Tupel von TVL als Datenstruktur für Boolesche Funktionen

Von der Fakultät für Mathematik und Informatik

der Technischen Universität Bergakademie Freiberg

genehmigte

DISSERTATION

zur Erlangung des akademischen Grades

Doktor-Ingenieur

(Dr.-Ing.)

vorgelegt

von Diplom-Geophysiker Galina Kempe

geboren am 07.06.1962 in Glasuchino

Gutachter: Prof. Dr.-Ing. habil. Bernd Steinbach, Freiberg

Prof. Dr.-Ing. habil. Jochen Beister, Kaiserslautern

Prof. Dr.-Ing. habil. Michael Gössel, Potsdam

Tag der Verleihung: 20.06.2003

Zusammenfassung

In der vorliegenden Arbeit wird eine Datenstruktur zur Darstellung einer Booleschen Funktion "TVL-Tupel" präsentiert, die im Ergebnis einer Kombination der bekannten Datenstrukturen Entscheidungsgraph und Ternärvektorliste entsteht. Zuerst wird untersucht, wie lokale Phasenlisten sich als Elemente des Tupels eignen. Weiterhin wird die neue Dekompositionsart ("Tupel-Dekomposition") einer Boolesche Funktion in drei bzw. vier Teilfunktionen vorgestellt. Die Besonderheit der Teilfunktionen der Dekomposition besteht in ihrer Orthogonalität zueinander. Der Vorteil der Dekomposition von Funktionen mit einer hohen Anzahl von Konjunktionen besteht im geringeren Speicherplatzbedarf. Des weiteren wurden Algorithmen für Realisierung der Operation entwickelt, die für eine Handhabung der zerlegten Funktionen erforderlich sind. Der detaillierte Vergleich der Berechnungszeiten für die Operationen erbringt den Nachweis, dass eine Verringerung des Zeitbedarfs als Folge der Zerlegung zu erwarten ist. Weiterhin bietet die Dekomposition einen Ansatz für den Entwurf von Algorithmen, die eine parallele Bearbeitung auf der Grundlage verteilter Rechentechnik zulassen. Die Erkenntnisse der Untersuchungen der Tupel-Dekomposition einschließlich der Verwendung der verteilen Verarbeitung können beispielsweise für die Suche der Variablenmengen der OR-Bi-Decomposition verwendet werden.

Inhaltsverzeichnis

Symbole und Abkürzungen

Symbole

$\underline{x}$	Vektor Boolescher Variablen
B^k	Boolescher Raum
$f(\underline{x})$	Boolesche Funktion
$\underline{h}(\underline{x})$	Vektor bzw. Tupel aus Booleschen Funktionen
$\overline{Expr}$	Negation des Ausdrucks $Expr$
$\wedge$	konjunktive Verknüpfung
$\cdot$	konjunktive Verknüpfung
$\vee$	disjunktive Verknüpfung
$\oplus$	antivalente Verknüpfung
$\odot$	äquivalente Verknüpfung
$\equiv$	identisch gleich
$L1(f)$	Lösungsmenge einer homogenen charakteristischen Booleschen Gleichung
$\frac{\partial f(\underline{x})}{\partial x_i}$	partielle Ableitung einer Booleschen Funktion $f(\underline{x})$ nach der Variablen x_i
$\frac{\partial^k f(\underline{x})}{\partial x_{i_1} \cdots \partial x_{i_k}}$	k-fache Ableitung einer Booleschen Funktion $f(\underline{x})$ nach den Variablen $x_{i_1} \cdots x_{i_k}$
$\min_{x_i} f(\underline{x})$	partielles Minimum einer Booleschen Funktion $f(\underline{x})$ nach der Variablen x_i
$\max_{x_i} f(\underline{x})$	partielles Maximum einer Booleschen Funktion $f(\underline{x})$ nach der Variablen x_i
$\min^k_{\underline{x_i}} f(\underline{x})$	k-faches Minimum einer Booleschen Funktion $f(\underline{x})$ nach den Variablen $\underline{x_i}$
$\max^k_{\underline{x_i}} f(\underline{x})$	k-faches Maximum einer Booleschen Funktion $f(\underline{x})$ nach den Variablen $\underline{x_i}$
f_{S0}	Cofaktor 0 der Shannon-Dekomposition
f_{S1}	Cofaktor 1 der Shannon-Dekomposition
f_D	Funktion der Davio-Dekomposition
f^-	Strichfunktion der Tupel-Dekomposition
f^0	Nullfunktion der Tupel-Dekomposition
f^1	Einsfunktion der Tupel-Dekomposition
$f^=$	Doppelstrichfunktion der Tupel-Dekomposition
$O_{wc}(Expr)$	Komplexität eines Algorithmus in Abhängigkeit vom Ausdruck $Expr$ in worst case
$O_{ac}(Expr)$	Komplexität eines Algorithmus in Abhängigkeit vom Ausdruck $Expr$ in average case
f_{wc}	Berechnungsaufwand einer Booleschen Operation in worst case
f_{ac}	Berechnungsaufwand einer Booleschen Operation in average case

N	Anzahl der Ternärvektoren einer TVL
n	Anzahl der Variablen einer Booleschen Funktion
m	Anzahl der rekursiven Zerlegungen einer Booleschen Funktion
c	Konstante
$\times$	Multiplikation
$\cup$	Vereinigung
$-$	Differenz
$\setminus$	Differenz

Abkürzungen

BDD	Binary Decision Diagram
BSD	Berkeley System Distribution
OBDD	Ordered Binary Decision Diagram
FDD	Functional Decision Diagram
OFDD	Ordered Functional Decision Diagram
OKDD	Ordered Kronecker Decision Diagram
TDD	Ternary Decision Diagram
ETDD	kanonische Form des EXOR Ternary Decision Diagram
EXOR-TDD	EXOR Ternary Decision Diagram
AND-OR-TDD	AND-OR Ternary Decision Diagram
TV	Ternärvektor
TVL	Ternärvektorliste
OTVL	geordnete Ternärvektorliste
PHL	Phasenliste
DF	disjunktive Form
KF	konjunktive Form
AF	Antivalenzform
EF	Äquivalenzform
ODA	orthogonale AF bzw. DF
OKE	orthogonale EF bzw. KF
ER	Speicherung in einem Booleschen Raum
VR	Speicherung in verschiedenen Booleschen Räumen
DERK	XBOOLE-Operation k-fache Ableitung
ISC	XBOOLE-Operation Konjunktion
DIF	XBOOLE-Operation Differenz
NEG	XBOOLE-Operation Negation

Kapitel 1

Einleitung

Der verstärkte Einsatz elektronischer Geräte und der Computertechnik in allen Bereichen des Lebens, der Wirtschaft und Wissenschaft innerhalb der letzten Jahrzehnte hat zu einem ständigen Ansteigen der Bedeutung der Mikroelektronik geführt. Bereits im Jahre 1965, gerade vier Jahre nach der Entwicklung der ersten integrierten Schaltung, beobachtete Gordon E. Moore eine Tendenz, die als "Moore-Gesetz" bekannt wurde. In einem Artikel [Moo65] sagte Moore voraus, dass sich die Zahl der Transistoren pro integrierte Schaltung bis zum Jahre 1975 alle 18 Monate verdoppeln wird. Diese Aussage hat bis zum gegenwärtigen Zeitpunkt seine Bedeutung nicht verloren, denn auch in der heutigen Zeit verdoppelt sich die Leistungsfähigkeit der Computertechnik in vergleichbaren Zeiträumen. Einerseits ist diese Tatsache mit der Entwicklung von verbesserten Technologien zu erklären. Andererseits wurden die Fortschritte auf dem Gebiet der Mikroelektronik durch neue Erfolge des rechnergestützten Schaltungsentwurfs möglich, der sich von der Analyse und Synthese bis zur Testbarkeit erstreckt.

Die Probleme des Entwurfs digitaler elektronischer Schaltungen können durch eine Folge von Operationen mit Booleschen Funktionen formuliert werden, wobei in der Regel nicht die einzelnen Funktionen sondern die Funktionsverbände zu betrachten sind. Die Darstellung der Funktionsverbände kann wiederum auf die Darstellung einzelner Funktionen zurückgeführt werden (siehe Abschnitt 2.1.3). Die praktische Verwendung von Booleschen Funktionen und Gleichungen führt zu hohen Anforderungen an die Datenstrukturen zu ihrer Darstellung und an die Algorithmen zu ihrer Handhabung.

Der Aufwand für die Lösung einer Booleschen Gleichung wird entscheidend von der Anzahl in ihr vorkommender unterschiedlicher Boolescher Variablen bestimmt. Da eine Boolesche Variable nur Werte aus der Menge $B = \{0, 1\}$ annehmen kann, müssen für die Suche der Lösungsmenge einer Gleichnung $f(\underline{x}) = g(\underline{x})$ mit n verschiedenen Variablen 2^n Belegungsvektoren ausgewertet werden. Der dazu benötigte Algorithmus besitzt eine exponentielle Berechnungskomplexität $O(2^n)$ und auch die Menge der zu speichernden Lösungsvektoren kann im worst case 2^n betragen. Die exponentielle Komplexität ist folglich sowohl in der Darstellung einer Booleschen Funktion als auch in den Algorithmen zu ihrer Handhabung wiederzufinden.

Wie auch auf anderen Gebieten der Informatik besteht ein möglicher Lösungsansatz zur effizienten Darstellung einer Booleschen Funktion in der Anwendung des Prinzips "Teile-und-Herrsche", oder auf eine Boolesche Funktion bezogen in der Dekomposition einer Booleschen Funktion. Allgemein bekannt sind die Shannon- und Davio-Dekompositionen einer Booleschen Funktionen in zwei Funktionen. Rekursiv angewandt bilden diese Dekompo-

sitionsprinzipien die Grundlage für einige als Entscheidungsbäume bekannte Datenstrukturen [SaFu96]. So wurden ausgehend von der Shannon-Dekomposition (1.1) (siehe auch Abschnitt 2.1.4) die Binary Decision Diagrams (BDD) bzw. Ordered Binary Decision Diagrams (OBDD) entwickelt [Ake78, Bry86, BRB90, Bry92].

$$\begin{aligned} f &= \overline{x}\, f_{S0} \vee x f_{S1} \\ &= \overline{x}\, f(x=0) \vee x f(x=1) \end{aligned} \tag{1.1}$$

Das OBDD ist eine kanonische Darstellung einer Booleschen Funktion. Die Kanonizität der Darstellung wird durch eine feste Variablenordnung auf jedem Pfad des OBDD gesichert. Die einheitliche Variablenordnung, die einerseits für die Kanonizität sorgt, schränkt anderseits die Optimierung eines OBDD ein. Wird auf verschiedenen Pfaden eines BDD unterschiedliche Variablenordnung zugelassen, so entstehen Free Binary Decision Diagrams (Free BDD) [BGMS94]. Die Einführung eines $\oplus$-Knotens neben regulären branching Knoten in OBDD hat ein $\oplus$-OBDD [GeMe96, MeSa99, MeSa01.1, MeSa01.2] zur Folge.

Die alternative Verwendung der positiven (1.2) bzw. negativen Davio-Zerlegung (1.3) (siehe auch Abschnitt 2.1.4) anstelle der Shannon-Zerlegung führt zu Functional Decision Diagrams (FDD) bzw. Ordered Functional Decision Diagrams (OFDD) [Mul54, Ree54, KSR92, BDT93]. Es hängt von der konkreten Booleschen Funktion ab, in welcher der beiden Graphendarstellungen ein geringerer Speicheraufwand benötigt wird.

$$\begin{aligned} f &= f_{S0} \oplus x f_D \\ &= f(x=0) \oplus x(f(x=0) \oplus f(x=1)) \end{aligned} \tag{1.2}$$

$$\begin{aligned} f &= f_{S1} \oplus \overline{x}\, f_D \\ &= f(x=1) \oplus \overline{x}\,(f(x=0) \oplus f(x=1)) \end{aligned} \tag{1.3}$$

Eine Kombination OBDD/OFDD stellt das Ordered Kronecker Decision Diagram (OKDD) dar, in dem alle drei in OBDD und OFDD verwendeten Zerlegungen zugelassen werden [DSTBP94, BDT95, DrBe95]. Für jede Boolesche Variable der Funktion wird im einzelnen über die Zerlegungsart entschieden. Da die Variablenordnung und die Art der Zerlegung für alle Pfade eines OKDD gleich ist, handelt es sich beim OKDD ebenfalls um eine kanonische Darstellung. Bei den pseudo Kronecker Decision Diagrams (pseudo KDD) wird die Dekompositionsart nicht für jede einzelne Variable sondern für jeden einzelnen Knoten des Diagramms festgelegt [LDB00, MaSa01].

Durch eine rekursive Anwendung der Dekomposition einer Booleschen Funktion in drei Funktionen entstehen die Ternary Decision Diagrams (TDD) [Sas93, RoBa94, Sas97, SaFu96]. Bekannt ist die kanonische Form der EXOR Ternary Decision Diagrams ein ETDD. Die drei Kanten eines ETDD führen zu den Funktionen f_{S0}, f_{S1} und f_D. Weitere nicht kanonische Formen des EXOR-TDD und AND-OR-TDD entstehen durch Dekomposition einer Booleschen Funktion nach (1.4) und (1.5) (siehe [Sas93, Yas95]), wobei die Funktionen F_0, F_1 und F_2 keine disjunkte Funktionen sind.

$$F = \overline{x}\, F_0 \oplus x F_1 \oplus 1 \cdot F_2 \tag{1.4}$$

$$F = \overline{x}\, F_0 \vee x F_1 \vee 1 \cdot F_2 \tag{1.5}$$

Eine zum Entscheidungsgraph alternative Darstellungsform ist die Ternärvektorliste (TVL). Der zur Darstellung Boolescher Funktionen als Ternärvektorliste bestehende Lösungsansatz ist auszugsweise im Abschitt 2.2 der vorliegenden Arbeit und in [PoSt79.1,

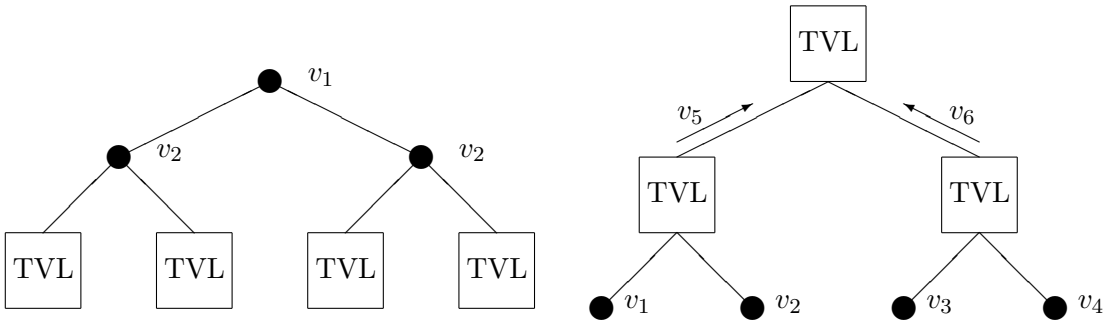


Abbildung 1.1: Mögliche Entscheidungsgraphen

PoSt79.8, Ste84, BoSt91, DKSW92] beschrieben. Zwei Binärvektoren werden zu einem Ternärvektor zusammengefasst, wenn sie sich nur an einer Stelle unterscheiden. Der entstehende Ternärvektor erhält an dieser Stelle ein Strichelement. Analoge Zusammenfassungen sind auch zwischen Ternärvektoren möglich. Ein Ternärvektor mit s Strichelementen beschreibt 2^s Binärvektoren. Damit wird dem exponentiellen Anwachsen der Problemgröße eine ebenfalls exponentielle Abschwächung in der Darstellungskomplexität entgegengesetzt. Eine besondere Form der TVL ist die orthogonale TVL. Alle Ternärvektoren einer orthogonalen TVL sind zueinander disjunkt. Die Weiterentwicklung einer orthogonalen TVL ist die geordnete TVL (OTVL) [StDo99, StDo00, StDo01]. Die Einordnung von Ternärvektoren einer OTVL in Klassen und Subklassen soll das schnellere Vergleichen Boolescher Funktionen ermöglichen.

Alle oben aufgeführte Datenstrukturen lassen sich in zwei Gruppen aufteilen. Eine Gruppe umfasst die Entscheidungsgraphen, bei denen die Komplexität einer Booleschen Funktion durch die Dekomposition der Funktion reduziert wird. Die zweite Gruppe enthält verschiedene Formen der TVL. Die Vorteile der TVL sind in der kompakten Speicherung, die als Ergebnis des Zusammenfassens von mehreren Ternärvektoren gegeben ist, und in der Möglichkeit der Parallelverarbeitung der Vektoren der TVL (siehe [BoSt91] und [DKSW92]) zu finden. Vergleichende Untersuchungen wie z. B. in [KeSt94, Ste95] zeigten auch, dass bei einer Beschränkung der Variablenzahl einer Booleschen Funktion die bezüglich des Speicherplatzbedarfs günstigere Speicherung einer Booleschen Funktion als TVL im Vergleich zu BDD möglich ist. Es ist damit zu erwarten, dass die Kombination der Lösungsansätze der Datenstrukturen beider Gruppen, d. h. die Dekomposition einer Booleschen Funktion in mehrere Teilfunktionen und die Darstellung der Teilfunktionen durch TVL zu einer effizienten Datenstruktur führen kann. Der Untersuchungsgegenstand dieser Arbeit ist die Datenstruktur "TVL-Tupel", die auf der Dekomposition der Booleschen Funktionen und der TVL-Darstellung basiert. Mit der Dekomposition Boolescher Funktionen wird eine Reduzierung der Komplexität der Darstellung und eine vereinfachte Handhabung angestrebt.

Die Dekomposition einer Booleschen Funktion führt zu einem Graph, dessen einzelne Knoten entweder Variablen oder Ternärvektorlisten sind. Dabei werden in der vorliegenden Arbeit betreffs Aussehen des Graphen folgende Möglichkeiten betrachet:

- Die Boolesche Funktion kann rekursiv bis zu einer bestimmten Funktionsgröße der Teilfunktionen zerlegt werden. Die durch TVL dargestellten im Ergebnis der Dekomposition entstandenen Teilfunktionen bilden die Blätter des Graphen (siehe Abbildung 1.1 links).

- Als Blätter des Graphen können die unabhängigen Variablen gespeichert und in die Ternärvektorlisten der übergeordneten Knoten übernommen werden. Die Funktionswerte der durch diese TVL dargestellten Funktionen sind die abhängigen Variablen der jeweiligen TVL und dienen ihrerseits als unbhängige Variablen für die darüberliegenden TVL (siehe Abbildung 1.1 rechts).

Bei der Durchführung der Dekomposition stehen weiterhin die Fragen nach dem Dekompositionsprinzip und der Tiefe der Dekomposition. Die bisher aufgeführten bekannten Dekompositionsarten schließen nicht aus, dass weitere Dekompositionsprinzipien möglich sind. Die Lösung der Frage über die Tiefe der Zerlegung besteht in der Suche des optimalen Verhältnisses zwischen dem Umfang und damit auch der Kompliziertheit des Graphen und der Größe der TVL an den Knoten des Graphen.

Neben der Reduzierung der Komplexität der Darstellung und der vereinfachten Handhabung einer Booleschen Funktion ermöglicht die Dekomposition die Optimierung der TVL-Darstellung durch das Speichern von Elementen des TVL-Tupels in verschiedenen Booleschen Räumen. Die Voraussetzung dazu ist durch das Raumkonzept des XBOOLE-Systems gegeben [BoSt91, DKSW92], das dem Anwender erlaubt die Dimensionen von beliebig vielen Booleschen Räumen selbst festzulegen und damit hochdimensionale Boolesche Probleme zu lösen.

Weiterhin bietet die Dekomposition einen Ansatz für den Entwurf von Algorithmen für Operationen mit Booleschen Funktionen, die eine parallele Bearbeitung auf der Grundlage verteilter Rechentechnik zulassen. Ein möglicher Einsatz der nebenläufigen Verarbeitung ist durch die interne Speicherstruktur einer TVL im XBOOLE-System gegeben [DKSW92] und besteht in der Entwicklung von speziellen Algorithmen zur parallelen Verarbeitung einzelner Ternärvektoren der TVL. Die Verwendung dieser Algorithmen kann unter Einsatz eines speziell dazu entwickelten Rechners [Hess94, Hess98, Hess99] oder mehrerer Rechner [StDob00, SDD01] erfolgen. In dieser Arbeit werden die Möglichkeiten der nebenläufigen Berechnungen nicht zwischen einzelnen Vektoren sondern zwischen Ternärvektorlisten von Elementen des Tupels analysiert.

Die Arbeit geht von der dargestellten Problemstellung aus und ist wie nachfolgend beschrieben aufgebaut.

Zum besseren Verständnis nachfolgender Untersuchungen wird zunächst im Kapitel 2 eine Einführung in Boolesche Funktionen und Gleichungen gegeben. Ein zweiter Themenkomplex dieses Kapitels befasst sich mit grundlegenden Aussagen zur verteilten Verarbeitung parallel realisierbarer Aufgabenstellungen, da diese Möglichkeiten innerhalb einer prototypischen Lösung Anwendung finden.

Im Kapitel 3 wird auf die Beschreibung des Verhaltens Boolescher Systeme mittels Phasenlisten (PHL) eingegangen. Als mögliche Darstellungsformen werden globale PHL und Tupel aus lokalen Phasenlisten unterschieden. Durch die Komposition mehrerer lokaler Phasenlisten zu einer Phasenliste kann die Anzahl der Elemente des Tupels verringert werden. Ein wesentlicher Untersuchungsgegenstand des Kapitels ist die Eignung der genannten Darstellungen als Datenmodell für die Beschreibung einer Booleschen Funktion. Weitere Untersuchungen befassen sich mit der Optimierung der Darstellung durch das Speichern von einzelnen Phasenlisten in verschiedenen Booleschen Räumen und mit der Optimierung der Größe der Phasenlisten-TVL.

Im Kapitel 4 wird die Möglichkeit der Darstellung einer Boolschen Funktion durch Tupel aus drei und vier Funktionen analysiert. Aufbauend auf den im Kapitel 4 eingeführten

Zerlegungsprinzipien werden spezielle Algorithmen für die Operationen Negation, Konjunktion, Disjunktion, Antivalenz und Äquivalenz entwickelt. Einen Schwerpunkt des Kapitels bildet die Beweisführung für die Korrektheit der Zerlegungsprinzipien und der erstellten Operationsalgorithmen, wobei die gewonnenen Erkenntnisse mit Beispielen unterlegt werden. Daneben werden in diesem Kapitel der für die Darstellung der Funktionen benötigte Speicherplatz, die Berechnungszeiten und die Komplexitätsordnung der Operationen analysiert.

Die Möglichkeit des Einsatzs der parallelen Bearbeitung bei der Durchführung von Booleschen Operationen werden im Kapitel 5 analysiert und innerhalb eines Prototyps umgesetzt. Den Ausgangspunkt für die verteilte Verarbeitung bildet die Tatsache, dass bei der Darstellung einer Booleschen Funktion durch ein Tupel aus einzelnen Funktionen mehrere Verknüpfungen auf die Elemente des Tupels angewandt werden, die sich unabhängig voneinander ausführen lassen.

Die im Kapitel 4 und 5 erzielten Erkenntnisse finden im 6 Kapitel für eine praxisnahe Aufgabestellung Anwendung, die die Suche der Variablenmengen der OR-Bi-Decomposition zum Gegenstand hat. An das zusammenfassende Kapitel 7 schliessen sich die Bibliographie und einige Anhänge an, die mit den weiterführenden Informationen den Hauptteil der Arbeit vertiefen.

Kapitel 2

Grundlagen

2.1 Boolesche Funktion

2.1.1 Boolesche Funktionen und Gleichungen

Ein **Binärvektor** der Länge k ist ein k-Tupel, dessen Elemente nur die Werte 0 oder 1 annehmen können (2.1). Die Menge aller möglichen Binärvektoren der Länge k nennt man den **Booleschen Raum** B^k (2.2).

$$\underline{x} = (x_1, x_2, ..., x_k) \quad \text{mit} \quad x_i \in \{0, 1\} \tag{2.1}$$

$$B^k = \{\underline{x} | \underline{x} = (x_1, x_2, ..., x_k) \quad \text{mit} \quad x_i \in \{0, 1\}\} \tag{2.2}$$

Eine **binäre Funktion** $y = f(x_1, x_2, ..., x_k) = f(\underline{x})$ ist eine Abbildungsvorschrift $B^k \rightarrow B$, die jedem Binärvektor $\underline{x} = (x_1, x_2, ..., x_k)$ einen binären Wert y eindeutig zuordnet.

Die binäre Funktion kann durch einen Booleschen Ausdruck beschrieben werden. Binäre Ausdrücke sind definiert durch:

1. Die Konstanten 0 und 1 und die Variablen $x_1, x_2, ..., x_k$ sind binäre Ausdrücke.

2. Durch beliebig häufige Anwendung von Verknüpfungsoperationen auf binäre Ausdrücke entstehen weitere binäre Ausdrücke. Als Verknüpfungsoperationen können die in der Tabelle 2.1 zusammengefassten Grundoperation verwendet werden.

   Für diese Operationen gelten die Gesetze der Idempotenz, Verknüpfung mit Null und Eins, Verknüpfung mit dem Komplement, die Gesetze der Kommutativität, Assoziativität und Distributivität und die Sätze von de Morgan und von Stone, die in der Literatur ausführlich beschrieben sind [BoSt91, PoBo86].

Tabelle 2.1: Binäre Grundoperationen

Operation	Schreibweise
Negation	$\overline{x_1}$
Konjunktion	$x_1 \cdot x_2$
Disjunktion	$x_1 \vee x_2$
Antivalenz	$x_1 \oplus x_2$
Äquivalenz	$x_1 \odot x_2$

Eine besondere Bedeutung in der Booleschen Algebra haben die folgenden vier Ausdrucksformen der Booleschen Funktion :

- **Disjunktive Form (DF)** - ist eine Disjunktion einzelner Konjunktionen,
- **Antivalenzform (AF)** - ist eine Antivalenz einzelner Konjunktionen,
- **Konjunktive Form (KF)** - ist eine Konjunktion einzelner Disjunktionen,
- **Äquivalenzform (EF)** - ist eine Äquivalenz einzelner Disjunktionen.

Zum Beispiel sind die Funktionen $f_1(\underline{x})$, $f_2(\underline{x})$, $f_3(\underline{x})$ und $f_4(\underline{x})$ (2.3) in den Formen DF, AF, KF bzw. EF dargestellt.

$$\begin{array}{l} f_1(\underline{x}) = \overline{x}_1 x_2 \vee \overline{x}_3 \\ f_2(\underline{x}) = \overline{x}_1 x_2 \oplus \overline{x}_3 \\ f_3(\underline{x}) = (\overline{x}_1 \vee x_2) \cdot \overline{x}_3 \\ f_4(\underline{x}) = (\overline{x}_1 \vee x_2) \odot \overline{x}_3 \end{array} \tag{2.3}$$

Für die Funktionen in DF, AF, KF und EF gelten folgende **Orthogonalitätssätze**.

- Wenn für die Konjunktionen einer Funktion in der DF oder AF die Bedingung

$$K_i \cdot K_j \equiv 0 \quad , \quad \forall i \neq j \tag{2.4}$$

gilt, dann folgt

$$\bigvee_i K_i = \bigoplus_i K_i$$

- Wenn für die Disjunktionen einer Funktion in der KF oder EF die Bedingung

$$D_i \vee D_j \equiv 1 \quad , \quad \forall i \neq j \tag{2.5}$$

gilt, dann folgt

$$\bigwedge_i D_i = \bigodot_i D_i$$

Wurden die Bedingungen (2.4) oder (2.5) erfüllt, sind die einzelnen Konjunktionen der Funktionen in DF oder AF (oder die einzelnen Disjunktionen der Funktionen in KF oder EF) **orthogonal** oder **disjunkt** zueinander.

Eine **Boolesche Gleichung** entsteht durch das Gleichsetzen zweier Boolescher Funktionen. Es sind drei Formen der Boolescher Gleichungen zu unterscheiden (2.6) - (2.8).

$$f(x_1, \cdots, x_k) = g(x_1, \cdots, x_k) \tag{2.6}$$

$$f(x_1, \cdots, x_k) = 0 \tag{2.7}$$

$$f(x_1, \cdots, x_k) = 1 \tag{2.8}$$

Gleichungen, die auf einer Seite des Gleichheitszeichens nur einen konstanten Ausdruck enthalten, werden als homogene Gleichungen bezeichnet. Die Gleichung 2.7 ist eine restriktive homogene Boolesche Gleichung, und die Gleichung 2.8 eine charakteristische homogene Boolesche Gleichung.

Tabelle 2.2: Boolesche Operationen und Mengenoperationen

Boolesche Operation	Operation für Lösungsmenge der charakteristischen Gleichung
$f(\underline{x}) = \overline{h(\underline{x})}$	$L1(f)$ ist Komplement von $L1(h)$
$f(\underline{x}) = h(\underline{x}) \cdot g(\underline{x})$	$L1(f)$ ist Durchschnit von $L1(h)$ und $L1(g)$
$f(\underline{x}) = h(\underline{x}) \vee g(\underline{x})$	$L1(f)$ ist Vereinigung von $L1(h)$ und $L1(g)$
$f(\underline{x}) = h(\underline{x}) \oplus g(\underline{x})$	$L1(f)$ ist symmetrische Differenz von $L1(h)$ und $L1(g)$
$f(\underline{x}) = h(\underline{x}) \odot g(\underline{x})$	$L1(f)$ ist Komplement der symmetrischen Differenz von $L1(h)$ und $L1(g)$

Die Lösungsmenge einer Booleschen Gleichung bilden alle Binärvektoren

$$\underline{x} = (x_1, \cdots, x_k),$$

die die Gleichung in die Identität $1 \equiv 1$ bzw. $0 \equiv 0$ überführen. Im Fall einer homogenen charakteristischen Gleichung (2.8) ist die Lösungsmenge $L1(f)$ die Menge aller 1-Belegungen der Funktion $f(x_1, \cdots, x_k)$. Ist die Funktion $f(x_1, \cdots, x_k)$ in einer zusammengesetzten Ausdrucksform gegeben, so kann die Lösungsmenge $L1(f)$ wie in der Tabelle 2.2 beschrieben mit Hilfe der Mengenoperationen bestimmt werden.

