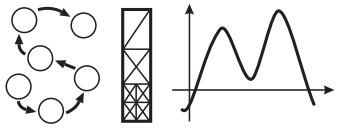
FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG
INSTITUT FÜR INFORMATIK (MATHEMATISCHE MASCHINEN UND DATENVERARBEITUNG)

Lehrstuhl für Informatik 10 (Systemsimulation)



Schnellere Simulation durch Zustandsraumreduktion

Michael Luber
geb. am 1. Dezember 1976 in Sulzbach-Rosenberg

Betreuer:

Dr. Graham Horton (IMMD 10),
Dipl.-Inf. Stefan Heller (DaimlerChrysler AG)

Beginn der Arbeit: 01.03.2000
Abgabe der Arbeit: 30.11.2000

Studienarbeit im Fach Informatik

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat oder von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den 30. November 2000 .....

Zusammenfassung

In dieser Studienarbeit wird zunächst der Aufbau eines zustandsorientierten, ereignisgesteuerten Simulationssystems vorgestellt und der Simulationsablauf erläutert. Außerdem wird darauf eingegangen, warum eine Reduktion des Zustandsraums zur Simulationsbeschleunigung wünschenswert ist.
Anschließend werden verschiedene Ansätze, die unter gewissen Umständen eine solche Reduktion ermöglichen, vorgestellt und erklärt. Es sind dies die Parallelkantenverschmelzung, die Berechnung von Verzweigungswahrscheinlichkeiten, und das Verändern der Verbindungswege zwischen den Zuständen, mit dem Ziel, bei der späteren Simulation Zustände, die als uninteressant deklariert sind, überspringen zu können und somit Rechenzeit einzusparen. Speziell für das Überspringen eines Zustands, der an einer Verzweigung beteiligt ist, wurde eine zufriedenstellende Lösung gefunden.
Auf diesen Grundlagen aufbauend wird ein realisierter Simulator beschrieben, der über die Simulation hinaus in der Lage ist, diese Reduktionsschritte auszuführen. Der Sinn der damit durchgeführten Experimente lag hauptsächlich darin, die Simulationsgeschwindigkeit vor und nach einer Reduktion des Zustandsraums zu vergleichen, und zu entscheiden, ob sich die in die Reduktion investierte Rechenzeit auszahlt.
Die Resultate dieser Versuche zeigen, dass die Ersparnis an Rechenzeit und der Aufwand für die Reduktion von verschiedenen Faktoren, etwa der Grafkonnektivität und dem Anteil der beobachteten Zustände, abhängig sind. Auch müssen für die Simulation des reduzierten Systems Einbußen in der Genauigkeit hingenommen werden. Je nach Beschaffenheit der Parameter des zu simulierenden Grafen stellt sich jedoch mitunter ein beträchtliches Einsparpotenzial für die Simulationsrechenzeit ein.
Zuletzt wird noch auf weiterführende Ideen und Ansätze eingegangen, wie zum Beispiel eine effizientere Darstellung von diskreten Dichten und Verteilungsfunktionen. Auch führt eine Ausführung der Reduktionsschritte in verschiedenen Reihenfolgen im allgemeinen auch zu verschiedenen Ergebnissen, somit erscheint hier eine genauere Analyse ebenfalls lohnenswert.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Unter Simulation versteht man die Nachahmung des Ablaufs eines Prozesses aus der realen Welt, um interessierende Eigenschaften des untersuchten Prozesses herauszufinden. Man bildet dazu typischerweise ein reales zu analysierendes System mittels Abstrahierung in ein Modell ab, in welchem nur noch für die Simulation relevante Eigenschaften vorhanden sein sollen. Damit kann dann, ohne Beeinflussung des realen Systems selbst, der Einfluss von Systemänderungen untersucht werden. Eine der wichtigsten Anwendungen von Simulation ist außerdem die Analyse eines in Planung befindlichen, noch nicht vorhandenes Systems. In speziellen Fällen kann es möglich sein, das Modell analytisch zu lösen, aber im allgemeinen Fall muss dazu eine üblicherweise sehr rechenaufwändige numerische Simulation durchgeführt werden, bei der überdies gewisse Einschränkungen, zum Beispiel für die Genauigkeit, hingenommen werden müssen.
Die Verfügbarkeit entsprechender Modellbeschreibungssprachen und insbesondere die immer höhere und billigere Rechenleistung moderner Computer machen die Systemsimulation zu einem immer weiter verbreiteten Werkzeug der Systemanalyse. Simulation erlaubt die Untersuchung komplexer Systeme und deren Interaktion als Teile eines großen Gesamtsystems, die Vorhersage des Effektes von Änderungen an den Eingangsgrößen eines Systems und gestattet die Beurteilung der Wichtigkeit dieser Eingangsgrößen. Darüber hinaus können sich bereits bei der Erstellung eines Simulationsmodells wertvolle Erkenntnisse über eventuelle Verbesserungsmöglichkeiten gewinnen lassen. Nicht zuletzt lassen sich bereits vorhandene analytische Lösungen verifizieren, um festzustellen, ob man wirklich alle relevanten Größen in Betracht gezogen hat.
Der Einsatz der Systemsimulation zur Systemanalyse bietet eine Vielfalt von Vorteilen: Beispielsweise kann, ohne den laufenden Betrieb eines realen Systems zu stören und somit zusätzlich Kosten zu verursachen, der Effekt neuer Strategien und Arbeitsweisen unter-

sucht werden. Außerdem können in eine Simulation beliebig teure Komponenten eingefügt werden, da diese ja nur virtuell vorhanden sind. Weiterhin lässt sich die Zeit beliebig dehnen oder komprimieren, was zum Beispiel die Vorhersage langwieriger Prozesse ermöglicht. Es kann die Relevanz einzelner Variablen innerhalb des Systems und ihrer Interaktion ermittelt werden, und eine Simulationsstudie kann dazu beitragen, das ganze System und dessen Arbeitsweise besser zu verstehen.
Neben diesen Vorzügen gilt es allerdings auch einige Nachteile nicht aus den Augen zu verlieren: Die Erstellung von Modellen zur Repräsentation eines realen Systems ist nicht automatisierbar. Sie erfordert eine spezielle Ausbildung und viel Erfahrung, und ist überdies nicht eindeutig. Gleiches gilt für die Interpretation von Simulationsergebnissen, da diese meist Zufallsprozessen unterliegen. Dies führt dazu, dass ein Simulationsprojekt zeitaufwändig wird und damit hohe Kosten verursacht. Diese Nachteile relativieren sich jedoch, wenn spezielle Softwarepakete zur Verfügung stehen, die meist vorgefertigte Modellschablonen besitzen und den Simulationsexperten durch diverse Datenanalysefunktionen unterstützen. Außerdem steht zu erwarten, dass der Trend der immer schneller werdenden Hardware bis auf weiteres anhalten wird.
Die Anwendungsgebiete für die Systemsimulation sind unüberschaubar, deshalb seien hier nur einige genannt: Simulation findet zum Beispiel Anwendung in der Produktion, in Transportsystemen, in Konstruktion, Analyse von Rechnernetzwerken, in der Unterhaltungsbranche, etc. Für weiterführende Informationen zum Thema Systemsimulation sei der Leser auf [LK00] verwiesen.
Generell lassen sich Systeme einteilen in kontinuierliche Systeme, in welchen sich die Zustandsgrößen kontinuierlich ändern können, und diskrete Systeme, bei denen Zustandswechsel zu bestimmten Zeitpunkten auftreten. Diese Arbeit befasst sich ausschließlich mit rein diskreten Systemen, die darüber hinaus durch einen Zustandsraum darstellbar sein müssen. Dabei wurden Verfahren zur Modellvereinfachung entwickelt und ihre Eignung zur Simulationsbeschleunigung untersucht. Es wird vorausgesetzt, dass der Leser mit den Grundlagen der Wahrscheinlichkeitsrechnung vertraut ist, als Literaturhinweise seien hierfür [Fel68] und [Mat90] angegeben.
Die einzelnen Kapitel sind wie folgt unterteilt: Im zweiten Kapitel werden zunächst der Zustandsraum und die Simulation im Zustandsraum eingeführt. Das dritte Kapitel stellt den realisierten Simulator vor, der im Rahmen dieser Arbeit erstellt wurde. Dabei wird besonders auf die interne Repräsentation der erforderlichen Dichten eingegangen. Als nächstes werden die möglichen Transformationen beschrieben, mit deren Hilfe ein Graf vereinfacht und die Simulation somit beschleunigt werden kann, was im vierten Kapitel näher erläutert wird. Das fünfte Kapitel nennt die durchgeführten Experimente und legt die gewonnenen

Ergebnisse dar. Das sechste und letzte Kapitel schließlich gibt einen Überblick über anstehende Probleme und weiterführende Ideen.
Die Implementierung erfolgte in *Microsoft Visual C++ 5.0*, als Rechnerplattform diente ein *AMD Athlon* mit *600 MHz* und *128 MB* Hauptspeicher.

Kapitel 2

Simulation im Zustandsraum

In diesem Kapitel werden Aufbau und Ablauf von Simulationen im Zustandsraum beschrieben. Außerdem wird auf Anwendungsmöglichkeiten und die dabei häufig auftretenden Schwierigkeiten eingegangen.

2.1 Der Zustandsraum

Der Zustandsraum wird repräsentiert durch einen gerichteten Grafen, bestehend aus einer Zustandsmenge V und einer Kantenmenge $E \subseteq V \times V$. Dabei muss ein bestimmter Zustand $S \in V$ als Startzustand gewählt sein. Es gibt zu jedem Zeitpunkt genau einen Zustand Z_i aus V, der als besetzt markiert ist, was man synonym auch als 'Aufenthalt' in diesem bezeichnen kann. Dieser Zustand ist zur Simulationszeit $\tau = 0$ mit dem Startzustand identisch.
Jede Kante ist mit einer Wahrscheinlichkeitsverteilung versehen, die angibt, wieviel Simulationszeit verstreichen muss, bis die Kante verwendet werden darf. Dabei sind nur Verteilungen zugelassen, die nicht-negative Zeiten liefern. Ein einfaches Beispiel für einen solchen Zustandsraum zeigt die Abbildung 2.1.
Die Symbole X_1 bis X_5 bezeichnen hier beliebig wählbare, nicht-negative Verteilungsfunktionen, der besetzte Zustand ist durch einen Punkt markiert.

2.2 Ablauf eines Simulationsexperiments

Die Simulation beginnt zur Simulationszeit $\tau = 0$, wobei der besetzte Zustand mit dem Startzustand identisch ist. Jetzt werden alle Kanten, die den besetzten Zustand verlassen, ausgewertet, d.h. für jede solche Kante wird eine Zufallszahl entsprechend der jeweiligen

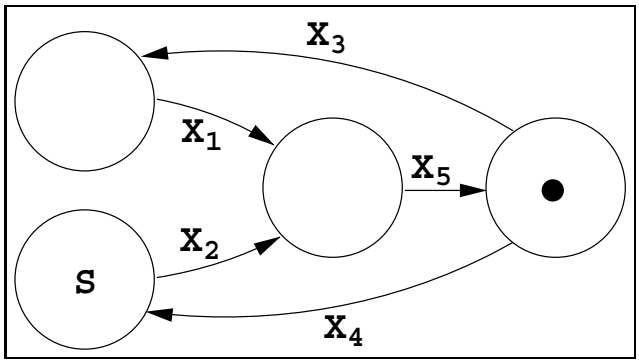


Abbildung 2.1: Beispiel für einen Zustandsgrafen

Verteilungsfunktion generiert. Diese Zufallszahl repräsentiert die jeweilige Zeitdauer, die beim Durchlaufen der entsprechenden Kante verbraucht werden würde. Nun bestimmt man diejenige Kante, welche die kleinste Zufallszahl, also die kürzeste Durchlaufdauer, geliefert hat; diese wird im Folgenden die schnellste Kante genannt. Daraufhin kann der eigentliche Simulationsschritt erfolgen, man addiert jetzt jene kürzeste Durchlaufdauer zur Simulationzeit τ hinzu, und versetzt den Marker, der auf den besetzten Zustand verweist, auf den Zielzustand der verwendeten Kante. Damit ist der erste Simulationsschritt beendet. Im nächsten Schritt werden wieder alle aus dem markierten Zustand ausgehenden Kanten ausgewertet, die schnellste Kante wieder ermittelt, usw. Dies wird solange wiederholt, bis ein zu definierendes Abbruchkriterium erfüllt ist, zum Beispiel kann man die Gesamtsimulationszeit vorgeben, oder die Höchstanzahl der Durchgänge durch eine Kante oder einen Zustand beschränken. Dabei ist darauf zu achten, dass das System nicht in eine Endlosschleife gerät. Für die Implementierung wurde die Vorgabe einer Gesamtsimulationszeit gewählt.
Um nach Beendigung eines Simulationslaufs verwertbare Daten zu erhalten, müssen diese natürlich während des Laufs mitprotokolliert werden. Welche Werte hierfür von Belang sind, hängt wiederum von der Anwendung ab. Denkbar sind zum Beispiel die Anzahl der Durchgänge durch bestimmte oder auch alle Zustände bzw. Kanten, oder auch welchen Anteil an der gesamten Simulationszeit ein Zustand besetzt war.

2.3 Einsatzmöglichkeiten in der Systemsimulation

Ein beliebtes Beispiel in der Systemsimulation ist die Warteschlange. Um eine Warteschlange in einem Zustandsraum darzustellen, benötigt man für jede mögliche Systemkonfigu-

ration, also für jede mögliche Warteschlangenlänge, einen eigenen Zustand. [1] Der dabei entstehende Graf besteht aus einer Kette von Zuständen, die, mit Ausnahme des ersten und letzten in der Kette, mit ihren beiden Nachbarzuständen durch jeweils eine eingehende und eine verlassende Kante verbunden sind. Abbildung 2.2 zeigt einen solchen Grafen für $N = 4$. Die Werte in den Zuständen stehen in der Abbildung für die Warteschlangenlänge, die durch den betreffenden Zustand repräsentiert wird.

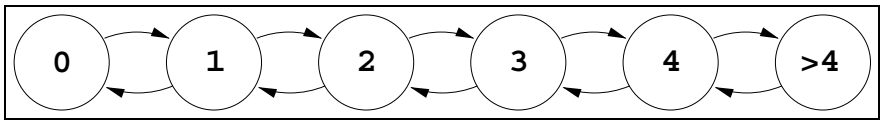


Abbildung 2.2: Zustandsgraf für eine Warteschlange

Im allgemeinen lässt sich die Simulation im Zustandsraum auf Systeme anwenden, deren relevante Eigenschaften sich durch diskrete Ereignisse ändern. Dabei entspricht jeder Zustand des Grafen einer möglichen Kombination der Systemeigenschaften und jede Kante einem solchen diskreten Ereignis.

2.4 Typische Merkmale und Probleme

Da für jede erdenkliche Konfiguration des zu simulierenden realen Systems ein eigener Zustand im Grafen existieren muss, kann die Anzahl der Zustände leicht sehr groß werden. Andererseits möchte man natürlich Ergebnisse mit einer gewissen Genauigkeit erhalten, was es beispielsweise erforderlich machen kann, dass ein spezieller interessiender Zustand mindestens eine gewisse Anzahl oft durchlaufen werden muss. Dies kann sich bei sehr großen Grafen negativ auf die aufzubringende Rechenleistung auswirken. Möchte man darüber hinaus auch noch mehrere Replikationen fahren, können die Rechenzeiten mitunter sehr lange werden.
Beobachtet man nun einen Anwender eines Simulationssystems, so kann man feststellen, dass sich dieser häufig nur für bestimmte Teilaspekte des Systems interessiert und daher nur bestimmte Zustände beobachtet, andere jedoch außer Acht lässt. Er würde also nichts davon merken, wenn die nicht beobachteten Zustände nicht mehr vorhanden wären, solange sichergestellt ist, dass sich die Ablaufcharakteristik der Simulation bezüglich der beobach-

[1] Da die Warteschlange potenziell unendlich lang werden kann, man aber endlich viele Zustände erhalten will, behilft man sich damit, alle Zustände über einer bestimmten Schlangenlänge N zu einem gemeinsamen Zustand zusammenzufassen.

teten Zustände nicht verändert.
Ein Beispiel hierfür wäre die Simulation einer industriellen Maschine, die in nicht näher bestimmten Abständen aus verschiedenen Gründen ausfallen kann und dann wieder mit Zeit- und Kostenaufwand repariert werden muss. Möchte man hierfür einen Zustandsgrafen erstellen, muss man alle möglichen Konfigurationen in Betracht ziehen, das heisst in diesem Beispiel alle möglichen Kombinationen von ausgefallenen Einzelkomponenten. Die zugehörigen Kanten ergeben sich dann aus den Wahrscheinlichkeitsverteilungen für die Lebensdauer der einzelnen Komponenten. Möchte man für diesen Fall einen auch nur annähernd ausreichenden Zustandsgrafen modellieren, kann die Anzahl der Zustände und Kanten sehr hoch werden. Obwohl es viele verschiedene Ausfallarten geben kann, die im allgemeinen auch verschiedene Ausfallzeiten nach sich ziehen können, liegt das eigentliche Interesse des Anwenders möglicherweise jedoch nur darauf, ob die Maschine arbeitet oder nicht. Das heisst, es muss einen Grafen geben, der nur zwei Zustände enthält, der jedoch zur Lösung der gestellten Aufgabe völlig ausreichend ist. Die Simulation dieses kleineren Grafen würde natürlich wesentlich weniger Rechenzeit in Anspruch nehmen als die des ursprünglichen Grafen. Andererseits benötigt natürlich das Finden dieses neuen Grafen wiederum Rechenzeit. Allgemein liegt es also nahe, nach Möglichkeiten zu suchen, einen Grafen so zu transformieren, dass eine Simulation desselben mit weniger Rechenleistung durchgeführt werden kann. Ein eventueller Gewinn an Rechenzeit würde natürlich durch eine solche Transformation wieder vermindert, die ja ihrerseits selbst Rechenleistung verbraucht. Es stellt sich also die Frage, ob und wenn ja unter welchen Umständen sich diese Vorgehensweise lohnt. Dies soll im Folgenden untersucht werden.

