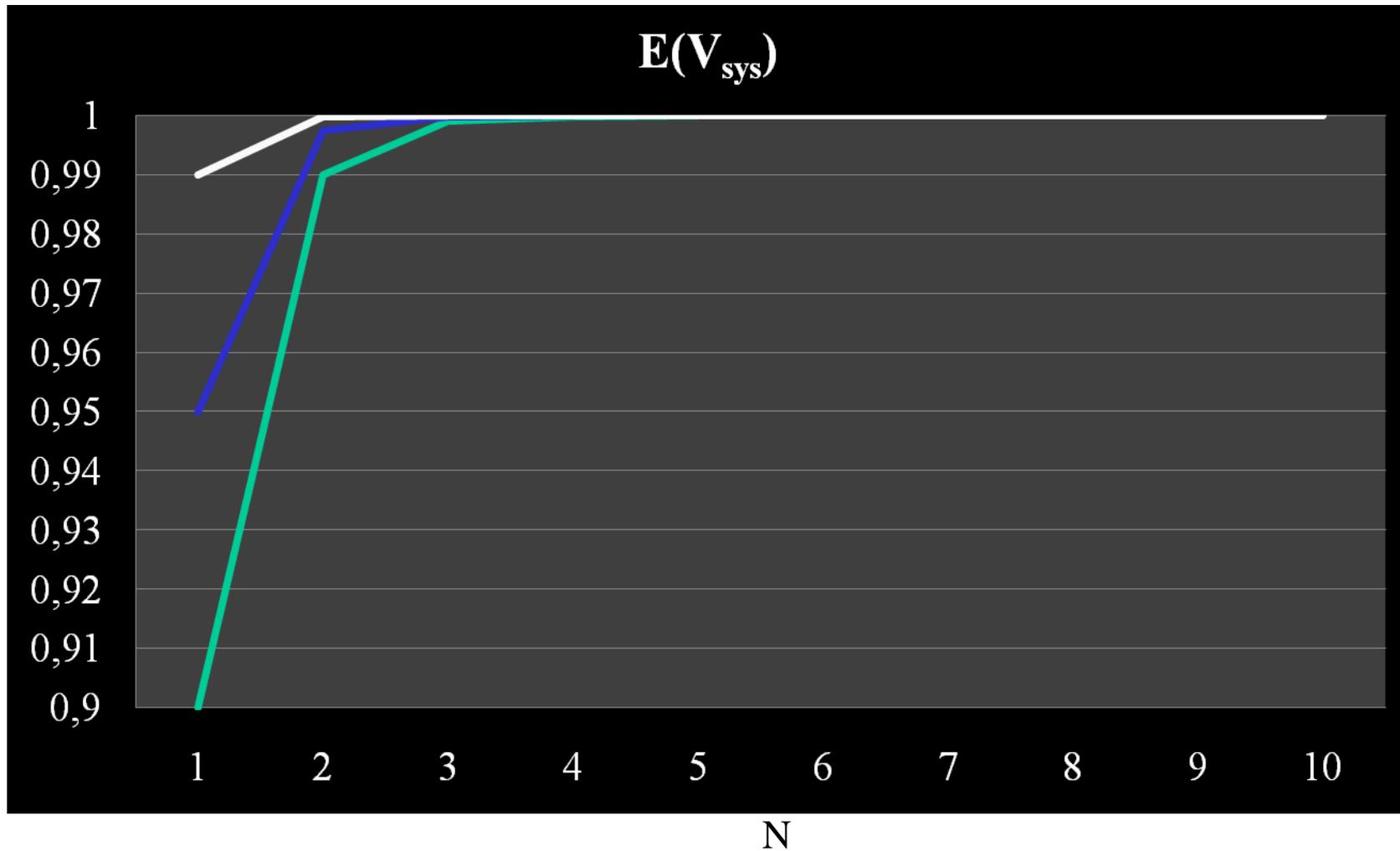
Falls alle Komponenten unabhängig repariert werden gilt

$$E(V_{sys}) = 1.0 - \prod_{i=1}^N (1.0 - E(V_i))$$

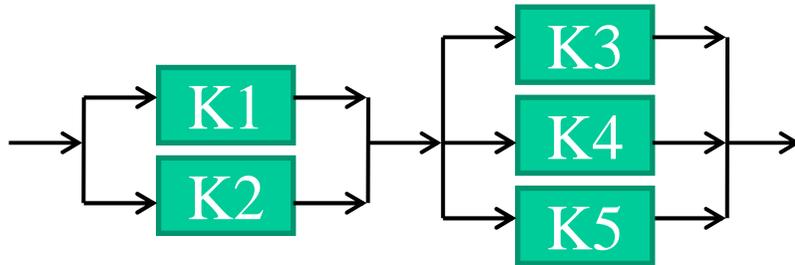
$E(V_i)$  liegt zwischen 0 und 1!

# Verfügbarkeit von Parallelschaltungen identischer Komponenten:

$E(V_i)$  Komponentenverfügbarkeit,  $E(V_{sys})$  Systemverfügbarkeit



## Serien- und Parallelschaltungen sind kombinierbar



System arbeitet, wenn

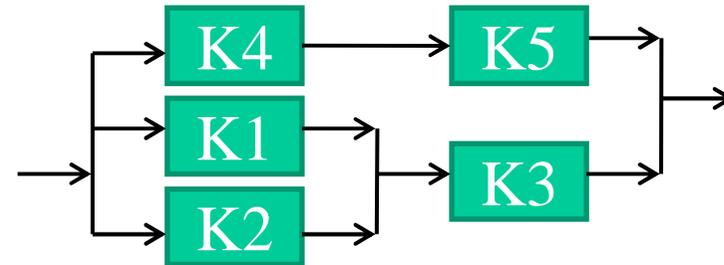
$$(K_1 \vee K_2) \wedge (K_3 \vee K_4 \vee K_5)$$

Es gilt:

$$F_{\text{sys}}(t) = \dots$$

und

$$E(V_{\text{sys}}) = \dots$$



System arbeitet, wenn

....

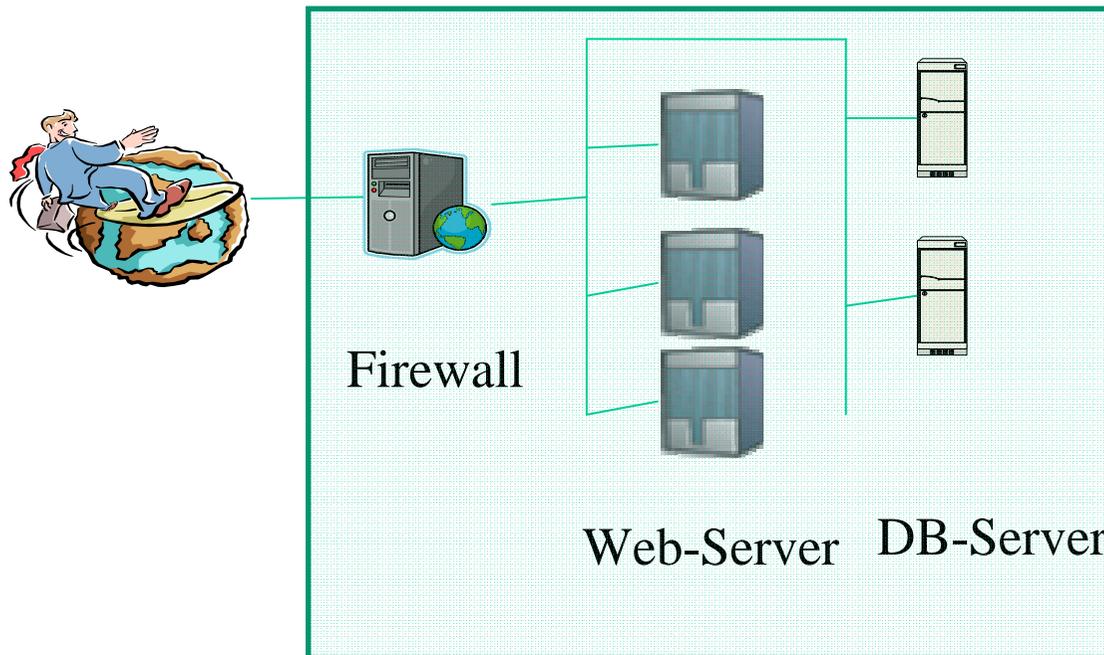
Es gilt:

$$F_{\text{sys}}(t) = \dots$$

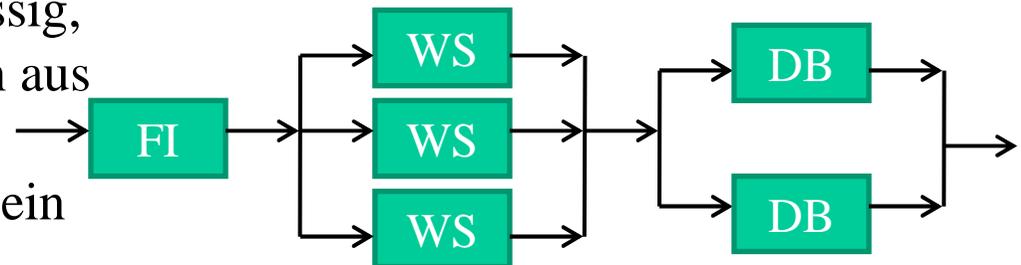
und

$$E(V_{\text{sys}}) = \dots$$

## Ein Beispiel: Verfügbarkeit eines Server-Systems



Annahme: Netzinfrastruktur ist zuverlässig,  
 nur Server bzw. Firewall fallen aus  
 System ist verfügbar,  
 wenn der Firewall und jeweils ein  
 Web- und DB-Server arbeiten



Anforderung: Systemverfügbarkeit von mindestens 99.99%

Systemkonfigurationen:

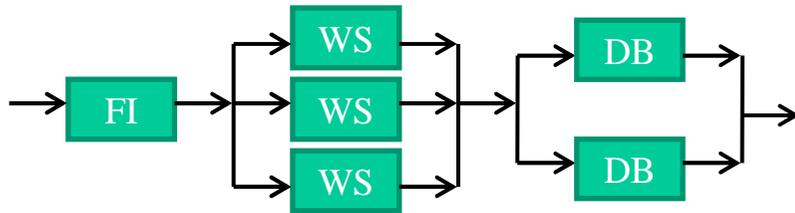
➤ Verfügbarkeit des Firewall > 99.99% notwendig!

Wir nehmen an, dass ein Firewall mit Verfügbarkeit 99.999% zur Verfügung steht

➤  $n_{WS}$  Web-Server und  $n_{DB}$  DB-Server sollen eingesetzt werden

➤ Für beide Server stehen einfache Systeme mit einer Verfügbarkeit von 85% und hochzuverlässige Systeme mit einer Verfügbarkeit von 99.9% zur Verfügung

Welche Konfigurationen erreichen die geforderte Verfügbarkeit und was ist die beste dieser Konfigurationen?



$$E(V_{sys}) = E(V_{FI}) \cdot (1 - (1 - E(V_{WS}))^{n_{WS}}) \cdot (1 - (1 - E(V_{DB}))^{n_{DB}})$$

Da  $E(V_{sys})=0.9999$  muss gelten

$$0.99991 \leq (1 - (1 - E(V_{WS}))^{n_{WS}}) \cdot (1 - (1 - E(V_{DB}))^{n_{DB}})$$

Damit gilt z.B. auch

$$n_{WS} = \left\lceil \frac{\log(1 - 0.99991 / (1 - (1 - E(V_{DB}))^{n_{DB}}))}{\log(1 - E(V_{WS}))} \right\rceil$$

Ferner muss natürlich gelten

$$(1 - E(V_{WS}))^{n_{WS}} \leq 0.00009 \quad \text{und} \quad (1 - E(V_{DB}))^{n_{DB}} \leq 0.00009$$

Einige Resultate:

$E(V_{WS})$	$E(V_{DB})$	$n_{WS}$	$n_{DB}$	$E(V_{sys})$
0.850	0.850	6	5	0.99990
0.850	0.999	5	2	0.99991
0.999	0.999	2	2	0.99999

Auswahl der besten Konfiguration hängt ab von

➤ Kosten

➤ Leistung

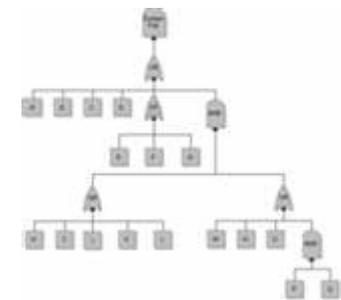
bzw. deren Kombination

Fehlerbäume:

Solide Basis zur Beschreibung von Zuverlässigkeitsmodellen, die logische und quantitative (probabilistische) Zusammenhänge beschreiben

Eigenschaften

- Graphische Darstellung einer logischen Funktion
- Darstellung des Zusammenhangs zwischen Ereignissen und Systemfehlern
- Unterstützung bei Fehleranalyse
- Im Sinne der Informatik nicht zwangsläufig Bäume
- Darstellung und Definition nicht immer eindeutig
- Ähnliche Modellierungsmächtigkeit (und Annahmen) wie Zuverlässigkeitsblockdiagramme



## Komponenten eines Fehlerbaums



Elementares Fehlerereignis, für das die Fehlerrate bekannt ist



Nicht vollständig spezifiziertes Fehlerereignis,  
für das eine Fehlerrate angenommen wird



Oder-Gatter



Und-Gatter

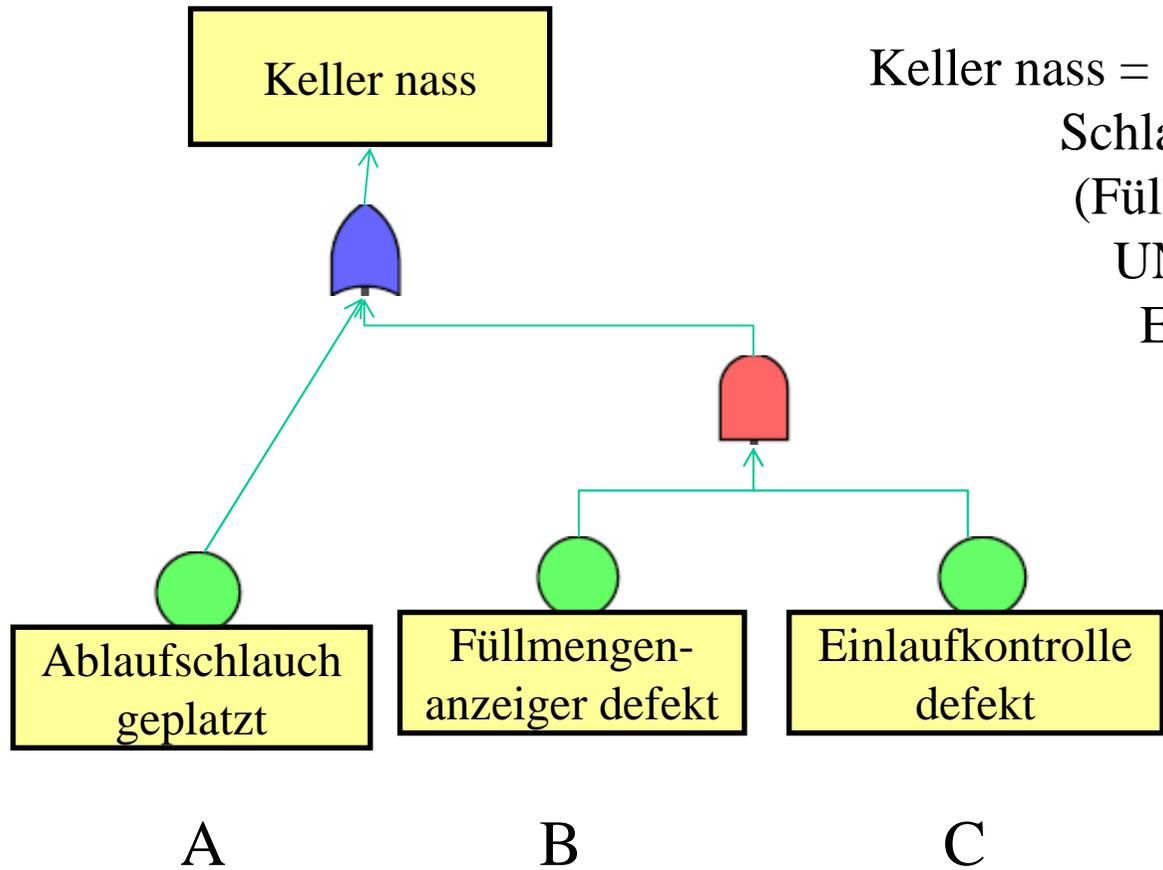


m aus n-Gatter, schaltet, wenn m von n Eingängen belegt sind



Kommentare

# Beispiel



Keller nass =  
Schlauch geplatzt ODER  
(Füllmengenanzeiger defekt  
UND  
Einlaufkontrolle defekt)

$$\text{Fehler} = A + BC$$

## Quantitative Analyse über Wahrscheinlichkeiten

$$P[A] = 0.01$$

$$P[B] = 0.05$$

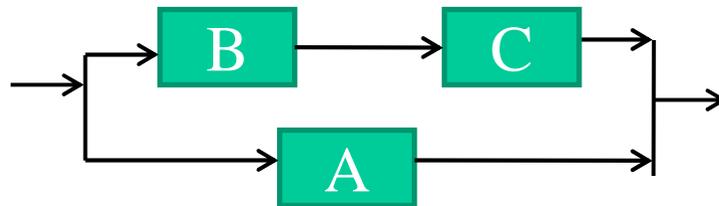
$$P[C] = 0.075$$

(unabhängige Fehler)