2.1.2 Ableitungsoperationen Boolescher Funktionen

Außer den im Abschnitt 2.1.1 genannten Grundoperationen (siehe Tabelle 2.2) haben die Ableitungen der Booleschen Funktionen eine große Bedeutung. Im Mittelpunkt dieser Operationen steht die Änderung der binären Größen. Einer Booleschen Funktion wird durch die Änderung einzelner Variablen (oder Variablenvektoren) eine gleichartige Funktion zugeordnet. Die antivalente, disjunktive oder konjunktive Verknüpfungen dieser Funktionen bilden die Grundlage für die Ableitungsoperationen.

Z. B. kann der Funktion $f(x_1, x_2, x_3) = x_1x_3 \oplus x_2\overline{x_3}$ die Funktion $f(\overline{x_1}, x_2, x_3) = \overline{x_1}\, x_3 \oplus x_2\overline{x_3}$ zugeordnet werden, indem die Variable x_1 durch $\overline{x_1}$ ersetzt wird. Die Funktionswerte verändern sich dabei so, wie im Karnaugh-Plan (Abbildung 2.1) ersichtlich ist. Verknüpft man die Funktionen $f(x_1, x_2, x_3)$ und $f(\overline{x_1}, x_2, x_3)$ antivalent, so entsteht eine Funktion mit dem Wert 1, wenn die Änderung von x_1 die Änderung der Funktion $f(x_1, x_2, x_3)$ bewirkt, und dem Wert 0, wenn die Funktion $f(x_1, x_2, x_3)$ unverändert bleibt (siehe Abbildung 2.1). Es ist ersichtlich, dass keine Abhängigkeit der Funktion $f(x_1, x_2, x_3) \oplus f(\overline{x_1}, x_2, x_3)$ von der Variable x_1 besteht.

Allgemein ist die **partielle Ableitung** einer Booleschen Funktion $f(\underline{x})$ nach der Variable x_i nach (2.9) definiert und ist damit auch eine Boolesche Funktion.

$$\frac{\partial f(\underline{x})}{\partial x_i} = f(x_i, \underline{x_0}) \oplus f(\overline{x_i}, \underline{x_0}) \tag{2.9}$$

Die Ableitung der Booleschen Funktion hat folgende wichtige Eigenschaften:

1. $\frac{\partial f(\underline{x})}{\partial x_i} = f(x_i = 0, \underline{x_0}) \oplus f(x_i = 1, \underline{x_0})$;
2. die Ableitung einer Booleschen Funktion nach der Variable x_i ist von x_i unabhängig;

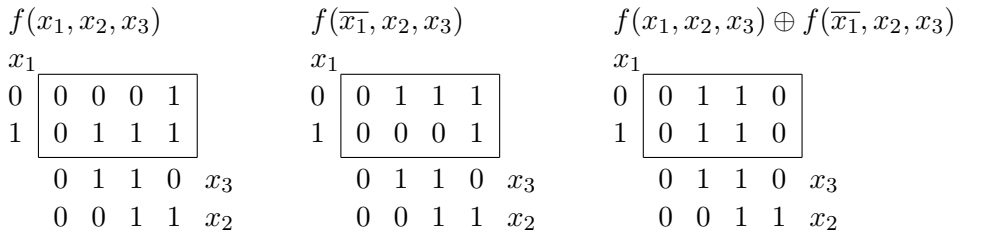


Abbildung 2.1: Funktionen $f(x_1, x_2, x_3)$, $f(\overline{x_1}, x_2, x_3)$ und $f(x_1, x_2, x_3) \oplus f(\overline{x_1}, x_2, x_3)$

3. $\frac{\partial f(\underline{x})}{\partial x_i} = 0$, wenn die Funktion $f(\underline{x})$ von der Variable x_i unabhängig ist, d. h. wenn die Änderung der Variable x_i keine Veränderung des Funktionswertes der Funktion $f(\underline{x})$ hervorruft,
$\frac{\partial f(\underline{x})}{\partial x_i} \neq 0$, wenn die Änderung der Variable x_i eine Änderung des Funktionswertes der Funktion $f(\underline{x})$ hervorruft,
und $\frac{\partial f(\underline{x})}{\partial x_i} = 1$, wenn die Funktion $f(\underline{x})$ in der Variable x_i linear ist.

Die Beweise für die Richtigkeit dieser Eigenschaften und weitere Eigenschaften der Ableitungen der Booleschen Funktionen wurden in [BoPo81] beschrieben.

Durch die wiederholte partielle Ableitung nach mehreren Variablen $x_{i_1}, ..., x_{i_k}$ (2.10) entsteht die **k-fache Ableitung**.

$$\frac{\partial^k f(\underline{x})}{\partial x_{i_1} \cdots \partial x_{i_k}} = \frac{\partial}{\partial x_{i_1}} \frac{\partial}{\partial x_{i_2}} \cdots \frac{\partial f(\underline{x})}{\partial x_{i_k}} \tag{2.10}$$

Die **k-fache Ableitung** nach den Variablen $x_{i_1}, ..., x_{i_k}$ ist von der Reihenfolge der Variablen unabhängig, d. h. für eine Permutation $\pi(\underline{x_1}) = (\pi(x_{i_1}), ..., \pi(x_{i_k}))$ der Variablen $\underline{x_1} = (x_{i_1}, ..., x_{i_k})$ gilt:

$$\frac{\partial^k f(\underline{x})}{\partial x_{i_1} \cdots \partial x_{i_k}} = \frac{\partial^k f(\underline{x})}{\partial \pi(x_{i_1}) \cdots \partial \pi(x_{i_k})} \tag{2.11}$$

Analog zur partiellen Ableitung werden das **partielle Minimum** und das **partielle Maximum** einer Booleschen Funktion $f(\underline{x})$ nach der Variable x_i nach (2.12) und (2.13) definiert.

$$\min_{x_i} f(\underline{x}) = f(x_i, \underline{x_0}) \cdot f(\overline{x_i}, \underline{x_0}) \tag{2.12}$$

$$\max_{x_i} f(\underline{x}) = f(x_i, \underline{x_0}) \vee f(\overline{x_i}, \underline{x_0}) \tag{2.13}$$

Minimum und Maximum einer Booleschen Funktion haben folgende wichtige Eigenschaften:

1. $\min_{x_i} f(\underline{x}) = f(x_i = 0, \underline{x_0}) \cdot f(x_i = 1, \underline{x_0})$ und
$\max_{x_i} f(\underline{x}) = f(x_i = 0, \underline{x_0}) \vee f(x_i = 1, \underline{x_0})$;

2. Minimum und Maximum einer Booleschen Funktion nach der Variable x_i sind von x_i unabhängig;

Die Beweise für die Richtigkeit dieser Eigenschaften werden in [BoPo81] geführt.

Das **k-fache Minimum** (2.14) und das **k-fache Maximum** (2.15) entstehen durch die Ausführung der partiellen Minima und Maxima nach mehreren Variablen $x_{i_1}, ..., x_{i_k}$.

$$\min_{\underline{x_i}}{}^k f(\underline{x}) = \min_{x_{i_1}}(\min_{x_{i_2}}(\cdots \min_{x_{i_k}} f(\underline{x}) \cdots)) \tag{2.14}$$

$$\max_{\underline{x_i}}{}^k f(\underline{x}) = \max_{x_{i_1}}(\max_{x_{i_2}}(\cdots \max_{x_{i_k}} f(\underline{x}) \cdots)) \tag{2.15}$$

Zwischen den Ableitungsoperationen einer Booleschen Funktion existieren folgende Zusammenhänge [BoPo81]:

1.
$$\min_{x_i} f(\underline{x}) = f(\underline{x})\overline{\frac{\partial f(\underline{x})}{\partial x_i}} \tag{2.16}$$

2.
$$\max_{x_i} f(\underline{x}) = f(\underline{x}) \vee \frac{\partial f(\underline{x})}{\partial x_i} \tag{2.17}$$

3.
$$\min_{x_i} f(\underline{x}) \oplus \frac{\partial f(\underline{x})}{\partial x_i} \oplus \max_{x_i} f(\underline{x}) = 0 \tag{2.18}$$

Wesentlich sind einige weitere in [BoPo81] und [BoSt91] beschriebene Eigenschaften der Ableitungsoperationen der Booleschen Funktionen. So gelten für die gegebenen Booleschen Funktionen $f(\underline{x})$ und $g(\underline{x})$ und die Konstante a folgende Aussagen:

1.
$$\min_{\underline{x_i}}{}^k a = a \tag{2.19}$$

2.
$$\max_{\underline{x_i}}{}^k a = a \tag{2.20}$$

3.
$$\min_{\underline{x_i}}{}^k (a \cdot f(\underline{x})) = a \cdot \min_{\underline{x_i}}{}^k f(\underline{x}) \tag{2.21}$$

4.
$$\max_{\underline{x_i}}{}^k (a \cdot f(\underline{x})) = a \cdot \max_{\underline{x_i}}{}^k f(\underline{x}) \tag{2.22}$$

5.
$$\frac{\partial^k (f(\underline{x}) \oplus g(\underline{x}))}{\partial x_{i_1} \cdots \partial x_{i_k}} = \frac{\partial^k f(\underline{x})}{\partial x_{i_1} \cdots \partial x_{i_k}} \oplus \frac{\partial^k g(\underline{x})}{\partial x_{i_1} \cdots \partial x_{i_k}} \tag{2.23}$$

6.
$$\min_{\underline{x_i}}{}^k (f(\underline{x}) \cdot g(\underline{x})) = \min_{\underline{x_i}}{}^k f(\underline{x}) \cdot \min_{\underline{x_i}}{}^k f(\underline{x}) \tag{2.24}$$

7.
$$\max_{\underline{x_i}}{}^k (f(\underline{x}) \vee g(\underline{x})) = \max_{\underline{x_i}}{}^k f(\underline{x}) \vee \max_{\underline{x_i}}{}^k f(\underline{x}) \tag{2.25}$$

$q(\underline{x})$	Funktionswert = 1	Infimum, Einsmenge M1
$r(\underline{x})$	Funktionswert = 0	Nullmenge M0
$\varphi(\underline{x})$	Funktionswert = -	Sperrfunktion, MS
$p(\underline{x})$	Funktionswert = 1 oder -	Supremum

Tabelle 2.3: Definition der charakteristischen Verbandsfunktionen

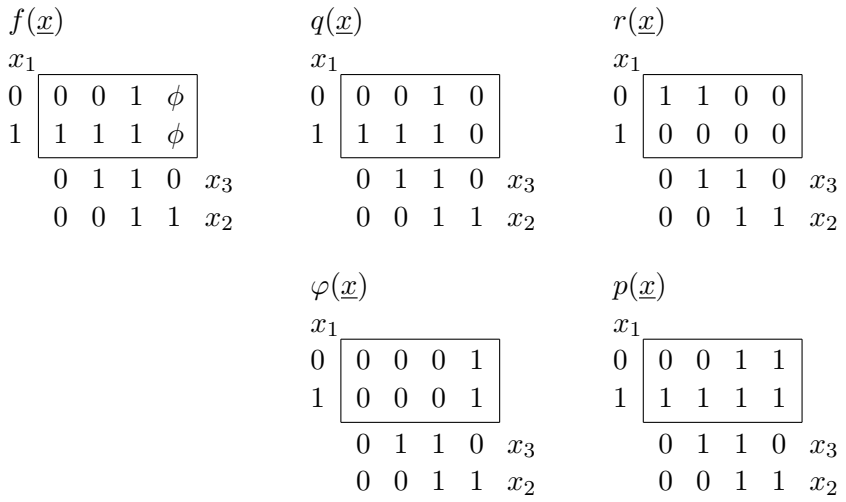


Abbildung 2.2: Beispiel für charakteristische Verbandsfunktionen

2.1.3 Partiell definierte Funktionen

Anstelle vollständig definierter Boolescher Funktionen werden in vielen Anwendungen partiell definierte Boolesche Funktionen betrachtet. Ihr Definitionsbereich ist auf Teile von B^k beschränkt. Eine partiell definierte Boolesche Funktion charakterisiert eine Menge von Booleschen Funktionen, die die Eigenschaft eines Funktionsverbänds bzw. eines Funktionsintervalls besitzt.

Ein Funktionsverband kann durch die Funktionen $q(\underline{x})$, $r(\underline{x})$, $\varphi(\underline{x})$ und $p(\underline{x})$ charakterisiert werden, die wie in der Tabelle 2.3 beschrieben definiert sind. Für jede Funktion $f(\underline{x})$ des Verbands gilt, dass $q(\underline{x}) \leq f(\underline{x}) \leq p(\underline{x})$ (siehe auch Beispiel in der Abbildung 2.2). Um ein Funktionsverband zu charakterisieren, werden von den vier genannten Funktionen stets nur zwei ($p(\underline{x})$ und $q(\underline{x})$, $p(\underline{x})$ und $\varphi(\underline{x})$, $q(\underline{x})$ und $r(\underline{x})$, $q(\underline{x})$ und $\varphi(\underline{x})$ oder $r(\underline{x})$ und $\varphi(\underline{x})$) benötigt, die anderen beide können aus den vorhandenen nach (2.26)-(2.29) berechnet werden.

$$q(\underline{x}) = p(\underline{x}) \cdot \overline{\varphi(\underline{x})} = \overline{r(\underline{x})} \cdot \overline{\varphi(\underline{x})} \tag{2.26}$$

$$r(\underline{x}) = \overline{p(\underline{x})} = \overline{q(\underline{x})} \cdot \overline{\varphi(\underline{x})} \tag{2.27}$$

$$\varphi(\underline{x}) = p(\underline{x}) \cdot \overline{q(\underline{x})} = \overline{q(\underline{x})} \cdot \overline{r(\underline{x})} \tag{2.28}$$

$$p(\underline{x}) = \overline{r(\underline{x})} = q(\underline{x}) \vee \varphi(\underline{x}) \tag{2.29}$$

Wegen (2.27) ist nur das Paar $p(\underline{x})$ und $r(\underline{x})$ zur Charakteristik eines Funktionsverbands nicht verwendbar.

2.1.4 Shannon- und Davio-Dekomposition

Jede Boolesche Funktion kann in zwei Boolesche Funktionen zerlegt werden (siehe auch Abschnitt 4.1.1). Die allgemein bekannten und in der Literatur ausreichend beschriebenen Dekompositionen dieser Art sind die Shannon- und Davio-Dekompositionen.

Shannon-Dekomposition: Ist $f(\underline{x}) = f(x_1, x_2, \ldots, x_k)$ eine Boolesche Funktion $B^k \rightarrow B$, so gilt für alle $x_i \in (x_1, x_2, \ldots, x_k)$:

$$\begin{aligned} f(x_i, \underline{x_0}) &= \overline{x_i}\, f_{S0}(\underline{x_0}) \vee x_i f_{S1}(\underline{x_0}) \\ &= \overline{x_i}\, f(x_i = 0, \underline{x_0}) \vee x_i f(x_i = 1, \underline{x_0}) \end{aligned} \tag{2.30}$$

Die Funktionen $f_{S0}(\underline{x_0})$ und $f_{S1}(\underline{x_0})$ werden als Cofaktoren bezeichnet und beschreiben die Funktion $f(x_i, \underline{x_0})$ an den Stellen $x_i = 0$ und $x_i = 1$.

Davio-Dekomposition: Ist $f(\underline{x}) = f(x_1, x_2, \ldots, x_k)$ eine Boolesche Funktion $B^k \rightarrow B$, so gilt für alle $x_i \in (x_1, x_2, \ldots, x_k)$:

$$\begin{aligned} f(\underline{x_0}, x_i) &= f_{S0}(\underline{x_0}) \oplus x_i f_D(\underline{x_0}) \\ &= f(x_i = 0, \underline{x_0}) \oplus x_i (f(x_i = 0, \underline{x_0}) \oplus f(x_i = 1, \underline{x_0})) \\ &= f(x_i = 0, \underline{x_0}) \oplus x_i \frac{\partial f(x_i, \underline{x_0})}{\partial x_i} \end{aligned} \tag{2.31}$$

$$\begin{aligned} f(\underline{x_0}, x_i) &= f_{S1}(\underline{x_0}) \oplus \overline{x_i}\, f_D(\underline{x_0}) \\ &= f(x_i = 1, \underline{x_0}) \oplus \overline{x_i}\, (f(x_i = 0, \underline{x_0}) \oplus f(x_i = 1, \underline{x_0})) \\ &= f(x_i = 1, \underline{x_0}) \oplus \overline{x_i}\, \frac{\partial f(x_i, \underline{x_0})}{\partial x_i} \end{aligned} \tag{2.32}$$

2.2 Ternärvektorliste

2.2.1 Kodieren der Booleschen Funktion durch TVL

Bei der Abbildung einer Booleschen Funktion durch eine Ternärvektorliste werden Konjunktionen von Variablen einer Funktion in der disjunktiven Form oder Antivalenzform bzw. Disjunktionen von Variablen einer Funktion in der konjunktiven Form oder Äquivalenzform als Ternärvektoren (TV) dargestellt. Ein TV beschreibt durch "0", dass eine Variable negiert auftritt, durch "1", dass eine Variable nicht negiert auftritt, oder durch "-", dass eine Variable nicht vorhanden ist. Die ausführliche Beschreibung der Kodierung ist in [BoSt91, DKSW92] zu finden. Jeder Ternärvektor, der s "-"-Elemente enthält, repräsentiert 2^s Binärvektoren.

Ternärvektoren werden zu Ternärvektorlisten (TVL) zusammengefügt. Die Form der auf diese Weise dargestellten Funktion wird in dem Formparameter vermerkt.
Die TVL der Funktionen aus dem Abschnitt 2.1.1 sind:

$$DF(f_1(\underline{x})) = \begin{array}{ccc} x_1 & x_2 & x_3 \\ 0 & 1 & - \\ - & - & 0 \end{array} \qquad AF(f_2(\underline{x})) = \begin{array}{ccc} x_1 & x_2 & x_3 \\ 0 & 1 & - \\ - & - & 0 \end{array}$$

$$KF(f_3(\underline{x})) = \begin{array}{ccc} x_1 & x_2 & x_3 \\ 0 & 1 & - \\ - & - & 0 \end{array} \qquad EF(f_4(\underline{x})) = \begin{array}{ccc} x_1 & x_2 & x_3 \\ 0 & 1 & - \\ - & - & 0 \end{array}$$

2.2.2 Orthogonale TVL

Zwei Ternärvektoren sind orthogonal, wenn für mindestens eine Variable die Kombination "01" auftritt. Sind alle TV orthogonal zueinander, ist die gesamte TVL orthogonal. In der orthogonalen Ternärvektorliste gibt es keinen Binärvektor, der in mehr als einer Zeile (TV) vorkommt.

Die orthogonalen Ternärvektorlisten haben eine besondere Bedeutung, weil bei den orthogonalen TVL nicht zwischen den DF und AF bzw. KF und EF unterschieden werden muss. Man spricht von den Funktionen in ODA-Form (orthogonale DF bzw. AF) und in OKE-Form (orthogonale KF bzw. EF). Die TVL der Funktionen aus dem Abschnitt 2.1.1 in ODA- und OKE-Form sind:

$$ODA(f_1(\underline{x})) = \begin{array}{ccc} x_1 & x_2 & x_3 \\ 0 & 1 & 1 \\ - & - & 0 \end{array} \qquad ODA(f_2(\underline{x})) = \begin{array}{ccc} x_1 & x_2 & x_3 \\ 0 & 1 & 1 \\ 1 & - & 0 \\ 0 & 0 & 0 \end{array}$$

$$OKE(f_3(\underline{x})) = \begin{array}{ccc} x_1 & x_2 & x_3 \\ 0 & 1 & 1 \\ - & - & 0 \end{array} \qquad OKE(f_4(\underline{x})) = \begin{array}{ccc} x_1 & x_2 & x_3 \\ 0 & 1 & 1 \\ 1 & - & 0 \\ 0 & 0 & 0 \end{array}$$

2.2.3 Raumkonzept

Das Raumkonzept des XBOOLE-Programmsystems ermöglicht die Berechnungen von Ternärvektorlisten im mehreren Booleschen Räumen beliebiger Dimension [BoSt91, DKSW92].

Werden die Variablen aller Raumdefinitionen in einer Variablenliste zusammengefasst, bilden sie den sogenannten Metaraum. Die Anzahl der Variablen im Metaraum ist dabei unbeschränkt. Der Metaraum umfasst alle aktuell verwendeten Variablen, gestattet aber nicht ihre unmittelbare Verarbeitung. Als Voraussetzung für die Verarbeitung werden beliebig viele Boolesche Räume gebildet, die nur eine beschränkte Anzahl der Variablen aus dem Metaraum enthalten. In jedem Booleschen Raum können beliebig viele Ternärvektorlisten eingebettet werden.

Diese zweistufige Variablen Zuordnung verbindet die Vorteile der extrem großen Gesamtvariablenanzahl im Metaraum mit einer sinnvoll vordefinierten Variablenzahl in jedem Booleschen Raum mit der kompatiblen Spaltenzuordnung. Alle Booleschen Operationen können unmittelbar nur mit den Ternärvektorlisten eines speziellen Booleschen Raums durchgeführt werden. Die Ternärvektorlisten eines Boooleschen Raums enthalten alle in diesem Raum definierten Variablen, so dass das Durchführen von Operationen keine zusätzliche Zeit für das Anpassen von TVL-Spalten benötigt, und damit die schnelle spaltengerechte Verarbeitung gewährleistet wird. Um die Berechnungen zwischen Ternärvektorlisten verschiedener Boolescher Räume durchführen zu können, werden die zu verknüpfenden TVL durch eine spezielle Operation spaltengerecht in einen gemeinsamen Booleschen Raum übertragen.

Das Raumkonzept gestattet es somit, die Lösung beliebig großer binärer Aufgaben mit der schnellen Verarbeitung in separaten Booleschen Räumen zu verbinden.

$$PHL(\underline{x}, y) = \begin{array}{cccc} x_1 & x_2 & x_3 & y \\ 0 & 1 & 1 & 1 \\ - & - & 0 & 1 \\ 1 & - & 1 & 0 \\ 0 & 0 & 1 & 0 \end{array} \qquad ODA(f(\underline{x})) = \begin{array}{ccc} x_1 & x_2 & x_3 \\ 0 & 1 & 1 \\ - & - & 0 \end{array}$$

Abbildung 2.3: Phasenliste

2.3 Phasenliste

Das Verhalten eines Booleschen Systems wie z. B. eines Schaltnetzwerks kann durch die Phasenliste (PHL) beschrieben werden (siehe auch [BoSt91]). Eine Phase beschreibt dabei das Verhalten von Teilen des Systems zu einem definierten Zeitpunkt, z. B. die Belegungen, die an bestimmten Leitungen einer Schaltung zu einem bestimmten Zeitpunkt vorliegen. Eine Phase kann durch einen Binärvektor dargestellt werden. Alle an einem Booleschen System beobachtbaren Phasen werden in der PHL als Verhaltensmodell zusammengefasst. Die PHL einer kombinatorischen Schaltung mit n Eingängen und einem Ausgang enthält 2^n Binärvektoren, d. h. für jede mögliche Belegung der Eingänge genau eine Phase. Binärvektoren lassen sich zu Ternärvektoren zusammenfassen. Im Vergleich zur Funktions-TVL enthält die Phasenliste zusätzlich die Spalte für den Funktionswert.

Für die gegebene Funktion $f(\underline{x})$ können die einzelnen Phasen als Lösungen einer expliziten Booleschen Funktionsgleichung (2.33) oder einer impliziten Booleschen Gleichung (2.34) betrachtet werden.

$$y = f(\underline{x}) \tag{2.33}$$

$$F(\underline{x}, y) = 1 \tag{2.34}$$

Zusammengefasst zu einer PHL bilden die Phasen die Lösungsmenge $L1(\underline{x}, y)$ der Gleichungen. In der Abbildung 2.3 links ist die Lösungsmenge der Gleichung (2.35) als PHL dargestellt. Die Transformation zwischen der Phasenliste und der Funktions-TVL (siehe Abbildung 2.3 rechts) ist jederzeit möglich.

$$y = \overline{x}_1 x_2 \vee \overline{x}_3 \tag{2.35}$$

Eine Phasenliste kann weiterhin um die Belegungen an den ausgewählten bzw. allen inneren Leitungen einer Schaltnetzwerks erweitert werden.

Durch eine PHL lässt sich das Verhalten eines Booleschen Systems in der Gesamtheit bescheiben. Man spricht in diesem Fall von globalen Phasenlisten. Für ein kombinatorisches Schaltnetzwerk mit einem Ausgang beschreibt die globale PHL die Lösungsmenge der Gleichungen (2.33) oder (2.34). Werden die Teilsysteme einzeln beobachtet, können sie mit lokalen PHL beschrieben werden. In diesem Fall beschreiben die lokalen PHL die Lösungsmengen der einzelnen Gleichungen des Gleichungssystems (2.36).

$$\left.\begin{array}{l} y = f(\underline{x}, \underline{g}) \\ g_1 = f_1(\underline{x}, \underline{g}) \\ g_2 = f_2(\underline{x}, \underline{g}) \\ \cdots\cdots\cdots\cdots \\ g_k = f_k(\underline{x}, \underline{g}) \end{array}\right| \tag{2.36}$$

Zur Modellierung eines Schaltnetzwerks mit l Ausgängen werden alle l Ausgangsleitungen beobachtet. Eine Phase umfasst dabei die Belegungen aller n Eingänge und l Ausgänge, die zu wenigstens einem Zeitpunkt beobachtet werden. Die gesamte PHL enhält in diesem Fall 2^n Binärvektoren. Die Phasenliste entsteht in diesem verallgemeinerten Fall als Lösung des Booleschen Gleichungssystem (2.37).

$$\left.\begin{array}{l} y_1 = f_1(\underline{x}) \\ y_2 = f_2(\underline{x}) \\ \ldots\ldots\ldots \\ y_l = f_l(\underline{x}) \end{array}\right| \tag{2.37}$$

Eine Phasenliste kann weiterhin zu einer Verbandsphasenliste verallgemeinert werden um die Funktionsverbände zu beschreiben. Im Vergleich zur bisherigen Phasenliste können in der Verbandsphasenliste in der Spalte des Ausgangs auch Strichelemente auftreten. Diese Phasen sind so zu interpretieren, dass für die entsprechende Eingangsbelegung sowohl "0" als auch "1" am Schaltnetzwerkausgang auftreten darf.

2.4 Verteilte Systeme

2.4.1 Softwarekonzept

Eine verteilte Anwendung ist eine Anwendung, die parallel auf mehreren Rechnern bzw. auch Prozessoren abgearbeitet wird, wobei die einzelnen Komponenten durch Informationsaustausch miteinander in Verbindung stehen. Ein System mit mehreren Rechnern bzw. Prozessoren, auf denen mindestens eine verteilte Anwendung lauffähig installiert ist, wird verteiltes System genannt [Web98].

Die Voraussetzung für die Realisierung verteilter Anwendungen ist eine Unterstützung durch die genutzten Betriebssysteme. Ein Betriebssystem, das um einen Kommunikationsmechanismus erweitert wurde, so dass Kommunikation und Kooperation zwischen Betriebssystemen möglich ist, nennt man Netzwerkbetriebssystem [LaSch94]. Folgerichtig müssen in Netzwerkbetriebssystemen Komponenten zur Realisierung der Kommunikation integriert sein. Als Beispiel sind Sockets zu nennen, die einen Zugang zur Transportschicht des Kommunikationssystems ermöglichen und einen Datenaustausch auf der Grundlage unterschiedlicher Protokolle erlauben. Unter Sockets sind Kommunikationsendpunkte zu verstehen, die eine Kommunikation insbesondere zwischen entfernten Prozessen ermöglichen. Die Semantik der Socket sieht eine Abbildung auf der Basis der Ein- und Ausgabefunktionen des Betriebssystems vor, so dass eine einfache Schnittstelle für den Nutzer zur Verfügung steht.

2.4.2 Kommunikationsmodelle

Bei der Arbeit in verteilten Systemen kommunizieren mindesten zwei Teilnehmer (Prozesse) miteinander. Eine Gleichberechtigung der Partner ist möglich, wobei jedoch in der Regel den Kommunikationspartnern spezifische Aufgaben zugeordnet werden, die zu einer Rollenverteilung führen. Mit der Client-Server Architektur wird diese Form des Zusammenwirkens von Komponenten im verteilten System beschrieben. Kennzeichnend für diese Architektur ist, dass Server Aufträge sogenannter Clients entgegennehmen und bearbeiten. Im Regelfall werden Server und Client nicht nur in unterschiedlichen Prozessen abgearbeitet, sondern auch auf entfernten Rechner laufen. Ein Informationsaustausch ist daher nur

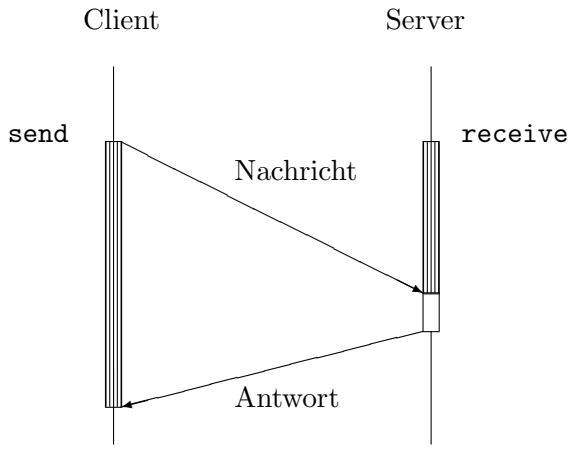


Abbildung 2.4: Synchrone Kommunikation

auf der Grundlage von Kommunikationsmechanismen (z. B. Sockets) möglich. Bezüglich des konkreten Szenarium des Zusammenwirkens von Clients und Servern können Unterschiede bestehen. Beispielsweise kann ein Server parallel von mehreren Clients angesprochen werden. Eine Umkehrung dieses Prinzips erfordert dagegen die Aufgabenstellung in dieser Arbeit. Ein Client vergibt parallel Aufträge an mehrere gleichartige Server, um durch Parallelisierung ein günstigeres Verarbeitungsverhalten zu erreichen.