Kapitel 3

Der realisierte Simulator

Im Rahmen dieser Arbeit wurde ein Programm entwickelt, das es erlaubt, Zustandsgrafen der beschriebenen Form einzulesen, Simulationen durchzuführen und die im nächsten Kapitel beschriebenen Reduktionsverfahren auf den Grafen anzuwenden. In diesem Kapitel soll zunächst nur auf die Simulation eingegangen werden. Dabei geht es hauptsächlich um die Realisierung der mit den Kanten verknüpften Verteilungsfunktionen bzw. Dichten sowie um die Generierung von entsprechenden Zufallszahlen. Abschließend werden die Ergebnisse eines Verifikationsexperiments angegeben.

3.1 Das Format der einzulesenden Grafen

Der zu simulierende Graf muss in einer ASCII-Datei der folgenden Form gegeben sein:

```
HEADER 6 9
E 0 1 n 9.757690 1.646210
E 1 2 e 1.664063
E 2 0 n 0.796814 0.512604
...
E 5 0 e 8.514709
S 2
S 5
```

Die erste Zeile muss das Schlüsselwort `HEADER`, gefolgt von der Anzahl der enthaltenen Zustände und der Anzahl der Kanten, enthalten. Darunter folgt die Liste aller Kanten, wobei jede Kante in einer eigenen Zeile stehen muss. Ein Kanteneintrag besteht aus dem Schlüsselbuchstaben `E`, den Indizes der Zustände an Kantenanfang und Kantenende, und

Kennbuchstabe	Verteilungstyp	Parameter
e	Exponential	λ
n	Normal	μ, σ
w	Weibull	α, β
g	Erlang	λ, k
o	Hypoexponential	λ_1, λ_2
r	Hyperexponential	λ_1, λ_2

Tabelle 3.1: Übersicht über die unterstützten Verteilungstypen

dem Kennbuchstaben für die Art der Wahrscheinlichkeitsverteilung gefolgt von den zugehörigen Parametern. In Tabelle 3.1 ist eine Übersicht der implementierten Verteilungstypen gegeben.
Nach der Auflistung aller Kanten des Grafen müssen nun nur noch die Indizes aller als *wichtig* markierten Zustände folgen. Für jeden solchen Zustand muss in einer eigenen Zeile der Buchstabe `S`, gefolgt von dem Index des betreffenden Zustands stehen. Dadurch erhält man die Möglichkeit, die wichtigen und unwichtigen Teile des Grafen festzulegen. Jeder Zustand, der bei der Simulation beobachtet werden soll, muss in dieser Liste enthalten sein. Damit teilt man dem Programm mit, dass es nicht erlaubt ist, Reduktionsverfahren anzuwenden, die die Aufenthaltswahrscheinlichkeit und Durchgangshäufigkeit dieses Zustandes ändern können. Für alle Zustände, die nicht in dieser Liste enthalten sind, ist dies jedoch erlaubt.

3.2 Interne Repräsentation der Verteilungsfunktionen

Hinsichtlich der Weiterverarbeitbarkeit der Verteilungsfunktionen bzw. der Dichten reicht es nicht aus, nur den Verteilungstyp und die zugehörigen Parameter zu speichern. Dies liegt zum einen daran, dass nicht für alle Typen eine analytische Beschreibung der zugehörigen Verteilungsfunktion möglich ist, diese jedoch zur Generierung einer entsprechenden Zufallszahl benötigt wird. Vor allem aber muss man für die Faltung zweier analytisch gegebener Dichten, auf die im nächsten Kapitel aufgebaut wird, ein Integral lösen, für welches nur in speziellen Fällen eine geschlossene Lösung bekannt ist. Diese Berechnungen müssen also numerisch durchgeführt werden. Um dies zu ermöglichen, müssen auch die Dichten in numerischer Form gegeben sein. Um diese numerische Form zu erhalten, muss die zu verarbeitenden Funktion an verschiedenen Stellen ausgewertet werden, um statt mit der geschlossenen Form mit diesem Satz an Funktionswerten weiterzurechnen. Diese

Überführung in eine gewisse Anzahl von *Stützstellen* nennt man *Diskretisierung*.

3.2.1 Diskretisierung

Es liegt nahe, eine zu diskretisierende Funktion mittels einer Treppenfunktion zu approximieren. Ein bestimmter Bereich der Zeitachse wird also in eine gewisse Zahl an Intervallen aufgeteilt, wobei der Funktionswert innerhalb eines Intervalls konstant ist. Dabei geht man folgendermaßen vor: Zunächst muss man sich entscheiden, wie oft und an welchen Stellen die Originalfunktion ausgewertet werden soll. Eine solche Stelle soll im folgenden auch 'Ort der Stützstelle' genannt werden. Dann führt man eben diese Auswertungen durch und ersetzt die Funktion durch eine stückweise konstante Funktion, wobei der Funktionswert jedes konstanten Stücks gleich dem Funktionswert der Originalfunktion in der jeweiligen Intervallmitte ist (siehe Abbildungen 3.1 und 3.2). Das Ergebnis einer solchen Funktionsauswertung einer Stützstelle heißt auch 'Wert der Stützstelle'.

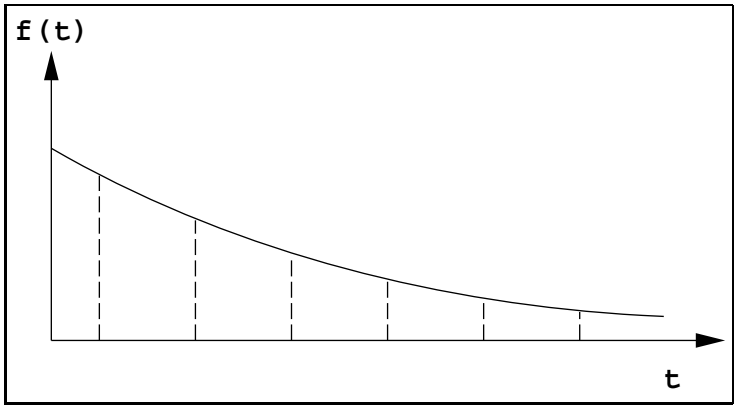


Abbildung 3.1: Auswertung der Dichtefunktion an sechs äquidistanten Stützstellen

Jetzt kann anstatt mit der ursprünglichen Dichtefunktion mit der entstandenen Treppenfunktion gerechnet werden.
Es stellt sich heraus, dass es in Hinblick auf eine später durchzuführende Faltung sinnvoll ist, die zu berechnenden Stützstellen so zu wählen, dass diese auf der Zeitachse alle den gleichen Abstand voneinander haben, also *äquidistant* zueinander sind. Weiterhin ist es, um zwei diskrete Funktionen zu falten, erforderlich, dass die Intervallbreiten beider Funktionen gleich sind. Dieses ist natürlich a priori nicht gegeben. Um das sehr rechenaufwändige Umrechnen mittels Interpolation zu vermeiden, wird so diskretisiert, dass sich die Intervallbreiten sämtlicher möglichen Kantenpaare nur um den Faktor 2^k ($k \in N_0$), unterscheiden. Um dies zu erreichen, wird unter Betrachtung aller im Grafen enthaltenen Kanten eine

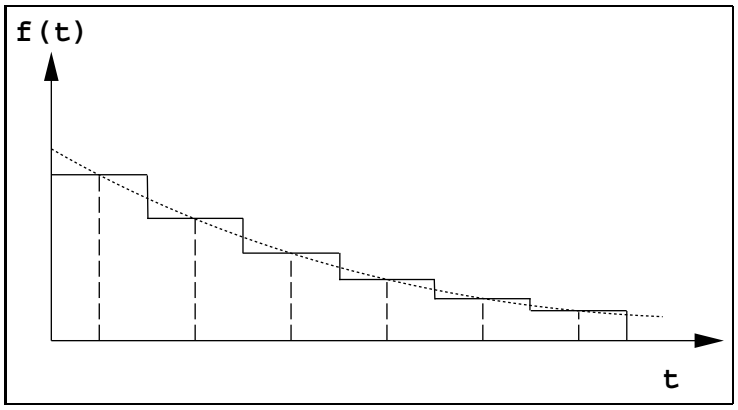


Abbildung 3.2: Ersetzen der Dichtefunktion durch Treppenfunktion

elementare Intervallbreite δ ermittelt; danach darf als tatsächliche Intervallbreite für eine Dichte nur noch $2^k\delta$, $(k \in N_0)$, verwendet werden. Damit wird die geforderte Bedingung erfüllt. Wie die verschiedenen Intervallbreiten zweier Dichten dann tatsächlich angeglichen werden, wird weiter unten in Abschnitt 4.2.1 erklärt.
Es ist außerdem darauf zu achten, dass das Integral über eine Dichtefunktion immer den Wert eins haben muss. Diese Eigenschaft kann sich durch den Approximationsfehler der Diskretisierung ändern, was wiederum durch Umskalierung ausgeglichen werden muss. Desweiteren muss man beachten, dass die Treppenfunktion einerseits einen begrenzten Träger hat und andererseits auch nur eine begrenzte Auflösung; man sollte also Überlegungen anstellen, wie die beiden Parameter Stützstellenzahl und Trägerbreite gewählt werden müssen, um den Approximationsfehler gering zu halten. Für die Exponentialverteilung mit dem Parameter λ etwa wurden die Stützstellen so gewählt, dass sie das Intervall $[0; \frac{10}{\lambda}]$ überdecken. Bei $t = \frac{10}{\lambda}$ bricht die Treppenfunktion also ab, und somit bleibt die Dichtefunktion für alle höheren Zeiten unberücksichtigt. Die Wahrscheinlichkeit, bei einer Exponentialverteilung mit dem Parameter λ eine Zufallszahl $t > \frac{10}{\lambda}$ zu erhalten, beträgt jedoch nur $1 - F_\lambda(\frac{10}{\lambda}) = e^{-10} \approx 4.54 \cdot 10^{-5}$ (unabhängig von λ), und kann damit als vernachlässigbar gelten. $F_\lambda(t)$ bezeichnet dabei die zugehörige Verteilungsfunktion, weiterführende Informationen dazu findet man zum Beispiel in [Fel68].

3.2.2 Auswertung

Möchte man zu einer gegebenen Verteilungsfunktion $F(t)$ eine entsprechende Zufallszahl t_0 generieren, geht man üblicherweise so vor, dass man eine uniform zwischen 0 und 1 verteilte Zufallszahl u_0 generiert und dann $t_0 = F^{-1}(u_0)$ berechnet. Das Problem dabei ist, das F^{-1} nicht für alle Typen von Verteilungsfunktionen berechnet werden kann. In unserem Fall,

nämlich der stückweise konstanten Dichte, ist dies jedoch nicht notwendig. Man erhält die zu einer solchen Dichte zugehörige Verteilungsfunktion durch einfaches Aufaddieren der Intervalle der Dichte (siehe Abbildung 3.3). Es muss lediglich noch so skaliert werden, dass das letzte Intervall die Höhe eins hat.

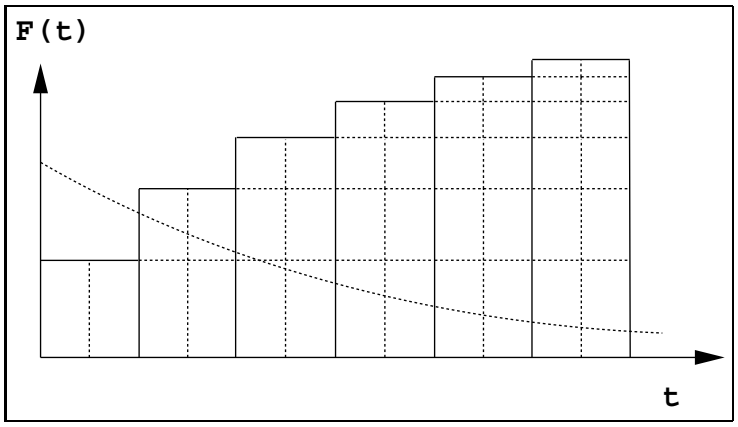


Abbildung 3.3: Von der Dichte zur Verteilungsfunktion

Um nun einen zufälligen Wert t_0 zu erhalten, wird wiederum eine uniform zwischen 0 und 1 verteilte Zufallsvariable ausgewertet, die den Wert u_0 liefert. Danach muss dasjenige benachbarte Intervallpaar gefunden werden, von denen eines kleiner und das andere größer als u_0 ist. Den endgültigen gesuchten Wert t_0 erhält man schließlich durch lineare Interpolation (siehe Abbildung 3.4).

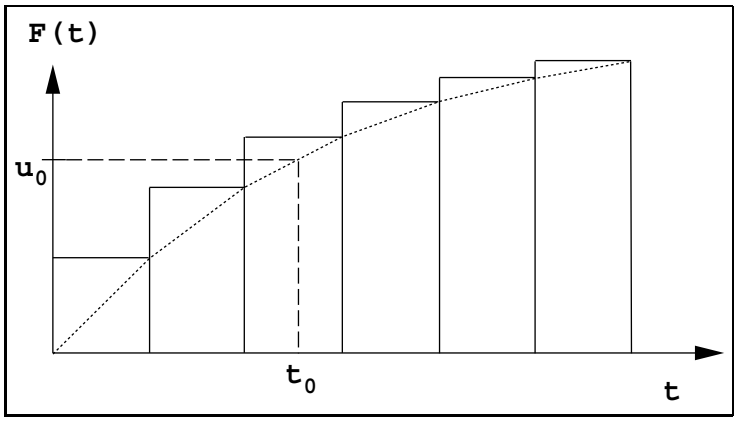


Abbildung 3.4: Auswertung der Zufallsvariable durch lineare Interpolation

3.3 Durchführung der Simulation

Startet man nun die Simulation, führt das Programm beginnend mit dem Startzustand solange Zustandswechsel durch, bis die vorgegebene Simulationszeit erreicht worden ist. Nach Beendigung der Berechnungen wird für jeden einzelnen Zustand die Aufenthaltshäufigkeit ausgegeben, also der Anteil an der gesamten Simulationszeit, in dem ein bestimmter Zustand als besetzt markiert war. Außerdem erhält man noch die Durchlaufanzahl für jeden Zustand, also die Anzahl wie oft er besucht wurde.
Darüber hinaus bietet das Programm die Möglichkeit, mehrere Replikationen zu fahren. Dabei erhält man nach der Durchführung von R Replikationen für jeden Zustand Z_i R Werte $X_{i,j}$ $(j = 1, .., R)$ für die Aufenthaltshäufigkeit in Zustand i bei Replikation j. Aus diesen Werten wird dann die mittlere Aufenthaltshäufigkeit

$$\hat{\theta}_i = \frac{1}{R} \sum_{j=1}^{R} X_{i,j} \tag{3.1}$$

und ein Schätzwert für die Varianz dieses Mittelwerts

$$\hat{\sigma}^2(\hat{\theta}_i) = \frac{1}{R(R-1)} \left(\sum_{j=1}^{R} X_{i,j}^2 - \frac{1}{R} (\sum_{j=1}^{R} X_{i,j})^2 \right) \tag{3.2}$$

für jeden einzelnen Zustand Z_i berechnet. Die Quellen für die Formeln 3.1 und 3.2 sowie weitere Informationen zum Thema Konfidenzintervalle sind in [BCN96] zu finden.

3.4 Verifikation des Simulators anhand eines analytisch lösbaren Grafen

Es ist naheliegend, die Ergebnisse des Simulators mit der analytischen Lösung eines lösbaren Grafen vergleichen zu wollen. Hierfür wurde ein Graf mit vier Zuständen gewählt, dessen Kanten ausschließlich mit exponentiell verteilten Zufallsvariablen versehen sind. Der jeweilige Parameter λ ist bei jeder Kante angegeben (Abbildung 3.5).
Berechnet und gemessen wurde jeweils die Aufenthaltswahrscheinlichkeit bzw. -häufigkeit in einem Zustand. Um den Einfluss der Vorbelegung [1] vernachlässigbar zu machen, muss die Simulationszeit hinreichend lang sein. Sie wurde hier so gewählt, dass der Graf mindestens 30000-mal durchlaufen wird.
In Tabelle 3.2 sind in der ersten Zeile die analytischen Lösungen, in allen weiteren Zeilen die Messwerte, jeweils in Prozent, für verschiedene N angegeben. N bezeichnet die Anzahl

[1] Eine solche Vorbelegung ergibt sich in erster Linie durch den Startzustand.

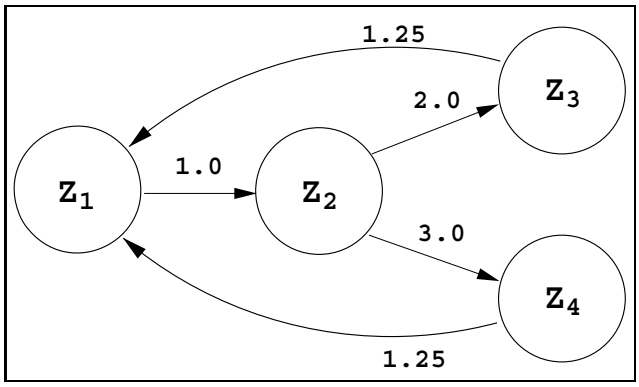


Abbildung 3.5: Graf zur Verifikation

N	Z_1	Z_2	Z_3	Z_4
analyt.	50,000	10,000	16,000	24,000
20	49.516698	10,088748	15,652620	24,741934
40	49,873814	10,035147	15,902862	24,188176
80	49,938833	10,011621	15,998564	24,050984
160	49,951484	10,012473	16,003035	24,033009

Tabelle 3.2: Ergebnis der Verifikation

der Stützstellen, die zur internen Repräsentation einer einzelnen Dichte verwendet werden, und ist somit ein Maß dafür, wie gut die generierten Zufallszahlen den analytischen Dichten entsprechen.
Erwartungsgemäß nähert sich der Messwert mit steigendem N der analytischen Lösung an.