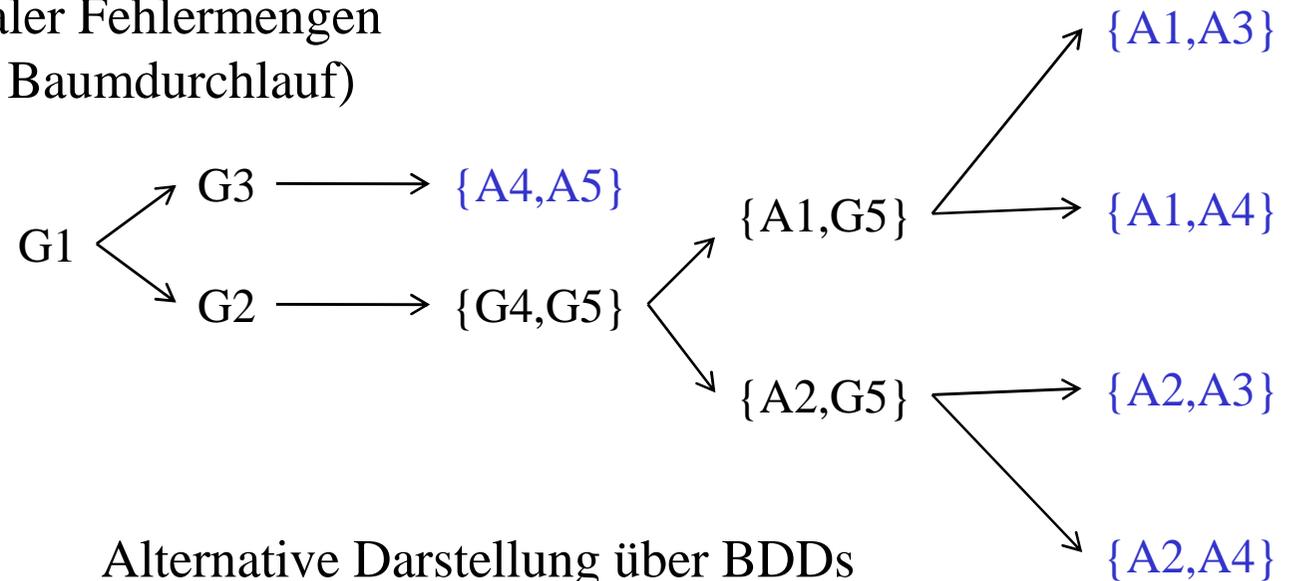
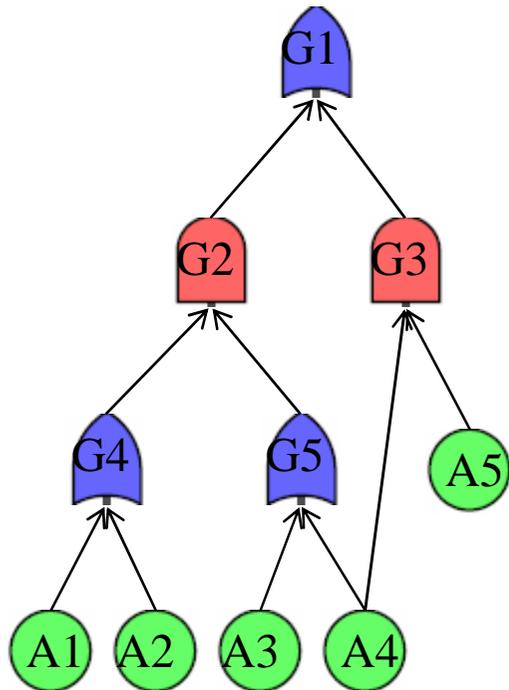
$$\begin{aligned} P[F] &= P[A + BC] \\ &= P[A] + P[AB] - P[ABC] \\ &= 0.01 + 0.00375 - 0.0000375 \\ &= 0.0137125 \end{aligned}$$

Minimale Fehlermengen:  $\{A\}$ ,  $\{BC\}$

Als Zuverlässigkeitsblockdigramm (natürliche Transformation)



## Bestimmung minimaler Fehlermengen (durch schrittweisen Baumdurchlauf)



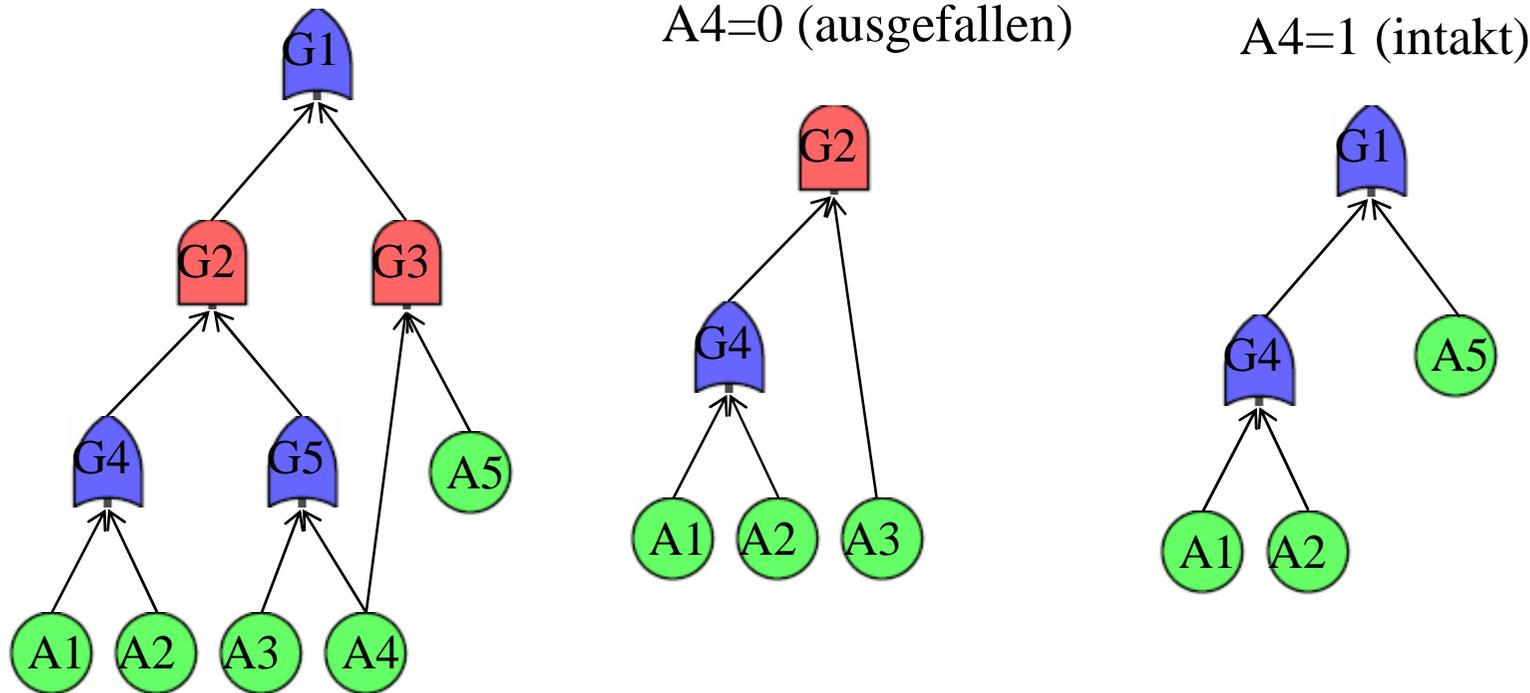
Alternative Darstellung über BDDs

Transformation in Zuverlässigkeitsblockdiagramm  
nicht so einfach

A4 muss in G3 und G5 immer den gleichen Wert haben  
und kann deshalb nicht durch unterschiedliche Knoten  
repräsentiert werden!