Nachfolgend sollen in Form einer Klassifikation grundsätzliche Möglichkeiten des Nachrichtenaustauschs gegeneinander abgegrenzt werden. Eine erste Unterscheidung besteht in der Unterteilung in einen blockierenden (synchronen) und nichtblockierenden (asynchronen) Nachrichtenaustausch. Bei blockierender Kommunikation ist der Nachrichtenaustausch automatisch mit einer Synchronisation der Teilnehmerprozesse verbunden. Ein Sender ist blockiert, bis die Routine **send** von der Empfängerseite eine Empfangsbestätigung (**acknowledge**) erhalten hat, da erst dann von einer erfolgreichen Datenübertragung ausgegangen werden kann. Ein korrespondierendes **receive** bei Empfänger blockiert dagegen bis eine Nachricht des Senders eingegangen ist. Das Zusammenwirken des Clientes und des Servers bei der blockierenden Kommunikation ist in der Abbildung 2.4 zu sehen. Die Blockierzeiten des Senders und des Empfängers sind dabei als gestrichelte Balken dargestellt. Da durch den blockierenden Nachrichtenaustausch eine Synchronisation von Sender und Empfänger stattfindet, sind keine zusätzlichen Mechanismen zur zeitlichen Abstimmung der Übertragungsvorgänge notwendig.

Im Gegensatz zur blockierenden werden bei einer nichtblockierenden Übertragung keine Rückmeldungen erwartet. Nach dem Auslösen des Sendevorgangs wird sofort der Verarbeitungsablauf fortgesetzt, so dass auch keine Rückschlüsse auf einen erfolgreichen Verlauf des Übertragungsvorgangs möglich sind.

Eine weitere Klassifikation besteht in der Unterscheidung in meldungs- und auftragsorientierte Kommunikationsformen. Dabei sind Einwegnachrichten typisch für die meldungsorientierte Kommunikation. Der Sender übermittelt eine Nachricht und erwartet höchstens eine Bestätigung, dass die Nachricht empfangen wurde. Ein weiterer Informationsgehalt ist mit der Rückmeldung nicht verbunden. Die auftragsorientierte Kommunikation ist durch die Übermittlung von Hin- und Rücknachrichten gekennzeichnet. Der Sender verschickt

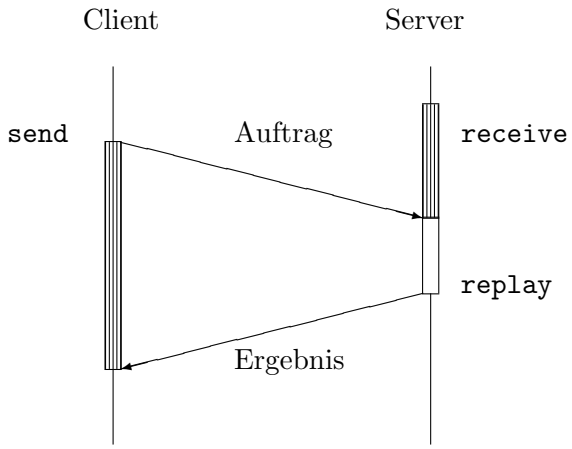


Abbildung 2.5: Synchroner entfernter Dienstaufruf

innerhalb einer Nachricht einen Auftrag und wartet auf die Rücksendung der Ergebnisse. Diese Kommunikationsformen werden auch als asynchroner entfernter Dienstaufruf (asynchronous remote service invocation, ARSI) bzw. synchroner entfernter Dienstaufruf (synchronous remote service invocation, SRSI) bezeichnet.

Der zeitliche Ablauf des synchronen entfernten Dienstaufrufs ist im Bild 2.5 dargestellt. Der Client übermittelt mit **send** über einen Datenkanal den Auftrag an den Server und wartet anschließend auf eine Antwort. Auf der Serverseite wird nach dem Eingang des Auftrags eine Bearbeitung eingeleitet und das Ergebnis an den Client geschickt. Die Blockierzeiten des Senders und des Empfängers sind hier wie im Bild 2.4 als gestrichelte Balken dargestellt, während durch den leeren Balken die Verarbeitungzeit auf der Serverseite beschrieben wird. Der Gesamtablauf der Client-Server-Kommunikation ist also auf die Grundoperationen **send**, **receive** und **replay** zurückzuführen, die das Senden, Empfangen und Bearbeiten der Nachrichten realisieren. Die Bereitstellung der Ergebnisse am Client wird gleichzeitig als Empfangsbestätigung gewertet.

2.4.3 Prozessmanagement

Eine grundlegende Anforderung an einen Server besteht in der Fähigkeit, auf mehrere unabhängige Clientanforderungen reagieren zu können. Als Realisierungsmöglichkeiten sind die iterative und nebenläufige Arbeitsweise zu unterscheiden, die schließlich auch zu einem unterschiedlichen Arbeitsverhalten des Servers führen. Relativ einfach ist die iterative Bearbeitung von Clientanforderungen zu implementieren. Die am Server eintreffenden Clientanforderungen werden in eine Warteschlange übernommen und sequentiell abgearbeitet. Ein neuer Auftrag wird erst dann bearbeitet, wenn der Vorgänger abgeschlossen wurde. Für sehr einfache Dienste mit geringen Auftragsbearbeitungszeiten und kleiner Anfragehäufigkeit sind iterativ ausgelegte Server durchaus ausreichend. Die Antwortzeit für einen einzelnen Auftrag kann aber insbesondere dann sehr lang werden, wenn eine Vielzahl von nicht erledigten Anforderungen bereits vorliegt. Zu berücksichtigen ist, dass die Abarbeitung der Aufträge nach der Regel First-Come-First-Served erfolgt. Eine Berücksichtigung von Prioritäten bzw. eine Optimierung der Bearbeitung ist dabei nicht vorgesehen.

Ein nebenläufiger Server ist durch die gleichzeitige Bearbeitung mehrerer Aufträge gekennzeichnet. Implementierungsmöglichkeiten sind durch das Erzeugen mehrerer Prozesse bzw. Threads (leichtgewichtige Prozesse) gegeben. Der Scheduler des Betriebssystems vergibt z. B. unter UNIX in Zeitscheiben die Steuerung an die einzelnen Prozesse, so dass ein quasi paralleler Verarbeitungsablauf entsteht. Auf die Abarbeitungszeiten nehmen dann Faktoren wie der Prozessortyp, die Anzahl der Prozessoren, der verfügbarer Arbeitsspeicher und die aktuelle Auslastung der Maschine Einfluss.

Die prinzipielle Arbeitsweise eines nebenläufigen Servers wird nachfolgend dargestellt:

```
forever
       receive (request,client);// Vaterprozess
       if (fork=0) then
              perform request;// Sohnprozess
              send(reply,client);
       endif;
endforever;
```

Nach dem Empfang einer Clientanforderung wird mit dem Systemruf **fork()** ein neuer Prozess erzeugt, der eine Kopie des Ausgangsprozesses darstellt. Innerhalb dieses Sohnprozesses erfolgt die Bearbeitung des Auftrags und das Versenden der Ergebnisse. Gleichzeitig wartet der Vaterprozess auf weitere Anforderungen, die in analoger Form behandelt werden.

Kapitel 3

Funktionale Komposition

3.1 Möglichkeiten der Darstellung einer Booleschen Funktion durch Phasenlisten

3.1.1 Analyse

Wie im Abschitt 2.3 dargelegt wurde, kann das Verhalten eines Booleschen Systems durch die Phasenliste (PHL) beschrieben werden. Als mögliche Darstellungsformen der Booleschen Funktion sind dabei die globale und die lokale Phasenlisten zu unterscheiden. Im weiteren ist zu analysieren, welche dieser Darstellungen als Datenmodell für eine Boolesche Funktion am besten geeignet ist.

Die globalen Phasenlisten beschreiben das Boolesche System in der Gesamtheit und stellen die Lösungsmenge der Gleichung (3.1) dar. Die Teilsysteme des Booleschen Systems können mit den lokalen PHL beschrieben werden. In diesem Fall beschreiben die Phasenlisten die Lösungsmengen der einzelnen Gleichungen des Gleichungssystems (3.2).

$$y = f(\underline{x}) \tag{3.1}$$

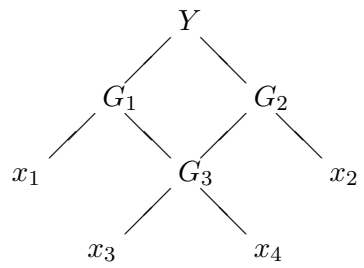
$$\left|\begin{array}{l} y = f(\underline{x}, \underline{g}) \\ g_1 = f_1(\underline{x}, \underline{g}) \\ g_2 = f_2(\underline{x}, \underline{g}) \\ \cdots\cdots\cdots\cdots \\ g_k = f_k(\underline{x}, \underline{g}) \end{array}\right. \tag{3.2}$$

Die Funktionen $f(\underline{x}, \underline{g})$, $f_1(\underline{x}, \underline{g}), \cdots, f_k(\underline{x}, \underline{g})$ der einzelnen Gleichungen des Gleichungssystems (3.2) sind im Unterschied zur der Funktion $f(\underline{x})$ der Gleichung (3.1) nicht von allen Variablen $\underline{x}$ und $\underline{g}$ abhängig. Somit ist die Lösungsmenge jeder einzelnen Gleichung des Systems (3.2) kleiner als die Lösungsmenge der Gleichung (3.1).

Wie schon erwähnt wurde, werden bei der Darstellung der Booleschen Funktion durch Phasenlisten zwei Grenzfälle betrachtet:

- Darstellung durch nur eine globale Phasenliste
- Darstellung durch viele lokale Phasenlisten

Für die Beantwortung der Frage, welche der Darstellungen weniger speicherplatz- und zeitintensiv ist und damit bevorzugt eingesetzt werden sollte, müssen folgende Vor- und Nachteile betrachtet werden:



$$Y = PHL(g_1, g_2, y)$$

$$G_1 = PHL(x_1, g_3, g_1)$$

$$G_2 = PHL(g_3, x_2, g_2)$$

$$G_3 = PHL(x_3, x_4, g_3)$$

Abbildung 3.1: Strukturgraph der Funktion der Benchmark-Schaltung C17

Bei der Darstellung durch nur eine Phasenliste:

- Man verwendet nur eine einfache Datenstruktur, die leicht zu handhaben ist.
- Bei einer extrem großen Anzahl von Variablen in der PHL, wie sie bei der Darstellung durch globale PHL zu erwarten sind, können Phasenlisten-TVL sehr speicheraufwendig werden.

Bei der Darstellung durch viele lokale Phasenlisten:

- Jede der Phasenlisten-TVL hat nur eine begrenzte Variablenzahl und benötigt damit wenig Speicherplatz.
- Für die Darstellung einer Funktion als PHL-Tupel ist eine Struktur notwendig, die die Verbindungen zwischen den lokalen PHL der Funktion realisiert und damit die gesamte Funktion beschreibt. Mit der wachsenden Anzahl der Elemente wird auch die Handhabung der Struktur aufwendiger. So wird zum Beispiel bei der Durchführung von Operationen zusätzliche Zeit für die Suche in der Struktur benötigt.

Eine weitere Möglichkeit der Darstellung einer Booleschen Funktion durch PHL besteht in der Komposition von lokalen Phasenlisten. Durch die Komposition mehrerer lokaler Phasenlisten zu einer Phasenliste kann die Anzahl der Elemente des Tupels verringern werden. Die entstandenen Phasenlisten werden voraussichtlich größer, so dass die Gegenüberstellung des Speicherplatzbedarfs für zusammengefasste und nicht zusammengefasste PHL notwendig wird. Führt die Komposition zu den positiven Ergebnissen, muss weiterhin untersucht werden, wieweit das Zusammenfassen fortgesetzt werden soll, und nach welchen Kriterien die zusammenfassenden PHL auszusuchen sind.

3.1.2 Strukturgraph über Phasenlisten

Die Darstellung einer Booleschen Funktion über mehrere Phasenlisten erfordert eine Datenstruktur, die die Zusammenhänge zwischen den einzelnen Phasenlisten beschreibt. Als eine geeignete Datenstruktur kann hier ein einem logischen Netzwerk ähnlicher Graph verwendet werden (siehe [deMi94, HaSo96, MoSc99]). Als Knoten dieses Strukturgraphen werden die Phasenlisten und Eingangsvariablen einer Funktion erfasst.

```
struct nodes { unsigned int knoten_nummer,
                            *phl;
               struct list_of_nodes   *nachknoten,
                                      *vorknoten;
               struct nodes *naechster_knoten;
               };
struct list_of_nodes { struct nodes *knoten,
                       struct list_of_nodes *naechster_eintrag;
                       };
```

Abbildung 3.2: Implementierung der Struktur "Knoten"

Als Beispiel ist in der Abbildung 3.1 der Strukturgraph der Funktion, die einen der Ausgänge der Benchmark-Schaltung C17 beschreibt, dargestellt. An der Spitze der Struktur wird die Phasenliste mit der Ausgangsvariablen y als Wurzelknoten Y abgespeichert. Die Knoten G_1 und G_2 in der Abbildung 3.1 sind die Nachknoten des Knotens Y. Der Knoten Y ist wiederum der Vorknoten für die Knoten G_1 und G_2. Die Knoten Y und G_1 haben in ihren Phasenlisten die gemeinsame Variable g_1, die in der PHL des Knotens Y als eine Eingangsvariable und in der PHL des Knotens G_1 als Ausgangsvariable auftritt und als abhängige Variable bezeichnet wird. Diese Aussage gilt entsprechend auch für alle anderen Nach- und Vorknotenpaare. Die terminalen Knoten x_1, x_2, x_3 und x_4 beschreiben die primären Eingänge der Schaltung bzw. allgemein die unabhängigen Variablen der Gesamtfunktion.

Die Abbildung 3.2 zeigt vereinfacht die Implementierung eines Knotens `nodes` in der Programmiersprache C. Jeder Knoten beinhaltet die Identifikationsnummer, den Zeiger auf die PHL, die Zeiger auf die Listen der Nach- und Vorknoten und den Zeiger auf den nächsten Knoten der Struktur. Die Listen der Nach- und Vorknoten werden mit der Struktur `list_of_nodes` beschrieben (siehe Abbildung 3.2). Jedes Element dieser Struktur enthält die Adresse des Nach- bzw. Vorknotens und den Zeiger auf den nächsten Eintrag in der Liste.

Der Strukturgraph wird zuerst als eine Liste aus einzelnen Knoten (siehe Abbildung 3.3) implementiert. Diese scheinbar überflüssige Verkettung der Knoten dient dazu, die Suche nach den schon vorhandenen Knoten während des Strukturaufbaus zu ermöglichen. Obwohl jeder Knoten der Struktur die Verweise auf die Nach- und Vorknoten enthält, können die Adressen der Nachknoten erst zu einem späteren Zeitpunkt in die Struktur eingetragen werden, da zum Zeitpunkt des Anlegen eines Knotens seine Nachknoten noch nicht bekannt sind. (Siehe den Algorithmus zum Aufbau der Struktur im Abschnitt 3.1.4 und im Anhang A.1).

3.1.3 Operationen mit den Strukturgraphen

Für die gegebenen Funktionen $g_1 = f_1(\underline{a})$ und $g_2 = f_2(\underline{b})$, dargestellt durch die Strukturgraphen mit den Wurzelknoten G_1 und G_2, werden die Funktionen gesucht:

- die Negation der Funktion $f_1(\underline{a})$: $y = \overline{f_1(\underline{a})}$,

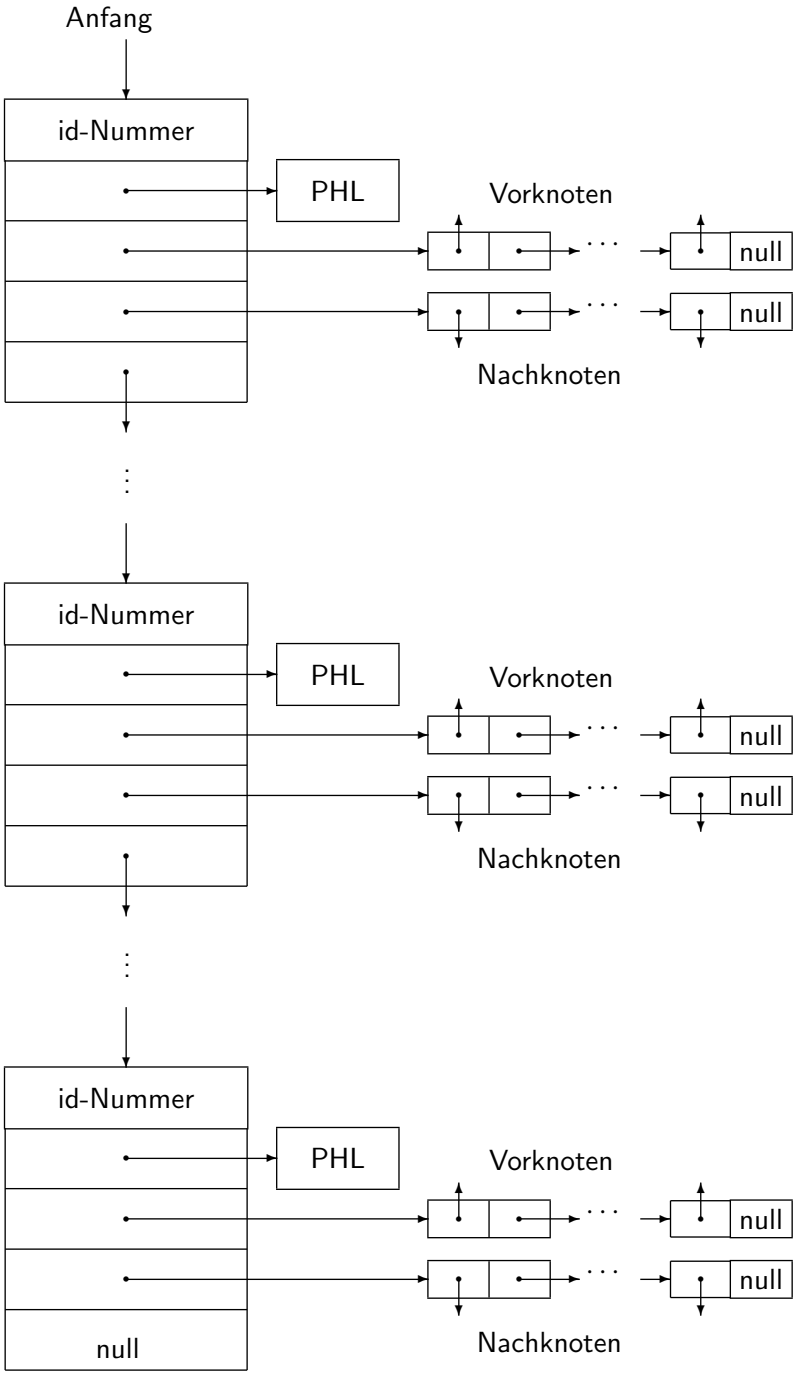


Abbildung 3.3: Datenstruktur “Knotenliste“

- die Konjunktion der Funktionen $f_1(\underline{a})$ und $f_2(\underline{b})$: $y = f_1(\underline{a}) \cdot f_2(\underline{b})$,
- die Disjunktion der Funktionen $f_1(\underline{a})$ und $f_2(\underline{b})$: $y = f_1(\underline{a}) \vee f_2(\underline{b})$,
- die Antivalenz der Funktionen $f_1(\underline{a})$ und $f_2(\underline{b})$: $y = f_1(\underline{a}) \oplus f_2(\underline{b})$,
- die Äquivalenz der Funktionen $f_1(\underline{a})$ und $f_2(\underline{b})$: $y = f_1(\underline{a}) \odot f_2(\underline{b})$.

Die Ergebnisfunktion der Negation der Funktion $f_1(\underline{a})$ kann als ein Strukturgraph dargestellt werden, der als Wurzelknoten die Phasenliste $Y = PHL(g_1, y)$ (3.3) erhält. Als Nachknoten des Wurzelknoten Y wird der Knoten G_1 eingetragen und alle weiteren Knoten der Funktion $f_1(\underline{a})$ werden übernommen. Soll die Ergebnisfunktion erneut negiert werden, ist das durch Entfernen des Wurzelknotens $Y = PHL(g_1, y)$ aus der Funktion erreichbar.

$$PHL(g_1, y) = \begin{array}{cc} g_1 & y \\ 0 & 1 \\ 1 & 0 \end{array} \tag{3.3}$$

Die Ergebnisfunktionen der Operationen Konjunktion, Disjunktion, Antivalenz und Äquivalenz der Funktionen $f_1(\underline{a})$ und $f_2(\underline{b})$ können auch als Strukturgraph mit dem Wurzelknoten $Y = PHL(g_1, g_2, y)$ dargestellt werden. Die Knoten G_1 und G_2 der Funktionen $f_1(\underline{a})$ und $f_2(\underline{b})$ werden dabei als Nachknoten des Knotens Y übernommen. Die Phasenlisten des Knoten Y sind für die Operation Konjunktion, Disjunktion, Antivalenz und Äquivalenz mit den Ternärvektorlisten (3.4), (3.5), (3.6) und (3.7) zu beschreiben. In Analogie zur Negation werden auch hier alle weiteren Knoten der Funktionen $f_1(\underline{a})$ und $f_2(\underline{b})$ übernommen.

$$PHL(g_1, g_2, y) = \begin{array}{ccc} g_1 & g_2 & y \\ 0 & - & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{array} \tag{3.4}$$

$$PHL(g_1, g_2, y) = \begin{array}{ccc} g_1 & g_2 & y \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & - & 1 \end{array} \tag{3.5}$$

$$PHL(g_1, g_2, y) = \begin{array}{ccc} g_1 & g_2 & y \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array} \tag{3.6}$$

$$PHL(g_1, g_2, y) = \begin{array}{ccc} g_1 & g_2 & y \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{array} \tag{3.7}$$

Bei der Durchführung von Operationen mit den Funktionen $f_1(\underline{a}, \underline{c})$ und $f_2(\underline{b}, \underline{c})$, die über eine oder mehrere gleiche unabhängige Variablen verfügen, muss darüber hinaus geprüft werden, ob die zu verknüpfenden Funktionen auch weitere gemeisame Knoten besitzen. Diese Knoten sollen nur einmal gespeichert und als Nachknoten der Knoten beider Funktionen eingetragen werden.

3.1.4 Tupel von lokalen Phasenlisten

Beim Darstellen von Booleschen Funktionen durch einen Tupel von lokalen Phasenlisten kann für das Speichern von lokalen Phasenlisten die Datenstruktur "Knoten" verwendet werden. Das in C implementierte Testsystem baut die Liste aus einzelnen Knoten entsprechend der Schaltungsstruktur wie folgt auf:

- Als Knoten werden die durch TVL dargestellten Phasenlisten und die unabhängigen Variablen erfasst.
- Jede Boolesche Funktionen wird unabhängig von anderen Funktionen dargestellt, d. h. es wird für jede Funktion eine eigene Liste aufgebaut.
- Bei dem Aufbau der Liste wird darauf geachtet, dass jeder Knoten nur einmal abgespeichert wird.

Die ausführliche Beschreibung von Algorithmen zum Aufbau einer Liste ist im Anhang A.1 zu finden.

Die Darstellung der Funktionen unabhängig von den anderen Funktionen eines Booleschen Systems ist mit einem Nachteil verbunden. Enthalten die verschiedenen Funktionen die gleichen Knoten, werden diese als unterschiedliche Knoten bewertet. Für die unabhängige Darstellung der Funktionen spricht dagegen, dass die Funktionen in diesem Fall auch in getrennten Booleschen Räumen gespeichert werden können (siehe Abschnitt 2.2.3), woraus eine Reduzierung des zur Darstellung der Booleschen Funktionen benötigten Speicherplatzes folgt.

Die Anzahl der durch Ternärvektolisten dargestellen Phasenlisten in einem Booleschen Raum ist bekanntlich logisch unbegrenzt. Damit ist es prinzipiell möglich alle Phasenlisten in einem Booleschen Raum einzubetten. Alle in diesem Raum eingebetteten Ternärvektorlisten haben damit die gleiche Variablenanzahl, d. h. in jeder TVL sind alle Variablen des Booleschen Raums enthalten. Das gilt auch für die Ternärvektorlisten, in denen tatsächlich nur wenige Variablen vorkommen. Diese Tatsache wirkt sich jedoch negativ auf den Speicherplatzbedarf aus. Der negative Effekt kann vermieden werden, wenn jede TVL in einem eigenen angepassten Booleschen Raum eingebettet wird. Um die Operationen zwischen den Ternärvektorlisten bei dieser Darstellungsart durchführen zu können, müssen die zu verknüpfenden Ternärvektorlisten unmittelbar vor der eigentlichen Operationsdurchführung zunächst in einen gemeinsamen Booleschen Raum transformiert werden. Damit werden die Variablenmengen der Ternärvektorlisten aneinander angepasst.

3.1.5 Globale Phasenliste

Für die Darstellung einer Booleschen Funktion durch eine globale Phasenliste ist keine zusätzliche Datenstruktur notwendig. Die Phasenliste lässt sich unmittelbar durch eine TVL darstellen.

Die globale Phasenliste kann aus den lokalen Phasenlisten durch das Verknüpfen von Phasenlisten-TVL mit der Operation Konjunktion und das damit verbundenen Entfernen aller abhängigen Variablen erstellt werden. Das Entfernen der abhängigen Variablen ist dann möglich, wenn alle Phasenlisten, die diese Variable als Eingangsvariable enthalten durch die Konjunktion verknüpft wurden. Um diesen Zeitpunkt zu bestimmen, wird für jede Zwischenvariable die Häufigkeit ihres Auftretens als Eingangsvariable in allen Knoten

der Funktion erfasst. Jeweils nach dem Verknüpfen von zwei Phasenlisten wird für die gemeinsame Variable geprüft, wie oft sie noch als Eingangsvariable auftritt. Ist sie nur noch in der gerade zu verknüpfenden Phasenliste vertreten, kann auch diese Variable nach dem Abschluss der Verknüpfung entfernt werden. Anderfalls wird der Zähler für die Häufigkeit des Auftretens um 1 reduziert (siehe auch den Algorithmus im Anhang A.2).

3.1.6 Tupel von zusammengefassten Phasenlisten

Bei der Darstellung einer Booleschen Funktion durch einen Tupel aus begrenzt zusammengefassten Phasenlisten werden die Ternärvektorlisten der lokalen Phasenlisten durch die Operation Konjunktion miteinander verknüpft. Nach dem Ausführen der Operation kann es bei den Ergebnis-TVL sowohl zur Vergrößerung wie auch zur Reduzierung der Zeilenzahl gegenüber den ursprunglichen Ternärvektorlisten kommen.

Die Konjunktion zweier Ternärvektorlisten mit gleichen Variablen führt im allgemeinen zur Reduzierung der Zeilenzahl der Ergebnis-TVL. Es wird der Durchschnitt von je zwei Ternärvektoren aus verschiedenen TVL gebildet und zur Ergebnis-TVL hinzugefügt. Da die Wahrscheinlichkeit, dass die Ternärvektoren der Ternärvektorlisten mit gleichen Variablen orthogonal zueinander sind, sehr gross ist, reduziert sich die Zeilenzahl der Ergebnis-TVL. Haben die zu verknüpfenden Ternärvektorlisten keine gemeinsamen Variablen, kommt es zur Bildung eines Kreuzproduktes und demzufolge zur Vergrößerung des Speicherplatzbedarfes.

Die folgenden Beispiele verdeutlichen den oben beschriebenen Sachverhalt. Im ersten Beispiel (Formel (3.8)) werden zwei Ternärvektorlisten mit gleichen Variablen konjunktiv verknüpft. Die Ergebnis-TVL enthält dabei weniger Ternärvektoren als die zu verknüpfenden Ternärvektorlisten. Das zweite Beispiel (Formel (3.9)) zeigt die konjunktive Verknüpfung zweier Ternärvektorlisten mit verschiedenen Variablen. Die Ergebnis-TVL wird als Kreuzprodukt aus den Ternärvektorlisten gebildet und enthält mehr Ternärvektoren als die ursprünglichen Ternärvektorlisten.

$$\begin{array}{ccc} x_1 & x_2 & x_3 \\ 0 & 0 & 1 \\ - & 1 & 0 \end{array} \wedge \begin{array}{ccc} x_1 & x_2 & x_3 \\ - & 1 & 1 \\ 0 & 0 & 1 \end{array} = \begin{array}{ccc} x_1 & x_2 & x_3 \\ 0 & 0 & 1 \end{array} \tag{3.8}$$

$$\begin{array}{cccc} x_1 & x_2 & x_3 & \\ 0 & 0 & 1 & \wedge \\ - & 1 & 0 & \end{array} \begin{array}{ccc} x_4 & x_5 & x_6 \\ - & 1 & 1 \\ 0 & 0 & 1 \end{array} = \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ 0 & 0 & 1 & - & - & - \\ - & 1 & 0 & - & - & - \end{array} \wedge \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ - & - & - & - & 1 & 1 \\ - & - & - & 0 & 0 & 1 \end{array}$$

$$= \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ 0 & 0 & 1 & - & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ - & 1 & 0 & - & 1 & 1 \\ - & 1 & 0 & 0 & 0 & 1 \end{array} \tag{3.9}$$

Bereits die Anwesenheit einer gemeinsamen Variable kann zur Reduzierung der Zeilenzahl der Ergebnis-TVL führen (siehe Formel (3.10)). Im allgemeinen gilt, dass mit steigender Anzahl gemeinsamer Variablen in den zu verknüpfenden Ternärvektorlisten immer kleinere

Ergebnis-TVL zu erwarten sind.

$$\begin{array}{ccc|c|ccc} x_1 & x_2 & x_3 & & x_3 & x_4 & x_5 \\ 0 & 0 & 1 & \wedge & - & 1 & 1 \\ - & 1 & 0 & & 0 & 0 & 1 \end{array} = \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \\ 0 & 0 & 1 & 1 & 1 \\ - & 1 & 0 & 1 & 1 \\ - & 1 & 0 & 0 & 1 \end{array} \tag{3.10}$$

Bei der Durchführung der Komposition ist es wichtig, die zu verknüpfenden Phasenlisten so zu wählen, dass die Durchschnittsbildung zur Reduzierung des Speicherplatzbedarfes führt. Für die Entscheidung, welche PHL zusammenzufassen sind, ist als Kriterium zu berücksichtigen, dass die zu verknüpfenden Knoten mindestens eine gemeinsame Variable enthalten. Da alle im Vorknoten-Nachknoten-Verhältnis zueinander stehende Knoten mindestens über eine gemeinsame Variable verfügen, ist dieses Kriterium für alle Vor- und Nachknoten erfüllt. D. h. ein Knoten sollte stets mit seinen Nachknoten verknüpft werden.