Kapitel 4

Reduktionsschritte und ihre Realisierung

Dieses Kapitel befasst sich mit verschiedenen Verfahren zur Einsparung von Rechenzeit für die Simulation durch Reduktion des Zustandsgrafen. Dabei werden neben bereits bekannten Reduktionen wie zum Beispiel die der reinen Sequenz auch Lösungen für bislang unreduzierbare Strukturen, insbesondere des Splits, präsentiert. Eine zentrale Rolle spielt dabei die Erkenntnis, dass die Auswahl einer von mehreren Kanten, die einen Zustand verlassen, separiert werden kann von der Auswertung der Kante, also der Betimmung einer entsprechenden Zufallszahl für die Zeit, die durch den anschließenden Zustandswechsel entlang dieser Kante verbraucht wird.
Die einzelnen Reduktionsverfahren werden dabei danach unterschieden, ob die ursprüngliche Ablaufstruktur der durchzuführenden Simulation erhalten bleiben muss, oder ob zum Beispiel Zustände übersprungen werden dürfen. Dabei muss der Anwender, also derjenige, der den Grafen zu Verfügung stellt, für jeden einzelnen Zustand vorgeben, ob er ihn beobachten will, ihn also die Aufenthaltswahrscheinlichkeit des Zustands interessiert, oder ob er gegebenenfalls übersprungen werden darf. Auch für Kanten wäre eine solche Differenzierung denkbar, in der erstellten Implementierung wurde jedoch darauf verzichtet, da der verwendete Simulator ohnehin nur die Beobachtung der entsprechend markierten Zustände erlaubt (siehe Abschnitt 3.3).
Als Abschluss dieses Kapitels werden die Grenzen der verwendeten Reduziermethoden aufgezeigt, es ist nämlich noch nicht gelungen, auch Strukturen mit Schleifen, also Zuständen, die ihr eigener Nachfolgezustand sind, zu vereinfachen.

4.1 Ablauferhaltende Methoden

Sollte es der Fall sein, dass untypischerweise die statistischen Eigenschaften sämtlicher Zustände und Kanten für die Systemanalyse unverzichtbar sind, gibt es dennoch Möglichkeiten, den Simulationsablauf zu beschleunigen: Die im folgenden beschriebene Verwendung von Verzweigungswahrscheinlichkeiten ist weder von den Zuständen noch von den Kanten aus sichtbar, gleichgültig ob diese relevant sind oder nicht. Sind hingegen zwar alle Zustände, jedoch nicht alle Kanten von Belang, kann zusätzlich auf das in Abschnitt 4.1.2 beschriebene Verschmelzen von Parallelkanten zurückgegriffen werden. Da im erstellten Simulator ohnehin die Kanten nicht beobachtet werden, werden beide Verfahren als ablauferhaltend eingestuft, und deshalb beide in diesem Abschnitt beschrieben.

4.1.1 Berechnung und Verwendung von Verzweigungswahrscheinlichkeiten

Wie im zweiten Kapitel beschrieben, muss, um einen Zustand verlassen zu können, für jede von ihm ausgehende Kante eine der zugehörigen Wahrscheinlichkeitsverteilung entsprechende Zufallszahl generiert werden, was im folgenden kurz als 'Kante auswerten' bezeichnet werden soll. Dies wirkt sich bei Grafen mit hoher Konnektivität, also hohem Verhältnis von Kantenanzahl zu Zustandsanzahl, negativ auf die Rechenzeit aus, da, um einen Zustand verlassen zu können, entsprechend viele Kanten ausgewertet werden müssen. Betrachtet man diesen Mechanismus genauer, so kann man feststellen, dass darin eigentlich zwei Schritte enthalten sind, nämlich zum einen die Auswahl derjenigen Kante, über die der Zustand letztlich verlassen werden soll, und zum anderen die Auswertung eben dieser Kante. Nun wird ein Verfahren vorgestellt, wie sich diese beiden Schritte von einander trennen lassen.

Es existiert offensichtlich für jede Kante eine Wahrscheinlichkeit dafür, dass sie benutzt wird, sobald der zugehörige Ausgangszustand besetzt ist. [1] Diese ist gleich der Wahrscheinlichkeit, dass die zugehörige Zufallsvariable einer bestimmten Kante einen kleineren Wert liefert als die Zufallsvariablen aller anderen Kanten. Besitzt ein Zustand n von ihm ausgehende Kanten mit zugeordneten Zufallsvariablen X_i, $(i = 1, .., n)$, und möchte man wissen, mit welcher Wahrscheinlichkeit p_{X_j} die Kante j benutzt wird so muss die Wahrscheinlichkeit dafür bestimmt werden, dass X_j kleiner ist als alle anderen Variablen

[1] Dabei summieren sich die Wahrscheinlichkeiten über alle verlassenden Kanten eines Zustands natürlich zu eins auf.

$X_1, .., X_{j-1}, X_{j+1}, .., X_n$. Es gilt hierfür:

$$p_{X_j} = P\left((X_j < X_1) \wedge ... \wedge (X_j < X_{j-1}) \wedge (X_j < X_{j+1}) \wedge ... \wedge (X_j < X_n)\right) \tag{4.1}$$

$$p_{X_j} = \int_{t=0}^{t=\infty} f_{X_j}(t) \cdot (1 - F_{X_1}(t)) \cdot ... \cdot (1 - F_{X_{j-1}}(t)) \cdot (1 - F_{X_{j+1}}(t)) \cdot ... \cdot (1 - F_{X_n}(t)) dt \tag{4.2}$$

Dabei bezeichne f_{X_j} die Dichtefunktion der Kante X_j, und F_{X_j} ihre Verteilungsfunktion. Da nur nicht-negative Zeiten erlaubt sind, reicht es aus, als untere Integrationsgrenze 0 zu setzen.

Man kann sich nun mit Hilfe der Formel 4.2 für jede verlassende Kante eines Zustands deren Verwendungswahrscheinlichkeit, im folgenden Verzweigungswahrscheinlichkeit genannt, berechnen. Bei der anschließenden Simulation kann dann, wenn dieser Zustand verlassen werden soll, auf das Auswerten aller in Frage kommenden Kanten verzichtet werden. Man braucht nur mit Hilfe einer uniform verteilten Zufallszahl, die dabei generiert werden muss, zu bestimmen, welche Kante nun Verwendung finden soll, um dann nur diese eine Kante auszuwerten. Man muss also nur noch maximal zwei Zufallszahlen generieren. Je mehr Kanten einen Zustand verlassen, also je höher der Konnektivitätsgrad des Grafen ist, um so mehr Rechenzeit lässt sich mit Hilfe dieser Transformation einsparen.

Für die Realisierung im Simulationsprogramm muss aufgrund der diskreten Darstellung der Dichten und Verteilungsfunktionen das Integral in Formel 4.2 in eine Summe umgewandelt werden. Summiert wird dabei über alle N Stützstellen der betreffenden Kante. Bezeichnet man den Ort der i-ten Stützstelle von f_{X_j} mit t_i, und ihren Wert mit $f_{X_j}[i]$, dann ergibt sich für die gesuchte Wahrscheinlichkeit

$$p_{X_j} = \sum_{i=1}^{N} f_{X_j}[i] \cdot (1 - F_{X_1}(t_i)) \cdot ... \cdot (1 - F_{X_{j-1}}(t_i)) \cdot (1 - F_{X_{j+1}}(t_i)) \cdot ... \cdot (1 - F_{X_n}(t_i)). \tag{4.3}$$

Die Werte der verwendeten Verteilungsfunktionen $F_{X_\cdot}(t_i)$ müssen durch lineare Interpolation der diskreten Funktionen $F_{X_\cdot}$ gewonnen werden.

Es stellt sich heraus, dass es nicht korrekt ist, nur diese Verzweigungswahrscheinlichkeiten zu berechnen und dann unter deren Verwendung zu simulieren. Wird nämlich im ursprünglichen Zustandsgrafen eine bestimmte Kante durchlaufen, geschieht dies implizit unter der Bedingung, dass alle anderen Kanten, die vom selben Zustand ausgehen, langsamer waren, also die den Verteilungsfunktionen entsprechenden Zufallszahlengeneratoren alle einen jeweils höheren Wert geliefert haben als derjenige der verwendeten Kante.

Hat also ein Zustand wie in Abbildung 4.1 mehr als eine von ihm ausgehende Kante, dann müssen, um Verzweigungswahrscheinlichkeiten verwenden zu können, die jeweiligen Verteilungsfunktionen durch ihre bedingten Versionen ersetzt werden.

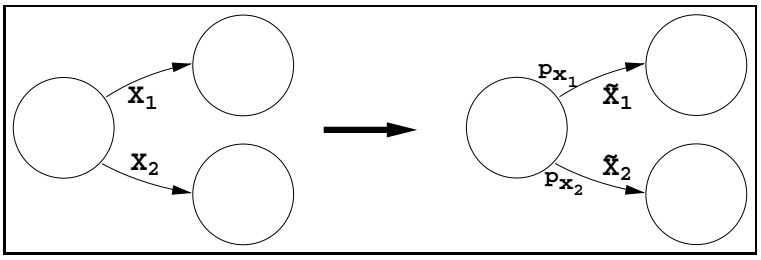


Abbildung 4.1: Die Verwendung von Verzweigungswahrscheinlichkeiten erfordert die Anpassung der Kanten

Die Notwendigkeit für diese Anpassung kann man sich anschaulich klar machen, indem man einen Zustand betrachtet, der nur über zwei Kanten mit identischen Verteilungsfunktionen $X_1 = X_2$ verlassen werden kann. Die Verweildauer in diesem Zustand ergibt sich nun zu $min(X_1, X_2)$. Verwendet man jedoch während des Simulationslaufs zur Kantenauswahl die Verzweigungswahrscheinlichkeiten ($p_{X_1} = p_{X_2} = 0.5$), erhält man für die Verweildauer gerade X_1 bzw. X_2, was natürlich nicht mehr richtig ist. Ein anschauliches Beispiel stellt das Werfen von zwei Laplace-Würfeln dar: Wirft man beide Würfel und nimmt die kleinere der beiden Zahlen als Ergebnis, entspricht dies der Vorgehensweise des in Kapitel 2 vorgestellten Simulators, wenn ein Zustand über zwei entsprechende Kanten verlassen werden kann und die Aufenthaltsdauer im Zustand durch die schnellere, das heisst die Kante mit der kürzeren Zeit, bestimmt wird. Legt man sich nun vorher entsprechend der Verzweigungswahrscheinlichkeiten für eine Kante fest, um diese dann zu verwenden, hieße dies im Beispiel mit den Würfeln, sich erst mit der Wahrscheinlichkeit $p = 0.5$ für einen der beiden Würfel zu entscheiden, ihn dann zu werfen, und seinen Wert als Ergebnis zu nehmen. Dies liefert natürlich eine andere Wahrscheinlichkeitsverteilung als das Minimum zweier Würfel. Die Wahrscheinlichkeit, dass eine Kante eine bestimmte Zeit beansprucht, muss also ersetzt werden durch eben diese Wahrscheinlichkeit unter der Bedingung, dass die Zeiten aller anderen Kanten länger waren. Im gezeigten Beispiel muss also X_1 ersetzt werden durch $\tilde{X}_1$ mit der zugehörigen Dichtefunktion

$$f_{\tilde{X_1}}(t) = f_{X_1}(t)|(X_1 < X_2) \tag{4.4}$$

$$\Rightarrow f_{\tilde{X_1}}(t) = f_{X_1}(t)\frac{(1 - F_{X_2}(t))}{P(X_1 < X_2)}, \tag{4.5}$$

also der Wahrscheinlichkeit für X_1, einen bestimmten Wert t anzunehmen, unter der Bedingung, dass X_1 kleiner ist als die konkurrierende Kante X_2.

Für den allgemeinen Fall ergibt sich

$$f_{\tilde{X_j}}(t) = f_{X_j}(t)|((X_j < X_1) \wedge ... \wedge (X_j < X_{j-1}) \wedge (X_j < X_{j+1}) \wedge ... \wedge (X_j < X_n)) \quad (4.6)$$

$$\Rightarrow f_{\tilde{X_j}}(t) = f_{X_j}(t)\frac{(1 - F_{X_1}(t)) \cdot ... \cdot (1 - F_{X_{j-1}}(t)) \cdot (1 - F_{X_{j+1}}(t)) \cdot (1 - F_{X_n}(t))}{p_{X_j}}. \quad (4.7)$$

Diese Anpassung der Kanten kostet natürlich wiederum nicht vernachlässigbare Rechenzeit. Es sei nochmals darauf hingewiesen, dass diese Transformation, einmal durchgeführt, weder von den Zuständen noch von den Kanten aus sichtbar ist. Die Ablaufcharakteristik bleibt also unangetastet. Auftretende Abweichungen sind auf die Diskretisierung zurückzuführen, bei der die Orte der Stützstellen unverändert bleiben und nur deren Werte, wieder mittels linearer Interpolation, angepasst werden.

4.1.2 Vereinigung von Parallelkanten

Interessiert sich der Anwender nur für Eigenschaften von Zuständen, ist es ihm gleichgültig, zum Beispiel wie oft eine Kante durchlaufen wird, da ja der Simulationsablauf nur hinsichtlich der Zustände erhalten werden muss. Dies eröffnet eine weitere Möglichkeit zur Vereinfachung des Zustandsraums, nämlich die Vereinigung von zwei oder mehr parallelen Kanten, also Kanten, die sowohl denselben Ausgangszustand wie auch denselben Zielzustand besitzen, wie in Abbildung 4.2 gezeigt.

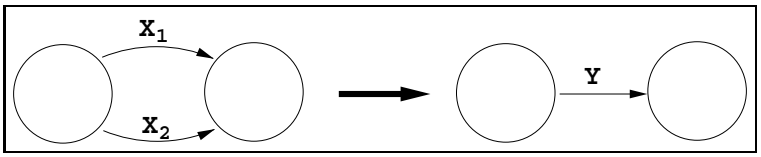


Abbildung 4.2: Zusammenfassung von parallelen Kanten

Liegt der Graf in seiner ursprünglichen Form vor, d.h. ohne Verzweigungswahrscheinlichkeiten, können die parallelen Kanten durch eine einzige neue Kante ersetzt werden, deren Verteilungsfunktion sich aus dem Minimum aller parallelen Kanten berechnen lässt. Möchte man wie im dargestellten Beispiel zwei Parallelkanten vereinigen, deren zugeordnete Verteilungsfunktionen $F_{X_1}(t)$ und $F_{X_2}(t)$ gegeben sind, lässt sich die entsprechende Funktion $F_Y(t)$ für die neue Kante folgendermaßen berechnen:
Für

$$Y := min\{X_1, X_2\} \quad (4.8)$$

gilt

$$P(Y > t) = P(X_1 > t)P(X_2 > t)$$

$$\Rightarrow 1 - P(Y \leq t) = (1 - P(X_1 \leq t))(1 - P(X_2 \leq t))$$

und damit

$$F_Y(t) = 1 - (1 - F_{X_1}(t))(1 - F_{X_2}(t)). \tag{4.9}$$

Für den allgemeinen Fall von n Kanten ergibt sich

$$F_Y(t) = 1 - \prod_{i=1}^{n}(1 - F_{X_i}(t)). \tag{4.10}$$

Die zugehörige Dichtefunktion $f_Y(t)$ kann, falls benötigt, durch Ableitung von $F_Y(t)$ gewonnen werden.

Um die Formel 4.10 in einem Simulator verwenden zu können, muss natürlich wieder eine diskrete Version angegeben werden. Zunächst bestimmt man den Ort und die Abstände der Stützstellen der neuen Kante. Es erweist sich als zweckmäßig, für den Ort der ersten Stützstelle das Minimum der Orte aller ersten Stützstellen, und für den Ort der letzten Stützstelle das Minimum der Orte aller letzten Stützstellen zu wählen. Bezeichnet man die j-te Stützstelle der Verteilungsfunktion F_{X_i} mit t_{j,X_i}, dann ergibt sich für die erste und letzte Stützstelle von F_Y

$$t_{1,Y} = \min_i(t_{1,X_i}), \qquad t_{N,Y} = \min_i(t_{N,X_i}). \tag{4.11}$$

Dadurch deckt man nämlich genau den Bereich aller möglichen Zufallswerte ab. Natürlich muss die bei der Diskretisierung gemachte Forderung, nämlich dass sich die Intervallbreiten zweier beliebiger Kanten nur durch eine Potenz des Faktors zwei unterscheiden, weiterhin erfüllt bleiben. Das macht es im allgemeinen notwendig, den abgedeckten Bereich zu vergrößern, also $t_{1,Y}$ kleiner und $t_{N,Y}$ größer zu wählen, und zwar so, dass die Bedingung erfüllt wird. Alle weiteren Stützstellen dazwischen ergeben sich dann zwingend aus der Äquidistanzeigenschaft.
Nun werden die Stützstellen einfach analog der Formel für die kontinuierliche Version gefüllt, wobei die benötigten Werte für $F_{X_i}(t)$ durch lineare Interpolation zwischen den entsprechenden Stützstellen gewonnen werden.
Die beschriebene Vorgehensweise ist nur korrekt, wenn der Zustandsgraf, oder zumindest der betroffene Teil, in seiner ursprünglichen Form vorliegt. Wurden die betrachteten Kanten, wie in Abschnitt 4.1.1 beschrieben, bereits mittels Berechnung von Verzweigungswahrscheinlichkeiten modifiziert, dann wird jede Kante, und damit auch die ihr zugeordnete Zufallsvariable mit der gegebenen Wahrscheinlichkeit benutzt. Das bedeutet, dass sowohl

die Verteilungsfunktion als auch die Dichte der neuen Kante durch gewichtete Addition der jeweiligen Funktionen der ursprünglichen Kanten ermittelt werden können. Als Gewichte hierfür sind die gegebenen Verzweigungswahrscheinlichkeiten zu verwenden, also ergibt sich für n Kanten:

$$F_Y(t) = \sum_{i=1}^{N} p_{X_i} F_{X_i}(t) \tag{4.12}$$

$$f_Y(t) = \sum_{i=1}^{N} p_{X_i} f_{X_i}(t) \tag{4.13}$$

Die Wahrscheinlichkeit, mit der die neue Kante beschritten wird, ergibt sich aus der Summe der ursprünglichen Verzweigungswahrscheinlichkeiten, also

$$p_Y = \sum_{i=1}^{N} p_{X_i}. \tag{4.14}$$

In der diskreten Version sind zunächst wieder die Orte der Stützstellen festzulegen. Im Gegensatz zum ersten beschriebenen Fall ohne Verzweigungswahrscheinlichkeitenergibt sich hier für die erste und letzte Stützstelle

$$t_{1,Y} = \min_i(t_{1,X_i}), \qquad t_{N,Y} = \max_i(t_{N,X_i}), \tag{4.15}$$

es müssen also alle Stützstellen aller beteiligten parallelen Kanten mit den neuen Stützstellen abgedeckt werden. Alles weitere wird analog zum ersten Fall berechnet, also mittels linearer Interpolation, etc.