# Faktorisierung mehrfach genutzter Eingaben

Jeweils ein Baum für jede mögliche Kombination von Variablenbelegungen!!



Fehlerbäume ohne mehrfach genutzte Eingaben, können auf natürliche Weise in Zuverlässigkeitsdiagramme überführt werden

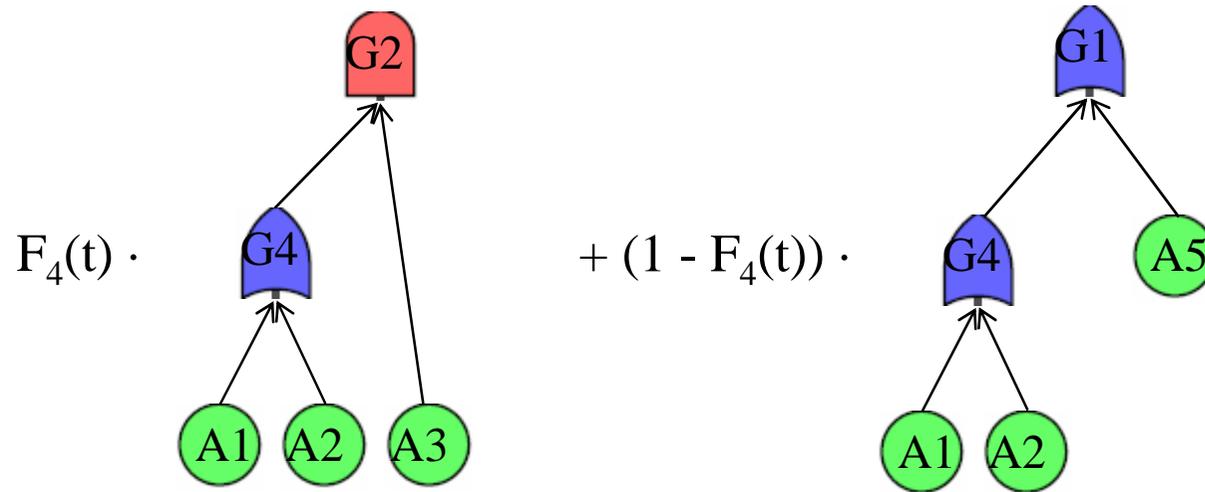
Aus einen Fehlerbaum mit  $K$  mehrfach genutzten Eingaben entstehen durch Faktorisierung  $2^K$  Fehlerbäume ohne Mehrfachnutzung der Eingaben

Sei  $F_i(t)$  die Verteilungsfunktion der Zeit bis zum Fehler am  $i$ -ten Eingang eines Gatters, dann gilt für  $F_G(t)$  die Verteilungsfunktion für den Gatterausgang

$$F_G(t) = \begin{cases} \prod_{i=1}^n F_i(t) & \text{für and-Gatter} \\ 1 - \prod_{i=1}^n (1 - F_i(t)) & \text{für or-Gatter} \\ \sum_{i=m}^n \binom{n}{i} (F(t))^i (1 - F(t))^{n-i} & \text{für m-aus-n-Gatter} \\ & \text{mit identischen Verteilungen} \\ \sum_{|J| \geq m} \left( \prod_{j \in J} F_j(t) \right) \left( \prod_{j \notin J} (1 - F_j(t)) \right) & \text{für m-aus-n-Gatter} \\ & \text{mit individuellen Verteilungen} \end{cases}$$

$n$  ist die Anzahl der Gattereingänge

## Berechnung für unser Beispiel



$$F_{\text{sys}}(t) = F_4(t) \cdot ((1 - (1 - F_1(t)) \cdot (1 - F_2(t))) \cdot F_3(t)) + \\
 (1 - F_4(t)) \cdot (1 - (1 - (1 - (1 - F_1(t)) \cdot (1 - F_2(t))) \cdot (1 - F_5(t))))$$

Analoge Berechnung von  $E(V_{\text{sys}})$

## Grenzen der beiden vorgestellten Ansätze:

- Abhängigkeiten bei Ausfällen  
z.B. Defekt einer Komponente führt zum Defekt einer anderen, Komponenten weisen ähnliche Produktionsfehler auf, ...
- Abhängigkeiten bei der Reparatur  
z.B. es gibt nur eine beschränkte Anzahl von Reparatereinheiten, ..
- Ausfälle nicht eindeutig erkennbar  
z.B. Komponente liefert falsche Ergebnisse, ...
- Ausfallverhalten hängt von der Belastung ab  
z.B. kalte Reserven, die erst im Einsatz ausfallen, ...
- Komponenten mit und ohne Reparatur
- Berechnung einer Maße  
z.B. bedingte Ausfallwahrscheinlichkeit, ...

## Q6.3 Markov-Prozesse zur Zuverlässigkeitsmodellierung

Interpretation des Basismodells als Markov-Prozess



Exponentiell verteilte Ausfall- und Reparaturzeiten

Analyse:

1. Aufbau des Zustandsraums und der Matrix  $\mathbf{Q}$

Rewards: 1 falls Zustand intaktes System beschreibt, 0 sonst

2. Stationäre oder transiente Analyse

liefert Verfügbarkeit, Ausfallzeiten, ....

Siehe  
Q5.3 und  
Q5.4

# Systeme ohne Reparatur



Insgesamt  $2^K$  Zustände

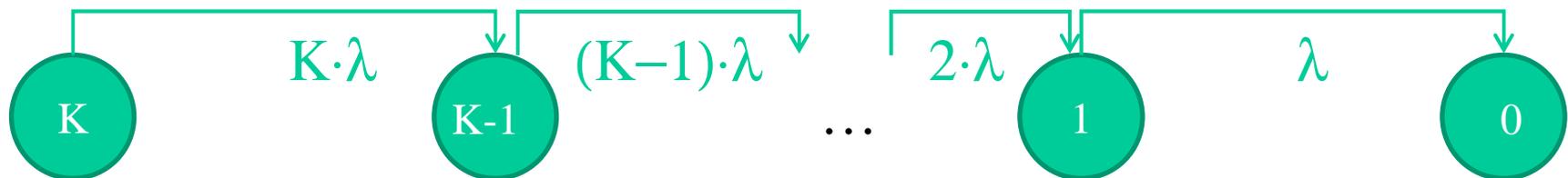
Zustandsraumreduktion durch:

➤ Zusammenfassen von Fehlerzuständen

z.B. L von K Komponenten müssen arbeiten  $\Rightarrow \sum_{l=1}^L \binom{K}{l} + 1$  Zustände

(wenn System ausgefallen, fallen keine weiteren Komponenten aus)

➤ Identische Komponenten, d.h.  $\lambda_1 = \lambda_2 = \dots = \lambda_K \Rightarrow K+1$  Zustände



$K$  unterschiedliche Komponenten, die jeweils  $N_k$  mal repliziert sind

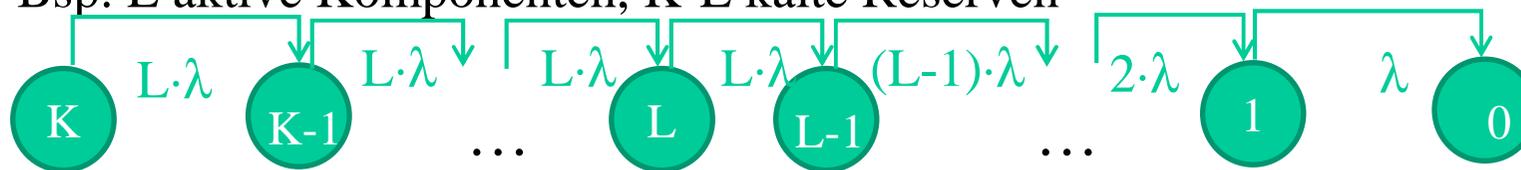
$\Rightarrow \prod_{k=1, \dots, K} (N_k + 1)$  Zustände

Weitere Modellierungsmöglichkeiten:

1. Kalte Reserven

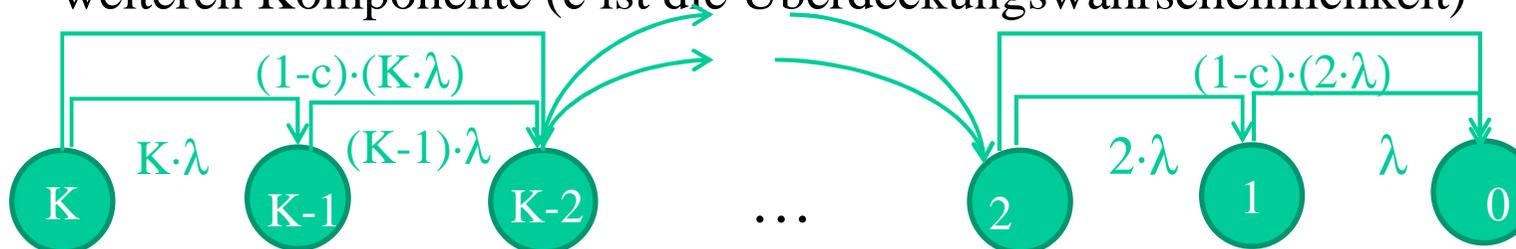
werden beim Ausfall zugeschaltet, können erst ausfallen, wenn aktiv

Bsp.  $L$  aktive Komponenten,  $K-L$  kalte Reserven



2. Überdeckung (coverage)

d.h. Ausfall einer Komponente bedingt mit Wahrsch.  $(1-c)$  Ausfall einer weiteren Komponente ( $c$  ist die Überdeckungswahrscheinlichkeit)



Systeme ohne Reparatur führen zu azyklische Markov-Prozessen,  
d.h. Matrix  $\mathbf{Q}$  lässt sich zu einer oberen Dreiecksmatrix umordnen!

Letzter Zustand ist absorbierend

Reward = 0 für den letzten Zustand und 1 für alle anderen Zustände

Beispielmatrix für ein System mit 3 identischen Komponenten

$$\mathbf{Q} = \begin{pmatrix} -3\lambda & 3\lambda & 0 & 0 \\ 0 & -2\lambda & 2\lambda & 0 \\ 0 & 0 & -\lambda & \lambda \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

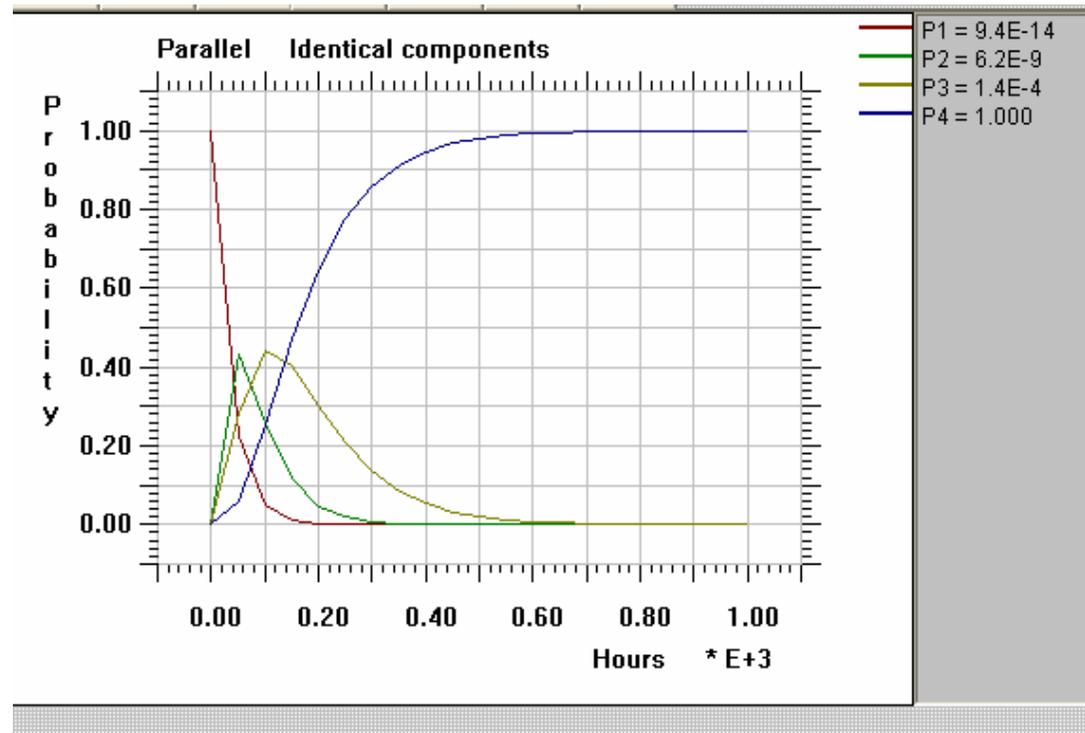
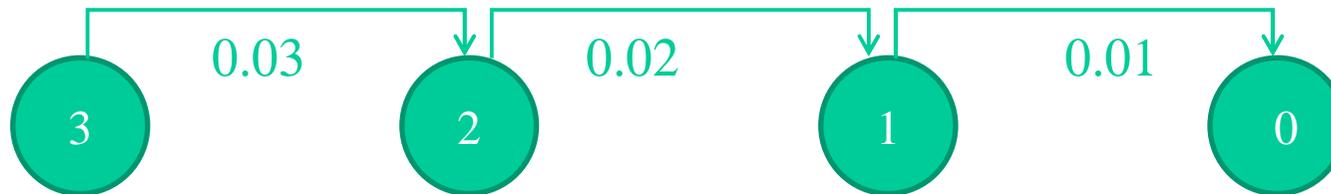
Bei  $n$  Zuständen  $S=\{1,\dots,n\}$  gilt:

➤  $\pi_1(0) = 1.0$  und  $\pi_i(0) = 0$  für  $1 < i \leq n$

➤  $\lim_{t \rightarrow \infty} \pi_n(t) = 1.0$  und  $\lim_{t \rightarrow \infty} \pi_i(t) = 0.0$  für  $1 \leq i < n$

➤  $F_{\text{sys}}(t) = \pi_n(t) \Rightarrow$  transiente Analyse zur Berechnung der Verteilungsfunktion der Ausfallzeit

Beispiel: 3 identische Komponenten mit Ausfallrate ...,  
System ist defekt, wenn alle Komponenten ausgefallen

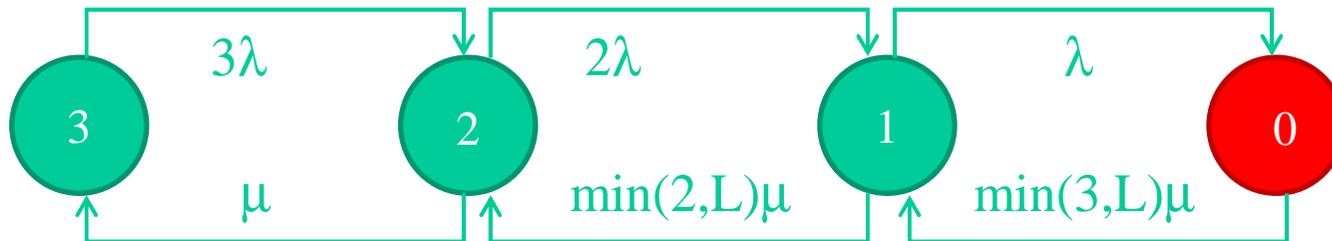


## Systeme mit Reparatur

Versionen wie Systeme ohne Reparatur mit zusätzlichen Reparaturtransitionen:

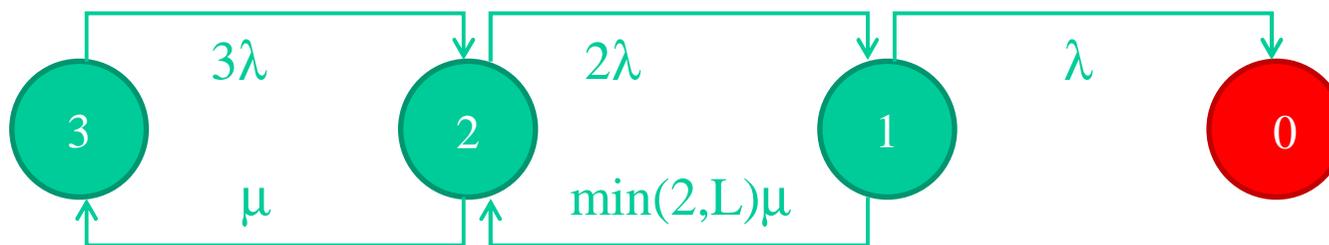
- Bei individueller Reparatur, Transitionen mit Rate  $k \cdot \mu$  bei  $k$  ausgefallenen Komponenten und Reparaturrate  $\mu$
- Bei  $L$  vorhandenen Reparatereinheiten, Transitionen mit  $\min(k, L) \cdot \mu$  bei  $k$  ausgefallenen Komponenten und Reparaturrate  $\mu$
- Falls unterschiedliche Komponenten ausfallen und nur eine beschränkte Anzahl von Reparatereinheiten vorhanden ist, so muss eine Strategie festgelegt werden, nach der ausgefallene Komponenten repariert werden  
z.B. FCFS, Prioritäten, Random,..... (beeinflusst die Zustandsraumgröße!)
- Reward 1 wird Zuständen zugewiesen, in denen das System arbeitet, Reward 0 den übrigen Zuständen
- Resultierende Markov-Prozesse sind nicht mehr azyklisch!