Für die Verknüpfung eines Knoten mit seinen Nachknoten spricht auch die Tatsache, dass einige der abhängigen Variablen nach dem Verknüpfen aus der Ergebnis-PHL entfernt werden können. Diese Aussage gilt aber nur für die Variablen, die außer in den zusammenzufassenden in keinen weiteren Phasenlisten vertreten sind. Zur Überprüfung, ob ein Entfernen der abhängigen Variable nach der Verknüpfung möglich ist, kann der schon im Abschitt 3.1.5 beschriebene Algorithmus (die ausführliche Beschreibung siehe Anhang A.2) herangezogen werden. Der Algorithmus sieht vor, dass für jede abhängige Variable die Häufigkeit ihres Auftretens als Eingangsvariable in allen Knoten der Funktion ermittelt wird. Nach dem Verknüpfen von zwei Phasenlisten wird für die gemeinsame Variable geprüft, wie oft sie noch als Eingangsvariable auftritt. Falls die Variable in keinen anderen Phasenlisten als Eingangsvariable auftritt, kann diese Variable entfernt werden. Da aber nicht alle Phasenlisten einer Funktion zusammengefasst werden, wird dieser Zustand für einige abhängige Variable nie erreicht. Die einfachere Möglichkeit das Auftreten einer Variable in den anderen Phasenlisten zu ermitteln, besteht in der Überprüfung, ob der zu verknüpfende Nachknoten auch ein Nachknoten anderer Knoten ist. Da bei der Darstellung einer Funktion durch den Tupel aus begrenzt zusammengefassten Phasenlisten die Knoten nur mit ihren Nachknoten verknüpft werden, genügt dieser einfachere und schnellere Algorithmus bereits den Anforderungen.

Zum Speichern von Phasenlisten kann die gleiche Datenstruktur “Knoten“ wie bei der Darstellung durch lokale Phasenlisten verwendet werden (siehe die Abbildung 3.2). Zusätzlich kann diese Datenstruktur weitere Informationen über Anzahl der Vor- und Nachknoten enthalten, um das schnellere Zusammenfassen von Phasenlisten zu ermöglichen.

Um die Größe der Phasenlisten-TVL zu begrenzen, werden die Knoten mit der großen Anzahl der Nachknoten in die Komposition nicht einbezogen.

Die mögliche Zusammenfassung von Knoten geht aus Abbildung 3.4 hervor. Die Abbildung enthält den Strukturgraph einer Booleschen Funktion, wobei die als gefüllte Kreise dargestellten Phasenlistenknoten, die zusammengefasst werden können, durch umschließende Rechtecke gekennzeichnet sind. Um die Phasenlisten-TVL in der Größe zu begrenzen, wurde der Knoten N mit keinen weiteren Knoten verknüpft. Im Beispiel wäre prinzipiell ein Verknüpfen des Knotens N mit einem seiner Nachknoten möglich. Der erforderliche Zeitaufwand für die Beurteilung dieser Möglichkeit wird aber durch die erreichbare geringfügige Verminderung des Speicherplatzbedarfs nicht gerechtfertigt.

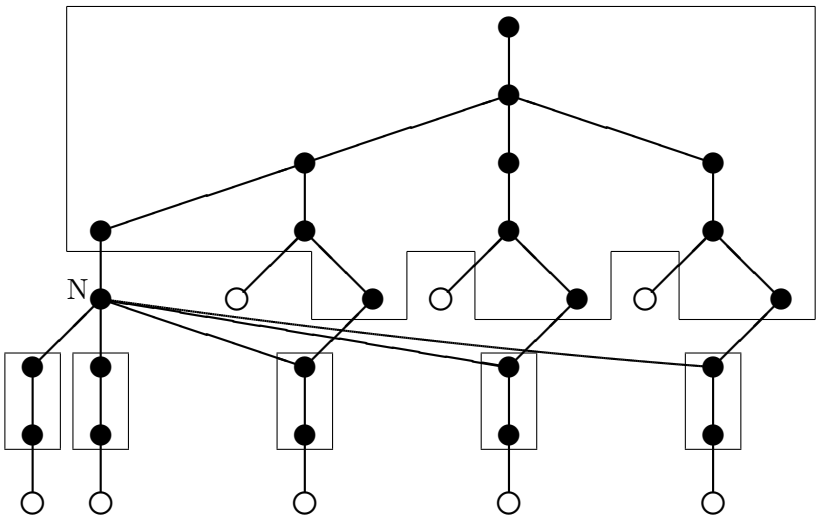


Abbildung 3.4: Erlaubte Komposition von Knoten

3.2 Ergebnisse der Darstellung einer Booleschen Funktion durch Phasenlisten

3.2.1 Tupel von lokalen Phasenlisten

Bei den Untersuchungen der Eignung des Tupels von lokalen Phasenlisten für die Darstellung einer Booleschen Funktion wurden als Testdaten die Benchmark-Schaltungen LGSynth91 [ATPG] verwendet. Die Funktionen einer Benchmark-Schaltung wurden wie im Abschnitt 3.1.4 beschrieben getrennt als Tupel aus mehreren lokalen PHL dargestellt. In der Untersuchungsreihe erfolgte die Darstellung aller Funktionen einer Benchmark-Schaltung in einem Booleschen Raum. Zum Vergleich wurden die gleichen Funktionen in verschiedenen Booleschen Räumen gespeichert.

Die Ergebnisse der Darstellung einer Booleschen Funktion durch den Tupel aus mehreren lokalen PHL sind in der Tabelle 3.1 zusammengefasst. In die vierte Spalte der Tabelle wurde die Anzahl der Elemente des PHL-Tupels (Summe aus PHL- und Variablenknoten) aller Funktionen der jeweiligen Benchmark-Schaltung aufgenommen. Die Werte der fünften und der sechsten Spalten der Tabelle beziehen sich auf den Speicherplatzbedarf (in Byte), der für die Darstellungen von Funktonen in einem Booleschen Raum (5. Spalte der Tabelle) und in verschiedenen Booleschen Räumen (6. Spalte) benötigt wird.

Die in den folgenden Tabellen für eine Benchmark-Schaltung aufgeführten Angaben wurden im Ergebnis folgender Schritte bereitgestellt:

- Der Speicherplatzbedarf und die Anzahl von Elementen des Tupels ergibt sich aus den Summen über alle Funktionen der Schaltung.

- Der Speicherplatzbedarf einer Funktion resultiert aus der Summe über alle Phasenlisten-TVL.

Tabelle 3.1: Ergebnisse der Darstellung durch lokale PHL

Benchmark	Eingänge	Ausgänge	Knoten	Byte	
				ER	VR
C432	36	7	958	136 980	27 196
C499	41	32	13 856	5 150 208	414 464
C880	60	26	1 366	138 932	34 172
C1355	41	32	11 616	3 991 808	370 432
C1908	33	25	13 369	6 069 576	394 944
C2670	233	140	7 179	3 304 848	195 784
C5315	178	123	18 687	6 594 396	533 828
C6288	32	32	77 483	-	2 448 284
C7552	207	108	31 909	19 869 556	902 588

- Schließlich wird für eine Phasenlisten-TVL der Speicherplatzbedarf wie folgt ermittelt:
  $(NTV(TVL)+1) \cdot GET\_TYP(TVL) \cdot 4Byte + 4Byte$, wobei
  $NTV(TVL)+1$ - die Anzahl der TV in der TVL und
  $GET\_TYP(TVL)$ - die Anzahl der zur Abspeicherung eines TV benötigten Maschinenwörter ist.

Wie aus den Messergebnissen der Tabelle ersichtlich, wird der Speicherplatzbedarf durch die Darstellung in verschieden Räumen wesentlich reduziert. Die Größe der Einsparung steigt dabei mit zunehmender Größe die Booleschen Funktionen. Während die Reduzierung bei einer kleineren Schaltungen den Faktor 4 annahm, konnte bei der Benchmark-Schaltung C7552 eine Reduzierung auf den 22-ten Teil erreicht werden. Bei der Benchmark-Schaltung C6288 übersteigt die gesamte Anzahl der Variablen (der unabhängigen und abhängigen Eingangsvariablen aller Funktionen) die Beschränkung der Variablenzahl der XBOOLE-Bibliothek, so dass hier auf die Darstellung in einem Booleschen Raum verzichtet wurde.

3.2.2 Globale Phasenliste

Aus früheren Untersuchungen zu Möglichkeiten der Darstellung einer Booleschen Funktionen durch eine Ternärvektorliste (siehe [KeSt94] und [Ste95]) ist bekannt, dass die Abbildung von extrem großen Funktionen als TVL eher ungünstig ist. Aus diesem Grund wurde auf die Darstellung aller Booleschen Funktionen durch globale PHL verzichtet.

Die Benchmark-Schaltungen enthalten mehrere Funktionen unterschiedlicher Größe. Eine Zusammenfassung der lokalen zu globalen PHL wurde wie im Abschnitt 3.1.5 beschrieben für alle die Funktionen einer Benchmark-Schaltung vorgenommen, wo kein Überschreiten der kritische Anzahl der PHL-Knoten auftritt. Andernfalls erfolgt eine Darstellung mittels lokaler PHL (Abschnitt 3.1.4). Als Grenzwerte für die kritische Knotenzahl wurden innerhalb der Untersuchungen die Werte 10, 20, 50, 60 und 80 gewählt.

In Tabelle 3.2 wurde der benötigte Speicherplatzbedarf in Byte für die Funktionen der Benchmark-Schaltungen C880 und C5315 aufgeführt. Die Angaben der Spalte 1 wurden aus der Tabelle 3.1 übernommen und beinhalten den Speicherplatzbedarf für die Funktionen entsprechender Schaltungen bei der Darstellung durch Tupel aus lokalen PHL.

Tabelle 3.2: Ergebnisse der Zusammenfassung ausgewählter Funktionen zu globalen PHL

Benchmark-Schaltung	kritische Knotenzahl					
	1	10	20	50	60	80
C880	138 932	138 000	138 000	166 408	166 408	-
C5315	6 594 396	6 592 944	6 591 632	6 590 940	6 620 484	7 234 544

Wie aus der Tabelle 3.2 ersichtlich, führt die Abbildung von Funktionen mit einer Knotenanzahl bis 20 bzw. 50 durch eine globale PHL zu verbesserten Ergebnissen. Bei der Darstellung von größeren Funktionen durch eine globale PHL musste dagegen eine Verschlechterung der Ergebnisse beobachtet werden. Auf die weitere Erhöhung der kritischen Knotenzahl und damit verbundene Abbildung der noch größeren Funktionen durch eine globale PHL wurde verzichtet, da die Entwicklungstendenz bereits aus den Werten der Tabelle 3.2 zu erkennen ist. Darüber hinaus wäre die Berechnung der TVL von globalen PHL dieser Größenordnung zeitaufwendig.

3.2.3 Tupel von zusammengefassten Phasenlisten

Zur Beantwortung der Frage, in wieweit die Komposition mehrerer lokaler PHL zu einer PHL den Speicherplatzbedarf einer Funktion beeinflusst, wurden die Durchschnitte über die PHL-Knoten zweier benachbarten Ebenen (Ebenen 1 und 2, 3 und 4, 5 und 6, usw. bis zum Wurzelknoten) wie im Abschnitt 3.1.6 beschrieben gebildet.

Der Speicherplatzbedarf der untersuchten Funktionen kann der Tabelle 3.3 Zeile 2 entnommen werden, wobei alle Speicherplatzangaben in der Tabelle in KByte erfolgen. Zum Vergleich enthält die Zeile 1 (Tabelle 3.3) die Werte für die Funktionen, die durch einen Tupel aus nichtzusammengefassten lokalen PHL dargestellt wurden. Bei allen Benchmark-Schaltungen ist eine Verbesserung der Ergebnisse in Zeile 2 gegenüber der Zeile 1 zu beobachten.

Im Rahmen weiterer Untersuchungen wurde das oben beschriebene Verfahren des Zusammenfassens der Knoten auf über mehrere (3, 4, ... ,12) Ebenen ausgedehnt. Die Zeilen 3 bis 12 der Tabelle beinhalten die Ergebnisse dieser Messungen. In den letzten Spalten der Tabelle sind die Summen über den gesamten Schaltungssatz und die Reduzierung des Speicherplatzbedarfs gegenüber der Zeile 1 in % dargestellt. Folgende tendenziellen Aussagen sind herauszustellen:

- Das Zusammenfassen führt zunächst (über 2-6 Ebenen) zu einer Verminderung des Speicherplatzbedarfs der Funktionen aller Schaltungen. Bezüglich des Minimums besteht eine Abhängigkeit zu speziellen Schaltungen.

- Bei einer Komposition über 7-12 Ebenen ist die Umkehrung Tendenz zu beobachten. Ein Zusammenfassen über 7-11 Ebenen verschlechtert das Ergebnis nur geringfügig (außer bei der Schaltung C1355). In der Regel benötigen die Funktionen immer noch weniger Speicherplatz als für den Tupel aus nicht zusammengefassten PHL. Dagegen wird beim Zusammenfassen über 12 Ebenen für die Funktionen einiger Benchmark-Schaltungen der vielfache Speicherplatz im Vergleich zu den minimalen Werten benötigt.

Tabelle 3.3: Speicherplatzbedarf der durch einen Tupel aus über n Ebenen zusammengefassten PHL dargestellten Funktionen

n	C432 KByte	C499 KByte	C880 KByte	C1355 KByte	C1908 KByte	C2670 KByte	C5315 KByte	C7552 KByte	Summe KByte	Reduz. %
1	137	5 150	139	3 992	6 070	3 309	6 594	19 870	45 256	
2	112	3 348	105	2 643	3 885	2 237	4 927	13 553	30 809	31,92
3	115	3 177	98	1 491	2 956	1 931	4 400	10 428	24 596	45,65
4	106	2 056	104	1 926	2 574	1 547	3 473	9 403	21 188	53,18
5	91	2 334	99	1 988	2 506	1 441	3 383	8 637	20 479	54,75
6	78	2 925	120	1 494	1 852	1 462	3 968	7 491	19 391	57,15
7	79	2 488	136	4 246	1 949	1 834	3 623	8 392	22 747	49,74
8	81	2 820	588	4 045	1 975	1 644	4 095	9 320	24 568	45,71
9	88	4 243	823	3 871	1 997	1 563	4 349	9 139	26 071	42,39
10	63	4 278	1 027	9 574	2 158	1 636	4 970	9 034	32 742	27,65
11	63	4 392	1 424	9 658	1 763	1 495	5 806	7 296	31 898	29,52
12	64	9 887	1 631	9 473	1 768	1 676	15 195	12 006	51 700	-14,24

- Die für die einzelnen Schaltungen beschriebenen Tendenzen zeichnen sich auch in den Werten für den gesamten Schaltungssatz ab. Die besten Ergebnisse sind bei einer Zusammenfassung über 6 Ebenen zu beobachten. Die Reduzierung des Speicherplatzbedarfs beträgt in diesem Fall 57,15%.

Die Untersuchungen mit lokalen Phasenlisten haben gezeigt, dass die Darstellung von lokalen PHL in verschiedenen Booleschen Räumen günstiger ist als die Darstellung des gesamten PHL-Tupels in einem Booleschen Raum. Aus diesem Grund wurde im weiteren untersucht, inwieweit das Zusammenfassen von lokalen Phasenlisten auch bei der Darstellung in verschiedenen Räumen zu der gewünschten Reduzierung führt. Die Ergebnisse der Untersuchungen sind in der Tabelle 3.4 dargestellt. Aus der letzten Spalte der Tabelle ist zu ersehen, dass die Reduzierung des Speicherplatzbedarfs bei einer Darstellung durch den Tupel aus den begrenzt zusammengefassten Phasenlisten gegenüber einer Darstellung aus nicht zusammengefassten Phasenlisten in verschiedenen Booleschen Räumen 56,88% beträgt und damit sehr effektiv ist.

3.2.4 Zusammenfassung der Ergebnisse

In den Abschnitten 3.2.1 und 3.2.3 wurde gezeigt, dass die Darstellung einer Booleschen Funktion durch einen Tupel aus lokalen PHL prinzipiell möglich ist und im allgemeinen gegenüber einer Darstellung mit einer globalen PHL bevorzugt werden sollte. Dieser prinzipiellen Aussage stehen nur Funktionen entgegen, die bei der Darstellung durch einen Tupel aus lokalen Phasenlisten die geringe Anzahl der Elementen enthalten. Hier führt der Einsatz globaler PHL zu unbedeutend günstigeren Ergebnissen (siehe Abschitt 3.2.2).

Die Abbildung der Funktionen durch einen Tupel von lokalen Phasenlisten kann weiter verbessert werden. Möglichkeiten sind erstens die Darstellung von einzelnen Phasenlisten in verschiedenen Booleschen Räumen und zweitens die Optimierung der Größe der Phasenlisten-TVL.

Tabelle 3.4: Speicherplatzbedarf der Funktionen, dargestellt durch einen Tupel aus über n Ebenen zusammengefassten und in verschiedenen Räumen dargestellten PHL

n	C432 KByte	C499 KByte	C880 KByte	C1355 KByte	C1908 KByte	C2670 KByte	C5315 KByte	C7552 KByte	Summe KByte	Reduz. %
1	27	414	34	370	395	196	534	903	2 873	
2	22	263	25	241	250	133	392	618	1 944	32,36
3	22	249	23	136	187	114	348	466	1 546	46,21
4	21	159	24	170	162	90	268	420	1 313	54,31
5	18	180	23	176	157	84	268	384	1 288	55,17
6	15	219	27	131	115	85	323	324	1 239	56,88
7	15	185	31	592	120	105	286	339	1 672	41,80
8	15	208	235	574	121	105	321	410	1 989	30,80
9	17	441	323	558	122	98	407	398	2 364	17,72
10	12	442	403	1 492	131	106	439	400	3 424	-19,17

Tabelle 3.5: Vergleich des Speicherplatzbedarfs Funktionen der LGSynth91-Benchmark-Schaltungen, dargestellt durch verschiedene PHL-Tupel

	ER	VR	Reduzierung
lokale PHL	45 256	2 873	15,75-fach
zusammengefasste lokale PHL	19 391	1 239	15,65-fach
Reduzierung	2,33-fach	2,31-fach	36,53-fach

Die Tabelle 3.5 zeigt zusammenfassend, wie die Ergebnisse der Darstellung durch einen PHL-Tupel verbessert werden können. Die Werte in der Tabelle beziehen sich auf den Speicherplatz in KByte, der für die Darstellung der Funktionen des gesamten LGSynth91-Benchmark-Set benötigt wird. In der zweiten und dritten Spalte der Tabelle sind die Ergebnisse der Darstellung in einen Booleschen Raum und in verschiedenen Booleschen Räumen aufgenommen. Die zweite und dritte Zeile zeigen die Ergebnisse der Darstellungen durch ein Tupel aus den nicht optimierten und durch das Zusammenfassen optimierten Phasenlisten. Es ist ersichtlich, dass der größte Reduzierungseffekt schon durch das Aufteilen von Elementen des Tupels auf verschiedene Boolesche Räume erreicht werden kann. Weitere wesentliche Reduzierung erfolgt durch die Optimierung der Größe von Phasenlisten-TVL. Die erreichte Reduzierung beträgt fast das 37-fache.

Kapitel 4

Tupel-Dekomposition

4.1 Zerlegung in drei Funktionen

4.1.1 Zerlegungsprinzip

Die Dekomposition einer Booleschen Funktion $f(\underline{x_0}, x_i)$ entsprechend (4.1) sieht vor, dass im Ergebnis der Zerlegung die Teilfunktionen $\underline{h}(\underline{x_0})$, nur von den Variablen $\underline{x_0}$ und nicht von der Variable x_i abhängen.

$$f(\underline{x_0}, x_i) = g(\underline{h}(\underline{x_0}), x_i) \tag{4.1}$$

Der Vektor $\underline{h}(\underline{x_0})$ kann eine unterschiedliche Anzahl Boolescher Funktionen enthalten. Es besteht die Möglichkeit, dass nur eine einzige Boolesche Funktion $h(\underline{x_0})$ in den Vektor $\underline{h}(\underline{x_0})$ aufgenommen wird (4.2). Die Dekomposition (4.2) ist nicht für alle Boolesche Funktionen möglich.

$$f(\underline{x_0}, x_i) = g(h(\underline{x_0}), x_i) \tag{4.2}$$

Satz 4.1. *Es gibt mindestens eine Boolesche Funktion, für die die Dekomposition (4.2) nicht möglich ist.*

Beweis:

Gegeben ist die Funktion $f(\underline{x_0}, x_i)$, die nach (4.3) definiert ist.

$$f(x_{01}, x_{02}, x_i) = \bar{x_i} x_{01} x_{02} \vee x_i (x_{01} \vee x_{02}) \tag{4.3}$$

Setzt man in der Funktion $g(h(\underline{x_0}), x_i)$ die Variable x_i auf einen konstanten Wert, z. B. $x_i = 0$, so können die vier Funktionen (4.4)-(4.7) als Funktion $g'(h(\underline{x_0})) = g(h(\underline{x_0}), 0)$ betrachtet werden.

$$g'_1(h(\underline{x_0})) = 1 \tag{4.4}$$
$$g'_2(h(\underline{x_0})) = 0 \tag{4.5}$$
$$g'_3(h(\underline{x_0})) = h(\underline{x_0}) \tag{4.6}$$
$$g'_4(h(\underline{x_0})) = \overline{h(\underline{x_0})} \tag{4.7}$$

Die gleichen Funktionen werden für die Funktion $g'(h(\underline{x_0})) = g(h(\underline{x_0}), 1)$ erhalten.

Wird die Variable $x_i = 0$ in die Funktion (4.3) eingesetzt, ergibt sich für diese Funktion (4.8).

$$f(\underline{x_0}, 0) = x_{01} x_{02} \tag{4.8}$$

Die Funktion $x_{01}x_{02}$ ist ungleich 0 und ungleich 1, damit sind nur noch die Fälle (4.9)-(4.10) möglich.

$$h_1(\underline{x_0}) = x_{01}x_{02} \tag{4.9}$$
$$\overline{h_2(\underline{x_0})} = x_{01}x_{02} \tag{4.10}$$

Nach der Zuweisung $x_i = 1$ in (4.3), ergibt sich für die Funktion (4.3) der Ausdruck (4.11).

$$f(\underline{x_0}, 0) = x_{01} \vee x_{02} \tag{4.11}$$

Auch hier sind nur die Fälle (4.12)-(4.13) weiter zu betrachten.

$$h_3(\underline{x_0}) = x_{01} \vee x_{02} \tag{4.12}$$
$$\overline{h_4(\underline{x_0})} = x_{01} \vee x_{02} \tag{4.13}$$

Trifft eine der Aussagen (4.14)-(4.17) zu, existiert auch die Funktion $h(\underline{x_0})$, die für die Funktion (4.3) die Dekomposition (4.2) ermöglicht.

$$h_1(\underline{x_0}) = h_3(\underline{x_0}) \tag{4.14}$$
$$h_1(\underline{x_0}) = h_4(\underline{x_0}) \tag{4.15}$$
$$h_2(\underline{x_0}) = h_3(\underline{x_0}) \tag{4.16}$$
$$h_2(\underline{x_0}) = h_4(\underline{x_0}) \tag{4.17}$$

Alle Aussagen (4.14)-(4.17) führen dagegen zum Widerspruch (siehe (4.9)-(4.10) und (4.12)-(4.13)), damit existiert auch keine Funktion $h(\underline{x_0})$, die für die Funktion (4.3) die Dekomposition (4.2) ermöglicht. □

Offenbar genügt eine Funktion $h(\underline{x_0})$ im Funktionsvektor $\underline{h}(\underline{x_0})$ (siehe (4.1)) nicht, um jede Boolesche Funktion zu beschreiben. Besteht der Funktionsvektor $\underline{h}(\underline{x_0})$ aus zwei Funktionen, so können diese so gewählt werden, dass eine beliebige Funktion $f(\underline{x_0}, x_i)$ konstruiert werden kann.

Satz 4.2. *Jede Boolesche Funktion kann nach (4.18) zerlegt werden.*

$$f(\underline{x_0}, x_i) = g(h_1(\underline{x_0}), h_2(\underline{x_0}), x_i) \tag{4.18}$$

Die im Kapitel 2 beschriebenen Shannon- und Davio-Dekompositionen entsprechen der Zerlegung einer Booleschen Funktion nach (4.18). Eine weitere Zerlegung jeder der beiden Funktionen $h_1(\underline{x_0})$ und $h_2(\underline{x_0})$ ist möglich. Damit gilt allgemein, dass eine Boolesche Funktion durch einen Tupel von Funktionen nach (4.1) dargestellt werden kann.

Im weiteren wird die Darstellung einer Booleschen Funktionen durch drei Teilfunktionen (4.19) betrachtet.

$$f(\underline{x_0}, x_i) = g(h_1(\underline{x_0}), h_2(\underline{x_0}), h_3(\underline{x_0}), x_i) \tag{4.19}$$

Die drei Teilfunktionen können wie folgt definiert werden:

Definition 4.1. *Die Funktion $f^-(\underline{x_0})$ (4.20) ist die Konjunktion der Funktion $f(x_i, \underline{x_0})$ und der komplementären partiellen Ableitung dieser Funktion nach der Variable x_i.*

$$f^-(\underline{x_0}) = \overline{\frac{\partial f(x_i, \underline{x_0})}{\partial x_i}} f(x_i, \underline{x_0}) \tag{4.20}$$

Definition 4.2. *Die Funktion $f^0(\underline{x_0})$ (4.21) ist die Konjunktion der Funktion $f(x_i, \underline{x_0})$ an der Stelle $x_i = 0$ und der partiellen Ableitung dieser Funktion nach der Variable x_i.*

$$f^0(\underline{x_0}) = f(x_i = 0, \underline{x_0}) \frac{\partial f(x_i, \underline{x_0})}{\partial x_i} \tag{4.21}$$

Definition 4.3. *Die Funktion $f^1(\underline{x_0})$ (4.22) ist die Konjunktion der Funktion $f(x_i, \underline{x_0})$ an der Stelle $x_i = 1$ und der partiellen Ableitung dieser Funktion nach der Variable x_i.*

$$f^1(\underline{x_0}) = f(x_i = 1, \underline{x_0}) \frac{\partial f(x_i, \underline{x_0})}{\partial x_i} \tag{4.22}$$

Die Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ sind von der Variable x_i unabhängig. Die Ableitung einer Funktion nach einer Variable ist genau dann gleich 1, wenn eine Änderung des Werts dieser Variable die Änderung des Funktionswerts bewirkt. Das Komplement der Ableitung ist genau dann gleich 1, wenn die Änderung einer Variable keine Veränderung des Funktionswerts hervorruft. Durch konjunktive Verknüpfung der Funktion mit ihrer komplementärer Ableitung nach Variable x_i erhält man somit die von x_i unabhängige Funktion $f^-(\underline{x_0})$. Die Funktionen $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ sind von x_i unabhängig, da sowohl die Funktion $f(x_i = c, \underline{x_0})$ als auch die Ableitung der Funktion $f(x_i, \underline{x_0})$ nach x_i von x_i unabhängig sind.