4.2 Ablaufverändernde Methoden

Wie bereits erwähnt, gilt für die allermeisten Simulationsaufgaben, die mittels Zustandsgrafen repräsentiert werden können, dass nur ein Bruchteil der enthaltenen Zustände wirklich zur gesuchten Lösung beiträgt. Man versucht nun, den Grafen einer Transformation zu unterwerfen, die einerseits den Simulationsablauf bezüglich der beobachteten Zustände nicht verändert, andererseits aber dafür sorgt, dass unwichtige Zustände übersprungen werden. Im Folgenden wird dies zunächst am einfachstmöglichen Fall, der Sequenz, erläutert. Anschließend wird das Verfahren verallgemeinert, um auch kompliziertere Strukturen verarbeiten zu können. Insbesondere wird ein Verfahren vorgestellt, mit dem auch an Verzweigungen beteiligte Zustände übersprungen werden können, ohne die Entscheidungscharakteristik an der Verzweigung zu verändern.

Ob ein Zustand zum Simulationsergebnis beiträgt oder nicht, ist anwendungsabhängig und muss zusammen mit dem Grafen gegeben sein. Jeder Zustand hat also noch ein zusätzliches Attribut, welches ihn als *wichtig* oder *unwichtig* kennzeichnet. Dabei bedeutet *unwichtig*, dass weder die Wahrscheinlichkeit, sich in diesem Zustand aufzuhalten, noch die Anzahl der Durchgänge durch ihn von Belang sind.

4.2.1 Verkürzung von Sequenzen

Der einfachste Fall, bei dem ein Zustand übersprungen werden darf, ist eine Sequenz aus drei Zuständen (Z_1, Z_2 und Z_3), wobei die beiden ersteren, also Z_1 und Z_2, als *unwichtig* markiert sein müssen und nur jeweils genau eine Kante zum Verlassen besitzen dürfen. Außerdem sind keine weiteren Kanten erlaubt, über die man den mittleren Zustand Z_2 betreten könnte (Abbildung 4.3).

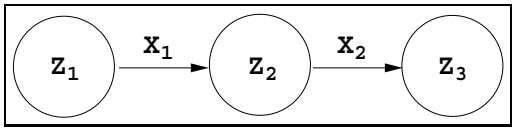


Abbildung 4.3: Eine Sequenz aus drei Zuständen

Es besteht nun zum Zweck der schnelleren Simulation die Möglichkeit, die beiden Kanten und den mittleren Zustand durch eine einzige neue Kante zu ersetzen. Bezeichnen wir die Zufallsvariablen der ursprünglichen Kanten mit X_1 und X_2, dann ergibt sich für die neue Kante $Y = X_1 + X_2$ (Abbildung 4.4).

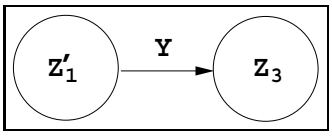


Abbildung 4.4: Vereinfachte Sequenz

Die Dichte von Y, f_Y, lässt sich mittels *Faltung* der beiden Dichten f_{X_1} und f_{X_2} berechnen. Es gilt

$$f_Y(t) = f_{X_1}(t) * f_{X_2}(t) = \int_0^t f_{X_1}(\tau) f_{X_2}(t-\tau)\, d\tau \tag{4.16}$$

Da nur nicht-negative Zufallsvariablen erlaubt sind, genügt es, statt über die gesamte Zeitachse nur von 0 bis t zu integrieren. Falls auch die zugehörige Verteilungsfunktion $F_Y(t)$ benötigt wird, kann diese durch Integrieren von $f_Y(t)$ ermittelt werden.

Im realisierten Simulator muss natürlich wieder mit diskreten Dichten, also mit Treppenfunktionen, gerechnet werden. Im folgenden wird ein Lösungsverfahren präsentiert, das es ermöglicht, zwei Dichten, die als diskrete Funktionen gegeben sind, zu falten. Wie bereits erwähnt, wird dabei vorausgesetzt, dass sich die Intervallbreiten der beiden Dichten nur um den Faktor 2^k ($k \in N_0$) unterscheiden dürfen. Diese Eigenschaft wurde jedoch bereits bei der Diskretisierung selbst sichergestellt.
Gegeben seien zwei diskrete Dichten f und g, mit jeweils N Stützstellen. Die Intervallbreiten seien $k\delta$ für f und $l\delta$ für g. Hierbei müssen k und l beide Potenzen von zwei sein. δ bezeichnet die kleinste erlaubte Intervallbreite und wurde bereits bei der Diskretisierung der beteiligten Dichten ermittelt. Die i-te Stützstelle von f bzw. g heiße r_i bzw. s_i. Unter Zuhilfenahme eines Rechtecksimpulses als Basisfunktion lassen sich beide Dichten durch jeweils eine Summenformel darstellen. Die Basisfunktion lautet:

$$B_a(t) = \begin{cases} 1 & \text{für} \quad 0 \leq t < a \\ 0 & \text{sonst} \end{cases} \tag{4.17}$$

Damit ergibt sich für die beiden Dichten:

$$f(t) = \sum_{\mu=0}^{N-1} r_\mu \cdot B_{k\delta}(t - o - \mu k\delta) \tag{4.18}$$

$$g(t) = \sum_{\nu=0}^{N-1} s_\nu \cdot B_{l\delta}(t - p - \nu l\delta) \tag{4.19}$$

Dabei bezeichnet o bzw. p jeweils den linken Rand des ersten Intervalls.

Um zwei diskrete Funktionen zu falten, ist es erforderlich, dass die Intervallbreiten beider Funktionen gleich sind. Dieses ist natürlich a priori nicht gegeben. Es wurde aber bereits bei der Diskretisierung der Dichten sichergestellt, dass sich die Intervallbreiten sämtlicher möglichen Kantenpaare nur um den Faktor 2^k ($k \in N_0$) unterscheiden. Im folgenden wird unter Ausnutzung dieser Eigenschaft ein effizientes Verfahren zur Berechnung der Faltung von f und g angegeben.

Es sei nun ohne Beschränkung der Allgemeinheit $k < l$, und es sei $m = \frac{l}{k}$, womit m natürlich auch eine Potenz von zwei ist.
Damit lassen sich beide Dichten mittels der gleichen Basisfunktion, also als Treppenfunktionen mit gleich breiten Intervallen, darstellen:

$$f(t) = \sum_{\mu=0}^{N-1} r_\mu \cdot B_{k\delta}(t - o - \mu k\delta) \tag{4.20}$$

$$g(t) = \sum_{\nu=0}^{Nm-1} s_{\lfloor \frac{\nu}{m} \rfloor} \cdot B_{k\delta}(t - p - \nu k\delta) \tag{4.21}$$

Man gleicht also die verschiedenen Intervallbreiten zweier Dichten an, indem man die Intervalle der jeweils gröberen entsprechend oft halbiert.

Wendet man nun die Definition für die Faltung an, so erhält man:

$$(f * g)(t) = \int_0^t \left(\left(\sum_{\mu=0}^{N-1} r_\mu \cdot B_{k\delta}(\tau - o - \mu k\delta) \right) \left(\sum_{\nu=0}^{Nm-1} s_{\lfloor \frac{\nu}{m} \rfloor} \cdot B_{k\delta}(t - \tau - p - \nu k\delta) \right) \right) d\tau \tag{4.22}$$

Multipliziert man die beiden Summen miteinander, ergibt sich:

$$(f * g)(t) = \int_0^t \sum_{\mu=0}^{N-1} \left(\sum_{\nu=0}^{Nm-1} \left(r_\mu \cdot s_{\lfloor \frac{\nu}{m} \rfloor} \cdot B_{k\delta}(\tau - o - \mu k\delta) \cdot B_{k\delta}(t - \tau - p - \nu k\delta) \right) \right) d\tau \tag{4.23}$$

und damit:

$$(f * g)(t) = \sum_{\mu=0}^{N-1} \sum_{\nu=0}^{Nm-1} \left(r_\mu \cdot s_{\lfloor \frac{\nu}{m} \rfloor} \cdot \int_0^t B_{k\delta}(\tau - o - \mu k\delta) \cdot B_{k\delta}(t - p - \tau - \nu k\delta) d\tau \right) \tag{4.24}$$

Nun enthält das Integral nur noch die beiden Basisfunktionen. Löst man das Integral, erhält man das Ergebnis der Faltung zweier gleich breiter Rechtecksimpulse. Es ergibt sich eine Dreiecksfunktion der Form

$$C_a(t) = \begin{cases} \frac{1}{a}t & \text{für} \quad 0 \leq t < a \\ 2 - \frac{1}{a}t & \text{für} \quad a \leq t < 2a \\ 0 & \text{sonst.} \end{cases} \tag{4.25}$$

Man erhält also für die Faltung aus f und g:

$$(f * g)(t) = \sum_{\mu=0}^{N-1} \sum_{\nu=0}^{Nm-1} \left(r_\mu \cdot s_{\lfloor \frac{\nu}{m} \rfloor} \cdot k\delta \cdot C_{k\delta}(t - p - o - (\mu + \nu)k\delta) \right) \tag{4.26}$$

Nun ordnet man die Summe so um, dass nicht mehr über alle Kombinationen von r_μ und $s_{\lfloor \frac{\nu}{m} \rfloor}$ summiert wird, sondern über alle verschiedenen Basisfunktionen $C_{k\delta}$. Dazu führt man eine neue Summationsvariable $\rho := \mu + \nu$ ein:

$$(f * g)(t) = \sum_{\rho=0}^{N+Nm-2} \left(\sum_{\mu+\nu=\rho} r_\mu \cdot s_{\lfloor \frac{\nu}{m} \rfloor} \right) \cdot k\delta \cdot C_{k\delta}(t - p - o - \rho k\delta) \tag{4.27}$$

Man erhält eine Funktion, die aus einer gewichteten Summe von verschobenen Dreiecksfunktionen besteht. Die Gewichte erhält man durch Berechnung der zugehörigen inneren Summe in Formel 4.27. Damit keine Multiplikation mehrfach ausgeführt werden muss, bietet es sich an, zu Anfang ein Wertetabelle zu erstellen, die die Produkte sämtlicher möglichen Kombinationen aus μ und ν enthält.
Man kann nun beobachten, dass an einer Stelle, wo eine beliebige Dreiecksfunktion maximal ist, alle anderen Dreiecksfunktionen null sind. Daraus folgt, dass der Funktionswert des Faltungsergebnisses an dieser Stelle genau dem Gewicht dieser Dreiecksfunktion entspricht. Es entsteht also eine stückweise lineare Funktion, deren Funktionswerte an den Knickstellen durch die berechneten Gewichte gegeben sind. Dadurch ist das Ergebnis vollständig bestimmt.
Ein Beispiel für zwei zu faltende Dichten ist in Abbildung 4.5 angegeben, jede Dichte wird durch je $N = 3$ Intervalle repräsentiert, es gilt $k = 1$ und $l = 2$, damit unterscheiden sich die Intervallbreiten um den Faktor $m = 2$. Die Intervalle der Funktion g müssen also einmal halbiert werden, was in der Abbildung durch gestrichelte Linien gekennzeichnet ist.

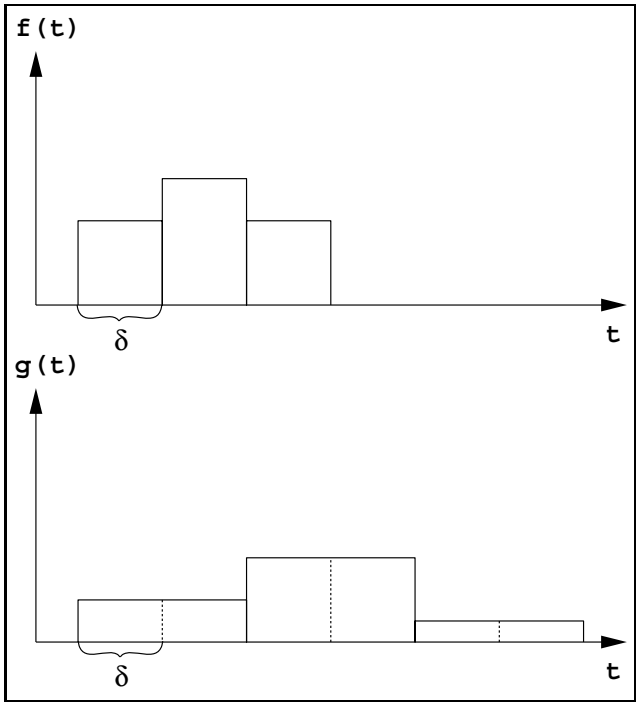


Abbildung 4.5: Angleichung der Intervallbreiten

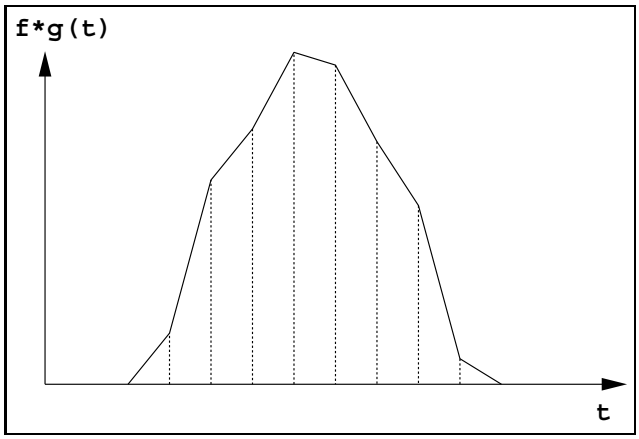


Abbildung 4.6: Ergebnis der Faltung der obigen Dichtefunktionen

Um nun die Faltung auszuführen, müssen wie beschrieben die Gewichte der Dreiecks-Basisfunktionen berechnet werden, was zu einer stückweise linearen Funktion führt (Abbildung 4.6).
Nun möchte man jedoch wiederum eine stückweise konstante Funktion erhalten, mit der dann eventuell noch weitere Faltungen ausgeführt werden können. Dazu berechnet man für jedes Intervall den Mittelwert, der sich wegen der Linearität aus dem Mittelwert der linken und rechten Grenze ergibt. (Abbildung 4.7)

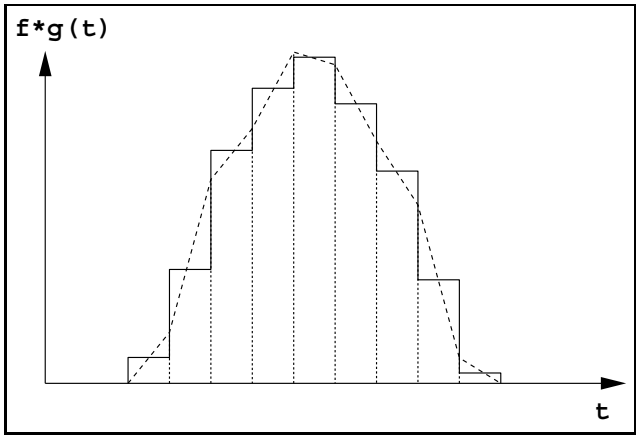


Abbildung 4.7: Endgültiges Ergebnis der Faltung

Es bleibt nur noch festzustellen, dass sich die Anzahl der Stützstellen durch die Faltung erhöht hat. Um dies zu kompensieren, kann man einerseits Randintervalle mit sehr geringen Funktionswerten, die sich bei der Faltung typischerweise ergeben, weglassen, oder die Intervallbreite verdoppeln, indem man Paare von benachbarten Intervallen verschmilzt.

Man erhält also am Ende als Ergebnis der Faltung wieder eine Treppenfunktion mit N Stützstellen und Intervallbreite $2^k\delta$, mit $(k \in N_0)$.