# 1) Berechnung der Verfügbarkeit (L Reparatureinheiten)



Stationäre oder transiente Analyse

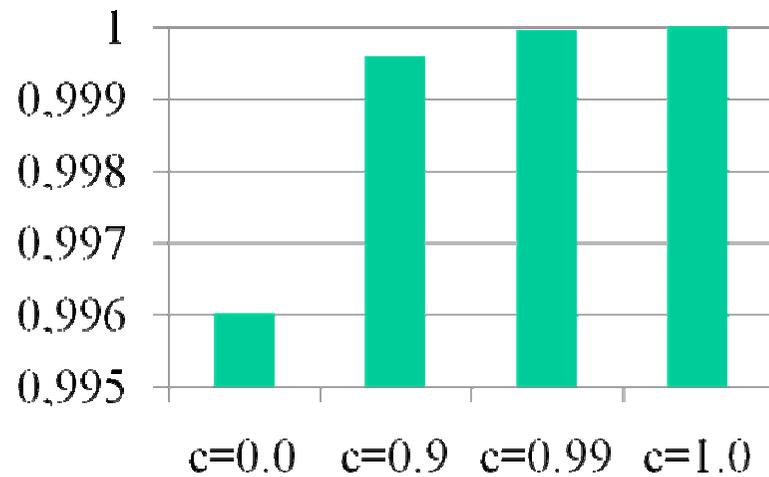
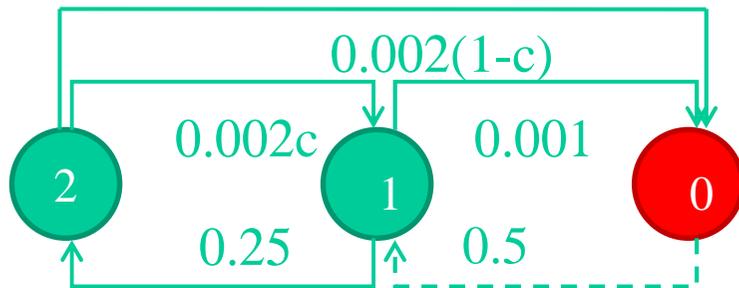
# 2) Berechnung der Zeit bis zum Systemausfall (L Reparatureinheiten)



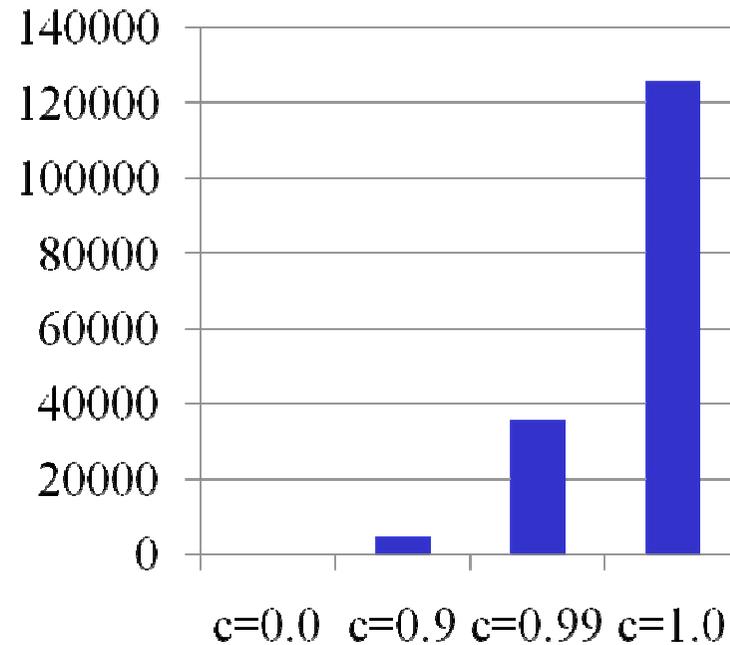
Transiente Analyse

Ein einfaches Beispiel:

2 Komponenten mit Ausfallrate  $\lambda=0.001$ , Reparaturrate  $\mu=0.25$  und Überdeckungswahrscheinlichkeit  $c$



■ E(V)



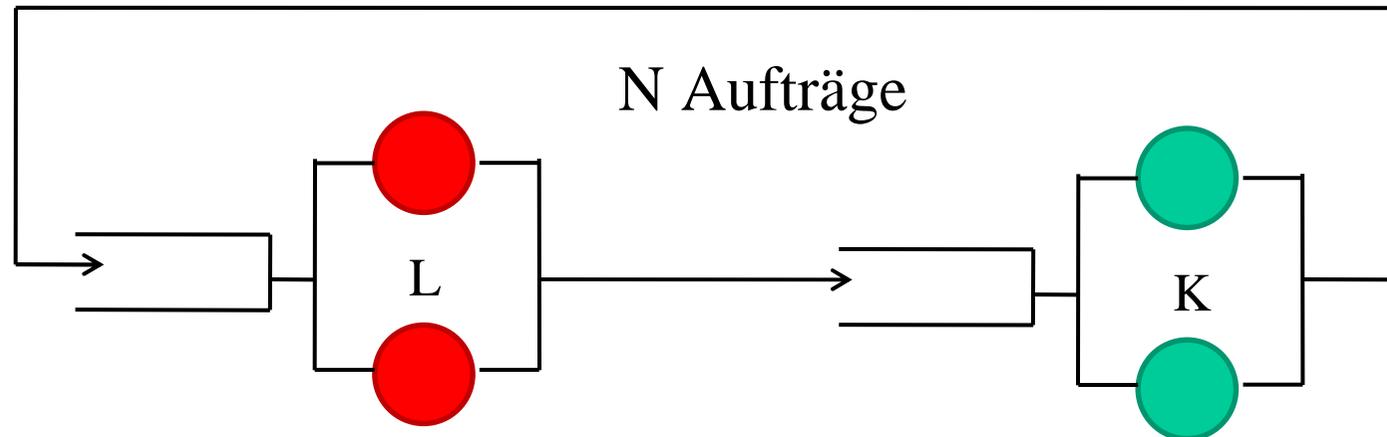
■ E(A)

## Spezifikation von Modellen:

- Auf Zustandsebene  
(sehr detailliert und aufwändig)
- Als Warteschlangennetz  
(sehr abstrakt, beschränkte Modellierungsmächtigkeit)
- Als stochastisches Petri-Netz  
(oft ein guter Kompromiss)
- Spezielle Eingabesprachen  
(bisher nur eingeschränkt vorhanden)
- Hierarchische Modelle  
(konzeptionell anspruchsvoll)

Beispiel Warteschlangennetz:

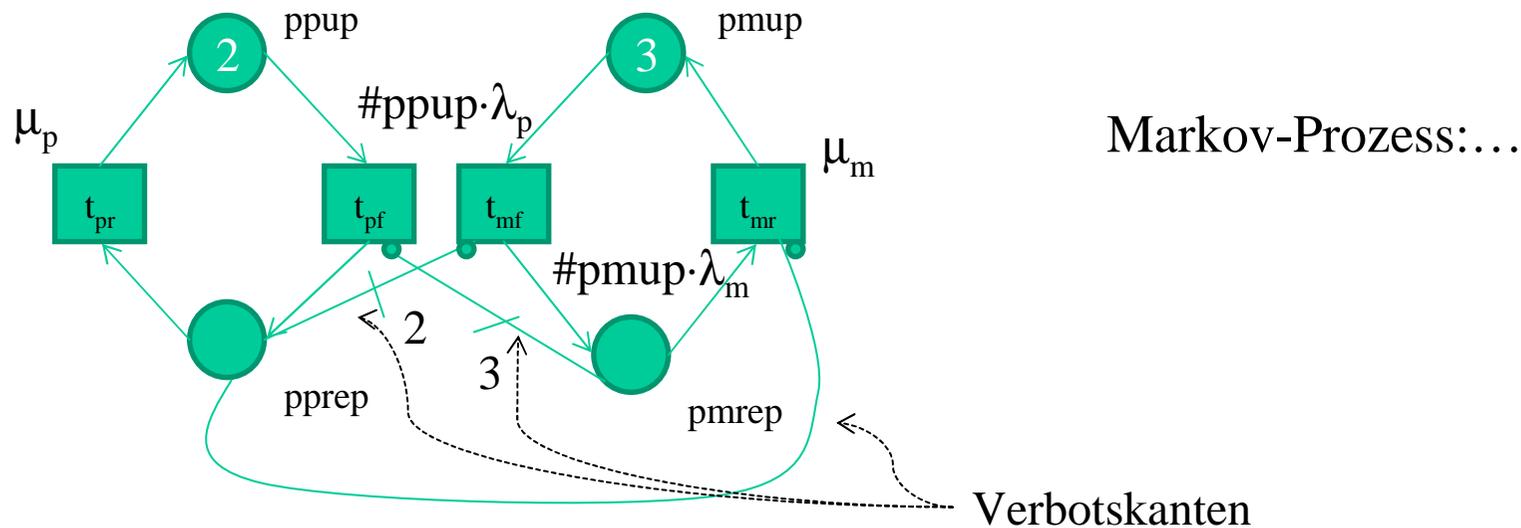
- $N$  Komponenten ( $N-K$ ) kalte Reserven, Ausfallzeit exp. Vert. mit Rate  $\lambda$
- $L$  Reparatureinheiten, Reparaturzeit exp. Vert. mit Rate  $\mu$



- Berechnung der Verfügbarkeit und Ausfallzeiten durch stationäre oder transiente Analyse des Warteschlangennetzes
- Effiziente stationäre Analyse bei exponentiellen Ausfall- und Reparaturzeiten (siehe Q5/Q7)

## Beispiel stochastisches Petri-Netz:

- Multiprozessorsystem mit 2 Prozessoren (Ausfallrate  $\lambda_p$ ) und
- 3 Speichermodulen (Ausfallrate  $\lambda_m$ )
- 1 Reparatereinheit (Reparaturraten  $\mu_p$  und  $\mu_m$ ), Prozessoren werden mit Priorität repariert
- System arbeitet, wenn mindestens ein Prozessor und ein Speichermodul verfügbar sind,
- Nur wenn das System arbeitet, können Komponenten ausfallen



Spezielle Eingabesprachen: z.B. SHARPE, USENUM, ....

Vorheriges Beispiel in USENUM:

CONSTANTS

MU\_P double 0.2

MU\_M double 0.25

LAMBDA\_P double 0.002

LAMBDA\_M double 0.001

MEM int 3

PROC int 2

%%

STATE

pup int

mup int

%%

INITIAL\_STATES

s->pup = c->PROC

s->mup = c->MEM

TRANSITION

t\_pfail

s->pup > 0 && s->mup > 0

%

next->pup = s->pup - 1

%

RATE

c->LAMBDA\_P \* (double) s->pup

%%

...

TRANSITION

T\_mrep

s->mup < c->MEM && s->pup == c->PROC

%

next->mup = s->mup + 1

%

RATE

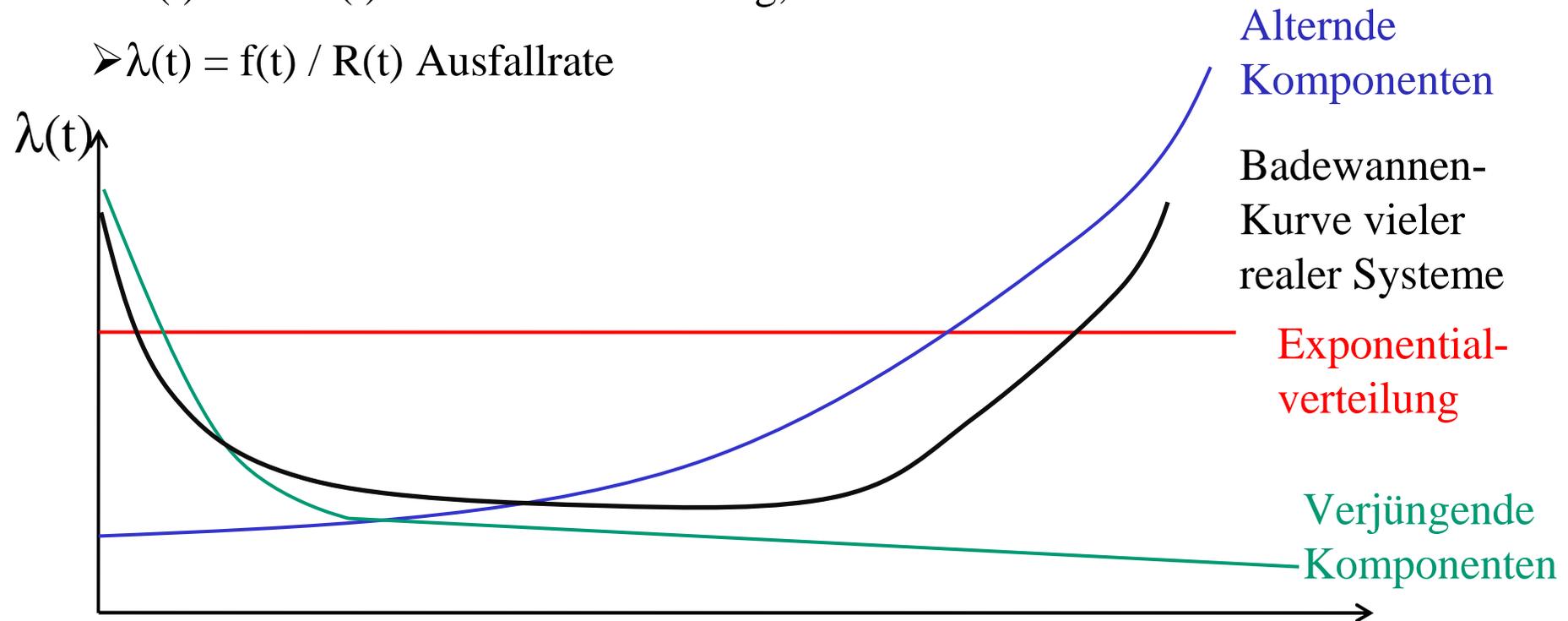
c->MU\_M

%%

## Q6.4 Nicht-Markovsche-Modelle zur Zuverlässigkeitsanalyse

Zur Erinnerung:

- $F(t)$  Verteilungsfunktion bis zum Ausfall (mit Dichtefunktion  $f(t)$ ),
- $R(t) = 1 - F(t)$  Überlebensverteilung,
- $\lambda(t) = f(t) / R(t)$  Ausfallrate



Modellierungsmöglichkeiten zur Darstellung nicht konstanter Ausfallraten:

1. Inhomogene Poisson-Prozesse

2. Weibull-Verteilungen

3. Phasenverteilungen

### Inhomogene Poisson-Prozesse

- Fehler treten mit Rate  $\lambda(t)$  zum Zeitpunkt  $t$  auf
- Verlauf der Ausfallrate kann sehr genau modelliert werden
- Analyse
  - Numerische Analyse des inhomogenen Markov-Prozesses, falls nur eine Verteilung mit zeitabhängiger Rate existiert
  - Approximation durch stückweise konstante Funktionen und die schrittweise transiente Analyse homogener Markov-Prozesse
  - Stochastische Simulation, spezieller Ansatz zur Erzeugung von Ausfallzeiten existiert

Weibull-Verteilung:

➤ Zwei Parameter  $\alpha, \beta > 0$

➤ Es gilt  $F(t) = 1 - e^{-\alpha t^\beta}$  und  $f(t) = \alpha \beta t^{\beta-1} e^{-\alpha t^\beta}$