Die Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ können als Funktionen $h_1(\underline{x_0})$, $h_2(\underline{x_0})$ und $h_3(\underline{x_0})$ in der Darstellung (4.19) verwendet werden, wobei für die Zuordnung die Gleichungen (4.23) genutzt werden.

$$\begin{aligned} h_1(\underline{x_0}) &= f^-(\underline{x_0}) \\ h_2(\underline{x_0}) &= f^0(\underline{x_0}) \\ h_3(\underline{x_0}) &= f^1(\underline{x_0}) \end{aligned} \tag{4.23}$$

Die Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ können durch die Cofaktoren der Shannon-Dekomposition (2.30) (siehe Abschnitt 2.1.4) nach (4.24)-(4.26) ausgedrückt werden. Im Mittelpunkt der Tupel-Dekomposition steht jedoch das dynamische Verhalten einer Booleschen Funktion. Die Funktion $f^-(\underline{x_0})$ besitzt für solche Belegungen $\underline{x_0}$ den Funktionswert 1, für die der Funktionswert 1 der Funktion $f(x_i, \underline{x_0})$ bei der Änderung des Wertes von x_i erhalten bleibt. Dagegen enthalten die Funktionen $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ die Teile der Funktion $f(x_i, \underline{x_0})$, die durch die Änderung des Variablenwertes den Funktionswert ändern.

$$f^-(\underline{x_0}) = f(x_i = 0, \underline{x_0}) \cdot f(x_i = 1, \underline{x_0}) \tag{4.24}$$

$$f^0(\underline{x_0}) = f(x_i = 0, \underline{x_0}) \cdot \overline{f(x_i = 1, \underline{x_0})} \tag{4.25}$$

$$f^1(\underline{x_0}) = \overline{f(x_i = 0, \underline{x_0})} \cdot f(x_i = 1, \underline{x_0}) \tag{4.26}$$

Satz 4.3. *Eine beliebige Boolesche Funktion $f(x_i, \underline{x_0})$ kann durch die drei Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ nach (4.27) dargestellt werden.*

$$f(x_i, \underline{x_0}) = f^-(\underline{x_0}) \vee \overline{x_i} f^0(\underline{x_0}) \vee x_i f^1(\underline{x_0}) \tag{4.27}$$

Beweis:

Nach dem Einsetzen der Definitionen der Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ (4.20-4.22) in (4.27) erhält man (4.28).

$$\begin{aligned} f(x_i,\underline{x_0}) &= \overline{\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}}\cdot f(x_i,\underline{x_0}) \\ &\quad \vee\overline{x_i}\cdot f(x_i=0,\underline{x_0})\cdot\frac{\partial f(x_i,\underline{x_0})}{\partial x_i} \\ &\quad \vee x_i\cdot f(x_i=1,\underline{x_0})\cdot\frac{\partial f(x_i,\underline{x_0})}{\partial x_i} \qquad (4.28) \\ &= \overline{\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}}\cdot f(x_i,\underline{x_0}) \\ &\quad \vee(\overline{x_i}\cdot f(x_i=0,\underline{x_0})\vee x_i\cdot f(x_i=1,\underline{x_0}))\cdot\frac{\partial f(x_i,\underline{x_0})}{\partial x_i} \qquad (4.29) \end{aligned}$$

Unter Anwendung der Shannon-Dekomposition (4.30) kann (4.29) in (4.31) überführt werden.

$$\overline{x_i}f(x_i=0,\underline{x_0})\vee x_i f(x_i=1,\underline{x_0}) = f(x_i,\underline{x_0}) \qquad (4.30)$$

$$\begin{aligned} f(x_i,\underline{x_0}) &= \overline{\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}}\cdot f(x_i,\underline{x_0})\vee f(x_i,\underline{x_0})\cdot\frac{\partial f(x_i,\underline{x_0})}{\partial x_i} \qquad (4.31) \\ &= \left(\overline{\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}}\vee\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}\right)\cdot f(x_i,\underline{x_0}) \\ &= 1\cdot f(x_i,\underline{x_0}) \qquad (4.32) \end{aligned}$$

□

Satz 4.4. *Die Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ sind zueinander paarweise disjunkt:*

$$f^\alpha(\underline{x_0})\cdot f^\beta(\underline{x_0}) = 0, \quad \textit{für} \quad \alpha,\beta\in\{-,0,1\} \quad \textit{und} \quad \alpha\neq\beta \qquad (4.33)$$

Beweis:

1. $$f^-(\underline{x_0})\cdot f^0(\underline{x_0}) = 0 \qquad (4.34)$$

   Nach dem Einsetzen der Definitionen der Funktionen $f^-(\underline{x_0})$ und $f^0(\underline{x_0})$ (siehe (4.20) und (4.21)) in (4.34) ergibt sich (4.35).

   $$\overline{\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}}f(x_i,\underline{x_0})\cdot f(x_i=0,\underline{x_0})\frac{\partial f(x_i,\underline{x_0})}{\partial x_i} = 0 \qquad (4.35)$$

   $$\overline{\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}}\cdot\frac{\partial f(x_i,\underline{x_0})}{\partial x_i} = 0 \qquad (4.36)$$

   $$0\cdot f(x_i,\underline{x_0})\cdot f(x_i=0,\underline{x_0}) = 0 \qquad (4.37)$$

   Auf Grund von (4.36) folgt (4.37) aus (4.35) .

   $$0 = 0 \qquad (4.38)$$

2.

$$f^-(\underline{x_0}) \cdot f^1(\underline{x_0}) = 0 \tag{4.39}$$

Nach dem Einsetzen der Definitionen der Funktionen $f^-(\underline{x_0})$ und $f^1(\underline{x_0})$ (siehe (4.20) und (4.22)) in (4.39) folgt (4.40).

$$\overline{\frac{\partial f(x_i, \underline{x_0})}{\partial x_i}} f(x_i, \underline{x_0}) \cdot f(x_i = 1, \underline{x_0}) \frac{\partial f(x_i, \underline{x_0})}{\partial x_i} = 0 \tag{4.40}$$

Wie (4.35) enthält auch (4.40) zueinander komplementäre Ableitungen, so dass dieser Teil des Satzes in Analogie zu Fall (4.34) bewiesen werden kann.

3.

$$f^0(\underline{x_0}) \cdot f^1(\underline{x_0}) = 0 \tag{4.41}$$

Nach dem Einsetzen der Definitionen der Funktionen $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ (siehe (4.21) und (4.22)) in (4.41) ergibt sich (4.42).

$$f(x_i = 0, \underline{x_0}) \frac{\partial f(x_i, \underline{x_0})}{\partial x_i} \cdot f(x_i = 1, \underline{x_0}) \frac{\partial f(x_i, \underline{x_0})}{\partial x_i} = 0 \tag{4.42}$$

$$f(x_i = 0, \underline{x_0}) \cdot f(x_i = 1, \underline{x_0}) \cdot \frac{\partial f(x_i, \underline{x_0})}{\partial x_i} = 0 \tag{4.43}$$

Das Einsetzen der Ableitungsdefinition in (4.43) führt zu (4.44)

$$f(x_i = 0, \underline{x_0}) \cdot f(x_i = 1, \underline{x_0}) \cdot (f(x_i = 0, \underline{x_0}) \oplus f(x_i = 1, \underline{x_0})) = 0 \tag{4.44}$$

$$f(x_i = 0, \underline{x_0}) f(x_i = 1, \underline{x_0}) \oplus f(x_i = 0, \underline{x_0}) f(x_i = 1, \underline{x_0}) = 0 \tag{4.45}$$

$$0 = 0 \tag{4.46}$$

□

Durch die konjunktive Verknüpfung der Funktion $f^0(\underline{x_0})$ mit $\bar{x_i}$ und der Funktion $f^1(\underline{x_0})$ mit x_i bleibt die Orthogonalität der disjunktiv verknüpften Funktionen in (4.27) erhalten. Nach dem Orthogonalitätssatz kann für die orthogonalen Funktionen $f^-(\underline{x_0})$, $\bar{x_i} f^0(\underline{x_0})$ und $x_i f^1(\underline{x_0})$ in (4.27) die disjunktive Verknüpfung durch die Antivalenzverknüpfung ersetzt werden. Somit gilt auch (4.47).

$$f(x_i, \underline{x_0}) = f^-(\underline{x_0}) \oplus \overline{x_i} f^0(\underline{x_0}) \oplus x_i f^1(\underline{x_0}) \tag{4.47}$$

Zum Beispiel entstehen beim Zerlegen der Funktion (4.48) nach Variable x_1 die Teilfunktionen (4.49) (siehe auch die Abbildung 4.1).

$$f(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_3 x_4 \vee x_3 \bar{x}_4 \tag{4.48}$$

$$\begin{array}{l} f^-(x_2, x_3, x_4) = x_3 \bar{x}_4 \\ f^0(x_2, x_3, x_4) = \bar{x}_2 \bar{x}_3 \\ f^1(x_2, x_3, x_4) = x_3 x_4 \end{array} \tag{4.49}$$

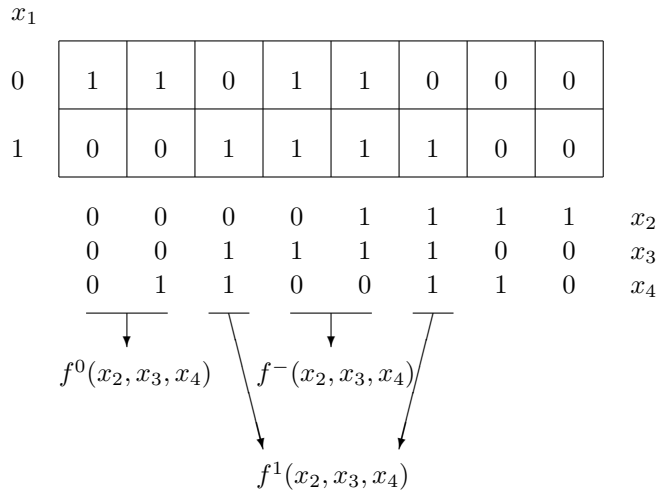


Abbildung 4.1: Zerlegung der Funktion $f(x_1, x_2, x_3, x_4)$ in drei Teilfunktionen

4.1.2 Operationen

Negation

Satz 4.5. *Für die gegebene Funktion f_1 mit jeweils drei Teilfunktionen ${f_1}^-$, ${f_1}^0$, ${f_1}^1$ (4.27) können die Teilfunktionen ${f_3}^-$, ${f_3}^0$, ${f_3}^1$ der negierten Funktion $f_3 = \bar{f}_1$ nach (4.50)-(4.52) berechnet werden.*

$$ {f_3}^- = \bar{{f_1}^-}\,\bar{{f_1}^0}\,\bar{{f_1}^1} \tag{4.50} $$

$$ {f_3}^0 = {f_1}^1 \tag{4.51} $$

$$ {f_3}^1 = {f_1}^0 \tag{4.52} $$

Beweis:

Wird die nach (4.27) dargestellte Funktion f_1 unter Annahme $x = x_i$ in (4.53) eingesetzt, entsteht (4.54).

$$ \begin{aligned} f_3 &= \bar{f}_1 && (4.53)\\ &= \overline{{f_1}^- \vee \bar{x}{f_1}^0 \vee x{f_1}^1} && (4.54)\\ &= \bar{{f_1}^-} \cdot (x \vee \bar{{f_1}^0}) \cdot (\bar{x} \vee \bar{{f_1}^1}) && (4.55)\\ &= \bar{{f_1}^-} \cdot (x\bar{x} \vee x\bar{{f_1}^1} \vee \bar{x}\bar{{f_1}^0} \vee \bar{{f_1}^0}\bar{{f_1}^1}) && (4.56)\\ &= \bar{{f_1}^-}\,\bar{{f_1}^0}\bar{{f_1}^1} \vee x\bar{{f_1}^-}\,\bar{{f_1}^1} \vee \bar{x}\bar{{f_1}^-}\,\bar{{f_1}^0} && (4.57) \end{aligned} $$

Unter Verwendung der Sätze von de Morgan kann (4.54) in (4.55) überführt werden. Die weiteren Umwandlungen führen zu (4.57).

Der Ausdruck $\bar{{f_1}^-}\bar{{f_1}^1}$ kann durch die elementaren Umwandlungen (4.59)-(4.62) unter Berücksichtigung der Orthogonalität der Funktionen ${f_1}^-$, ${f_1}^0$ und ${f_1}^1$ (4.58) in (4.62) nach (4.63) überführt werden.

$$ 0 = {f_1}^- {f_1}^0 = {f_1}^- {f_1}^1 = {f_1}^0 {f_1}^1 \tag{4.58} $$

$$\begin{aligned}
\bar{f_1^-}\bar{f_1^1} &= (f_1^0 \vee \bar{f_1^0}) \cdot \bar{f_1^-}\bar{f_1^1} && (4.59)\\
&= f_1^0\bar{f_1^-}\bar{f_1^1} \vee \bar{f_1^0}\bar{f_1^-}\bar{f_1^1} && (4.60)\\
&= f_1^0(1 \oplus f_1^-)(1 \oplus f_1^1) \vee \bar{f_1^-}\bar{f_1^0}\bar{f_1^1} && (4.61)\\
&= (f_1^0 \oplus f_1^0 f_1^- \oplus f_1^0 f_1^1 \oplus f_1^0 f_1^- f_1^1) \vee \bar{f_1^-}\bar{f_1^0}\bar{f_1^1} && (4.62)\\
&= f_1^0 \vee \bar{f_1^-}\bar{f_1^0}\bar{f_1^1} && (4.63)
\end{aligned}$$

Analog gilt auch (4.64).

$$\bar{f_1^-}\bar{f_1^0} = f_1^1 \vee \bar{f_1^-}\bar{f_1^0}\bar{f_1^1} \quad (4.64)$$

Nach dem Einsetzen von (4.63) und (4.64) in (4.57) wird (4.65) erhalten.

$$\begin{aligned}
\bar{f}_1 &= \bar{f_1^-}\bar{f_1^0}\bar{f_1^1} \vee x(f_1^0 \vee \bar{f_1^-}\bar{f_1^0}\bar{f_1^1}) \vee \bar{x}(f_1^1 \vee \bar{f_1^-}\bar{f_1^0}\bar{f_1^1}) && (4.65)\\
\bar{f}_1 &= \bar{f_1^-}\bar{f_1^0}\bar{f_1^1} \vee \bar{x}f_1^1 \vee xf_1^0 && (4.66)
\end{aligned}$$

Aus (4.66) gehen wegen (4.53) und (4.27) die Teilfunktionen (4.50), (4.51) und (4.52) der negierten Funktion $f_3 = \bar{f}_1$ unmittelbar hervor. □

Satz 4.6. [1] *Die nach (4.50), (4.51) und (4.52) berechneten Teilfunktionen f_3^-, f_3^0 und f_3^1 (4.27) der Funktion $f_3 = \bar{f}_1$ sind zueinander paarweise disjunkt:*

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \textit{für} \quad \alpha, \beta \in \{-, 0, 1\} \quad \textit{und} \quad \alpha \neq \beta \quad (4.67)$$

Im folgenden Beispiel werden für die Funktion (4.68) mit den Teilfunktionen (4.69) die Teilfunktionen der Funktion $f_3(\underline{x}) = \overline{f_1(\underline{x})}$ mit (4.70) beschrieben.

$$f_1(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1x_3x_4 \vee x_3\bar{x}_4 \quad (4.68)$$

$$\begin{aligned}
f_1^-(x_2, x_3, x_4) &= x_3\bar{x}_4\\
f_1^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3\\
f_1^1(x_2, x_3, x_4) &= x_3x_4
\end{aligned} \quad (4.69)$$

$$\begin{aligned}
f_3^-(x_2, x_3, x_4) &= x_2\bar{x}_3\\
f_3^0(x_2, x_3, x_4) &= x_3x_4\\
f_3^1(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3
\end{aligned} \quad (4.70)$$

Die Funktion $f_1(\underline{x})$ (4.68) mit den Teilfunktionen (4.69) und die Funktion $f_3(\underline{x})$ mit den Teilfunktionen (4.70) sind in der Abbildung 4.2 dargestellt. Es ist ersichtlich, dass die Funktion $f_3^0(\underline{x})$ sich aus der Funktion $f_1^1(\underline{x})$ ergibt, und die Funktion $f_3^1(\underline{x})$ der Funktion $f_1^0(\underline{x})$ entspricht. Die Funktion $f_3^-(\underline{x})$ enthält die Konjunktion, die in keiner der Funktionen $f_1^-(\underline{x})$, $f_1^0(\underline{x})$ oder $f_1^1(\underline{x})$ enthalten sind, und kann als $\overline{f_1^-(\underline{x})}\ \overline{f_1^0(\underline{x})}\ \overline{f_1^1(\underline{x})}$ berechnet werden.

Disjunktion

Satz 4.7. *Für die gegebenen Funktionen f_1 und f_2 mit jeweils drei Teilfunktionen ${f_1}^-$, ${f_1}^0$, ${f_1}^1$ und ${f_2}^-$, ${f_2}^0$, ${f_2}^1$ (4.27) können die Teilfunktionen ${f_3}^-$, ${f_3}^0$, ${f_3}^1$ der Funktion $f_3 = f_1 \vee f_2$ nach (4.71)-(4.73) berechnet werden.*

$${f_3}^- = {f_1}^- \vee {f_2}^- \vee {f_1}^0{f_2}^1 \vee {f_1}^1{f_2}^0 \quad (4.71)$$

[1] Beweis siehe Anhang B

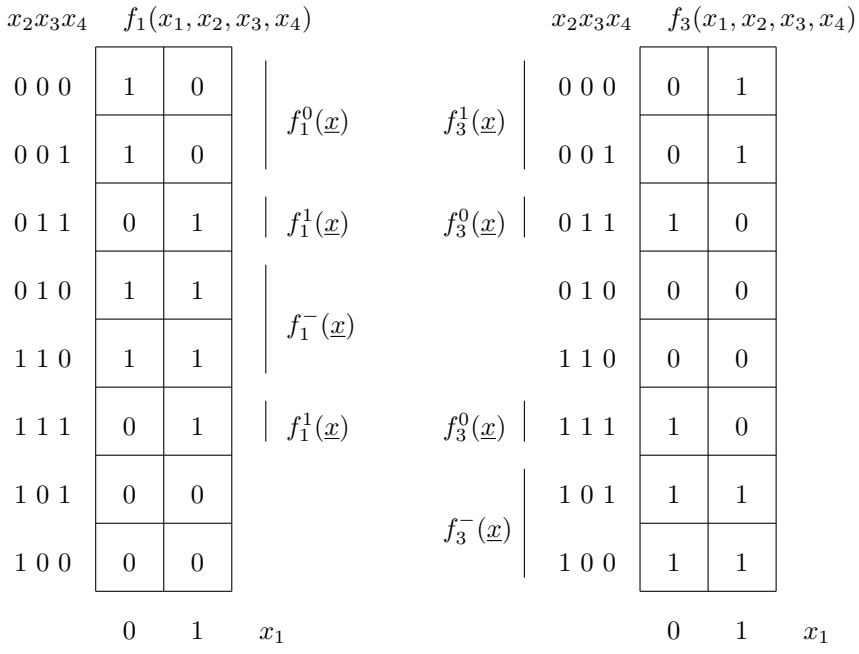


Abbildung 4.2: Funktion $f_3(x_1,x_2,x_3,x_4) = \overline{f_1(x_1,x_2,x_3,x_4)}$

$$f_3{}^0 = f_1^0 f_2^0 \vee f_1^0 \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee f_2^0 \bar{f_1^-} \bar{f_1^0} \bar{f_1^1} \tag{4.72}$$

$$f_3{}^1 = f_1^1 f_2^1 \vee f_1^1 \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee f_2^1 \bar{f_1^-} \bar{f_1^0} \bar{f_1^1} \tag{4.73}$$

Beweis:

Nach dem Einsetzen der nach (4.27) dargestellten Funktionen f_1 und f_2 (wobei $x = x_i$ ist) ergibt sich (4.75).

$$\begin{aligned} f_3 &= f_1 \vee f_2 && (4.74)\\ &= f_1^- \vee \bar{x} f_1^0 \vee x f_1^1 \vee f_2^- \vee \bar{x} f_2^0 \vee x f_2^1 && (4.75)\\ &= f_1^- \vee f_2^- \vee \bar{x}(f_1^0 \vee f_2^0) \vee x(f_1^1 \vee f_2^1) && (4.76) \end{aligned}$$

Die Disjunktion $f_1^0 \vee f_2^0$ kann nach (4.77) umgewandelt werden.

$$\begin{aligned} f_1^0 \vee f_2^0 &= f_1^0 \cdot 1 \vee f_2^0 \cdot 1\\ &= f_1^0(f_2^0 \vee \bar{f_2^0}) \vee f_2^0(f_1^0 \vee \bar{f_1^0})\\ &= f_1^0 f_2^0 \vee f_1^0 \bar{f_2^0} \vee f_2^0 f_1^0 \vee f_2^0 \bar{f_1^0}\\ &= f_1^0 f_2^0 \vee f_1^0 \bar{f_2^0} \vee f_2^0 \bar{f_1^0} && (4.77) \end{aligned}$$

Auf gleiche Weise lässt sich auch $f_1^1 \vee f_2^1$ umwandeln (siehe (4.78)).

$$f_1^1 \vee f_2^1 = f_1^1 f_2^1 \vee f_1^1 \bar{f_2^1} \vee f_2^1 \bar{f_1^1} \tag{4.78}$$

Die Funktion $\bar{f_1^0}$ kann durch elementare Umwandlungen von (4.79) in (4.83) überführt werden, wobei in (4.82) die Orthogonalität der Funktionen f_1^-, f_1^0 und f_1^1 (4.58) berücksichtigt wird.

$$\begin{aligned}
\bar{f}_1^0 &= 1 \cdot \bar{f}_1^0 && (4.79)\\
\bar{f}_1^0 &= (f_1^- \vee f_1^1 \vee \bar{f_1^-}\,\bar{f}_1^1)\bar{f}_1^0 && (4.80)\\
\bar{f}_1^0 &= f_1^-(1 \oplus f_1^0) \vee f_1^1(1 \oplus f_1^0) \vee \bar{f_1^-}\,\bar{f}_1^0\bar{f}_1^1 && (4.81)\\
\bar{f}_1^0 &= (f_1^- \oplus f_1^- f_1^0) \vee (f_1^1 \oplus f_1^0 f_1^1) \vee \bar{f_1^-}\,\bar{f}_1^0\bar{f}_1^1 && (4.82)\\
\bar{f}_1^0 &= f_1^- \vee f_1^1 \vee \bar{f_1^-}\,\bar{f}_1^0\bar{f}_1^1 && (4.83)
\end{aligned}$$

Analog entstehen auch (4.84), (4.85) und (4.86).

$$\bar{f}_2^0 = f_2^- \vee f_2^1 \vee \bar{f_2^-}\,\bar{f}_2^0\bar{f}_2^1 \quad (4.84)$$

$$\bar{f}_1^1 = f_1^- \vee f_1^0 \vee \bar{f_1^-}\,\bar{f}_1^0\bar{f}_1^1 \quad (4.85)$$

$$\bar{f}_2^1 = f_2^- \vee f_2^0 \vee \bar{f_2^-}\,\bar{f}_2^0\bar{f}_2^1 \quad (4.86)$$

Nach dem Einsetzen von (4.77), (4.78), (4.83)-(4.86) in (4.75) entsteht (4.87). Die weiteren elementaren Umstellungen führen zu (4.89).

$$\begin{aligned}
f_3 &= f_1^- \vee f_2^-\\
&\quad \vee\bar{x}(f_1^0 f_2^0 \vee f_1^0(f_2^- \vee f_2^1 \vee \bar{f_2^-}\,\bar{f}_2^0\bar{f}_2^1) \vee f_2^0(f_1^- \vee f_1^1 \vee \bar{f_1^-}\,\bar{f}_1^0\bar{f}_1^1))\\
&\quad \vee x(f_1^1 f_2^1 \vee f_1^1(f_2^- \vee f_2^0 \vee \bar{f_2^-}\,\bar{f}_2^0\bar{f}_2^1) \vee f_2^1(f_1^- \vee f_1^0 \vee \bar{f_1^-}\,\bar{f}_1^0\bar{f}_1^1)) \qquad (4.87)
\end{aligned}$$

$$\begin{aligned}
f_3 &= f_1^- \vee f_2^-\\
&\quad \vee\bar{x}(f_1^0 f_2^0 \vee f_1^0 f_2^- \vee f_1^0 f_2^1 \vee f_1^0 \bar{f_2^-}\,\bar{f}_2^0\bar{f}_2^1 \vee f_1^- f_2^0 \vee f_1^1 f_2^0 \vee \bar{f_1^-}\,\bar{f}_1^0\bar{f}_1^1 f_2^0) \vee\\
&\quad x(f_1^1 f_2^1 \vee f_1^1 f_2^- \vee f_1^1 f_2^0 \vee f_1^1 \bar{f_2^-}\,\bar{f}_2^0\bar{f}_2^1 \vee f_1^- f_2^1 \vee f_1^0 f_2^1 \vee \bar{f_1^-}\,\bar{f}_1^0\bar{f}_1^1 f_2^1) \qquad (4.88)\\
f_3 &= f_1^- \vee f_2^-\\
&\quad \vee\bar{x}f_1^0 f_2^- \vee \bar{x}f_1^0 f_2^1 \vee \bar{x}f_1^- f_2^0 \vee \bar{x}f_1^1 f_2^0\\
&\quad \vee\bar{x}(f_1^0 f_2^0 \vee f_1^0 \bar{f_2^-}\,\bar{f}_2^0\bar{f}_2^1 \vee \bar{f_1^-}\,\bar{f}_1^0\bar{f}_1^1 f_2^0)\\
&\quad \vee x f_1^1 f_2^- \vee x f_1^1 f_2^0 \vee x f_1^- f_2^1 \vee x f_1^0 f_2^1\\
&\quad \vee x(f_1^1 f_2^1 \vee f_1^1 \bar{f_2^-}\,\bar{f}_2^0\bar{f}_2^1 \vee \bar{f_1^-}\,\bar{f}_1^0\bar{f}_1^1 f_2^1) \qquad (4.89)
\end{aligned}$$

Unter Berücksichtigung der Umstellungen (4.90), (4.91), (4.92) und (4.93) kann der Ausdruck (4.89) in den Ausdruck (4.94) umgewandelt werden.

$$\bar{x}f_1^0 f_2^1 \vee x f_1^0 f_2^1 = (\bar{x} \vee x)f_1^0 f_2^1 = f_1^0 f_2^1 \quad (4.90)$$

$$\bar{x}f_1^1 f_2^0 \vee x f_1^1 f_2^0 = (\bar{x} \vee x)f_1^1 f_2^0 = f_1^1 f_2^0 \quad (4.91)$$

$$f_1^- \vee \bar{x}f_1^- f_2^0 \vee x f_1^- f_2^1 = (1 \vee \bar{x}f_2^0 \vee x f_2^1)f_1^- = f_1^- \quad (4.92)$$

$$f_2^- \vee \bar{x}f_1^0 f_2^- \vee x f_1^1 f_2^- = (1 \vee \bar{x}f_1^0 \vee x f_1^1)f_2^- = f_2^- \quad (4.93)$$

$$\begin{aligned}
f_3 &= f_1^- \vee f_2^- \vee f_1^0 f_2^1 \vee f_2^0 f_1^1\\
&\quad \vee\bar{x}(f_1^0 f_2^0 \vee f_1^0 \bar{f_2^-}\,\bar{f}_2^0\bar{f}_2^1 \vee f_2^0 \bar{f_1^-}\,\bar{f}_1^0\bar{f}_1^1)\\
&\quad \vee x(f_1^1 f_2^1 \vee f_1^1 \bar{f_2^-}\,\bar{f}_2^0\bar{f}_2^1 \vee f_2^1 \bar{f_1^-}\,\bar{f}_1^0\bar{f}_1^1) \qquad (4.94)
\end{aligned}$$

In (4.94) sind die Teilfunktionen (4.71), (4.72) und (4.73) der Funktion $f_3 = f_1 \vee f_2$ unmittelbar zu erkennen, wenn die Formeln (4.74) und (4.27) berücksichtigt werden. □

Satz 4.8. [2] *Die nach (4.71), (4.72) und (4.73) berechneten Teilfunktionen f_3^-, f_3^0 und f_3^1 (4.27) der Funktion $f_3 = f_1 \vee f_2$ sind zueinander paarweise disjunkt:*

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \textit{für} \quad \alpha, \beta \in \{-, 0, 1\} \quad \textit{und} \quad \alpha \neq \beta \tag{4.95}$$

Zum Beispiel werden für die Funktionen (4.96) und (4.98) mit den Teilfunktionen (4.97) und (4.99) die Teilfunktionen der Funktion $f_3(\underline{x}) = f_1(\underline{x}) \vee f_2(\underline{x})$ mit (4.100) beschrieben (siehe auch die Abbildung 4.3).

$$f_1(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1x_3x_4 \vee x_3\bar{x}_4 \tag{4.96}$$

$$\begin{aligned} f_1^-(x_2, x_3, x_4) &= x_3\bar{x}_4 \\ f_1^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3 \\ f_1^1(x_2, x_3, x_4) &= x_3x_4 \end{aligned} \tag{4.97}$$

$$f_2(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_3\bar{x}_4 \vee x_1\bar{x}_2x_4 \vee \bar{x}_2x_3\bar{x}_4 \vee x_1x_2\bar{x}_3x_4 \tag{4.98}$$

$$\begin{aligned} f_2^-(x_2, x_3, x_4) &= \bar{x}_2x_3\bar{x}_4 \\ f_2^0(x_2, x_3, x_4) &= \bar{x}_3\bar{x}_4 \\ f_2^1(x_2, x_3, x_4) &= \bar{x}_2x_4 \vee x_2\bar{x}_3x_4 \end{aligned} \tag{4.99}$$

$$\begin{aligned} f_3^-(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3x_4 \vee x_3\bar{x}_4 \\ f_3^0(x_2, x_3, x_4) &= \bar{x}_3\bar{x}_4 \\ f_3^1(x_2, x_3, x_4) &= \bar{x}_2x_3x_4 \vee x_2x_4 \end{aligned} \tag{4.100}$$

Die Abbildung 4.3 zeigt, dass die Funktion $f_3^-(\underline{x})$ aus der Disjunktion der Funktionen $f_1^-(\underline{x})$ und $f_2^-(\underline{x})$ und der Konjunktion $\bar{x}_2\bar{x}_3x_4$ entsteht, die sich aus der konjunktiven Verknüpfung $f_1^0(\underline{x})f_2^1(\underline{x})$ ergibt. Die in der Funktion $f_3^0(\underline{x})$ enthaltenen Konjunktionen werden durch die Verknüpfungen $f_1^0(\underline{x})f_2^0(\underline{x})$ (die Konjunktion $\bar{x}_2\bar{x}_3\bar{x}_4$) und $f_2^0(\underline{x})\ \overline{f_1^-(\underline{x})}\ \overline{f_1^0(\underline{x})}\ \overline{f_1^1(\underline{x})}$ (die Konjunktion $x_2\bar{x}_3\bar{x}_4$) gebildet. Die Konjunktion $\bar{x}_2x_3x_4$ der Funktion $f_3^1(\underline{x})$ entstand durch die Verknüpfung $f_1^1(\underline{x})f_2^1(\underline{x})$. Die weiteren Konjunktionen der Funktion $f_3^1(\underline{x})$ sind das Ergebnis der Verknüpfungen $f_1^1(\underline{x})\ \overline{f_2^-(\underline{x})}\ \overline{f_2^0(\underline{x})}\ \overline{f_2^1(\underline{x})}$ und $f_2^1(\underline{x})\ \overline{f_1^-(\underline{x})}\ \overline{f_1^0(\underline{x})}\ \overline{f_1^1(\underline{x})}$.