4.2.2 Verallgemeinerung der Sequenzverkürzung

Da die oben beschriebene Faltung nur auf den Spezialfall einer reinen Sequenz anwendbar ist, wird man sie in einem beliebigen Grafen nur selten einsetzen können. Wünschenswert wäre also ein Verfahren, das weniger strenge Forderungen an die Grafstruktur stellt.
Betrachtet man die Vereinfachung der reinen Sequenz in Abschnitt 4.2.1 genauer, so kann man feststellen, dass man den Zustand Z_2 genau genommen nicht entfernt hat, sondern ihn mit Zustand Z_1 verschmolzen hat. Das heisst, die Aufenthaltswahrscheinlichtkeit für den neuen Zustand Z_1' ergibt sich aus der Summe der beiden Wahrscheinlichkeiten für die ursprünglichen Zustände Z_1 und Z_2. Man hat also eine Verschiebung von Aufenthaltswahrscheinlichkeiten vorgenommen, und da die Restwahrscheinlichkeit des Zustands Z_2 gleich null war, konnte man ihn weglassen. Wenn man nun das Verschieben von Wahrscheinlichkeiten von der Löschung von Zuständen trennt, ergibt sich folgenden Vorgehensweise:
Zur Beschleunigung der Simulation allein reicht es aus, die Kante von Z_1 nach Z_2 zu entfernen, und eine neue Kante von Z_1 nach Z_3 in den Grafen einzufügen, versehen mit der durch die Faltung berechneten Zufallsvariablen. Sowohl der Zustand Z_2 als auch die Kante von Z_2 nach Z_3 können zunächst bestehen bleiben. Nun kann der Zustand Z_2 gelöscht werden, da er ja nicht mehr erreichbar ist. Führt man dagegen nur den ersten Schritt aus, kann eine der obigen Forderungen fallen gelassen werden; es ist nämlich nun erlaubt, dass der mittlere Zustand Z_2 auch von anderen Zuständen als von Z_1 erreichbar ist. In diesem Fall kann er dann eben nicht sofort nach der Faltung entfernt werden. Damit kann eine weitere elementare Figur vereinfacht werden, und zwar:

Die Merge-Figur

Als Merge-Figur wird der Fall bezeichnet, dass zwei als unwichtig markierte Zustände einen gemeinsamen Zustand als jeweils alleinigen Folgezustand haben, der ebenfalls unwichtig sein muss und seinerseits auch nur einen Nachfolgezustand besitzt (Abbildung 4.8).
Die Anwendung der Sequenzvereinfachung für die Sequenz $Z_1 \longrightarrow Z_3 \longrightarrow Z_4$ liefert zunächst das Ergebnis nach Abbildung 4.9. Der Zustand Z_3 darf jetzt natürlich nicht gelöscht werden, da er von Z_2 aus noch erreichbar ist und seine Aufenthaltswahrscheinlichkeit nicht 0 ist.
Nochmaliges Anwenden der Sequenzvereinfachung, diesmal auf $Z_2 \longrightarrow Z_3 \longrightarrow Z_4$, führt jedoch schließlich auf einen Teilgrafen nach Abbildung 4.10.

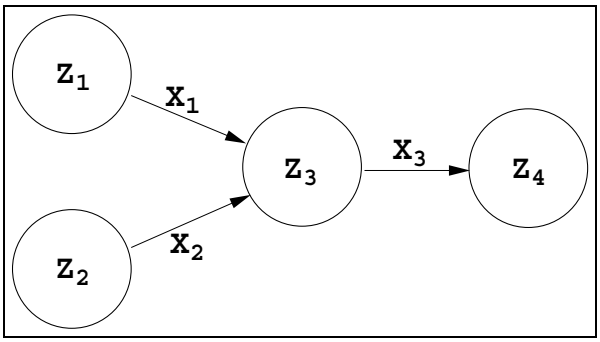


Abbildung 4.8: Die Merge-Figur

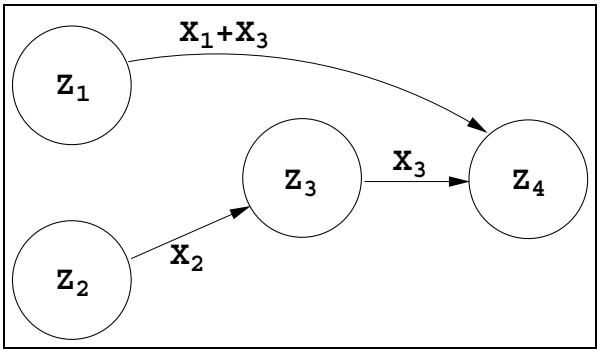


Abbildung 4.9: Die Merge-Figur nach Schritt 1

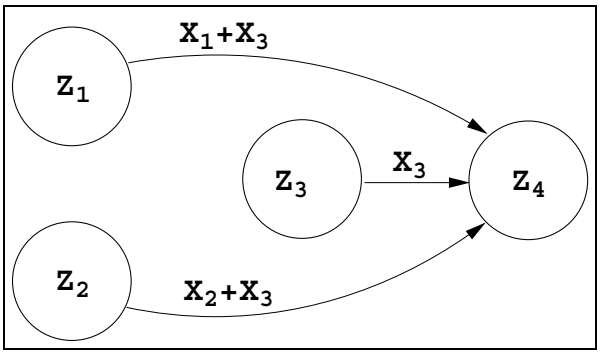


Abbildung 4.10: Die Merge-Figur nach Schritt 2

Nun ist Z_3 nicht mehr erreichbar, seine ursprüngliche Aufenthaltswahrscheinlichkeit wurde an die beiden Zustände Z_1 und Z_2 verteilt und beträgt nunmehr 0. Damit kann er letztlich doch weggelassen werden.
Eine weitere Einschränkung, die beim Auflösen einer Sequenz aus drei Zuständen $Z_1 \longrightarrow Z_2 \longrightarrow Z_3$ gemacht wurde, besteht darin, dass der Zustand Z_2 nur einen einzigen Nachfolgezustand besitzen darf. Möchte man auch diese Bedingung beseitigen, stößt man auf die

dritte Elementarfigur, nämlich:

Die Split-Figur

Besitzt der Mittelzustand Z_2 einer Sequenz $Z_1 \longrightarrow Z_2 \longrightarrow Z_3$ noch einen weiteren Nachfolgezustand Z_4, muss man sich bei der Simulation mit der Entscheidung für eine von mehreren möglichen Kanten befassen (Abbildung 4.11).

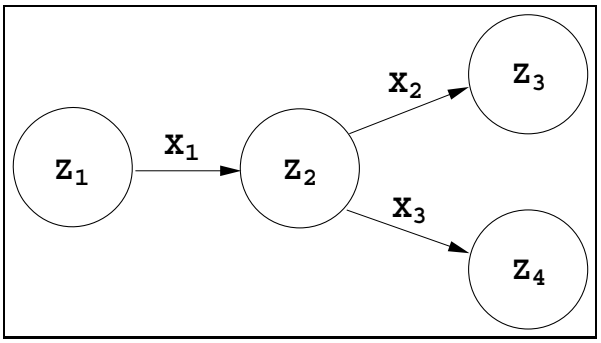


Abbildung 4.11: Die Split-Figur

Möchte man nun den Grafen mit dem Ziel, bei der Simulation den Zustand Z_2 zu überspringen, modifizieren, wäre ein naiver Ansatz, sowohl die Zustandsfolge $Z_1 \longrightarrow Z_2 \longrightarrow Z_3$ als auch die Folge $Z_1 \longrightarrow Z_2 \longrightarrow Z_4$ als einfache Sequenz zu bearbeiten, sprich, die jeweiligen Dichten zusammenzufalten, um die Figur in Abbildung 4.12 zu bekommen.

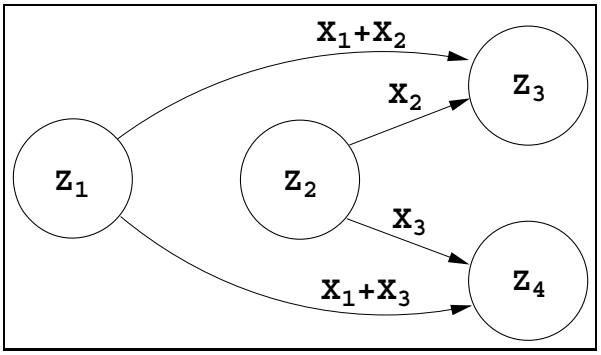


Abbildung 4.12: Die Split-Figur, vereinfacht nach dem naiven Ansatz

Man muss sich aber hierbei im klaren sein, dass man die Entscheidungscharakteristik, am Ende in Z_3 oder in Z_4 zu sein, verfälscht hat, wenn man so vorgeht. Im allgemeinen ist nämlich $P\left((X_1 + X_2) < (X_1 + X_3)\right)$ ungleich $P(X_2 < X_3)$, ebenfalls muss man von der Möglichkeit ausgehen, dass sich an die beiden Endzustände Z_3 und Z_4 Teile des Grafen

anschließen, die als *wichtig* markierte Zustände enthalten. Daher ist es wichtig, die Entscheidungscharakteristik zwischen Z_3 und Z_4 nicht zu verändern. Die obige Transformation verändert also die Ablauffolge der belegten Zustände und ist damit unzulässig.
Über einen Umweg ist es dennoch gelungen, auch für diese Grafstruktur eine korrekte Reduktionsmethode zu finden. Der Schlüssel zur Lösung liegt dabei in der Trennung von *Weg* und *Zeit*: Betrachtet man die Trennung von Kantenauswahl und Kantenauswertung bei den Verzweigungswahrscheinlichkeiten aus Abschnitt 4.1.1 dieses Kapitels, so fällt auf, dass sich dieses Prinzip auch hier zur Lösung des Problems einsetzen lässt. Führt man nämlich zunächst mit allen beteiligten Kanten die Berechnung der Verzweigungswahrscheinlichkeiten durch, und ersetzt die jeweiligen Dichten durch ihre bedingten Versionen, kann man die Auswahl des Weges, der durch den Grafen genommen wird, separieren von der Zeit, die dazu benötigt wird. Somit lässt sich die Vereinfachung des Splits dennoch ohne Verfälschung der anschließenden Simulation ausführen. In Abbildung 4.13 ist die mit Verzweigungswahrscheinlichkeiten versehene Version des ursprünglichen Grafen gegeben, darunter ist in Abbildung 4.14 das Ergebnis der korrekten Vereinfachung zu sehen. Die Verwendungswahrscheinlichkeiten sind jeweils neben den zugehörigen Kanten angegeben.

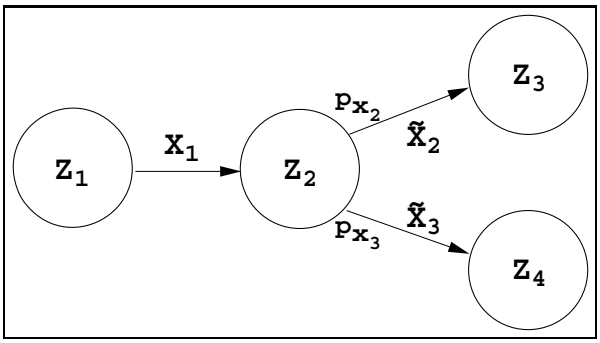


Abbildung 4.13: Die Split-Figur mit Verzweigungswahrscheinlichkeiten

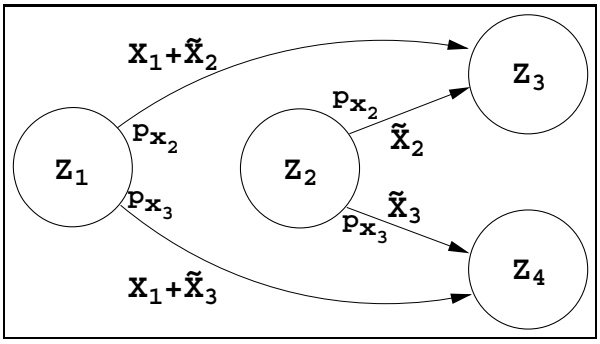


Abbildung 4.14: Die Split-Figur nach der korrekten Vereinfachung

Man beachte, dass die Anzahl der Kanten zunächst einmal ansteigt, da der Zustand Z_2 nicht weggelassen werden kann, sollte er noch weitere eingehende Kanten besitzen. Man kann jedoch im Hinblick auf die spätere Simulation dennoch von einer Vereinfachung sprechen, da man dabei dann einen Simulationsschritt einsparen wird, wenn man durch das betrachtete Grafstück hindurchlaufen will.
Da man mit dieser Methode die Auswahl des Weges von der Bestimmung der dafür benötigten Zeit getrennt hat, kann noch eine weitere Einschränkung, die bei der Behandlung der einfachen Sequenz gemacht wurde, fallen gelassen werden. Es ist unter Verwendung von Verzweigungswahrscheinlichkeiten nunmehr auch erlaubt, dass der erste Zustand des Splits Z_1 noch weitere verlassende Kanten haben darf. Ein Beispiel hierfür ist in Abbildung 4.15 angegeben, die Verzweigungswahrscheinlichkeiten wurden bereits berechnet. Die zugehörige Lösung ist in Abbildung 4.16 dargestellt.

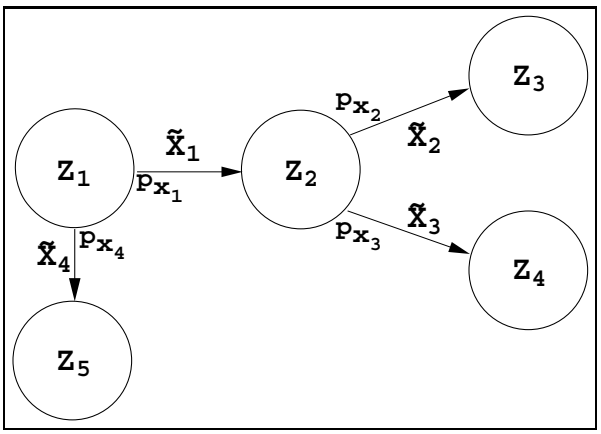


Abbildung 4.15: Die erweiterte Split-Figur

Als einzige Forderung bleibt bestehen, dass der mittlere Zustand der Sequenz keinen sogenannten *Loop*, eine Kante, die auf ihn selbst verweist, haben darf. Auf diese Bedingung wird in Abschnitt 4.2.3 noch näher eingegangen.

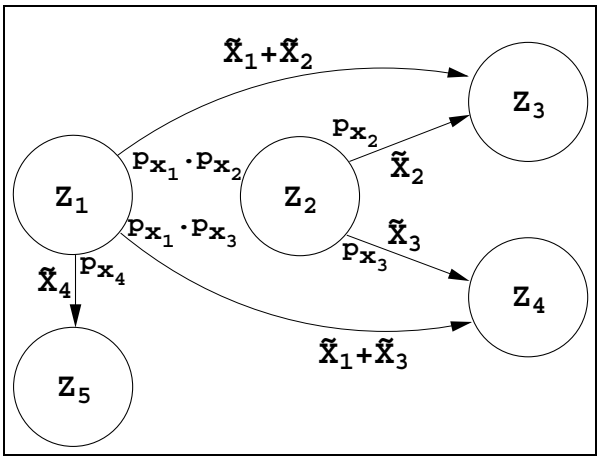


Abbildung 4.16: Die erweiterte Split-Figur nach der Vereinfachung

Zusammenfassend ergibt sich nun folgende Vorgehensweise:
Es müssen zwei Zustände Z_1 und Z_2 gefunden werden, die beide als *unwichtig* gekennzeichnet sind, und die durch eine Kante von Z_1 nach Z_2 verbunden sind (Abbildung 4.17).

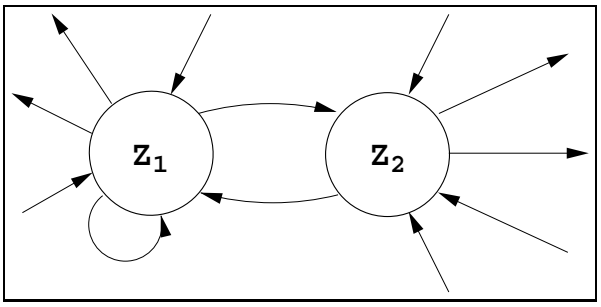


Abbildung 4.17: Der allgemeine Fall

Dabei darf keine der zum Zustand Z_2 gehörenden Kanten ein Loop sein. Als erstes werden, falls nicht bereits geschehen, für alle beteiligten Kanten, sprich für alle Kanten, die den Zustand Z_1 bzw. Z_2 verlassen, die jeweiligen Verzweigungswahrscheinlichkeiten berechnet und die den Kanten zugeordneten Verteilungsfunktionen wie oben beschrieben angepasst. Dann wird für alle Folgezustände Z_i von Z_2 die Faltung der Kante von Z_1 nach Z_2 mit der Kante von Z_2 nach Z_i durchgeführt, eine neue Kante von Z_1 nach Z_i eingefügt und mit dem Ergebnis der berechneten Faltung versehen. Danach werden für die neuen Kanten die Verzweigungswahrscheinlichkeiten berechnet und abschließend die Kante von Z_1 nach Z_2 entfernt.
Lässt man nun die Simulation laufen, kann der Marker für den besetzten Zustand von Zustand Z_1 direkt zu einem der Folgezustände Z_i gehen, ohne den Umweg über Z_2. Man

spart also einen Simulationsschritt ein.

4.2.3 Schleifen

Durch die Vereinfachung von ringförmigen Sequenzen oder selten auch schon in seinem Originalzustand kann ein Graf Schleifen, sogenannte *Loops* enthalten. Als Loop bezeichnet man eine Kante, deren Vorgängerzustand und Nachfolgezustand identisch sind. Es ist bisher nicht gelungen, eine befriedigende Lösung zur Aufgabe der Elimination einer solchen Kante zu finden. Enthält der Graf etwa eine Struktur wie in Abbildung 4.18 dargestellt, mit bereits berechneten Verzweigungswahrscheinlichkeiten und entsprechend angepassten Kanten, wäre eine Transformation in das in Abbildung 4.19 dargestellte Ergebnis wünschenswert.