➤  $\lambda(t) = \alpha \beta t^{\beta-1}$

➤ falls  $\beta=1$  entspricht dies der Exponentialverteilung mit Rate  $\alpha$

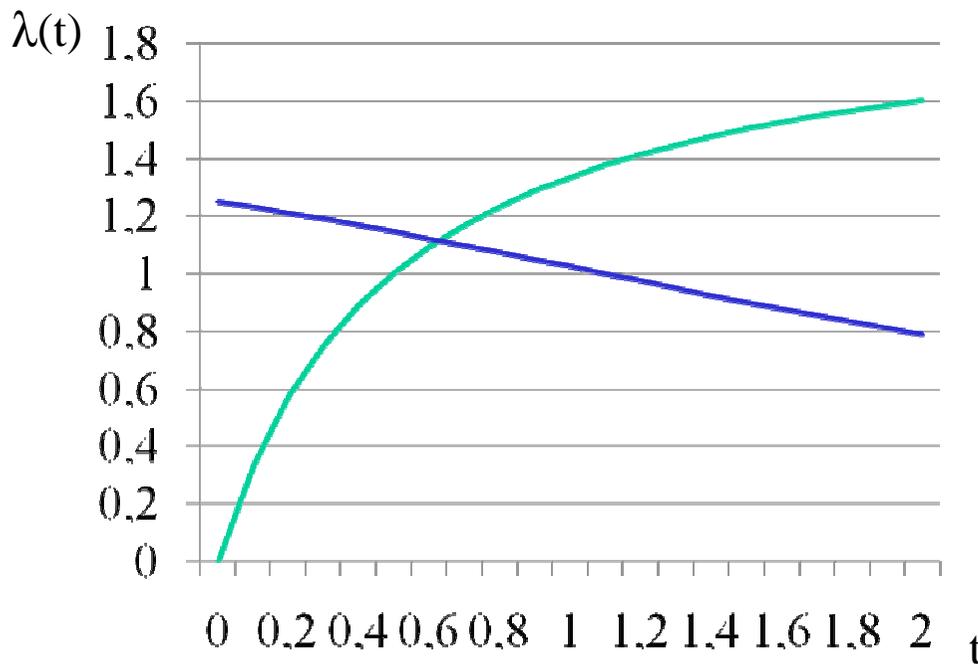
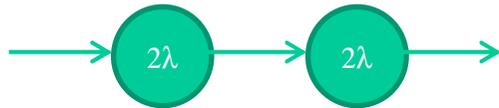
➤ Falls  $\beta < 1$  nimmt die Ausfallrate ab

➤ Falls  $\beta > 1$  nimmt die Ausfallrate zu

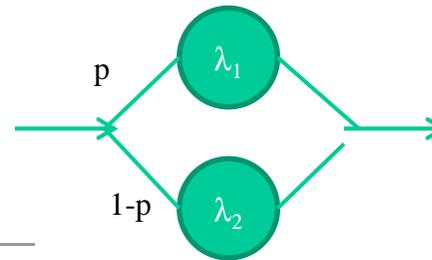
➤ Analyse nur mittels Simulation

# Phasenverteilungen: Netzwerke aus exponentiellen Phasen

Beispiel: Erlang-Verteilung  
(hier Erlang 2)



Beispiel: Hyperexponential-Verteilung  
(hier Hyperexponential 2)



Erlang 2 mit  $\lambda = 2.0$

Hyperexp. Mit  
 $\lambda_1 = 0.5,$   
 $\lambda_2 = 1.5$  und  
 $p = 0.25$

In beiden Fällen  
gilt:  $E(T) = 1.0$

## Phasenverteilungen

➤ Führen zu Markov-Prozessen, die mit den bekannten Methoden analysiert werden

➤ Führen aber auch zur so genannten *Zustandsraumexplosion*

Bsp. K unterschiedlich Komponenten, die jeweils  $N_i$  mal repliziert werden und deren Ausfallzeiten durch eine Phasenverteilung mit  $M_i$  Phasen realisiert wurde

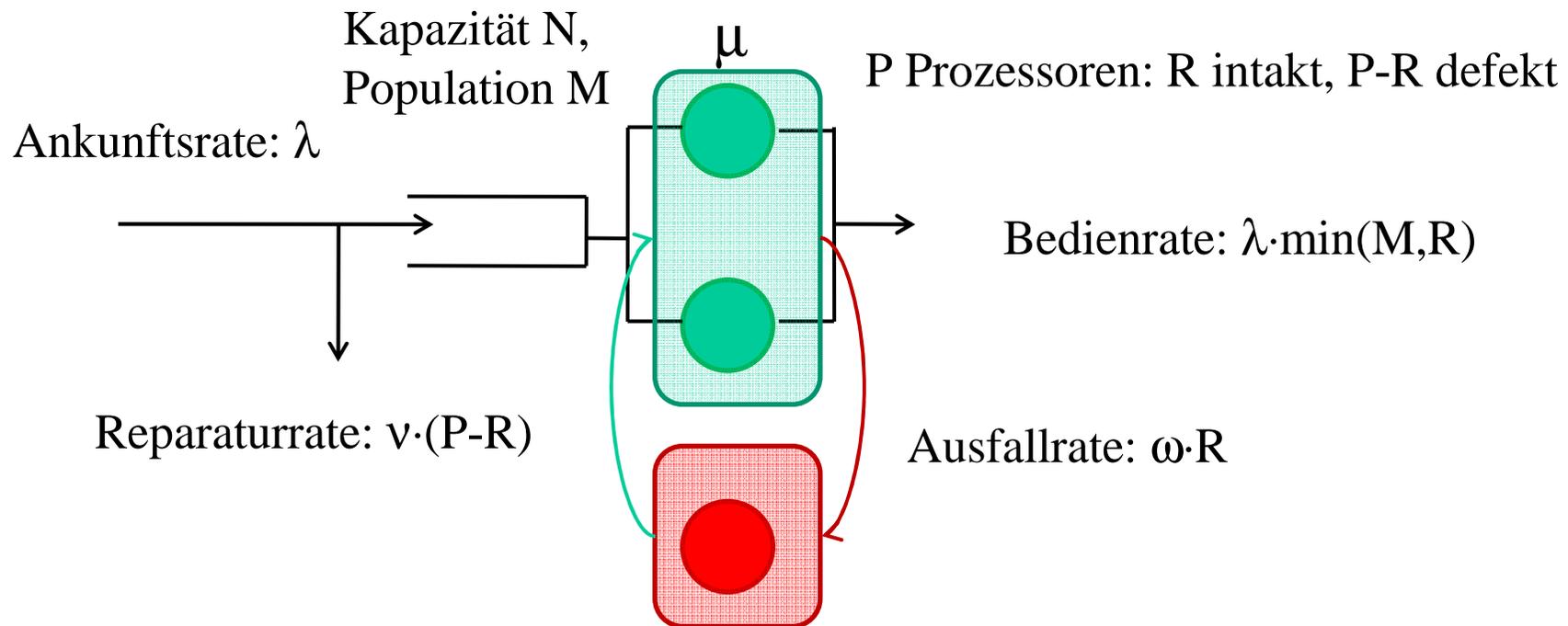
$$\Rightarrow \sum_{i=1}^K \binom{N_i + M_i - 1}{N_i} \text{ Zustände}$$

➤ Spezifikation von Phasenverteilungen in Modellierungswerkzeugen oft nur explizit möglich

## Q6.5 Zuverlässigkeit und Leistung

Im Englischen unter dem Kunstwort *performability* (*performance* und *dependability*) zusammengefasst

Beispiel: Multiprozessorsystem mit Ausfällen und Reparatur



Beispiel mit

➤  $(P+1) \cdot (N+1)$  Zuständen bei exponentiellen Zeiten

(entsprechende Aufblähung des Zustandsraums bei Nutzung von Phasenverteilungen)

➤ Es gilt  $\lambda, \mu \gg \omega, \nu$

➤ Markov-Prozess erreicht einen quasi stationären Zustand mit einer festen Zahl intakter Prozessoren, bevor ein Prozessor ausfällt oder repariert wird

➤ Ausfälle und Reparaturen sind unabhängig von der Beabreitung von Aufträgen

⇒ Analyse separat je ein Leistungsmodell mit  $R=1, \dots, P$  intakten Prozessoren und ein Zuverlässigkeitsmodell

# Markov-Prozess bei exponentiell verteilten Zeiten