Konjunktion

Satz 4.9. *Für die gegebenen Funktionen f_1 und f_2, die nach (4.27) durch die Teilfunktionen ${f_1}^-$, ${f_1}^0$, ${f_1}^1$ und ${f_2}^-$, ${f_2}^0$, ${f_2}^1$ dargestellt sind, können die Teilfunktionen ${f_3}^-$, ${f_3}^0$, ${f_3}^1$ der Funktion $f_3 = f_1 \cdot f_2$ nach (4.101)-(4.103) berechnet werden.*

$${f_3}^- = {f_1}^-{f_2}^- \tag{4.101}$$

$${f_3}^0 = {f_1}^-{f_2}^0 \vee {f_1}^0{f_2}^- \vee {f_1}^0{f_2}^0 \tag{4.102}$$

$${f_3}^1 = {f_1}^-{f_2}^1 \vee {f_1}^1{f_2}^- \vee {f_1}^1{f_2}^1 \tag{4.103}$$

Beweis:
Werden die nach (4.27) dargestellten Funktionen f_1 und f_2 in (4.104) eingesetzt (wobei $x =$

[2] Beweis siehe Anhang B

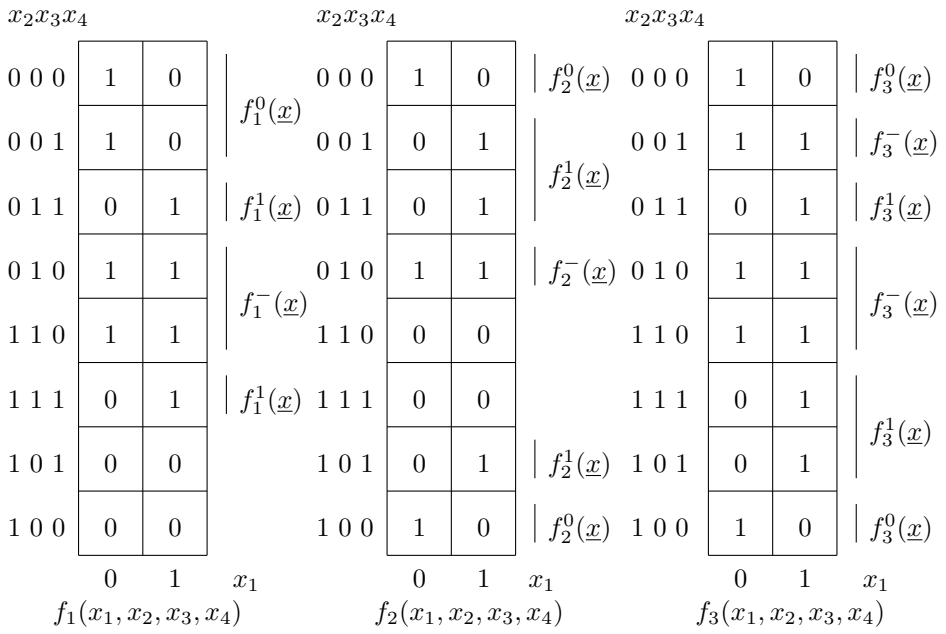


Abbildung 4.3: Funktion $f_3(x_1, x_2, x_3, x_4) = f_1(x_1, x_2, x_3, x_4) \vee f_2(x_1, x_2, x_3, x_4)$

x_i ist), erhält man (4.105). Nach weiteren elementaren Umformungen entsteht schließlich (4.108).

$$
\begin{aligned}
f_3 &= f_1 \cdot f_2 && (4.104)\\
&= (f_1^- \vee \bar{x} f_1^0 \vee x f_1^1) \cdot (f_2^- \vee \bar{x} f_2^0 \vee x f_2^1) && (4.105)\\
&= f_1^- f_2^- \vee \bar{x} f_1^- f_2^0 \vee x f_1^- f_2^1 \\
&\quad \bar{x} f_1^0 f_2^- \vee \bar{x} f_1^0 \bar{x} f_2^0 \vee \bar{x} f_1^0 x f_2^1 \\
&\quad \vee x f_1^1 f_2^- \vee x f_1^1 \bar{x} f_2^0 \vee x f_1^1 x f_2^1 && (4.106)\\
&= f_1^- f_2^- \\
&\quad \vee \bar{x} f_1^- f_2^0 \vee x f_1^- f_2^1 \vee \bar{x} f_1^0 f_2^- \\
&\quad \vee \bar{x} f_1^0 f_2^0 \vee x f_1^1 f_2^- \vee x f_1^1 f_2^1 && (4.107)
\end{aligned}
$$

$$
\begin{aligned}
f_3 &= f_1^- f_2^- \\
&\quad \vee \bar{x}(f_1^- f_2^0 \vee f_1^0 f_2^- \vee f_1^0 f_2^0) \\
&\quad \vee x(f_1^- f_2^1 \vee f_1^1 f_2^- \vee f_1^1 f_2^1) && (4.108)
\end{aligned}
$$

Aus (4.108) gehen infolge (4.104) und (4.27) die Teilfunktionen (4.101), (4.102) und (4.103) der Funktion $f_3 = f_1 \cdot f_2$ unmittelbar hervor. □

Satz 4.10. [3] *Die nach (4.101), (4.102) und (4.103) berechneten Teilfunktionen f_3^-, f_3^0 und f_3^1 (4.27) der Funktion $f_3 = f_1 \cdot f_2$ sind zueinander paarweise disjunkt:*

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \textit{für} \quad \alpha, \beta \in \{-, 0, 1\} \quad \textit{und} \quad \alpha \neq \beta \tag{4.109}$$

[3] Beweis siehe Anhang B

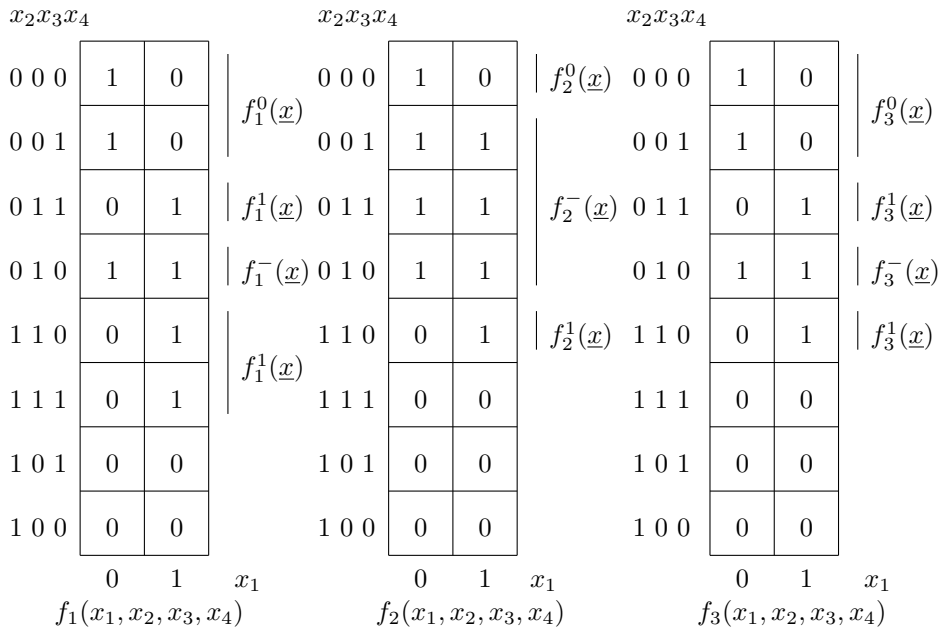


Abbildung 4.4: Funktion $f_3(x_1, x_2, x_3, x_4) = f_1(x_1, x_2, x_3, x_4) \cdot f_2(x_1, x_2, x_3, x_4)$

Für die Beispielfunktionen (4.110) und (4.112) mit den Teilfunktionen (4.111) und (4.113) werden die Teilfunktionen der Funktion $f_3(\underline{x}) = f_1(\underline{x}) \cdot f_2(\underline{x})$ mit (4.114) beschrieben.

$$f_1(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1x_3 \vee \bar{x}_1\bar{x}_2x_3\bar{x}_4 \tag{4.110}$$

$$\begin{aligned} f_1^-(x_2, x_3, x_4) &= \bar{x}_2x_3\bar{x}_4 \\ f_1^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3 \\ f_1^1(x_2, x_3, x_4) &= \bar{x}_2x_3x_4 \vee x_2x_3 \end{aligned} \tag{4.111}$$

$$f_2(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2 \vee x_1\bar{x}_2x_4 \vee x_1x_3\bar{x}_4 \tag{4.112}$$

$$\begin{aligned} f_2^-(x_2, x_3, x_4) &= \bar{x}_2x_4 \vee \bar{x}_2x_3\bar{x}_4 \\ f_2^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3\bar{x}_4 \\ f_2^1(x_2, x_3, x_4) &= x_2x_3\bar{x}_4 \end{aligned} \tag{4.113}$$

$$\begin{aligned} f_3^-(x_2, x_3, x_4) &= \bar{x}_2x_3\bar{x}_4 \\ f_3^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3 \\ f_3^1(x_2, x_3, x_4) &= \bar{x}_2x_3x_4 \vee x_2x_3\bar{x}_4 \end{aligned} \tag{4.114}$$

In der Abbildung 4.4 sind die oben beschiebenen Funktionen

$$f_1(x_1, x_2, x_3, x_4),\ f_2(x_1, x_2, x_3, x_4) \quad \text{und} \quad f_3(x_1, x_2, x_3, x_4)$$

und ihre Teilfunktionen mittels eines Karnaughplan dargestellt. Die Funktion $f_3^-(\underline{x})$ entsteht im Ergebnis der konjunktiven Verknüpfung der Funktionen $f_1^-(\underline{x})$ und $f_2^-(\underline{x})$. Die zwei Konjunktionen der Funktion $f_3^0(\underline{x})$ werden durch Verknüpfungen $f_1^0(\underline{x})f_2^0(\underline{x}) = \bar{x}_2\bar{x}_3\bar{x}_4$ und $f_1^0(\underline{x})f_2^-(\underline{x}) = \bar{x}_2\bar{x}_3x_4$ gebildet. Die Konjunktionen der Funktion $f_3^1(\underline{x})$ entstehen durch die Verknüpfungen $f_1^1(\underline{x})f_2^-(\underline{x})$ und $f_1^1(\underline{x})f_2^1(\underline{x})$.

Antivalenz

Satz 4.11. *Für die gegebenen Funktionen f_1 und f_2, die nach (4.27) durch die Teilfunktionen ${f_1}^-$, ${f_1}^0$, ${f_1}^1$ und ${f_2}^-$, ${f_2}^0$, ${f_2}^1$ dargestellt sind, können die Teilfunktionen ${f_3}^-$, ${f_3}^0$, ${f_3}^1$ der Funktion $f_3 = f_1 \oplus f_2$ nach (4.115), (4.116) und (4.103) berechnet werden.*

$$ {f_3}^- = {f_1}^0 {f_2}^1 \vee {f_1}^1 {f_2}^0 \vee {f_1}^- \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee {f_2}^- \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1 \tag{4.115} $$

$$ {f_3}^0 = {f_1}^1 {f_2}^- \vee {f_1}^- {f_2}^1 \vee {f_1}^0 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee {f_2}^0 \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1 \tag{4.116} $$

$$ {f_3}^1 = {f_1}^0 {f_2}^- \vee {f_1}^- {f_2}^0 \vee {f_1}^1 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee {f_2}^1 \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1 \tag{4.117} $$

Beweis:

Nach der Anwendung des Satzes von Stone auf (4.118) entsteht (4.119).

$$ \begin{aligned} f_3 &= f_1 \oplus f_2 && (4.118)\\ &= f_1 \bar{f}_2 \vee \bar{f}_1 f_2 && (4.119) \end{aligned} $$

Die Funktionen $\bar{f}_1$ und $\bar{f}_2$ können nach (4.66) als (4.120) und (4.121) dargestellt werden.

$$ \bar{f}_1 = \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1 \vee \bar{x} f_1^1 \vee x f_1^0 \tag{4.120} $$

$$ \bar{f}_2 = \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee \bar{x} f_2^1 \vee x f_2^0 \tag{4.121} $$

Das Einsetzen der nach (4.27) dargestellten Funktionen f_1 und f_2 (wobei $x = x_i$ ist) und der nach (4.120) und (4.121) dargestellten Funktionen $\bar{f}_1$ und $\bar{f}_2$ in (4.119) führt zu (4.122). Nach dem Ausmultiplizieren der Disjunktionen in (4.122) entsteht (4.123).

$$ \begin{aligned} f_3 &= (f_1^- \vee \bar{x} f_1^0 \vee x f_1^1)(\bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee \bar{x} f_2^1 \vee x f_2^0)\\ &\quad \vee (\bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1 \vee \bar{x} f_1^1 \vee x f_1^0)(f_2^- \vee \bar{x} f_2^0 \vee x f_2^1) \end{aligned} \tag{4.122} $$

$$ \begin{aligned} f_3 &= f_1^- \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee \bar{x} f_1^- f_2^1 \vee x f_1^- f_2^0\\ &\quad \vee \bar{x} f_1^0 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee \bar{x} f_1^0 f_2^1\\ &\quad \vee x f_1^1 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee x f_1^1 f_2^0\\ &\quad \vee \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1 f_2^- \vee \bar{x} \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1 f_2^0 \vee x \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1 f_2^1\\ &\quad \vee \bar{x} f_1^1 f_2^- \vee \bar{x} f_1^1 f_2^0\\ &\quad \vee x f_1^0 f_2^- \vee x f_1^0 f_2^1 \end{aligned} \tag{4.123} $$

Durch weitere Umformungen von (4.123) unter Berücksichtigung von (4.124) und (4.125) ergibt sich (4.126).

$$ x f_1^0 f_2^1 \vee \bar{x} f_1^0 f_2^1 = (x \vee \bar{x}) f_1^0 f_2^1 = f_1^0 f_2^1 \tag{4.124} $$

$$ x f_1^1 f_2^0 \vee \bar{x} f_1^1 f_2^0 = (x \vee \bar{x}) f_1^1 f_2^0 = f_1^1 f_2^0 \tag{4.125} $$

$$ \begin{aligned} f_3 &= f_1 \oplus f_2\\ &= {f_1}^0 {f_2}^1 \vee {f_1}^1 {f_2}^0 \vee {f_1}^- \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee {f_2}^- \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1\\ &\quad \vee \bar{x}({f_1}^1 {f_2}^- \vee {f_1}^- {f_2}^1 \vee {f_1}^0 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee {f_2}^0 \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1)\\ &\quad \vee x({f_1}^0 {f_2}^- \vee {f_1}^- {f_2}^0 \vee {f_1}^1 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee {f_2}^1 \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1) \end{aligned} \tag{4.126} $$

Die Teilfunktionen (4.115), (4.116) und (4.117) der Funktion $f_3 = f_1 \oplus f_2$ gehen wegen (4.27) aus (4.126) unmittelbar hervor. □

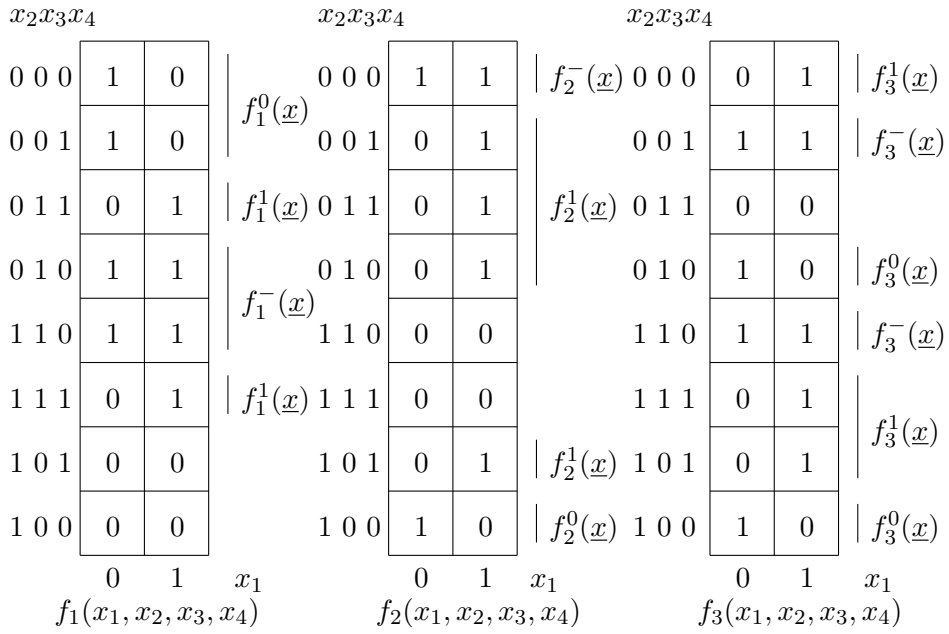


Abbildung 4.5: Funktion $f_3(x_1, x_2, x_3, x_4) = f_1(x_1, x_2, x_3, x_4) \oplus f_2(x_1, x_2, x_3, x_4)$

Satz 4.12. [4] *Die nach (4.115), (4.116) und (4.117) berechneten Teilfunktionen f_3^-, f_3^0 und f_3^1 (4.27) der Funktion $f_3 = f_1 \oplus f_2$ sind zueinander paarweise disjunkt:*

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \textit{für} \quad \alpha, \beta \in \{-, 0, 1\} \quad \textit{und} \quad \alpha \neq \beta \tag{4.127}$$

Im folgenden Beispiel werden die Funktionen (4.128) und (4.130) über die Teilfunktionen (4.129) und (4.131) antivalent verknüpft. Die dabei entstehenden Teilfunktionen der Ergebnisfunktion $f_3(\underline{x}) = f_1(\underline{x}) \oplus f_2(\underline{x})$ werden mit (4.132) beschrieben.

$$f_1(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1x_3x_4 \vee x_3\bar{x}_4 \tag{4.128}$$

$$\begin{aligned} f_1^-(x_2, x_3, x_4) &= x_3\bar{x}_4 \\ f_1^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3 \\ f_1^1(x_2, x_3, x_4) &= x_3x_4 \end{aligned} \tag{4.129}$$

$$f_2(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_3\bar{x}_4 \vee x_1\bar{x}_2 \vee x_1x_2\bar{x}_3x_4 \tag{4.130}$$

$$\begin{aligned} f_2^-(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3\bar{x}_4 \\ f_2^0(x_2, x_3, x_4) &= x_2\bar{x}_3\bar{x}_4 \\ f_2^1(x_2, x_3, x_4) &= \bar{x}_2x_3 \vee \bar{x}_3x_4 \end{aligned} \tag{4.131}$$

$$\begin{aligned} f_3^-(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3x_4 \vee x_2x_3\bar{x}_4 \\ f_3^0(x_2, x_3, x_4) &= \bar{x}_2x_3\bar{x}_4 \vee x_2\bar{x}_3\bar{x}_4 \\ f_3^1(x_2, x_3, x_4) &= x_2x_4 \vee \bar{x}_2\bar{x}_3\bar{x}_4 \end{aligned} \tag{4.132}$$

In der Abbildung 4.5 sind die Funktionen (4.128) und (4.130) und die Funktion $f_3(\underline{x})$

[4] Beweis siehe Anhang B

mit ihren Teilfunktionen abgebildet. Es ist erkennbar, dass eine der Konjunktionen der Funktion $f_3^-(\underline{x})$ durch die Verknüpfung $f_1^0(\underline{x})f_2^1(\underline{x})$ entsteht. Die andere Konjunktion der Funktion $f_3^-(\underline{x})$ ist der Teil der Funktion $f_1^-(\underline{x})$, der auf keine der Funktionen $f_2^-(\underline{x})$, $f_2^0(\underline{x})$ oder $f_2^1(\underline{x})$ trifft. Die Konjunktionen der Funktion $f_3^0(\underline{x})$ enstehen durch die Verknüpfungen $f_1^-(\underline{x})f_2^1(\underline{x})$ und $f_2^0(\underline{x})\,\overline{f_1^-(\underline{x})}\,\overline{f_1^0(\underline{x})}\,\overline{f_1^1(\underline{x})}$. Die Konjunktionen der Funktion $f_3^1(\underline{x})$ können als $f_1^0(\underline{x})f_2^-(\underline{x})$, $f_1^1(\underline{x})\,\overline{f_2^-(\underline{x})}\,\overline{f_2^0(\underline{x})}\,\overline{f_2^1(\underline{x})}$ und $f_2^1(\underline{x})\,\overline{f_1^-(\underline{x})}\,\overline{f_1^0(\underline{x})}\,\overline{f_1^1(\underline{x})}$ berechnet werden.

Äquivalenz

Satz 4.13. *Für die gegebenen Funktionen f_1 und f_2, die nach (4.27) durch die Teilfunktionen ${f_1}^-$, ${f_1}^0$, ${f_1}^1$ und ${f_2}^-$, ${f_2}^0$, ${f_2}^1$ dargestellt sind, können die Teilfunktionen ${f_3}^-$, ${f_3}^0$, ${f_3}^1$ der Funktion $f_3 = f_1 \odot f_2$ nach (4.133), (4.134) und (4.135) berechnet werden.*

$$ {f_3}^- = {f_1}^- {f_2}^- \vee {f_1}^0 {f_2}^0 \vee {f_1}^1 {f_2}^1 \vee \bar{f_1^-} \bar{f_1^0} \bar{f_1^1} \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \tag{4.133} $$

$$ {f_3}^0 = {f_1}^- {f_2}^0 \vee {f_1}^0 {f_2}^- \vee {f_1}^1 \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee {f_2}^1 \bar{f_1^-} \bar{f_1^0} \bar{f_1^1} \tag{4.134} $$

$$ {f_3}^1 = {f_1}^- {f_2}^1 \vee {f_1}^1 {f_2}^- \vee {f_1}^0 \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee {f_2}^0 \bar{f_1^-} \bar{f_1^0} \bar{f_1^1} \tag{4.135} $$

Beweis:

Durch die Anwendung des Satzes von Stone auf (4.136) entsteht (4.137). Die Funktionen $\bar{f_1}$ und $\bar{f_2}$ können als (4.120) und (4.121) dargestellt werden. Werden die nach (4.27) dargestellten Funktionen f_1 und f_2 (wobei $x = x_i$ ist) und die nach (4.120) und (4.121) dargestellten Funktionen $\bar{f_1}$ und $\bar{f_2}$ in (4.137) eingesetzt, wird (4.138) erhalten. Nach dem Ausmultiplizieren der Disjunktionen in (4.138) ergibt sich (4.139).

$$
\begin{aligned}
f_3 &= f_1 \odot f_2 && (4.136)\\
&= f_1 f_2 \vee \bar{f_1}\bar{f_2} && (4.137)\\
&= (f_1^- \vee \bar{x} f_1^0 \vee x f_1^1)(f_2^- \vee \bar{x} f_2^0 \vee x f_2^1) \\
&\quad \vee (\bar{f_1^-} \bar{f_1^0} \bar{f_1^1} \vee \bar{x} f_1^1 \vee x f_1^0)(\bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee \bar{x} f_2^1 \vee x f_2^0) && (4.138)\\
&= f_1^- f_2^- \vee \bar{x} f_1^- f_2^0 \vee x f_1^- f_2^1 \\
&\quad \vee \bar{x} f_1^0 f_2^- \vee \bar{x} f_1^0 f_2^0 \\
&\quad \vee x f_1^1 f_2^- \vee x f_1^1 f_2^1 \\
&\quad \vee \bar{f_1^-} \bar{f_1^0} \bar{f_1^1} \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee \bar{x} \bar{f_1^-} \bar{f_1^0} \bar{f_1^1} f_2^1 \vee x \bar{f_1^-} \bar{f_1^0} \bar{f_1^1} f_2^0 \\
&\quad \vee \bar{x} f_1^1 \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee \bar{x} f_1^1 f_2^1 \\
&\quad \vee x f_1^0 \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee x f_1^0 f_2^0 && (4.139)
\end{aligned}
$$

Durch weitere Umformungen von (4.139) unter Berücksichtigung von (4.140) und (4.141) folgt (4.142).

$$ x f_1^0 f_2^0 \vee \bar{x} f_1^0 f_2^0 = (x \vee \bar{x}) f_1^0 f_2^0 = f_1^0 f_2^0 \tag{4.140} $$

$$ x f_1^1 f_2^1 \vee \bar{x} f_1^1 f_2^1 = (x \vee \bar{x}) f_1^1 f_2^1 = f_1^1 f_2^1 \tag{4.141} $$

$$\begin{aligned} f_3 &= f_1 \odot f_2 \\ &= f_1{}^- f_2{}^- \vee f_1{}^0 f_2{}^0 \vee f_1{}^1 f_2{}^1 \vee \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \\ &\quad \vee \bar{x}(f_1{}^- f_2{}^0 \vee f_1{}^0 f_2{}^- \vee f_1{}^1 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee f_2{}^1 \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1) \\ &\quad \vee x(f_1{}^- f_2{}^1 \vee f_1{}^1 f_2{}^- \vee f_1{}^0 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee f_2{}^0 \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1) \end{aligned} \tag{4.142}$$

Die Teilfunktionen (4.133), (4.134) und (4.135) der Funktion $f_3 = f_1 \odot f_2$ verknüpft nach (4.27) sind aus (4.142) unmittelbar abzuleiten. □

Satz 4.14. [5] *Die nach (4.133), (4.134) und (4.135) berechneten Teilfunktionen f_3^-, f_3^0 und f_3^1 (4.27) der Funktion $f_3 = f_1 \odot f_2$ sind zueinander paarweise disjunkt:*

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \textit{für} \quad \alpha, \beta \in \{-, 0, 1\} \quad \textit{und} \quad \alpha \neq \beta \tag{4.143}$$

In der Abbildung 4.6 sind die Funktionen $f_1(\underline{x})$ (4.144) und $f_2(\underline{x})$ (4.146) mit ihren Teilfunktionen (4.145) und (4.147) und die Funktion $f_3(\underline{x}) = f_1(\underline{x}) \odot f_2(\underline{x})$ mit ihren Teilfunktionen (4.148) dargestellt.

$$f_1(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1x_3x_4 \vee x_3\bar{x}_4 \tag{4.144}$$

$$\begin{aligned} f_1^-(x_2, x_3, x_4) &= x_3\bar{x}_4 \\ f_1^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3 \\ f_1^1(x_2, x_3, x_4) &= x_3x_4 \end{aligned} \tag{4.145}$$

$$f_2(x_1, x_2, x_3, x_4) = \bar{x}_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1\bar{x}_2\bar{x}_3x_4 \vee \vee x_1\bar{x}_2x_3 \vee x_2x_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_3\bar{x}_4 \tag{4.146}$$

$$\begin{aligned} f_2^-(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3\bar{x}_4 \vee x_2x_3\bar{x}_4 \\ f_2^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3x_4 \vee x_2\bar{x}_3\bar{x}_4 \\ f_2^1(x_2, x_3, x_4) &= \bar{x}_2x_3 \end{aligned} \tag{4.147}$$

$$\begin{aligned} f_3^-(x_2, x_3, x_4) &= \bar{x}_2x_4 \vee x_2x_3\bar{x}_4 \vee x_2\bar{x}_3x_4 \\ f_3^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3\bar{x}_4 \vee x_2x_3x_4 \\ f_3^1(x_2, x_3, x_4) &= \bar{x}_2x_3\bar{x}_4 \vee x_2\bar{x}_3\bar{x}_4 \end{aligned} \tag{4.148}$$

Die Teilfunktion $f_3^-(\underline{x})$ besteht aus vier Konjunktionen, die durch die Verknüpfungen $f_1^0(\underline{x})f_2^0(\underline{x})$, $f_1^1(\underline{x})f_2^1(\underline{x})$, $f_1^-(\underline{x})f_2^-(\underline{x})$ und $\overline{f_1^-(\underline{x})}\ \overline{f_1^0(\underline{x})}\ \overline{f_1^1(\underline{x})}\ \overline{f_2^-(\underline{x})}\ \overline{f_2^0(\underline{x})}\ \overline{f_2^1(\underline{x})}$ gebildet werden. Beide Konjunktionen der Teilfunktion $f_3^0(\underline{x})$ lassen sich wie folgt berechnen: $\bar{x}_2\bar{x}_3\bar{x}_4 = f_1^0(\underline{x})f_2^-(\underline{x})$ und $x_2x_3x_4 = f_1^1(\underline{x})\overline{f_2^-(\underline{x})}\ \overline{f_2^0(\underline{x})}\ \overline{f_2^1(\underline{x})}$. Die Konjunktionen der Teilfunktion $f_3^1(\underline{x})$ sind das Ergebnis der Verknüpfungen $f_1^-(\underline{x})f_2^1(\underline{x}) = \bar{x}_2x_3\bar{x}_4$ und $\overline{f_1^-(\underline{x})}\ \overline{f_1^0(\underline{x})}\ \overline{f_1^1(\underline{x})}\ f_2^0(\underline{x}) = x_2\bar{x}_3\bar{x}_4$.

4.1.3 Analyse

Wie im Kapitel 1 beschrieben wurde, kann eine Boolesche Funktion durch verschiedene Datenstrukturen dargestellt werden. Eine der Darstellungsformen ist die Ternärvektorliste (siehe Abschnitt 2.2). Im Ergebnis der Zerlegung einer Booleschen Funktion in drei Teilfunktionen werden drei Ternärvektorlisten gebilden und als TVL-Tupel abgespeichert.

Mit der Bewertung einer neuen Datenstruktur ist die Frage verbunden, ob die Datenstruktur im Vergleich zur ursprünglichen Datenstruktur Vorteile aufweist. Dabei sind verschiedene Aspekte in Betracht zu ziehen. Die wichtigsten sind:

[5] Beweis siehe Anhang B

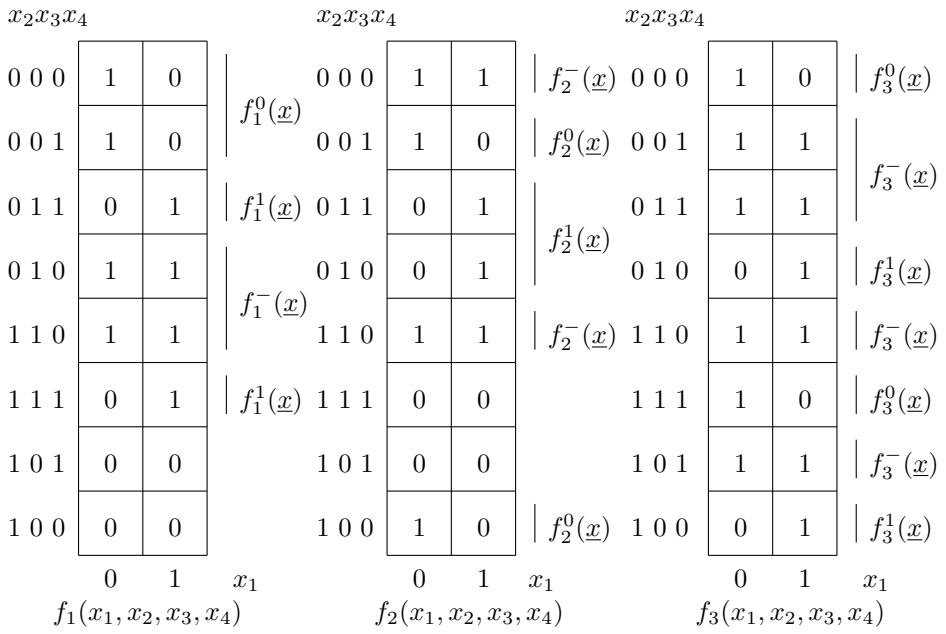


Abbildung 4.6: Funktion $f_3(x_1, x_2, x_3, x_4) = f_1(x_1, x_2, x_3, x_4) \odot f_2(x_1, x_2, x_3, x_4)$

- der Speicherplatz, der zur Abspeicherung der Booleschen Funktion benötigt wird,
- die Zeit, die für das Ausführen der Grundoperationen zwischen zwei Booleschen Funktionen notwendig ist.