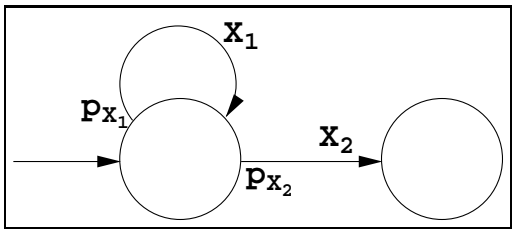


Abbildung 4.18: Teilgraf mit einer Loop-Kante

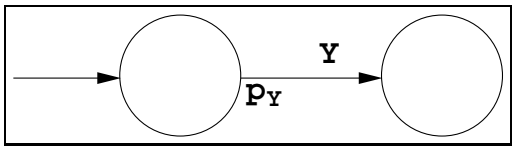


Abbildung 4.19: Gewünschtes Transformationsergebnis

Es kann für die Dichte f_Y von Y lediglich eine Lösung in Summenschreibweise angegeben werden:

$$f_Y = p_{X_2} f_{X_2} + p_{X_1} p_{X_2} (f_{X_1} * f_{X_2}) + p_{X_1}^2 p_{X_2} (f_{X_1} * f_{X_1} * f_{X_2}) + p_{X_1}^3 p_{X_2} (f_{X_1} * f_{X_1} * f_{X_1} * f_{X_2}) + ... \tag{4.28}$$

Da die Summe nicht bis ins Unendliche berechnet werden kann, muss man nach einer gewissen Anzahl von Summanden abbrechen, was natürlich dazu führt, das eine exakte Lösung, ganz abgesehen von der Diskretisierung der Dichten, unmöglich wird. Wieviele Summanden nötig sind, hängt von der geforderten Genauigkeit ab, und insbesondere von p_1. Im ungünstigsten Fall ist p_1 sehr nahe an der eins, mit dem Ergebnis, dass die Faktoren $p_1^i p_2$

vor den Summanden nur sehr langsam kleiner werden, es müssen also unverhältnismäßig viele Faltungen ausgeführt werden. Dies führt zu einem immensen Rechenaufwand, deshalb wurde auf eine Implementierung verzichtet.

4.2.4 Entfernung unbenutzter Grafelemente

Bei der wiederholten Anwendung der vorgestellten Reduktionsverfahren kommt es immer wieder vor, dass Zustände oder sogar ganze Grafteile nach der Transformation nicht mehr vom Startzustand aus erreichbar sind. Es empfiehlt sich, diese Zustände bzw. Grafteile sofort zu entfernen, sobald dies möglich ist. Zum einen reduziert sich damit der Verwaltungsaufwand, vor allem aber wird vermieden, dass möglicherweise aufwändige Faltungen berechnet werden, obwohl die betreffenden Zustände nie erreicht werden können.

4.3 Verifikation der Reduktionsschritte

Im folgenden sollen die im letzten Kapitel beschriebenen Reduktionsschritte auf geeignete Grafen angewendet werden, um die Ergebnisse der anschließenden Simulation mit der des Originalsystems zu vergleichen.

4.3.1 Verzweigungswahrscheinlichkeiten

Zur Verifikation der Verzweigungswahrscheinlichkeiten wird der gleiche Graf herangezogen wie zur Verifikation der Simulation selbst im letzten Kapitel. Es wird für die Verzweigung nach Zustand Z_2 die jeweilige Wahrscheinlichkeit berechnet, die obere oder untere Kante zu verwenden. Wie im Abschnitt 4.1.1 beschrieben, wird dadurch auch eine Kantenanpassung (bedingte Kanten) erforderlich. Darauf führt man die gleichen Schritte aus wie im letzten Abschnitt und erhält die Wertetabelle 4.1, wobei p die Wahrscheinlichkeit bezeichnet, die obere Kante zu benutzen. Es wurde dabei bis zu einer Simulationszeit von 10^5 Sekunden gerechnet, was einem Erwartungswert von 5×10^4 Grafdurchläufen entspricht.
Es lässt sich das gleiche Ergebnis ableiten wie bei der Verifikation der Simulation, allerdings ergibt ein Vergleich mit den Ergebnissen des ursprünglichen Grafen, dass die Genauigkeit nach der Berechnung der Verzweigungswahrscheinlichkeiten schlechter geworden ist. Insbesondere hat Zustand Z_3 zu Gunsten von Z_4 an Aufenthaltshäufigkeit verloren, was auf die Abweichung der Wahrscheinlichkeit p zurückzuführen ist. Diese Abweichung von der analytischen Lösung wird durch eine geringe Stützstellenzahl natürlich noch verstärkt.

N	p	Z_1	Z_2	Z_3	Z_4
analyt.	0.4	50,000	10,000	16,000	24,000
20	0,372499	49,218450	10,626561	14,945080	25,209906
40	0,392867	49,795836	10,181443	15,709134	24,313588
80	0,398200	49,931079	10,044071	15,926412	24,098437
160	0,399549	50,002143	10,010028	15,978151	24,009678

Tabelle 4.1: Verifikation der Verzweigungswahrscheinlichkeiten

4.3.2 Faltung

Zur Verifikation der Faltungsoperation wurde ein Ring mit drei Zuständen herangezogen, wie in Abbildung 4.20 dargestellt.

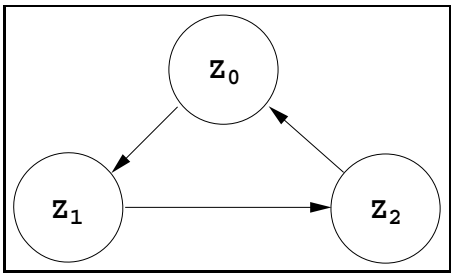


Abbildung 4.20: Ringgraf aus drei Zuständen

Ist nur der Zustand Z_0 interessant, so kann man die Kante von Z_1 nach Z_2 mit der Kante von Z_2 nach Z_0 falten, das Ergebnis als neue Kante von Z_1 nach Z_0 einfügen und den Zustand Z_2 entfernen.
Die Dichten für die Kanten wurden zufällig ausgewählt, wobei sichergestellt ist, dass die Intervallbreiten der zu faltenden Dichten verschieden sind. Es wurde wieder solange simuliert bis der Graf 5×10^4 Mal durchlaufen wurde. In Tabelle 4.2 sind jeweils die Aufenthaltshäufigkeiten für Z_0 vor und nach der Faltung zum Vergleich angegeben:

N	vorher	nachher
5	45,915226	45,924759
20	45,955254	45,924701
40	45,937043	45,937888

Tabelle 4.2: Verifikation der Faltung

Man kann beobachten, dass bereits für $N = 40$ Stützstellen die Faltung mit genügender Genauigkeit arbeitet. Offensichtlich eignet sich der implementierte Algorithmus recht gut, um im richtigen Intervall auf der Zeitachse der Dichte genügend Stützstellen zu berechnen, und damit den relevanten Bereich mit genügender Auflösung abzubilden.

4.3.3 Min-Term-Bildung

Zur Überprüfung der Min-Term-Bildung wurde der dafür einfachst mögliche Graf nach Abbildung 4.21 verwendet. Für die Kanten gilt wieder das gleiche wie bei der Faltung im letzten Abschnitt. Die Simulationszeit wurde wieder so gewählt, dass der Graf 5×10^4 Mal durchlaufen wird.

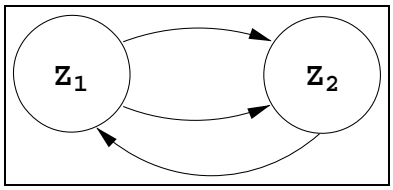


Abbildung 4.21: Minimalgraf mit einer Doppelkante

Die Ergebnisse sind in Tabelle 4.3 dargestellt.

N	vorher	nachher
20	31,768461	32,649244
40	31,433234	31,675353
80	31,383846	31,444243
160	31,360914	31,363852

Tabelle 4.3: Verifikation der Doppelkantenverschmelzung

Offensichtlich benötigt man für eine Min-Term-Bildung über 100 Stützstellen, also wesentlich mehr als bei der Faltung, um eine ausreichende Genauigkeit zu erreichen. Dies kann darauf zurückgeführt werden, dass die Dichte des Min-Terms zweier Zufallsvariablen die höchsten Werte im Bereich der Stützstellen hat, die für die kürzesten Zeiten stehen. Das heisst, die relevante Information verteilt sich nicht gleichmäßig über alle Stützstellen, sondern belegt nur einen Teil, während die Stützstellen für die höheren Zeitdauern weitgehend ungenutzt bleiben. Dieser Nachteil ließe sich nur durch nicht-äquidistante Verteilung der Stützstellen beheben, was jedoch der verwendete Faltungsalgorithmus verbietet.

Kapitel 5

Experimente und Ergebnisse

In diesem Kapitel werden die durchgeführten Experimente beschrieben und die gewonnenen Ergebnisse präsentiert. Zunächst wurde der Rechenaufwand für jeden einzelnen der im dritten Kapitel erklärten Reduktionsschritte ermittelt. Danach wurden nur noch zufällig generierte Grafen mit variablen Parametern wie Größe, Verzweigungsgrad, etc. betrachtet. Dabei wurde zunächst für verschiedene Grafen die Rechenzeit jeweils für die Simulation vor und nach der Reduktion sowie für die Reduktion selbst ermittelt. Anschließend wurden wiederum für verschiedene Grafen Genauigkeitsmessungen durchgeführt. Am Ende dieses Kapitels werden die gewonnen Ergebnisse zusammengefasst und bewertet.

5.1 Rechenaufwand der einzelnen Simulations- und Reduktionsschritte

Um einen Überblick darüber zu erlangen, in welchen Größenordnungen sich die Rechenzeiten für Simulation und Reduktion bewegen, werden diese in diesem Abschnitt, jeweils in Abhängigkeit von der Zahl der Stützstellen, angegeben.

5.1.1 Auswertung einer einzelnen Kante

Um den Rechenaufwand für einen einzelnen Zustandswechsel zu ermitteln, wurde ein ringförmiger Graf mit 1000 Zuständen herangezogen. Die Resultate sind in Tabelle 5.1 angegeben.
Die angegebene Rechenzeit versteht sich jeweils je Zustandswechsel. Zur Kantenauswertung muss ein bestimmtes Intervallpaar der diskreten Dichte gefunden werden (siehe Abbildung 3.4). Da für diese Suche ein logarithmischer Ansatz verwendet wurde, bleibt der Aufwand

Stützstellen	Rechenzeit in Sekunden
5	5.33e-07
20	8.12e-07
40	1.05e-06
80	1.33e-06
1000	2.93e-06

Tabelle 5.1: Rechenzeit für die Auswertung einer Kante

auch bei hohen Stützstellenzahlen gering. Die nach dem Finden des Intervallpaares erforderliche Interpolation hat einen von der Stützstellenzahl unabhängigen Aufwand.

5.1.2 Ermittlung der Verzweigungswahrscheinlichkeiten und Kantenanpassung

Hier wurde ein Zustand betrachtet, der eine eingehende und zwei verlassende Kanten besitzt. Es wurde für verschiedene Stützstellenanzahlen ermittelt, wieviel Zeit das Berechnen der zugehörigen Verzweigungswahrscheinlichkeiten und das Anpassen der Kanten benötigt (Tabelle 5.2).

Stützstellen	Rechenzeit in Sekunden
5	4.10e-05
20	6.28e-05
40	1.08e-04
80	2.72e-04
1000	1.92e-02

Tabelle 5.2: Rechenzeiten für die Verzweigungswahrscheinlichkeiten

Es ergibt sich ein starker Anstieg der Rechenzeit für große Stützstellenanzahlen, da für jede Stützstelle einer verlassenden Kante jeweils alle Stützstellen der anderen verlassenden Kante mit in die Berechnung einbezogen werden müssen. Wenn man sich dazu die Formeln 4.3 und 4.7 betrachtet, erkennt man, dass bei mehr Stützstellen sowohl die Anzahl der Summanden als auch der Aufwand für die Interpolation ansteigt. Es ergibt sich also ein superlineares Wachstum. Der schwächere Anstieg bei niedrigen Werten für die Anzahl

der Stützstellen ist darauf zurückzuführen, dass sich hier der Verwaltungsaufwand in der gleichen Größenordnung bewegt wie der eigentliche Rechenvorgang.

5.1.3 Die Faltung zweier Dichten

Bei der Messung der Rechenzeit für eine Faltung zweier Dichten wurde zusätzlich zur Stützstellenzahl auch noch der Unterschied der Intervallbreiten variiert, da dieser entscheidenden Einfluss auf die Zahl der durchzuführenden Multiplikationen hat. F bezeichnet in der Tabelle 5.3 den Faktor, um den sich die Intervallbreiten der beiden Dichten unterscheiden.

	Rechenzeit				
Stützstellen	$F = 1$	$F = 4$	$F = 16$	$F = 256$	$F = 1024$
5	1.3e-5	2.2e-5	5.0e-5	5.9e-4	2.4e-3
20	5.4e-5	1.5e-4	5.3e-4	8.4e-3	3.5e-2
40	1.8e-4	6.0e-4	2.0e-3	3.3e-2	1.4e-1
80	6.8e-4	2.6e-3	8.2e-3	1.3e-1	5.4e-1
1000	3.0e-1	7.8e-1	3.0e+0	4.1e+1	1.8e+2(!!)

Tabelle 5.3: Rechenzeit für eine Faltung in Sekunden

Erwartungsgemäß lässt sich feststellen, dass die Rechenzeit quadratisch von der Zahl der Stützstellen abhängt. Die zu hohe Zeit für eintausend Stützstellen rührt daher, dass bei der Ausführung zunächst eine Lookup-table angelegt wird, in der die Ergebnisse aller möglichen Multiplikationen von Stützstellen der einen und der anderen Kante abgelegt werden (vergleiche Abschnitt 4.2.1). Dies ist notwendig, um das mehrfache Ausführen von identischen Multiplikationen zu vermeiden. Aufgrund der Größe der Lookup-table von einer Million Fließkommawerten bei eintausend Stützstellen kann sie nicht mehr vollständig im schnelleren Cache-Speicher des verwendeten Rechners abgelegt werden. Die erforderliche Auslagerung in den normalen Hauptspeicher verbraucht die zusätzliche Rechenzeit.
Ebenfalls bemerkenswert ist der hohe Anstieg für stark unterschiedliche Intervallbreiten. Dieser kann dadurch erklärt werden, dass die Dichte mit der größeren Intervallbreite so viele Intervallhalbierungen, also Stützstellenverdopplungen, erfährt, bis die Intervalle beider Dichten gleich breit sind.

5.1.4 Die Minterm-Bildung

Hier wurde die zur Verschmelzung zweier paralleler Kanten benötigte Rechenzeit gemessen. Die dabei erhaltenen Resultate sind in Tabelle 5.4 angegeben.

Stützstellen	Rechenzeit in Sek.
5	1.4e-5
20	2.4e-5
40	3.7e-5
80	7.7e-5
160	2.0e-4
320	6.1e-4
640	2.1e-3
1000	5.1e-3
2000	1.9e-2

Tabelle 5.4: Rechenzeit für die Minterm-Bildung

Für große Stützstellenzahlen lässt sich eine quadratische Abhängigkeit für die Rechenzeit ablesen, für geringere Werte spielt der Verwaltungsaufwand eine stärkere Rolle.

5.2 Wirkung der Grafeigenschaften auf den Rechenaufwand

Der Rechenaufwand für Simulation und Reduktion und das Reduzierungspotenzial werden von verschiedenen Faktoren beeinflusst. Um einen Einblick in die gegenseitigen Beziehungen dieser Faktoren zu gewinnen, wurde eine Reihe von Experimenten durchgeführt, bei denen Größen wie die Anzahl der Zustände, die Grafkonnektivität, der Anteil der relevanten, also nicht überspringbaren Zustände, und die Anzahl der Stützstellen variiert wurden. Jede Zeile in einer der Tabellen entspricht dabei einem Experiment mit einem anderen Grafen. Die Bedeutung der Spaltenüberschriften ist in Tabelle 5.5 angegeben und ist für alle Tabellen dieses Abschnitts nahezu gleich, nur konstant gelassene Parameter sind in der Tabellenunterschrift angegeben. Für jedes Experiment wurde der zugehörige Graf einmal in seinem Originalzustand simuliert, dann wurden, soweit möglich, sämtliche in Kapitel 4 beschriebenen Reduktionen durchgeführt, um danach das reduzierte System zu simulieren.

$Stst.$	Anzahl der Stützstellen
$Zust.$	Anzahl der Zustände des Grafen
$Kanten$	Anzahl der Kanten des Grafen
$\%\ care$	Anteil der relevanten Zustände in Prozent
t_{davor}	Rechenzeit eines Simulationslaufs vor der Reduktion
t_{danach}	Rechenzeit eines Simulationslaufs nach der Reduktion
Z_{davor}	Anzahl der Zustandswechsel in der Simulation vor der Reduktion
Z_{danach}	Anzahl der Zustandswechsel in der Simulation nach der Reduktion
$t_{reduktion}$	Rechenzeit für die Reduktion
$\%\ profit$	Anteil der eingesparten an der ursprünglichen Rechenzeit in Prozent

Tabelle 5.5: Erklärung der Spaltenüberschriften der folgenden Tabellen

Die zentrale Frage, der hier nachgegangen werden soll, lautet, ob sich die Reduktion nun gelohnt hat, oder ob eine Simulation des Originalsystems effizienter gewesen wäre. Man könnte also für jedes Experiment entscheiden, ob der Gewinn an Rechenzeit $(t_{davor} - t_{danach})$ größer ist als die dafür investierten Kosten $t_{reduktion}$. Die Werte t_{davor} und t_{danach} und somit auch $(t_{davor} - t_{danach})$ sind jedoch linear abhängig von der Simulationszeit, wogegen die Rechenzeit für die Reduktion $t_{reduktion}$ konstant bleibt. Deshalb wurde auf eine direkte Auswertung verzichtet und statt dessen die Zeitersparnis relativ zur unreduzierten Simulation $profit = \frac{(t_{davor} - t_{danach})}{t_{davor}}$ bestimmt.
Es wurde bei allen Experimenten bis zur gleichen Simulationszeit gerechnet. Da bei konstanter Simulationszeit die Rechenzeit für die Simulation abhängig ist von den Verteilungsfunktionen, ist bei den Experimenten zum Zwecke der Reproduzierbarkeit zusätzlich die Anzahl der Zustandswechsel bei der Simulation jeweils vor und nach der Reduktion angegeben. Möchte man also nun mit einem zufällig generierten Grafen, der natürlich den jeweiligen Parametern des nachzuvollziehenden Experiments entsprechen muss, die Ergebnisse reproduzieren, muss nur die Simulationszeit so gewählt werden, dass sich die gleiche Anzahl an Zustandswechseln einstellt. Die Rechenzeiten sind dabei natürlich hardwareabhängig, die prozentuale Abweichung von den gegebenen Werten in den folgenden Tabellen sollte jedoch konstant sein.