Bezüglich des Speicherplatzbedarfs sind folgende Betrachtungen in die Bewertung einzubeziehen:

- Zur Darstellung einer Boolesche Funktion mit n Variablen durch eine orthogonale TVL werden im ungünstigsten Fall 2^{n-1} Ternärvektoren benötigt. Wird eine Boolesche Funktion in drei Funktionen nach dem beschriebenen Algorithmus zerlegt, wird in der neu entstandenen Funktionen die Anzahl der Variablen um 1 vermindert. Das bedeutet, dass die maximale Anzahl der Konjunktionen jeder einzelnen Funktion (oder Zeilenzahl bei der Darstellung als TVL) gleich 2^{n-2} ist. Die maximale Anzahl der Ternärvektoren aller drei Funktionen würde in diesem Fall $3 \cdot 2^{n-2}$ betragen, so dass die Zeilenanzahl der ursprünglichen Funktion überschritten würde.

  Eine der wichtigsten Eigenschaften der Teilfunktionen ist ihre Orthogonalität zueinander. Da die drei Funktionen paarweise orthogonal sind, sind auch die einzelnen Ternärvektoren orthogonal, d. h. alle Ternärvektoren enthalten unterschiedliche Binärvektoren. In einem Booleschen Raum mit $n-1$ Variablen sind maximal 2^{n-1} verschiedene Binärvektoren möglich. Folglich kann die Summe der Ternärvektoren der drei Funktionen den Wert 2^{n-1} nicht übersteigen, d. h. die maximale Anzahl der Ternärvektoren aller drei Funktionen kann auch nicht die maximale Zeilenzahl der nicht zerlegten Funktion überschreiten.

- Wird davon ausgegangen, dass eine TVL im Mittel nur über die Hälfte der maximal

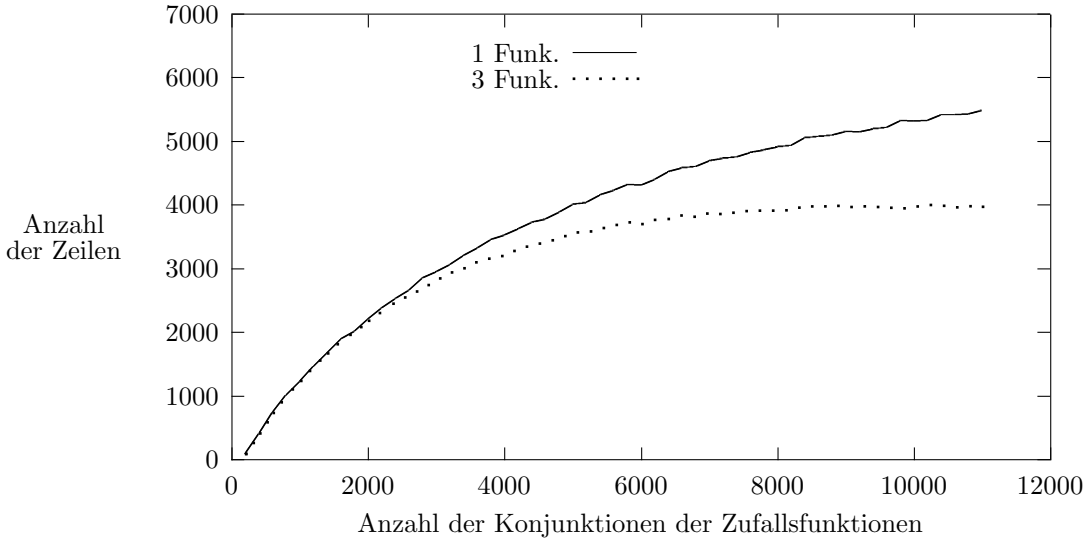


Abbildung 4.7: Anzahl der Konjunktionen der Funktionen vor und nach Dekomposition

möglichen Zeilen verfügt, ergibt sich für die nicht zerlegte Funktion eine TVL mit 2^{n-2} Zeilen.

Jede Teilfunktion einer zerlegten Funktion kann maximal 2^{n-2} Konjunktionen besitzen, weil sie von einer Variable weniger als die ursprüngliche Funktion abhängt. Mit der Annahme, dass die durchschnittliche TVL nur die Hälfte der maximal möglichenen Zeilen enthält, ergibt sich für die TVL einer Teilfunktion die Zeilenzahl 2^{n-3} und eine Zeilensumme über drei Teilfunktionen von $3 \cdot 2^{n-3}$. Das heißt, in einigen Fällen werden zur Darstellung einer Booleschen Funktion mit drei Funktionen mehr Ternärvektoren gebraucht als beim Einsatz einer TVL.

- Die Auswertung der zufällig erzeugten und als TVL dargestellten Funktionen hat gezeigt, dass die Summe der Konjunktionen (bei einer Darstellung als TVL entspricht das der Anzahl der Ternärvektoren) innerhalb der drei Teilfunktionen in der Regel kleiner ist als die Anzahl der Ternärvektoren der nicht zerlegten Funktion. Die Abbildung 4.7 enthält die Ergebnisse der Darstellung der Zufallsfunktionen mit 14 Variablen. Dabei wird die nicht zerlegte Funktion einer Zerlegung in drei Teilfunktionen gegenübergestellt. Die Funktionen mit einer kleineren Anzahl von Konjunktionen lassen sich als eine Funktion noch günstig darstellen. Mit der wachsenden Zahl der Konjunktionen in der Funktion werden die Vorteile der Dekomposition dagegen immer deutlicher. Diese Tatsache ist darauf zurückzuführen, dass bei kleineren TVL die Möglichkeiten des Zusammenfassens von TV durch das Teilen von TVL in drei Teile geringer werden.

Wie im Abschnitt 4.1.2 gezeigt, benötigen die Grundoperationen spezielle Algorithmen. Als Problemgröße wird bei der Abschätzung der Algorithmen hier wie in [Hess99] die Länge der TVL (Anzahl der enthaltenen Ternärvektoren) herangezogen. Die XBOOLE-Operationen (beschrieben in [DKSW92]) können in den Algorithmen der Tupeloperationen weiter verwendet werden. Mit einer Tupeloperation sind die drei Teilfunktionen der

Ergebnisfunktion zu berechnen, wobei für die Berechnung jeder Teilfunktion mehrere Verknüpfungen von Teilfunktionen der Operanden notwendig sind.

Der Aufwand für die Operationen Konjunktion, Negation und Disjunktion kann wie folgt abgeschätzt werden:

- Konjunktion
  - Worst case:
    Für die Berechnung der Konjunktion von zwei nicht zerlegten Funktionen sind zwei Ternärvektorlisten mit N_A und N_B Ternärvektoren zu verknüpfen. Der Berechnungsaufwand und die Komplexitätsordnung können in diesem Fall nach (4.149) ermittelt werden, wobei unter λ die Breite eines Ternärvektors (Anzahl der Spalten) verstanden wird. Da für die hier vorgenommenen Vergleiche diese Einflussgröße keine Bedeutung hat, und weil die Anzahl der Variablen (oder der Spalten) in einer TVL mit der Länge der TVL unmittelbar verbunden ist, wird λ in weiteren Aufwandabschätzungen vernachlässigt.

    $$f_{wc} = N_A \times N_B \times \lambda = O_{wc}(N_A \times N_B) \tag{4.149}$$

    Für die Berechnung der drei Teilfunktionen der Ergebnisfunktion aus zwei in drei Teile zerlegten Funktionen sind 7 konjunktive Verknüpfungen und 4 disjunktive Verknüpfungen (siehe Formeln (4.101), (4.102) und (4.103)) durchzuführen. Die konjunktiven Verknüpfungen werden mit jeweils zwei Ternärvektorlisten mit $N_A/3$ und $N_B/3$ Zeilen ausgeführt. Disjunktive Verknüpfungen können wegen der Orthogonalität der Teilfunktionen in konstanter Zeit durch Anhängen einer TVL an die anderen realisiert werden. Der gesamte Aufwand und die Komplexitätsordnung lassen sich damit nach (4.150) berechnen.

    $$f_{wc} = 7/9 \times N_A \times N_B + 4c = O_{wc}(N_A \times N_B) \tag{4.150}$$

  - Average case:
    Unter Berücksichtigung, dass die Zeilenzahlen der nicht zerlegten Funktion in diesem Fall $N_A/2$ und $N_B/2$ betragen und die Teilfunktionen aus jeweils $N_A/6$ und $N_B/6$ Zeilen bestehen, können hier analog worst case der Berechnungsaufwand und die Komplexitätsordnung für die nicht zerlegten Funktionen nach (4.151) und für die zerlegten Funktionen nach (4.152) bestimmt werden.

    $$f_{ac} = 1/4 \times N_A \times N_B = O_{ac}(N_A \times N_B) \tag{4.151}$$

    $$f_{ac} = 7/36 \times N_A \times N_B + 4c = O_{ac}(N_A \times N_B) \tag{4.152}$$

- Negation
  - Worst case:
    Die Operation Negation einer Funktion kann auf die Operation Differenz zwischen der 1-Funktion (in der TVL-Darstellung die Liste aus einem Tenärvektor aus Strichelementen) und dieser Funktion zurückgeführt werden [BoSt91]. Für die Durchführung der Negation mit der nicht zerlegten Funktion sind dabei die zwei Ternärvektorlisten zu verknüpfen. Wird unterstellt, dass der Strich-TV N_A

Binärvektoren enthält und die maximale Anzahl der Zeilen einer TVL N_B beträgt, können der Aufwand und die Komplexitätsordnung nach (4.153) berechnet werden (siehe [Hess99]).

$$f_{wc} = N_A \times N_B = O_{wc}(N_A \times N_B) \tag{4.153}$$

Die Berechnung der Teilfunktion f_3^- der negierten Funktion $f_3 = \bar{f}_1$ wird praktisch nach (4.154) durchgeführt.

$$f_3^- = \bar{f_1^-} \backslash f_1^0 \backslash f_1^1 \tag{4.154}$$

Zur Berechnung der Teilfunktion f_3^- ist zuerst ein Strich-TV aus $N_A/2$ Binärvektoren mit einer TVL aus $N_B/3$ Ternärvektoren zu verknüpfen. Der Aufwand für diese Berechnung beträgt $\frac{N_A}{2} \times \frac{N_B}{3}$. Im Ergebnis der Verknüpfung wird eine TVL aus maximal $N_A/2$ Binärvektoren oder $N_A/4$ Ternärvektoren entstehen. Diese TVL wird mit der TVL aus $N_B/3$ Zeilen, die die Funktion f_1^0 repräsentiert, verknüpft. Die Ergebnis-TVL aus $N_A/4$ Zeilen ist wiederum mit der TVL aus $N_B/3$ Zeilen, die die Funktion f_1^1 repräsentiert, zu verknüpfen. Der Berechungsaufwand der beiden Verknüpfungen beträgt damit $2 \times \frac{N_A}{4} \times \frac{N_B}{3}$. Die Berechnung der Teilfunktionen f_3^0 und f_3^1 geschieht in konstanter Zeit (siehe Formeln (4.51) und (4.52)). Der Aufwand und die Komplexitätsordnung zur Berechnung aller drei Teilfunktionen können somit nach (4.155) bestimmt werden.

$$\begin{aligned} f_{wc} &= \frac{N_A}{2} \times \frac{N_B}{3} \\ &\quad +2 \times \frac{N_A}{4} \times \frac{N_B}{3} + 2c \\ &= \frac{1}{3} \times N_A \times N_B + 2c \\ &= O_{wc}(N_A \times N_B) \end{aligned} \tag{4.155}$$

– Average case:
In diesem Fall ist für die nicht zerlegte Funktion ein Strich-TV aus N_A Zeilen mit einer TVL aus $N_B/2$ (die Hälfte von der maximalen Zeilenzahl) zu verknüpfen. Der Berechnungsaufwand und die Komplexitätsordnung betragen (4.156).

$$f_{ac} = \frac{1}{2} \times N_A \times N_B = O_{ac}(N_A \times N_B) \tag{4.156}$$

Für die zerlegte Funktion lässt sich der Aufwand zur Berechnung der Teilfunktionen f^-, f^0 und f^1 der Ergebnisfunktion und die Komplexitätsordnung nach (4.157) bestimmen. Hier wird davon ausgegangen, dass der Strich-TV aus $N_A/2$ Binärvektoren und die Ternärvektorlisten der Teilfunktionen aus $N_B/6$ Zeilen (die Hälfte von der maximalen Zeilenzahl) bestehen und die Zwischenergebnisse maximal $N_A/4$ Zeilen enthalten.

$$\begin{aligned} f_{ac} &= \frac{N_A}{2} \times \frac{N_B}{6} \\ &\quad +2 \times \frac{N_A}{4} \times \frac{N_B}{6} + 2c \\ &= \frac{1}{6} \times N_A \times N_B + 2c \\ &= O_{ac}(N_A \times N_B) \end{aligned} \tag{4.157}$$

- Disjunktion

  – Worst case:
  Der Berechnungsaufwand und die Komplexitätsordnung für die Berechnung der Disjunktion von zwei nicht zerlegten Funktionen können wie im Fall Konjunktion nach (4.149) ermittelt werden.

  Für die Berechnung der drei Teilfunktionen der Funktion, die durch Disjunktion zweier in drei Teile zerlegten Funktionen entsteht, sind 4 konjunktive Verknüpfungen $f_1^0 f_2^1$, $f_1^1 f_2^0$, $f_1^0 f_2^0$ und $f_1^1 f_2^1$ und eine disjunktive Verknüpfung $f_1^- \vee f_2^-$ durchzuführen (siehe Formeln (4.71), (4.72) und (4.73)). Die Ternärvektorlisten der Funktionen f_1^-, f_1^0, f_1^1, f_2^-, f_2^0 und f_2^1 enthalten $N_A/3$ (oder $N_B/3$) Zeilen, und der Berechnungsaufwand für diese Verknüpfungen beträgt somit $5 \times \frac{N_A}{3} \times \frac{N_B}{3}$.

  Wie den Formeln (4.71), (4.72) und (4.73) entnommen werden kann, müssen zusätzlich die Ausdrücke $\bar{f_1^-} \bar{f_1^0} \bar{f_1^1}$ und $\bar{f_2^-} \bar{f_2^0} \bar{f_2^1}$ berechnet, und die Ergebnisse der Berechnungen mit den Teilfunktionen f_1^0, f_1^1, f_2^0 und f_2^1 konjunktiv verknüpft werden. Der Berechnungsaufwand für die Ausdrücke $\bar{f_1^-} \bar{f_1^0} \bar{f_1^1}$ und $\bar{f_2^-} \bar{f_2^0} \bar{f_2^1}$ beträgt dabei $2 \times \frac{1}{3} \times N_A \times N_B$ (siehe Formel (4.155)). Die vier konjunktiven Verknüpfungen werden mit den TVL aus jeweils $N_A/3$ und $N_B/4$ Zeilen durchgeführt. Die Zahl $N_B/4$ folgt aus der Überlegung, dass im Ergebnis der Negation eine TVL aus maximal $N_A/4$ Zeilen entsteht, da im gegebenen Booleschen Raum maximal $N_A/2$ Binärvektoren möglich sind. Die sechs in den Formeln enthaltenen disjunktiven Verknüpfungen können wegen der Orthogonalität der Teilfunktionen in konstanter Zeit durch Anhängen einer TVL an die anderen realisiert werden.

  Damit kann der Gesamtaufwand und die Komplexitätsordnung nach (4.158) berechnet werden.

$$
\begin{aligned}
f_{wc} &= 5 \times \frac{N_A}{3} \times \frac{N_B}{3} \\
&\quad + \frac{2}{3} \times N_A \times N_B \\
&\quad + 4 \times \frac{N_A}{3} \times \frac{N_B}{4} + 6c \\
&= \frac{14}{9} \times N_A \times N_B + 6c \\
&= O_{wc}(N_A \times N_B)
\end{aligned}
\tag{4.158}
$$

  – Average case:
  Der Berechnungsaufwand und die Komplexitätsordnung für die Berechnung der Disjunktion von zwei nicht zerlegten Funktionen lässt sich auch hier nach (4.151) ermitteln.

  Unter Berücksichtigung der abweichenden Zeilenzahlen der Teilfunktionen von $N_A/6$ und $N_B/6$ in average case, wird in Analogie zu worst case der Berechnungsaufwand und die Komplexitätsordnung für die zerlegten Funktionen nach

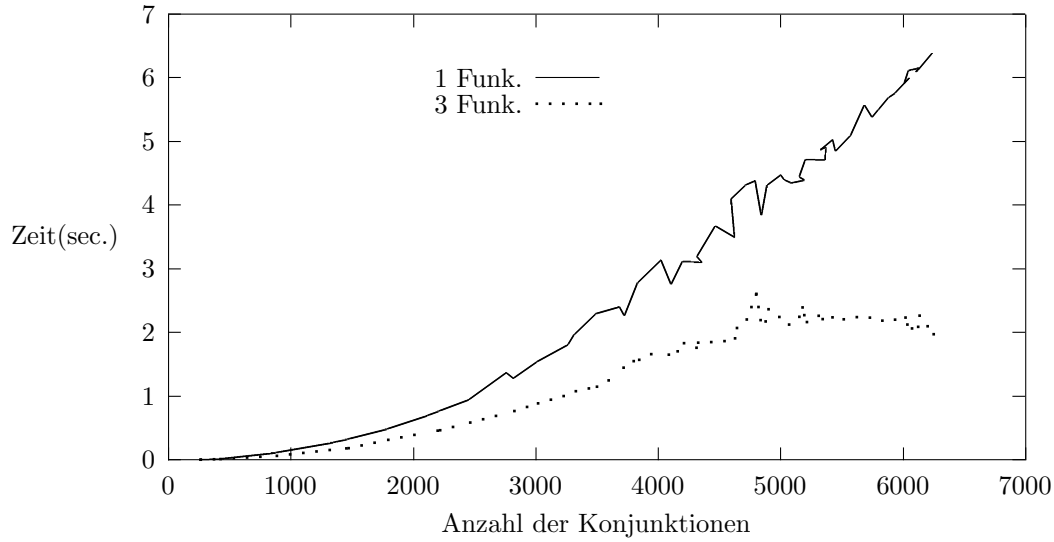


Abbildung 4.8: Berechnungszeit für Operation Konjunktion

(4.159) bestimmt.

$$\begin{aligned} f_{ac} &= 5 \times \frac{N_A}{6} \times \frac{N_B}{6} \\ &\quad + \frac{2}{6} \times N_A \times N_B \\ &\quad + 4 \times \frac{N_A}{6} \times \frac{N_B}{4} + 6c \\ &= \frac{23}{36} \times N_A \times N_B + 6c \\ &= O_{ac}(N_A \times N_B) \end{aligned} \tag{4.159}$$

Den Formeln (4.149)-(4.159) ist zu entnehmen, dass die Tupeloperationen die gleiche Komplexitätsordnung besitzen wie die Operationen mit nicht zerlegten Funktionen. Der detaillierte Vergleich der Formeln (4.149) mit (4.150) und (4.151) mit (4.152) für die Konjunktion und der Formeln (4.153) mit (4.155) und (4.156) mit (4.157) für die Negation zeigt, dass der Zeitaufwand für die Realisierung dieser Operationen durch Zerlegen der Funktionen verringert wird. Auch die Untersuchungen mit Zufallsfunktionen für die Konjunktion und Negation bestätigen diese Aussage.

In den Abbildungen 4.8 und 4.9 sind die Zeiten aufgetragen, die für die Durchführung der Konjunktion zweier Zufallsfunktionen und Negation einer Funktion benötigt werden. Die Operationen wurden mit nicht zerlegten Funktionen und mit den im Ergebnis der Zerlegung in 3 Teile entstandenen Funktionen durchgeführt. Aus der Gegenüberstellung der Kurven ist der wesentlich geringere Zeitbedarf für die Durchführung der Konjunktion und der Negation nach der Zerlegung der Funktionen erkennbar.

Die Komplexitätsordnung bleibt bei der Operation Disjunktion für zerlegte und nicht zerlegte Funktionen gleich, wogegen sich der Zeitaufwand für die Realisierung der Operation nach der Zerlegung vergrößert (siehe Formeln (4.149), (4.158), (4.151) und (4.159)). Eine

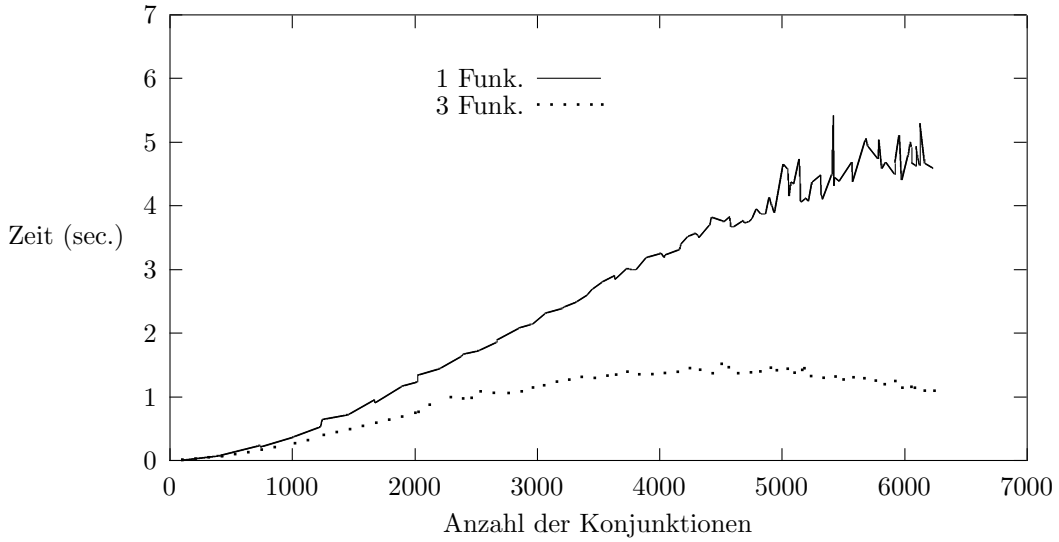


Abbildung 4.9: Berechnungszeit für Operation Negation

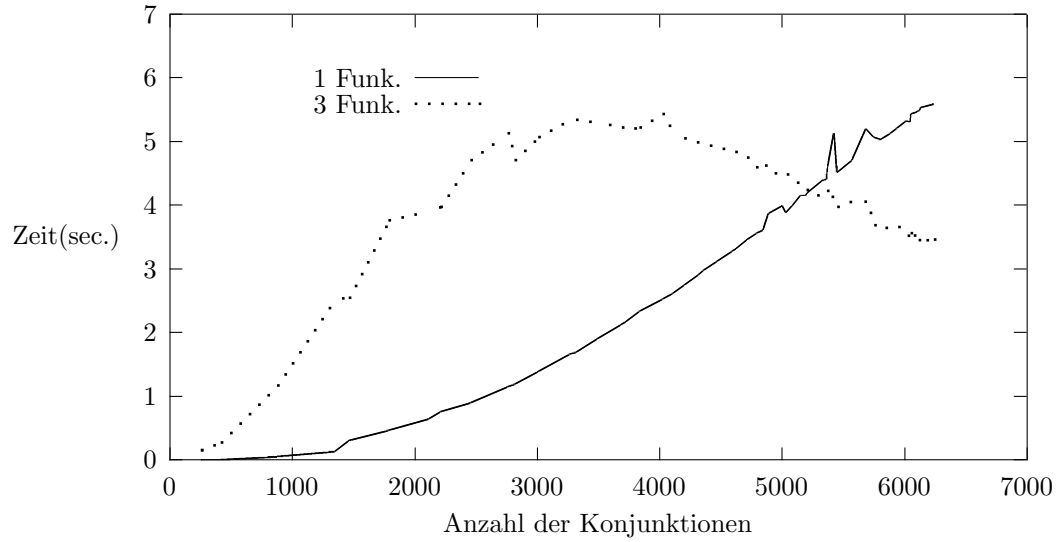


Abbildung 4.10: Berechnungszeit für Operation Disjunktion

Ursache für den erhöhten Zeitaufwand sind zusätzliche Negationen, die beim Durchführen der Disjunktion notwendig werden. In Analogie verwenden auch die Algorithmen der Tupeloperationen Antivalenz und Äquivalenz mehrere Negationen, so dass hier ebenfalls der Zeitbedarf gegenüber den entsprechenden Operationen für nicht zerlegte Funktionen geringfügig steigt.

Die in der Abbildung 4.10 dargestellten Ergebnisse der Disjunktion aus zwei Zufallsfunktionen bestätigen diese Aussage für die Funktionen mit einer begrenzten Anzahl von Konjunktionen. Erst bei den größeren Funktionen werden für die Tupeloperation bessere Ergebnisse erzielt, da dann die negierten Funktionen $\bar{f}^-$, $\bar{f}^0$ und $\bar{f}^1$ kleiner sind und der Ausdruck $\bar{f}^-\bar{f}^0\bar{f}^1$ schneller berechnet werden kann.

Zusammenfassend ist herauszustellen, dass für die Operationen Konjunktion und Negation die Zerlegung einer Funktion in drei Teilfunktionen zur verbesserten Berechnungszeit führt, während für Operationen, deren Algorithmen zusätzliche Negationen enthalten, die Zeitaufwendungen bei einer Zerlegung steigen. Der in Algorithmen von Negation, Disjunktion, Antivalenz und Äquivalenz enthaltene Ausdruck $\bar{f}^-\bar{f}^0\bar{f}^1$ wird beim Ausführen der Operationen stets neu berechnet. Wird dieser Ausdruck bereits beim Zerlegen der Funktionen berechnet und als vierte Teilfunktion im TVL-Tupel abgespeichert, ist eine wesentliche Verminderung des Zeitaufwands für die Ausführung der Operationen möglich. Für die Negation beschränkt sich die Berechnung von Teilfunktionen sogar auf das Vertauschen der Teilfunktionen.

4.2 Zerlegung in vier Funktionen

4.2.1 Zerlegungsprinzip

Aus der Analyse der Möglichkeiten der Darstellung einer Booleschen Funktion durch drei Teilfunktionen folgt, dass die Darstellung einer Funktion durch Tupel aus vier Funktionen (4.160) in Betracht zu ziehen ist.

$$f(\underline{x_0}, x_i) = g(h_1(\underline{x_0}), h_2(\underline{x_0}), h_3(\underline{x_0}), h_4(\underline{x_0}), x_i) \tag{4.160}$$

Die drei Funktionen $h_1(\underline{x_0})$, $h_2(\underline{x_0})$ und $h_3(\underline{x_0})$ sind im Abschnitt 4.1.1 als Funktionen $f^-(\underline{x_0})$ nach (4.20), $f^0(\underline{x_0})$ nach (4.21) und $f^1(\underline{x_0})$ nach (4.22) definiert worden. Die vierte Funktion $h_4(\underline{x_0})$ kann auf folgende Weise definiert werden:

Definition 4.4. *Die Funktion $f^=(\underline{x_0})$ (4.161) ist die Konjunktion der komplementären Funktion $\overline{f(x_i, \underline{x_0})}$ und der komplementären partiellen Ableitung dieser Funktion nach der Variable x_i.*

$$f^=(\underline{x_0}) = \overline{\frac{\partial f(x_i, \underline{x_0})}{\partial x_i}}\, \overline{f(x_i, \underline{x_0})} \tag{4.161}$$

Die Funktion $f^=(\underline{x_0})$ ist von der Variable x_i unabhängig. Das Komplement der Ableitung der Funktion $\overline{f(x_i, \underline{x_0})}$ ist genau dann gleich 1, wenn die Änderung der Variable x_i keine Veränderung des Funktionswerts hervorruft. Durch konjunktive Verknüpfung der Funktion $\overline{f(x_i, \underline{x_0})}$ mit ihrer komplementärer Ableitung nach Variable x_i werden in der Ergebnisfunktion $f^=(\underline{x_0})$ nur die von x_i unabhängigen Konjunktionen enthalten sein.

Satz 4.15. *Eine beliebige Boolesche Funktion $f(x_i, \underline{x_0})$ kann durch die vier Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$, $f^1(\underline{x_0})$ und $f^=(\underline{x_0})$ nach (4.162) dargestellt werden.*

$$f(x_i, \underline{x_0}) = 1 \cdot f^-(\underline{x_0}) \vee \overline{x_i} \cdot f^0(\underline{x_0}) \vee x_i \cdot f^1(\underline{x_0}) \vee 0 \cdot f^=(\underline{x_0}) \tag{4.162}$$

Beweis:

Durch die konjunktiven Verknüpfungen von $f^-(\underline{x_0})$ mit 1 und $f^=(\underline{x_0})$ mit 0 entsteht (4.163).

$$f(x_i, \underline{x_0}) = f^-(\underline{x_0}) \vee \overline{x_i} \cdot f^0(\underline{x_0}) \vee x_i \cdot f^1(\underline{x_0}) \tag{4.163}$$

$$f(x_i, \underline{x_0}) = f(x_i, \underline{x_0}) \tag{4.164}$$

Wegen (4.27) gilt (4.164). □

Wie im Abschnitt 4.1.1 gezeigt wurde, kann eine Boolesche Funktion durch drei Teilfunktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ beschrieben werden. Diese Funktionen werden auch in der Darstellung (4.162) unverändert genutzt. Die Funktion $f^=(\underline{x_0})$ ist in der Darstellung (4.162) mit 0 verknüpft enthalten, und könnte dadurch beliebig gewählt werden. Die nachfolgend beschriebenen Eigenschaften der nach (4.161) definierten Funktion $f^=(\underline{x_0})$ wurden aber so gewählt, dass sich Vereinfachungen beim Durchführen von Operationen zwischen zwei nach (4.162) dargestellten Funktionen ergeben. Die Funktion $f^=(\underline{x_0})$ kann analog der Funktionen der Drei-Tupel-Dekomposition durch die Cofaktoren der Shannon-Dekomposition (2.30) (siehe Abschnitt 2.1.4) nach (4.165) definiert werden.

$$f^=(\underline{x_0}) = \overline{f(x_i = 0, \underline{x_0})} \cdot \overline{f(x_i = 1, \underline{x_0})} \tag{4.165}$$

Satz 4.16. *Die Funktion $f^=(\underline{x_0})$ ist orthogonal zu der Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$:*

$$f^=(\underline{x_0}) \cdot f^\alpha(\underline{x_0}) = 0, \quad \textit{für} \quad \alpha \in \{-, 0, 1\} \tag{4.166}$$

Beweis:

1.
$$f^=(\underline{x_0}) \cdot f^-(\underline{x_0}) = 0 \tag{4.167}$$

Durch das Einsetzen der Definitionen der Funktionen $f^=(\underline{x_0})$ und $f^-(\underline{x_0})$ (siehe (4.161) und (4.20)) in (4.167) folgt (4.168).

$$\frac{\overline{\partial f(x_i, \underline{x_0})}}{\partial x_i} \overline{f(x_i, \underline{x_0})} \cdot \frac{\overline{\partial f(x_i, \underline{x_0})}}{\partial x_i} f(x_i, \underline{x_0}) = 0 \tag{4.168}$$

$$\frac{\overline{\partial f(x_i, \underline{x_0})}}{\partial x_i} \cdot 0 = 0 \tag{4.169}$$

$$0 = 0 \tag{4.170}$$

Die elementaren Umwandlungen von (4.168) führen zu (4.170).