5.2.1 Grafgröße und relevante Zustände

In der ersten Versuchsreihe wurde die Abhängigkeit von Reduktionspotenzial und -aufwand sowohl von der Anzahl der im Grafen enthaltenen Zustände als auch von dem Anteil der

relevanten Zustände untersucht. Als Grafgröße wurden die Werte 256, 1024, 4096 und 16384 gewählt, als Prozentsätze für den Anteil der relevanten Zustände wurden 5, 30, 55 und 80 zugelassen. Für alle Experimente wurde eine Stützstellenzahl von 20 verwendet, ebenfalls konstant gehalten wurde die Grafkonnektivität von 5 Kanten pro Zustand, was als Mittelwert über alle Zustände zu verstehen ist.

$Zust.$	$Kanten$	$\%\ care$	t_{davor}	t_{danach}	Z_{davor}	Z_{danach}	$t_{reduktion}$	$\%\ profit$
256	1280	5	5.42e–001	1.39e–001	227253	170228	7.24e–001	74.4
256	1280	30	4.87e–001	2.20e–001	205528	172759	6.96e–001	54.8
256	1280	55	6.88e–001	4.29e–001	271384	241673	2.19e–001	37.7
256	1280	80	5.00e–001	4.25e–001	199390	197555	7.33e–002	15.0
1024	5120	5	9.31e–001	2.50e–001	230051	170966	2.68e+000	73.2
1024	5120	30	1.24e+000	5.59e–001	300504	252264	2.56e+000	54.9
1024	5120	55	1.38e+000	9.23e–001	340221	317745	9.84e–001	33.1
1024	5120	80	9.29e–001	7.97e–001	235503	225969	4.14e–001	14.2
4096	20480	5	1.49e+000	4.55e–001	265039	203841	1.22e+001	69.5
4096	20480	30	1.65e+000	8.48e–001	294326	244577	2.10e+001	48.6
4096	20480	55	1.54e+000	1.07e+000	286241	259621	5.49e+000	30.5
4096	20480	80	1.59e+000	1.42e+000	281020	278564	1.42e+000	10.7
16384	81920	5	1.82e+000	8.51e–001	279724	215930	7.79e+001	53.2
16384	81920	30	1.78e+000	9.11e–001	281433	232251	4.35e+001	48.8
16384	81920	55	1.78e+000	1.25e+000	274490	247313	2.80e+001	29.8
16384	81920	80	1.96e+000	1.70e+000	292676	280422	7.66e+000	13.3

Tabelle 5.6: Rechenzeiten abhängig von der Grafgröße und dem Anteil der relevanten Zustände, $Stst. = 20$

Wie man aus der Tabelle 5.6 ersehen kann, steigt der Aufwand für die Reduktion in etwa linear mit der Grafgröße an. Ebenfalls den Erwartungen entspricht die Erkenntnis, dass sich um so mehr Rechenzeit durch die Reduktion einsparen lässt, je weniger Zustände für den Anwender von Bedeutung sind, da ja dann mehr Zustände zur Vereinfachung in Frage kommen. Interessant jedoch ist die Beobachtung, dass das Einsparpotenzial für größere Grafen geringer wird. Vergleicht man nämlich das Experiment in der ersten Zeile der Tabelle mit dem in der viertletzten Zeile, so unterscheiden sich die beiden Experimente nur in der Grafgröße, 256 bzw. 16384 Zustände, für den kleineren Grafen erhält man ein Einsparpotenzial von 74.4 %, beim großen Grafen dagegen nur 53.2 %.
Erhöht man nun bei konstanter Grafgröße den Anteil an relevanten Zuständen, verringert

sich natürlich das Einsparpotenzial, andererseits geht auch der dafür aufzubringende Rechenaufwand zurück. Bemerkenswert hierbei ist, dass der Rechenaufwand schneller sinkt als das Einsparpotenzial. Vergleicht man etwa die viertletzte mit der letzten Zeile, verringert sich der Rechenaufwand um den Faktor zehn, das Einsparpotenzial jedoch nur um den Faktor vier. Dies lässt sich damit begründen, dass bei vielen relevanten Zuständen die teuere Faltungsoperation weniger häufig angewendet werden kann, da sie nur bei zwei durch eine Kante verbundenen Zuständen, die beide als nicht relevant markiert sein müssen, eingesetzt werden kann. Muss man die Kosten, also den investierten Rechenaufwand im Auge behalten, ist die Reduktion eines Systems mit einem hohen Anteil an relevanten Zuständen günstiger. Andererseits ist das Reduktionspotenzial selbst natürlich höher, wenn dieser Anteil niedriger ist, da ja dann mehr Zustände für das Überspringen in Frage kommen.

5.2.2 Konnektivitätsgrad und relevante Zustände

In der zweiten Experimentreihe wurde im Vergleich zur ersten nicht die Grafgröße, sondern der Grafkonnektivitätsgrad variiert. Dieser Grad gibt an, über wie viele Kanten ein Zustand im Mittel verlassen werden kann. Die unterschiedlichen Werte für die Grafkonnektivität waren 2, 5, 10 und 20 Kanten pro Zustand. Damit ergaben sich bei einer konstanten Zustandsanzahl von 1024 für die Anzahl der Kanten die Werte 2048, 5120, 10240 und 20480. Als zweiter Parameter wurde wieder der Anteil der relevanten Zustände gewählt, mit den gleichen Werten wie in der ersten Versuchsreihe. Schließlich wurde wieder bei allen Experimenten eine Stützstellenzahl von 20 verwendet.
Simuliert man zwei Grafen mit unterschiedlichem Konnektivitätsgrad bis zu einer vorgegebenen Simulationszeit, benötigt man für den Grafen mit höherer Konnektivität wesentlich mehr Rechenzeit, da zum einen die durchschnittliche Aufenthaltsdauer in einem Zustand kürzer ist und somit mehr Zustandswechsel ausgeführt werden müssen, um die gegebene Simulationszeit zu erreichen, und zum anderen für einen Zustandswechsel mehr Kanten ausgewertet werden müssen. Dies lässt sich leicht an der Tabelle 5.7 verifizieren. Vergleicht man die Experimente mit Konnektivität 2 und 20, erkennt man, dass eine Verzehnfachung der Konnektivität einen Anstieg der Rechenzeit in etwa um den Faktor 1000 nach sich zieht. Andererseits birgt eine hohe Konnektivität auch mehr Einsparpotenzial hinsichtlich der Berechnung von Verzweigungswahrscheinlichkeiten. Am extremsten zeigt sich dies am Experiment in der viertletzten Zeile von Tabelle 5.7 mit Konnektivität 20, bei dem eine Reduktion der zur Simulation notwendigen Rechenzeit um mehr als 90 % ermittelt wurde. Bei niedrigeren Konnektivitäten geht das Einsparpotenzial zurück, bei im Mittel zwei Kanten pro Zustand beträgt es nur noch 56.4 %. Der Reduktionsaufwand steigt für höhere

$Zust.$	$Kanten$	$\%\ care$	t_{davor}	t_{danach}	Z_{davor}	Z_{danach}	$t_{reduktion}$	$\%\ profit$
1024	2048	5	5.71e–002	2.49e–002	54329	33386	9.21e–001	56.4
1024	2048	30	7.67e–002	4.46e–002	74237	53674	6.70e–001	41.9
1024	2048	55	7.56e–002	5.92e–002	73853	62633	1.85e–001	21.7
1024	2048	80	6.60e–002	6.04e–002	61739	59017	4.76e–002	8.5
1024	5120	5	1.29e+000	3.52e–001	318878	239694	3.83e+000	72.7
1024	5120	30	1.52e+000	6.86e–001	354810	297234	2.35e+000	54.9
1024	5120	55	1.20e+000	7.64e–001	284134	260611	1.08e+000	36.3
1024	5120	80	9.98e–001	8.25e–001	242361	231088	2.70e–001	17.3
1024	10240	5	1.21e+001	2.02e+000	1265369	924758	1.42e+001	83.3
1024	10240	30	1.36e+001	4.65e+000	1438577	1112876	5.95e+000	65.8
1024	10240	55	1.29e+001	7.21e+000	1360975	1161691	2.53e+000	44.1
1024	10240	80	1.32e+001	1.04e+001	1376286	1284551	1.04e+000	21.2
1024	20480	5	7.20e+001	6.26e+000	3570193	1781946	1.58e+001	91.3
1024	20480	30	7.02e+001	1.66e+001	3463595	2039506	9.89e+000	76.4
1024	20480	55	7.25e+001	3.05e+001	3583890	2452422	7.63e+000	57.9
1024	20480	80	7.42e+001	5.14e+001	3627960	3015994	4.17e+000	30.7

Tabelle 5.7: Rechenzeiten abhängig vom Konnektivitätsgrad und relevanten Zuständen, $Ststs. = 20$

Konnektivitäten zwar auch, jedoch wesentlich schwächer als die Rechenzeit des unreduzierten Systems. Erhöht man nämlich die Kantenzahl um den Faktor zehn, erhöht sich der Rechenaufwand für die Simulation des Originalsystems wie erwähnt um den Faktor 1000, der Anstieg des Reduktionsaufwands bleibt dagegen kleiner als Faktor 100. Je verzweigter das System also ist, desto mehr lohnt sich eine Vereinfachung desselben.
Bemerkenswert ist auch, dass selbst bei einem Anteil relevanter Zustände von 80 %, mit einer Konnektivität von 20, durch Reduktion noch etwa 30 % der Rechenzeit für die Simulation eingespart werden kann. Dieser Effekt ist darauf zurückzuführen, dass die Verwendung von Verzweigungswahrscheinlichkeiten, wie im Abschnitt 4.1.1 erläutert, den Ablauf der Simulation nicht ändert und somit auf jeden Zustand des Grafen anwendbar ist, gleichgültig ob er als relevant markiert ist oder nicht. Eine weitere Beobachtung, die hier gemacht werden kann, bestätigt diesen Rückschluss: vergleicht man den Gewinn bezüglich der Rechenzeit mit der Einsparung an Zustandswechseln, erkennt man, dass die Rechenzeit durch die Reduktion stärker gesunken ist als die Anzahl der Zustandswechsel. Daraus lässt sich folgern, dass die Rechenzeit pro Zustandswechsel ebenfalls gesunken sein muss, was wie-

derum auf die Berechnung der Verzweigungswahrscheinlichkeiten zurückzuführen ist.

5.2.3 Stützstellenzahl und Konnektivitätsgrad

In der dritten und letzten Versuchsreihe soll schließlich der Einfluss der Stützstellenzahl in Verbindung mit dem Konnektivitätsgrad untersucht werden. Dabei durchlief die Stützstellenzahl die Werte 5, 20, 40 und 80, für die Konnektivität wurde wieder 2, 5, 10 bzw. 20 gewählt. Die konstant gehaltenen Werte waren der Anteil der relevanten Zustände mit 30 % und die Grafgröße mit 1024 Zuständen.

$Ststs.$	$Zust.$	$Kanten$	t_{davor}	t_{danach}	Z_{davor}	Z_{danach}	$t_{reduktion}$	$\%\ profit$
5	1024	2048	5.52e–002	3.78e–002	68384	58356	2.26e–001	31.5
5	1024	5120	1.01e+000	4.17e–001	270866	213142	7.47e–001	58.7
5	1024	10240	6.96e+000	1.87e+000	798933	475984	1.77e+000	73.1
5	1024	20480	3.34e+001	5.60e+000	1756520	704412	3.45e+000	83.2
20	1024	2048	9.09e–002	5.07e–002	74167	52694	7.70e–001	44.2
20	1024	5120	1.89e+000	8.74e–001	355046	296876	2.67e+000	53.8
20	1024	10240	1.68e+001	5.79e+000	1437138	1110734	6.67e+000	65.5
20	1024	20480	8.55e+001	2.03e+001	3444538	1998757	1.15e+001	76.3
40	1024	2048	1.17e–001	6.66e–002	73943	53080	2.37e+000	43.1
40	1024	5120	2.33e+000	1.12e+000	361707	307462	8.28e+000	51.9
40	1024	10240	2.13e+001	8.13e+000	1559585	1330456	2.08e+001	61.8
40	1024	20480	1.13e+002	3.38e+001	3942255	2912228	3.26e+001	70.1
80	1024	2048	1.58e–001	7.93e–002	74728	52714	8.71e+000	49.8
80	1024	5120	3.09e+000	1.46e+000	366237	314656	3.05e+001	52.8
80	1024	10240	2.84e+001	1.06e+001	1600907	1432877	7.65e+001	62.7
80	1024	20480	1.52e+002	4.89e+001	4148872	3566076	1.20e+002	67.8

Tabelle 5.8: Rechenzeiten abhängig von der Anzahl der Stützstellen und vom Konnektivitätsgrad, $\%\ care = 30$

Zunächst lässt sich die Erwartung bestätigen, dass sich für alle Stützstellenzahlen nach einer Erhöhung der Konnektivität auch ein höheres Einsparpotenzial einstellt. Der Anstieg dieses Potenzials fällt jedoch um so geringer aus, je höher die Stützstellenzahl gewählt wird. Erhöht man die Konnektivität von zwei auf zehn, erhöht sich das Einsparpotenzial bei fünf Stützstellen von 31.5 % auf 83.2 %, bei 80 Stützstellen dagegen nur von 49.8 % auf 67.8 %. In der vorhergehenden Versuchsreihe wurde festgestellt, dass ein hoher Verzweigungsgrad

ein hohes Einsparpotenzial zur Folge hat. Bei höheren Stützstellenzahlen schwächt sich dieser Effekt demnach ab.

5.3 Wirkung der Grafeigenschaften auf die Genauigkeit

Neben den Rechenzeiten für Simulation und Reduktion dürfen natürlich auch die Auswirkungen der Reduktionsverfahren auf die Simulationsgenauigkeit nicht vernachlässigt werden. Im folgenden werden die Ergebnisse verschiedener Experimente in Tabellenform angegeben. Es wurden in allen Fällen jeweils 50 Replikationen durchgeführt. Um eine Reproduktion zu ermöglichen, ist bei jedem Experiment jeweils unter der Tabelle neben den eigentlichen Grafeigenschaften die Anzahl der durchgeführten Zustandswechsel, beschränkt auf relevante Zustände, mit angegeben. 'CARE=2000' bedeutet etwa, dass bei jeder Replikation im Durchschnitt 2000 relevante Zustände besucht wurden. Es sind die Aufenthaltshäufigkeiten für jeweils zehn relevante Zustände des jeweiligen Systems aufgelistet, und zwar einmal vor der Reduktion (links) und einmal danach (rechts). Außerdem wurde für jede Häufigkeit ein Schätzwert für dessen Standardabweichung berechnet und damit ein 90% -Konfidenzintervall bestimmt. Die somit erlaubte Abweichung ist jeweils neben der Häufigkeit angegeben. Die Berechnung von Aufenthaltshäufigkeit und Standardabweichung erfolgte dabei gemäß den Formeln 3.1 und 3.2. Nähere Informationen zum Thema Konfidenzintervalle findet man in [BCN96].

5.3.1 Grafgröße

Als erstes soll der Einfluss der Grafgröße untersucht werden. Dabei werden Konnektivität, Stützstellenzahl und der Anteil der relevanten Zustände konstant gelassen und nur die Anzahl der Zustände variiert.
Man kann nun für jedes Experiment bestimmen, für wieviele Zustände die Aufenthaltshäufigkeit nach der Reduktion noch im entsprechenden Konfidenzintervall des unreduzierten Systems liegt. Für den ersten Grafen mit 256 Zuständen ist dies für drei Zustände der Fall, beim zweiten Grafen mit 1024 Zuständen sind es fünf, und für den dritten Grafen mit 4096 Zuständen sechs Zustände. Daraus lässt sich keine eindeutige Abhängigkeit der Genauigkeit von der Grafgröße ableiten. Offensichtlich sind 40 Stützstellen jedoch zu wenig, um ein Konfidenzintervall von 90 % erfüllen zu können.

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.039551	0.001934
1	2.213187	0.037216
2	0.017535	0.002220
3	0.389082	0.006854
4	2.895576	0.099401
5	0.106819	0.002995
6	0.046082	0.004431
7	0.070397	0.002483
8	0.468942	0.014149
9	0.011158	0.000931

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.036448	0.001904
1	1.986107	0.033188
2	0.013900	0.001982
3	0.382885	0.008458
4	2.823071	0.086789
5	0.103493	0.002658
6	0.052287	0.005445
7	0.062057	0.002110
8	0.458765	0.013490
9	0.007064	0.000619

Tabelle 5.9: Genauigkeitsmessung für 256 Zustände, 1280 Kanten, % care=30, 40 Stützstellen, CARE=12051

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.137671	0.004835
3	0.006032	0.000898
4	0.055612	0.001887
5	0.039382	0.001553
7	0.027206	0.001580
9	0.051169	0.001893
10	0.004287	0.000347
11	0.045963	0.003167
12	0.018173	0.003241
14	0.005660	0.000484

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.134586	0.003935
3	0.006058	0.000977
4	0.059868	0.001792
5	0.040499	0.001299
7	0.026716	0.001895
9	0.052034	0.001699
10	0.004687	0.000431
11	0.039012	0.002679
12	0.007784	0.002372
14	0.006101	0.000681

Tabelle 5.10: Genauigkeitsmessung für 1024 Zustände, 5120 Kanten, % care=30, 40 Stützstellen, CARE=10139

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.005814	0.000919
1	0.017454	0.001804
2	0.008013	0.000683
3	0.011158	0.000969
4	0.019990	0.002227
6	0.019849	0.001499
8	0.063691	0.003824
9	0.003056	0.000328
10	0.001471	0.000282
11	0.004878	0.000495

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.006263	0.000803
1	0.016282	0.001830
2	0.006642	0.000525
3	0.011665	0.000948
4	0.018195	0.002287
6	0.020878	0.001277
8	0.067634	0.003125
9	0.002365	0.000354
10	0.001245	0.000255
11	0.003596	0.000449

Tabelle 5.11: Genauigkeitsmessung für 4096 Zustände, 20480 Kanten, % care=30, 40 Stützstellen, CARE=8616

5.3.2 Grafkonnektivität

Als nächstes wurde eine eventuelle Beziehung zwischen Genauigkeit der Simulation und der Grafkonnektivität untersucht. Dabei wurden wieder drei Experimente durchgeführt, jeweils mit 40 Stützstellen, 1024 Zuständen, einem Anteil an relevanten Zuständen von 30 % und mit verschiedenen Kantenanzahlen.

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.016541	0.003507
3	0.003402	0.000832
6	0.142350	0.008266
7	0.052784	0.013062
9	0.307816	0.019576
12	0.008770	0.001481
15	0.000615	0.000243
16	0.004728	0.000874
17	0.000136	0.000229
18	0.151671	0.004223

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.013829	0.003731
3	0.000019	0.000032
6	0.015466	0.012180
7	0.314091	0.023485
9	0.008029	0.001408
12	0.000519	0.000193
15	0.156410	0.004568
16	0.000354	0.000594
17	0.000000	0.000000
18	0.003686	0.000490

Tabelle 5.12: Genauigkeitsmessung für 1024 Zustände, 2048 Kanten, % care=30, 40 Stützstellen, CARE=2166

Für das erste Experiment ergibt sich für den Zustand mit Index 0 zwar ein Wert innerhalb

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.025183	0.001160
1	0.011844	0.000704
2	0.445050	0.009408
3	0.452081	0.030133
4	0.011098	0.000569
5	0.001082	0.000323
6	0.021127	0.001234
7	0.017167	0.000988
8	0.000675	0.000144
9	0.200458	0.009076

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.023911	0.001250
1	0.010892	0.000638
2	0.455704	0.009501
3	0.447490	0.026233
4	0.010219	0.000545
5	0.001021	0.000358
6	0.023305	0.001250
7	0.016322	0.000904
8	0.000549	0.000126
9	0.197300	0.008654

Tabelle 5.13: Genauigkeitsmessung für 1024 Zustände, 5120 Kanten, % care=30, 40 Stützstellen, CARE=12510

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.011514	0.000915
1	0.018248	0.000902
2	0.044926	0.001073
3	0.075133	0.002056
4	0.016054	0.000766
5	0.025798	0.000936
6	0.112818	0.002576
7	0.013321	0.000638
8	0.064817	0.001044
9	0.125253	0.002325

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.010286	0.000805
1	0.015762	0.000662
2	0.041755	0.001268
3	0.066400	0.001933
4	0.009874	0.000575
5	0.025502	0.000877
6	0.095437	0.001956
7	0.010458	0.000527
8	0.061426	0.001254
9	0.120623	0.001977

Tabelle 5.14: Genauigkeitsmessung für 1024 Zustände, 10240 Kanten, % care=30, 40 Stützstellen, CARE=44495

des entsprechenden Intervalls, jedoch zeigen alle anderen Zustände nach der Reduktion sehr starke Abweichungen von den Werten davor. Dies lässt sich dadurch erklären, dass bei einer Konnektivität von zwei die mittlere Verweildauer in einem Zustand länger ist als bei höheren Verzweigungsgraden. Das führt dazu, dass bei gleichbleibender Simulationszeit ein bestimmter Zustand seltener besucht wird. Beim zweiten Experiment mit einer Konnektivität von fünf befinden sich dagegen fünf von zehn Werten innerhalb des gegebenen Intervalls. Bei einem Verzweigungsgrad von zehn, im dritten Experiment, liegt wieder nur

ein Zustand innerhalb des Konfidenzintervalls, die Genauigkeit geht also für höhere Konnektivitätsgrade wieder leicht zurück. Die Abweichungen fallen jedoch nicht so stark aus wie im ersten Experiment.

5.3.3 Stützstellenzahl

Es ist zu erwarten, dass die Genauigkeit sowohl vor als auch nach der Reduktion direkt von der Anzahl der Stützstellen abhängt, da diese ja für die Genauigkeit der den Kanten zugeordneten Dichten abhängt. Diese Annahme wird bestätigt durch die Tabellen 5.15 bis 5.17:

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.024861	0.001391
1	0.011970	0.000728
2	0.461660	0.009221
3	0.419961	0.027112
4	0.012443	0.000525
6	0.021977	0.001352
7	0.018818	0.001083
8	0.000735	0.000153
9	0.207183	0.008385
10	0.023549	0.001363

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.028238	0.001553
1	0.009227	0.000606
2	0.490771	0.009530
3	0.445634	0.031668
4	0.011727	0.000669
6	0.023203	0.001350
7	0.017392	0.000886
8	0.000428	0.000108
9	0.226536	0.009228
10	0.016744	0.001187

Tabelle 5.15: Genauigkeitsmessung für 1024 Zustände, 5120 Kanten, % care=30, 20 Stützstellen, CARE=11848

Während für 20 Stützstellen nur zwei der zehn Zustände nach der Reduktion eine Aufenthaltshäufigkeit innerhalb des gegebenen Konfidenzintervalls aufweisen, sind es für 40 Stützstellen bereits acht, und für 80 Stützstellen sämtliche der betrachteten Zustände.

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.025584	0.001340
1	0.011679	0.000702
2	0.450730	0.008683
3	0.429354	0.030341
4	0.010666	0.000493
5	0.001155	0.000269
6	0.022889	0.001388
7	0.017496	0.000861
8	0.000639	0.000124
9	0.199629	0.007144

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.024927	0.001296
1	0.011525	0.000662
2	0.460063	0.008549
3	0.452590	0.031641
4	0.009818	0.000664
5	0.000912	0.000316
6	0.023832	0.001234
7	0.017016	0.000953
8	0.000590	0.000128
9	0.198349	0.007854

Tabelle 5.16: Genauigkeitsmessung für 1024 Zustände, 5120 Kanten, % care=30, 40 Stützstellen, CARE=12203

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.024388	0.001369
1	0.011302	0.000655
2	0.442574	0.007585
3	0.392759	0.024399
4	0.009654	0.000489
5	0.001362	0.000407
6	0.022714	0.001050
7	0.016135	0.000794
9	0.186258	0.007958
10	0.023930	0.001323

Zustand	$\hat{\theta}$	$\hat{\sigma}(\hat{\theta}) \cdot t_{0.05,49}$
0	0.023814	0.001149
1	0.011870	0.000493
2	0.437095	0.008387
3	0.410410	0.033469
4	0.009830	0.000607
5	0.001194	0.000357
6	0.022848	0.001151
7	0.016272	0.000994
9	0.183668	0.007744
10	0.023926	0.001534

Tabelle 5.17: Genauigkeitsmessung für 1024 Zustände, 5120 Kanten, % care=30, 80 Stützstellen, CARE=12203

5.4 Bewertung der Ergebnisse

Aus den Experimenten der Abschnitte 5.2 und 5.3 können verschiedene Schlüsse gezogen werden: Zunächst konnten bereits vor den Experimenten angestellte Vermutungen bestätigt werden. Dazu gehört zum Beispiel die Erkenntnis, dass der Reduktionsaufwand in etwa proportional zur Grafgröße ist. Auch erhält man ein höheres Einsparpotenzial, sowohl für höhere Grafkonnektivitäten als auch für geringere Anteile an relevanten Zuständen.

Jedoch gibt es auch einige Resultate, die sich erst aus den Experimenten ergeben haben: Zunächst konnte festgestellt werden, dass das Einsparpotenzial mit größer werdenden Grafen sinkt, ganz abgesehen vom höheren Reduktionsaufwand. Ebenfalls neu ist die Erkenntnis, dass bei höher werdendem Anteil an relevanten Zuständen der Rechenaufwand für die Reduktion schneller sinkt als das Einsparpotenzial, eine Reduktion eines Systems mit vielen relevanten Zuständen zahlt sich also früher aus als die eines Systems mit wenigen solchen Zuständen. Ein weiteres Ergebnis ist, dass bei höher werdendem Konnektivitätsgrad neben dem bereits erwähnten Anstieg des Reduktionspotenzials der Aufwand für die Reduktion langsamer ansteigt als die Rechenzeit für die Simulation des Originalsystems.
Die eindeutigste Folgerung betrifft die Genauigkeit der Darstellung der diskreten Dichten. Aus den Versuchen in Abschnitt 5.3.3 lässt sich ableiten, dass, um ein Konfidenzintervall von 90 % zu erfüllen, mindestens 80 Stützstellen für die interne Repräsentation der Wahrscheinlichkeitsdichten verwendet werden müssen, da schon bei dem Experiment mit 40 Stützstellen zwei von zehn Werten bei der Simulation des reduzierten Systems außerhalb des entsprechenden Konfidenzintervalls liegen.
Die Fragestellung, ob sich eine Reduktion denn nun lohnt, kann also nur in Abhängigkeit von den Grafparametern gelöst werden. Die Antwort hängt hauptsächlich von zwei Faktoren ab, nämlich von der Grafkonnektivität k, also der mittleren Anzahl von Kanten pro Zustand, und vom Anteil der relevanten Zustände *care*. Der Effekt dieser beiden Größen wurde in der zweiten Versuchsreihe in Abschnitt 5.2.2 untersucht. Wurden der Konnektivitätsgrad hoch ($k = 20$) und der Anteil der relevanten Zustände niedrig ($care = 5\%$) gewählt, ergab sich eine Beschleunigung der Simulation um den Faktor zehn, wogegen sich bei $k = 2$ und $care = 80\%$ lediglich eine Zeitersparnis von 8.5% (also ein Beschleunigungsfaktor von 1.093) einstellte. Mit Hilfe dieser beiden Parameter lässt sich also für einen Grafen die Frage nach der Rentabilität der Vereinfachung beantworten.

Kapitel 6

Ausblick

Das wohl dringendste Problem, das noch nicht gelöst werden konnte, stellt die Elimination einer Loop-kante dar, wie sie in Abschnitt 4.2.3 beschrieben wurde. Wäre man in der Lage, auch die gezeigt Reduktion mit vertretbarem Rechenaufwand ausführen zu können, könnte man jeden Grafen, der in der geforderten Form gegeben ist, auf den jeweiligen Minimalgrafen reduzieren, da es dann keine Einschränkungen mehr bezüglich der überspringbaren Zustände mehr gibt. Für diesen Minimalgrafen würde dann gelten, dass ein unrelevanter Zustand sowohl als Vorgänger als auch als Nachfolger ausschließlich relevante Zustände hat. Dass dieser Graf nicht mehr weiter reduziert werden kann, ohne die Aufenthaltswahrscheinlichkeit in einem relevanten Zustand zu verfälschen, ist anschaulich klar, da es kein Paar aus unrelevanten Zuständen mehr gibt, das direkt mittels einer Kante verbunden ist.
Die Repräsentation der Dichten beziehungsweise Verteilungsfunktionen mittels äquidistanter Stützstellen ermöglicht zwar einen effizienten Faltungsalgorithmus, ist aber für die Simulation selbst nicht die optimale Lösung, insbesondere was die Genauigkeit angeht. Eine Dichte ließe sich mit der gleichen Anzahl von Stützstellen genauer darstellen, wenn deren Abstände von einander in Bereichen, in denen sich die Dichte stark ändert, geringer wären, und dafür für nahezu konstante Abschnitte der Dichte weniger Stützstellen verwendet würden. Dies würde den Approximationsfehler bei der Diskretisierung stark reduzieren. In den Ergebnissen von Faltung und Min-Termbildung würde eine intelligente Verteilung der Stützstellen ebenfalls eine höhere Genauigkeit ermöglichen.
Auch über die Reihenfolge, in der die Zustände des Grafen bei der Reduktion bearbeitet werden, kann man Überlegungen anstellen. Wie in Abschnitt 4.3.2 gezeigt, verbraucht die Faltung von Dichten mit stark unterschiedlichen Intervallbreiten verhältnismäßig viel Rechenzeit. Man könnte also zu Anfang bevorzugt Kanten mit ähnlicher oder gleicher Intervallbreite verarbeiten. Auch die Verwendung eines Generationsindexes für jede Kante

wäre denkbar. Hintergrund für diesen Ansatz war der Versuch, einen Ringgrafen zu reduzieren. Im ungünstigsten Fall faltet man die Kanten der Reihe nach, es wäre jedoch viel effizienter, benachbarte Kanten jeweils paarweise zusammenzufassen, und im Endeffekt nur Dichten zu falten, deren Kanten einen gleichen oder ähnlichen Generationsindex besitzen. Die Eliminationsreihenfolge spielt nicht nur für den Rechenaufwand eine Rolle. Es kann mitunter der Fall sein, dass verschiedene Eliminationsreihenfolgen zu verschiedenen Grafstrukturen führen. Betrachte man zu diesem Zweck den Grafen nach Abbildung 6.1, in dem nur der Zustand Z_3 als relevant markiert sein soll:

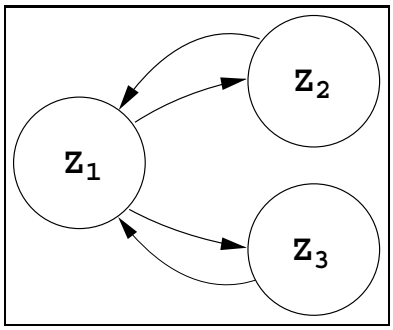


Abbildung 6.1: Ausgangsgraf

Beginnt man zur Reduktion damit, die Kante von Z_1 nach Z_2 zu entfernen, muss man die zugehörige Dichte mit der von der Kante von Z_2 nach Z_1 falten, und es entsteht eine Schleife bei Z_1 (siehe Abbildung 6.2). Z_2 ist ab sofort nicht mehr erreichbar und kann weggelassen werden.

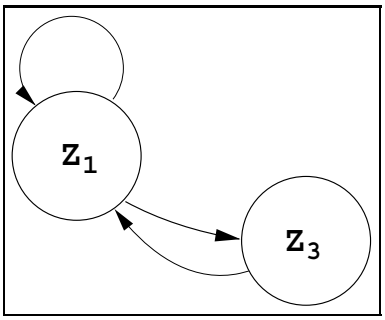


Abbildung 6.2: Reduktionsergebnis 1

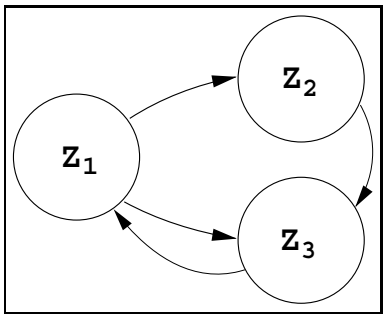


Abbildung 6.3: Reduktionsergebnis 2

Entfernt man dagegen als erstes die Kante von Z_2 nach Z_1, muss die Dichte gefaltet werden mit der Dichte der Kante von Z_1 nach Z_3, und der Graf nach Abbildung 6.3 entsteht. Die eine Methode führt also zu einem Loop, wogegen das Ergebnis der anderen Vorgehensweise einen Zustand mehr enthält als die erste. Je nach weiteren Gegebenheiten kann entweder

das eine oder das andere Ergebnis von Vorteil sein.
Ein völlig anderer Ansatz, den man verfolgen könnte, wäre eine Erweiterung der Definition des in Abschnitt 2.1 beschriebenen Zustandsraums und wenn möglich die dadurch notwendige Anpassung der Reduktionsschritte. Speziell vor dem Hintergrund der zum Beispiel in [MBC+95] beschriebenen *Petrinetze* wäre eine genauere Analyse der Übertragbarkeit von Reduktionstechniken auf andere Netztypen sicherlich lohnenswert. Da Petrinetze wesentlich komplexer und damit auch mächtiger in ihrer Systembeschreibungsfähigkeit sind, steht zu erwarten, dass die Anwendung noch zu findender Techniken, die ein solches Netz vereinfachen können, eine wesentliche Reduktion der für eine Simulation aufzubringende Rechenleistung zur Folge haben wird.

Literaturverzeichnis

[BCN96] Jerry Banks, John S. Carson, and Barry L. Nelson. *Discrete-Event System Simulation.* Prentice Hall Inc., 2nd edition, 1996.

[Fel68] William Feller. *An Introduction to Probability Theory and its Applications.* John Wiley & Sons, 1968.

[LK00] Averill M. Law and W. David Kelton. *Simulation Modeling and Analysis.* Mc Graw Hill, 3rd edition, 2000.

[Mat90] Rudolf Mathar. *Stochastik für Informatiker.* Teubner, Stuttgart, 1990.

[MBC+95] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets.* John Wiley & Sons, 1995.