2.

$$f^{=}(\underline{x_0}) \cdot f^{0}(\underline{x_0}) = 0 \tag{4.171}$$

Nach dem Einsetzen der Definitionen der Funktionen $f^{=}(\underline{x_0})$ (4.161) und $f^{0}(\underline{x_0})$ (4.21) in (4.171) ergibt sich (4.172).

$$\overline{\frac{\partial f(x_i, \underline{x_0})}{\partial x_i}}\,\overline{f(x_i, \underline{x_0})} \cdot f(x_i = 0, \underline{x_0}) \frac{\partial f(x_i, \underline{x_0})}{\partial x_i} = 0 \tag{4.172}$$

Da der Ausdruck (4.172) zueinander komplementäre Ableitungen enthält, kann (4.172) in (4.173) umgewandelt werden.

$$0 \cdot \overline{f(x_i, \underline{x_0})} \cdot f(x_i = 0, \underline{x_0}) = 0 \tag{4.173}$$

$$0 = 0 \tag{4.174}$$

3.

$$f^{=}(\underline{x_0}) \cdot f^{1}(\underline{x_0}) = 0 \tag{4.175}$$

Nach dem Einsetzen der Definitionen der Funktionen $f^{=}(\underline{x_0})$ und $f^{1}(\underline{x_0})$ (siehe (4.161) und (4.22)) in (4.175) erhält man (4.176).

$$\overline{\frac{\partial f(x_i, \underline{x_0})}{\partial x_i}}\,\overline{f(x_i, \underline{x_0})} \cdot f(x_i = 1, \underline{x_0}) \frac{\partial f(x_i, \underline{x_0})}{\partial x_i} = 0 \tag{4.176}$$

Analog (4.172) enthält auch (4.176) zueinander komplementäre Ableitungen, so dass dieser Teil des Satzes in Analogie zu Fall (4.171) bewiesen werden kann.

□

Im Satz 4.4 wurde bereits bewiesen, dass die Funktionen $f^{-}(\underline{x_0})$, $f^{0}(\underline{x_0})$ und $f^{1}(\underline{x_0})$ zueinander disjunkt sind. Als Folgerung der Sätze 4.4 und 4.16 gilt, dass alle vier Funktionen $f^{-}(\underline{x_0})$, $f^{0}(\underline{x_0})$, $f^{1}(\underline{x_0})$ und $f^{=}(\underline{x_0})$ zueinander orthogonal sind. Durch die konjunktive Verknüpfung der Funktion $f^{-}(\underline{x_0})$ mit 1, der Funktion $f^{0}(\underline{x_0})$ mit $\bar{x}_i$, der Funktion $f^{1}(\underline{x_0})$ mit x_i und der Funktion $f^{=}(\underline{x_0})$ mit 0 bleibt die Orthogonalität der disjunktiv verknüpften Konjunktionen der Funktionen in (4.162) erhalten. Analog (4.47) gilt für die orthogonalen Funktionen $1 \cdot f^{-}(\underline{x_0})$, $\bar{x}_i f^{0}(\underline{x_0})$, $x_i f^{1}(\underline{x_0})$ und $0 \cdot f^{=}(\underline{x_0})$ nach dem Orthogonalitätssatz, dass die disjunktive Verknüpfung in (4.162) durch die antivalente Verknüpfung ersetzt werden kann (4.177).

$$f(x_i, \underline{x_0}) = 1 \cdot f^{-}(\underline{x_0}) \oplus \overline{x_i} f^{0}(\underline{x_0}) \oplus x_i f^{1}(\underline{x_0}) \oplus 0 \cdot f^{=}(\underline{x_0}) \tag{4.177}$$

Satz 4.17. *Die Funktionen $f^{-}(\underline{x_0})$, $f^{0}(\underline{x_0})$, $f^{1}(\underline{x_0})$ und $f^{=}(\underline{x_0})$ überdecken den Booleschen Raum vollständig:*

$$f^{-}(\underline{x_0}) \vee f^{0}(\underline{x_0}) \vee f^{1}(\underline{x_0}) \vee f^{=}(\underline{x_0}) = 1 \tag{4.178}$$

Beweis:

Nach dem Einsetzen von Definitionen der Funktionen $f^{-}(\underline{x_0})$, $f^{0}(\underline{x_0})$, $f^{1}(\underline{x_0})$ und $f^{=}(\underline{x_0})$ in (4.178) wird (4.179) erhalten. Die elementaren Umformungen von (4.179) führen zu

(4.180).

$$\begin{aligned}
&\overline{\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}} f(x_i,\underline{x_0}) \vee \frac{\partial f(x_i,\underline{x_0})}{\partial x_i} f(x_i=0,\underline{x_0}) \\
\vee &\frac{\partial f(x_i,\underline{x_0})}{\partial x_i} f(x_i=1,\underline{x_0}) \vee \overline{\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}}\,\overline{f(x_i,\underline{x_0})} &= 1 \qquad (4.179)\\
&\overline{\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}} (f(x_i,\underline{x_0}) \vee \overline{f(x_i,\underline{x_0})}) \\
\vee &\frac{\partial f(x_i,\underline{x_0})}{\partial x_i} (f(x_i=0,\underline{x_0}) \vee f(x_i=1,\underline{x_0})) &= 1 \qquad (4.180)
\end{aligned}$$

Aus dem Einsetzen der Definition der Ableitung und dem Ersetzen der Disjunktion durch Antivalenz in der zweiten Zeile der Formel (4.180) resultiert (4.181). Das Ausmultiplizieren führt zu (4.182). Durch weitere elementare Umformungen entsteht (4.183).

$$\begin{aligned}
&\overline{\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}} \vee (f(x_i=0,\underline{x_0}) \oplus f(x_i=1,\underline{x_0})) \\
\cdot &(f(x_i=0,\underline{x_0}) \oplus f(x_i=1,\underline{x_0}) \oplus f(x_i=0,\underline{x_0}) f(x_i=1,\underline{x_0})) &= 1 \qquad (4.181)\\
&\overline{\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}} \vee (f(x_i=0,\underline{x_0}) \oplus f(x_i=1,\underline{x_0}) f(x_i=0,\underline{x_0}) \\
&\oplus f(x_i=0,\underline{x_0}) f(x_i=1,\underline{x_0}) \oplus f(x_i=1,\underline{x_0}) \\
\oplus &f(x_i=0,\underline{x_0}) f(x_i=1,\underline{x_0}) \oplus f(x_i=0,\underline{x_0}) f(x_i=1,\underline{x_0})) &= 1 \qquad (4.182)\\
&\overline{\frac{\partial f(x_i,\underline{x_0})}{\partial x_i}} \vee (f(x_i=0,\underline{x_0}) \oplus f(x_i=1,\underline{x_0})) &= 1 \qquad (4.183)\\
&1 &= 1 \qquad (4.184)
\end{aligned}$$

In der Formel (4.183) kann die Definition der Ableitung eingesetzt werden. Damit folgt aus (4.183) die Formel (4.184). □

Die in den Sätzen 4.4, 4.16 und 4.17 formulierten Eigenschaften der Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$, $f^1(\underline{x_0})$ und $f^=(\underline{x_0})$ sind deutlich zu erkennen, wenn die Funktionen durch die Cofaktoren der Shannon-Dekomposition beschrieben werden. Jeder der beiden Cofaktoren zerlegt zusammen mit seinem Komplement die Menge aller Binärvektoren des Booleschen Raums B^k. Wie aus der Abbildung 4.11 hervorgeht, werden diese beiden unabhängigen Zerlegungen von den vier gewällten Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$, $f^1(\underline{x_0})$ und $f^=(\underline{x_0})$ miteinander kombiniert, so dass diese vier Funktionen paarweise zueinander disjunkt alle Binärvektoren das Boolesche Raum B^k überdecken.

Folgerung der Sätze 4.4, 4.16 und 4.17: Die Dekomposition (4.162) ist die eindeutig bestimmte Zerlegung Z der Menge B der Binärvektoren eines Booleschen Raums. Die Binärvektoren für die die Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$, $f^1(\underline{x_0})$ und $f^=(\underline{x_0})$ den Wert 1 annehmen, sind die disjunkten Teilmengen der Menge B, deren Vereinigung die Menge B ist.

Satz 4.18. *Zwischen den Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$, $f^1(\underline{x_0})$ und $f^=(\underline{x_0})$ existieren folgende Zusammenhänge:*

$$f^-(\underline{x_0}) = \overline{f^0(\underline{x_0}) \vee f^1(\underline{x_0}) \vee f^=(\underline{x_0})} = \overline{f^0(\underline{x_0})}\;\overline{f^1(\underline{x_0})}\;\overline{f^=(\underline{x_0})} \qquad (4.185)$$

	$f(x_i = 0, \underline{x_0})$	$\overline{f(x_i = 0, \underline{x_0})}$
$f(x_i = 1, \underline{x_0})$	$f^-(\underline{x_0}) = f(x_i = 0, \underline{x_0}) \cdot f(x_i = 1, \underline{x_0})$	$f^1(\underline{x_0}) = \overline{f(x_i = 0, \underline{x_0})} \cdot f(x_i = 1, \underline{x_0})$
$\overline{f(x_i = 1, \underline{x_0})}$	$f^0(\underline{x_0}) = f(x_i = 0, \underline{x_0}) \cdot \overline{f(x_i = 1, \underline{x_0})}$	$f^=(\underline{x_0}) = \overline{f(x_i = 0, \underline{x_0})} \cdot \overline{f(x_i = 1, \underline{x_0})}$

Abbildung 4.11: Eigenschaften der Teilfunktionen

$$f^0(\underline{x_0}) = \overline{f^-(\underline{x_0}) \vee f^1(\underline{x_0}) \vee f^=(\underline{x_0})} = \overline{f^-(\underline{x_0})}\ \overline{f^1(\underline{x_0})}\ \overline{f^=(\underline{x_0})} \tag{4.186}$$

$$f^1(\underline{x_0}) = \overline{f^-(\underline{x_0}) \vee f^0(\underline{x_0}) \vee f^=(\underline{x_0})} = \overline{f^-(\underline{x_0})}\ \overline{f^0(\underline{x_0})}\ \overline{f^=(\underline{x_0})} \tag{4.187}$$

$$f^=(\underline{x_0}) = \overline{f^-(\underline{x_0}) \vee f^0(\underline{x_0}) \vee f^1(\underline{x_0})} = \overline{f^-(\underline{x_0})}\ \overline{f^0(\underline{x_0})}\ \overline{f^1(\underline{x_0})} \tag{4.188}$$

Beweis:

Nach dem Einsetzen von Definitionen der Funktion $f^0(\underline{x_0})$, $f^1(\underline{x_0})$ und $f^=(\underline{x_0})$ in den Ausdruck (4.189) (siehe auch (4.185)) und nach dem Ausklammern der Ableitung ergibt sich (4.190). Weitere Umformungen von (4.190) führen zu (4.193). Nach Definition 4.1 ist der Ausdruck (4.193) gleich die Funktion $f^-(\underline{x_0})$.

$$\overline{f^=(\underline{x_0})}\ \overline{f^0(\underline{x_0})}\ \overline{f^1(\underline{x_0})} = \overline{f^=(\underline{x_0}) \vee f^0(\underline{x_0}) \vee f^1(\underline{x_0})} \tag{4.189}$$

$$= \overline{\overline{\frac{\partial f(x_i, \underline{x_0})}{\partial x_i}}\,\overline{f(x_i, \underline{x_0})} \vee (f(x_i = 0, \underline{x_0}) \vee f(x_i = 1, \underline{x_0}))\frac{\partial f(x_i, \underline{x_0})}{\partial x_i}} \tag{4.190}$$

$$= \overline{\overline{\frac{\partial f(x_i, \underline{x_0})}{\partial x_i}}\,\overline{f(x_i, \underline{x_0})} \vee \frac{\partial f(x_i, \underline{x_0})}{\partial x_i}} \tag{4.191}$$

$$= \left(\frac{\partial f(x_i, \underline{x_0})}{\partial x_i} \vee f(x_i, \underline{x_0})\right) \overline{\frac{\partial f(x_i, \underline{x_0})}{\partial x_i}} \tag{4.192}$$

$$= \overline{\frac{\partial f(x_i, \underline{x_0})}{\partial x_i}} f(x_i, \underline{x_0}) \tag{4.193}$$

$$= f^-(\underline{x_0}) \tag{4.194}$$

Damit ist die Richtigkeit der Gleichung (4.185) bewiesen.

Die Richtigkeit der Gleichungen (4.186), (4.187) und (4.188) kann analog zur Gleichung (4.185) durch Einsetzen von Definitionen der Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$, $f^1(\underline{x_0})$ und $f^=(\underline{x_0})$ in den rechten Teil der jeweiligen Gleichung und durch weitere elementare Umformungen bewiesen werden. □

Aus den Zusammenhängen (4.185), (4.186), (4.187) und (4.188) folgt, dass die Konjunktion aus drei beliebigen negierten Teilfunktionen durch die vierte nicht negierte Teilfunktion ersetzt werden kann. Diese Aussage kann weiter verallgemeinert werden.

Satz 4.19. *Für die vier Teilfunktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$, $f^1(\underline{x_0})$ und $f^=(\underline{x_0})$ der Funktion $f(x_i, \underline{x_0})$ gilt allgemein, dass die negierten Teilfunktionen und ihre konjunktive Verknüpfungen durch die anderen nicht negierten Teilfunktionen oder ihre disjunktive Verknüpfungen ersetzt werden können:*

1.

$$\begin{array}{l}\overline{f^{\alpha}(\underline{x_0})} = f^{\beta}(\underline{x_0}) \vee f^{\gamma}(\underline{x_0}) \vee f^{\delta}(\underline{x_0}), \\ \textit{für} \quad \alpha,\beta,\gamma,\delta \in \{-,0,1,=\} \quad \textit{und} \quad \alpha \neq \beta \neq \gamma \neq \delta\end{array} \tag{4.195}$$

2.

$$\begin{array}{l}\overline{f^{\alpha}(\underline{x_0}) \vee f^{\beta}(\underline{x_0})} = \overline{f^{\alpha}(\underline{x_0})}\ \overline{f^{\beta}(\underline{x_0})} = f^{\gamma}(\underline{x_0}) \vee f^{\delta}(\underline{x_0}), \\ \textit{für} \quad \alpha,\beta,\gamma,\delta \in \{-,0,1,=\} \quad \textit{und} \quad \alpha \neq \beta \neq \gamma \neq \delta\end{array} \tag{4.196}$$

3.

$$\begin{array}{l}\overline{f^{\alpha}(\underline{x_0}) \vee f^{\beta}(\underline{x_0}) \vee f^{\gamma}(\underline{x_0})} = \overline{f^{\alpha}(\underline{x_0})}\ \overline{f^{\beta}(\underline{x_0})}\ \overline{f^{\gamma}(\underline{x_0})} = f^{\delta}(\underline{x_0}), \\ \textit{für} \quad \alpha,\beta,\gamma,\delta \in \{-,0,1,=\} \quad \textit{und} \quad \alpha \neq \beta \neq \gamma \neq \delta\end{array} \tag{4.197}$$

Beweis:

1. Die Ausdrücke (4.198), (4.199), (4.200) und (4.201) entstehen durch das Negieren beider Seiten der Gleichungen (4.185), (4.186), (4.187) und (4.188).

$$\overline{f^{-}(\underline{x_0})} = f^{0}(\underline{x_0}) \vee f^{1}(\underline{x_0}) \vee f^{=}(\underline{x_0}) \tag{4.198}$$

$$\overline{f^{0}(\underline{x_0})} = f^{-}(\underline{x_0}) \vee f^{1}(\underline{x_0}) \vee f^{=}(\underline{x_0}) \tag{4.199}$$

$$\overline{f^{1}(\underline{x_0})} = f^{-}(\underline{x_0}) \vee f^{0}(\underline{x_0}) \vee f^{=}(\underline{x_0}) \tag{4.200}$$

$$\overline{f^{=}(\underline{x_0})} = f^{-}(\underline{x_0}) \vee f^{0}(\underline{x_0}) \vee f^{1}(\underline{x_0}) \tag{4.201}$$

2. Werden die nach (4.198) und (4.199) dargestellten negierten Funktionen $\overline{f^{-}(\underline{x_0})}$ und $\overline{f^{0}(\underline{x_0})}$ miteinander konjunktiv verknüpft, wird für den Audruck $\overline{f^{-}(\underline{x_0})}\ \overline{f^{0}(\underline{x_0})}$ der Ausdruck (4.202) erhalten. Nach dem Ausmultiplizieren der Disjunktionen unter Berücksichtigung der Orthogonalität der Funktionen $f^{-}(\underline{x_0})$, $f^{0}(\underline{x_0})$, $f^{1}(\underline{x_0})$ und $f^{=}(\underline{x_0})$ entsteht (4.203).

$$\begin{array}{rcll}\overline{f^{-}(\underline{x_0}) \vee f^{0}(\underline{x_0})} & = & \overline{f^{-}(\underline{x_0})}\ \overline{f^{0}(\underline{x_0})} & \\ & = & (f^{0}(\underline{x_0}) \vee f^{1}(\underline{x_0}) \vee f^{=}(\underline{x_0})) & \\ & & \cdot(f^{-}(\underline{x_0}) \vee f^{1}(\underline{x_0}) \vee f^{=}(\underline{x_0})) & (4.202) \\ & = & f^{1}(\underline{x_0}) \vee f^{=}(\underline{x_0}) & (4.203)\end{array}$$

In analoger Weise entstehen für die Verknüpfungen $\overline{f^{-}(\underline{x_0})}\ \overline{f^{1}(\underline{x_0})}$, $\overline{f^{-}(\underline{x_0})}\ \overline{f^{=}(\underline{x_0})}$, $\overline{f^{0}(\underline{x_0})}\ \overline{f^{1}(\underline{x_0})}$, $\overline{f^{0}(\underline{x_0})}\ \overline{f^{=}(\underline{x_0})}$ und $\overline{f^{1}(\underline{x_0})}\ \overline{f^{=}(\underline{x_0})}$ die Ausdrucke (4.204), (4.205), (4.206), (4.207) und (4.208).

$$\overline{f^{-}(\underline{x_0}) \vee f^{1}(\underline{x_0})} = \overline{f^{-}(\underline{x_0})}\ \overline{f^{1}(\underline{x_0})} = f^{0}(\underline{x_0}) \vee f^{=}(\underline{x_0}) \tag{4.204}$$

$$\overline{f^{-}(\underline{x_0}) \vee f^{=}(\underline{x_0})} = \overline{f^{-}(\underline{x_0})}\ \overline{f^{=}(\underline{x_0})} = f^{0}(\underline{x_0}) \vee f^{1}(\underline{x_0}) \tag{4.205}$$

$$\overline{f^{0}(\underline{x_0}) \vee f^{1}(\underline{x_0})} = \overline{f^{0}(\underline{x_0})}\ \overline{f^{1}(\underline{x_0})} = f^{-}(\underline{x_0}) \vee f^{=}(\underline{x_0}) \tag{4.206}$$

$$\overline{f^{0}(\underline{x_0}) \vee f^{=}(\underline{x_0})} = \overline{f^{0}(\underline{x_0})}\ \overline{f^{=}(\underline{x_0})} = f^{-}(\underline{x_0}) \vee f^{1}(\underline{x_0}) \tag{4.207}$$

$$\overline{f^{1}(\underline{x_0}) \vee f^{=}(\underline{x_0})} = \overline{f^{1}(\underline{x_0})}\ \overline{f^{=}(\underline{x_0})} = f^{-}(\underline{x_0}) \vee f^{0}(\underline{x_0}) \tag{4.208}$$

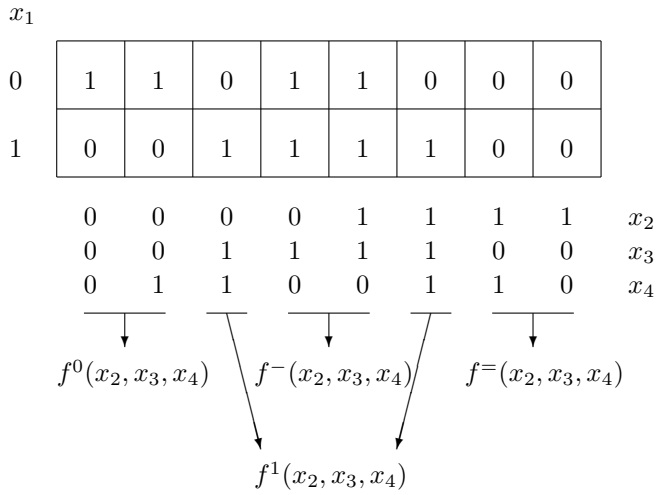


Abbildung 4.12: Zerlegung der Funktion $f(x_1, x_2, x_3, x_4)$ in vier Teilfunktionen

3. Die Richtigkeit der Aussage (4.197) geht aus den Gleichungen (4.185), (4.186), (4.187) und (4.188) unmittelbar hervor. □

Als Beispiel für die Darstellung einer Booleschen Funktion durch vier Funktionen kann an dieser Stelle die schon im Abschnitt 4.1.1 verwendete Funktion (4.48) betrachtet werden. Beim Zerlegen dieser Funktion nach Variable x_1 entstehen die vier Teilfunktionen (4.209) (siehe auch die Abbildung 4.12).

$$\begin{aligned} f^-(x_2, x_3, x_4) &= x_3\bar{x}_4 \\ f^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3 \\ f^1(x_2, x_3, x_4) &= x_3x_4 \\ f^=(x_2, x_3, x_4) &= x_2\bar{x}_3 \end{aligned} \tag{4.209}$$

4.2.2 Operationen

Negation

Satz 4.20. *Für die gegebene Funktion f_1 mit jeweils vier Teilfunktionen ${f_1}^-$, ${f_1}^0$, ${f_1}^1$ und ${f_1}^=$ (4.162) können die Teilfunktionen ${f_3}^-$, ${f_3}^0$, ${f_3}^1$ und ${f_3}^=$ der negierten Funktion $f_3 = \bar{f}_1$ nach (4.210)-(4.213) berechnet werden.*

$${f_3}^- = {f_1}^= \tag{4.210}$$

$${f_3}^0 = {f_1}^1 \tag{4.211}$$

$${f_3}^1 = {f_1}^0 \tag{4.212}$$

$${f_3}^= = {f_1}^- \tag{4.213}$$

Beweisschritt 1:

Die Teilfunktionen $f_3{}^{-}$, $f_3{}^{0}$ und $f_3{}^{1}$ der nach (4.162) dargestellten Funktion f_3 (wobei $x = x_i$) sind mit den gleichnamigen Funktionen der Zerlegung (4.27) identisch und können, wie bereits im Satz 4.5 bewiesen, nach (4.50)-(4.52) berechnet werden. Nachdem in (4.50) für den Ausdruck $\bar{f}_1^{-}\bar{f}_1^{0}\bar{f}_1^{1}$ die Teilfunktion $f_1^{=}$ nach (4.188) eingesetzt wird, ergibt sich (4.210). Damit ist die Richtigkeit der Aussagen (4.210)-(4.212) bewiesen.

Beweisschritt 2:

Die Funktion $f_3{}^{=}$ kann nach (4.188) aus den Funktionen $f_3{}^{-}$, $f_3{}^{0}$ und $f_3{}^{1}$ berechnet werden:

$$f_3^{=} = \overline{f_3^{-} \vee f_3^{0} \vee f_3^{1}} \tag{4.214}$$

Nach dem Einsetzen der Ausdrücke (4.210)-(4.212) in (4.214) für die Funktionen $f_3{}^{-}$, $f_3{}^{0}$ und $f_3{}^{1}$ ergibt sich (4.215).

$$f_3^{=} = \overline{f_1^{=} \vee f_1^{1} \vee f_1^{0}} \tag{4.215}$$

Nach (4.185) kann der Ausdruck (4.215) durch die Funktion $f_3{}^{-}$ ersetzt werden. Damit ist auch (4.213) bewiesen. □

Satz 4.21. [6] *Die nach (4.210), (4.211), (4.212) und (4.213) berechneten Teilfunktionen f_3^{-}, f_3^{0}, f_3^{1} und $f_3^{=}$ (4.162) der Funktion $f_3 = \bar{f}_1$ sind zueinander paarweise disjunkt:*

$$f_3^{\alpha} \cdot f_3^{\beta} = 0, \quad \textit{für} \quad \alpha, \beta \in \{-, 0, 1, =\} \quad \textit{und} \quad \alpha \neq \beta \tag{4.216}$$

Für die Funktion (4.68), die als Operand für die Negation in Abschnitt 4.1.2 verwendet wurde, sind die Teilfunktionen mit (4.217) zu beschreiben.

$$\begin{aligned} f_1^{-}(x_2, x_3, x_4) &= x_3\bar{x}_4 \\ f_1^{0}(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3 \\ f_1^{1}(x_2, x_3, x_4) &= x_3x_4 \\ f_1^{=}(x_2, x_3, x_4) &= x_2\bar{x}_3 \end{aligned} \tag{4.217}$$

Die Teilfunktionen der Ergebnisfunktion $f_3(\underline{x}) = \overline{f_1(\underline{x})}$ werden mit (4.218) beschrieben.

$$\begin{aligned} f_3^{-}(x_2, x_3, x_4) &= x_2\bar{x}_3 \\ f_3^{0}(x_2, x_3, x_4) &= x_3x_4 \\ f_3^{1}(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3 \\ f_3^{=}(x_2, x_3, x_4) &= x_3\bar{x}_4 \end{aligned} \tag{4.218}$$

In der Abbildung 4.13 sind die Funktionen $f_1(\underline{x})$ (4.68) mit den Teilfunktionen (4.217) und $f_3(\underline{x})$ mit den Teilfunktionen (4.218) dargestellt. Es ist erkennbar, dass die Funktion $f_3^{0}(\underline{x})$ sich aus der Funktion $f_1^{1}(\underline{x})$ ergibt und die Funktion $f_3^{1}(\underline{x})$ der Funktion $f_1^{0}(\underline{x})$ entspricht. Die Funktion $f_3^{-}(\underline{x})$ ist gleich die Funktion $f_1^{=}(\underline{x})$, und die Funktion $f_3^{=}(\underline{x})$ ergibt sich aus der Funktion $f_1^{-}(\underline{x})$.

[6] Beweis siehe Anhang B

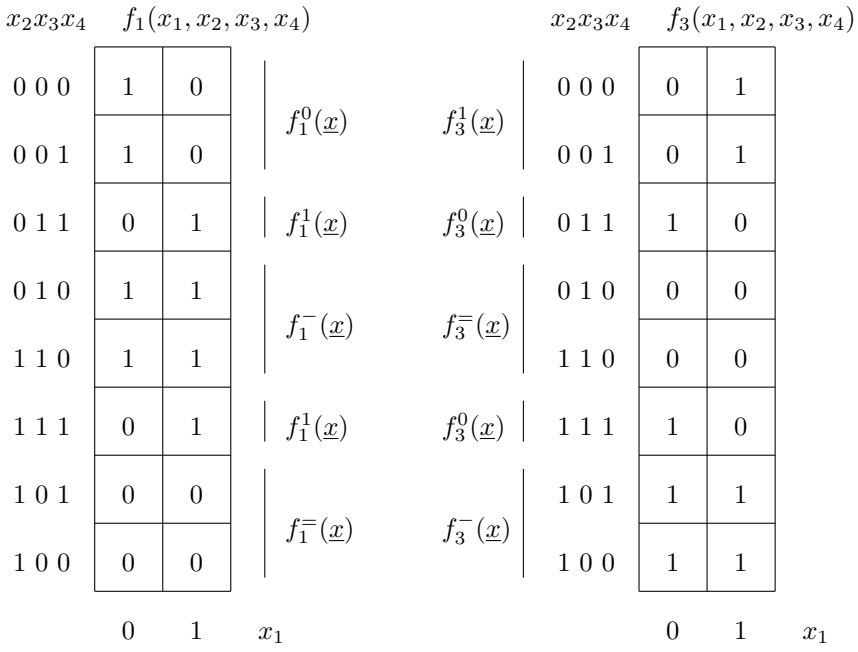


Abbildung 4.13: Funktion $f_3(x_1,x_2,x_3,x_4) = \overline{f_1(x_1,x_2,x_3,x_4)}$

Disjunktion

Satz 4.22. *Für die gegebenen Funktionen f_1 und f_2 mit jeweils vier Teilfunktionen ${f_1}^-$, ${f_1}^0$, ${f_1}^1$, ${f_1}^=$ und ${f_2}^-$, ${f_2}^0$, ${f_2}^1$, ${f_2}^=$ (4.162) können die Teilfunktionen ${f_3}^-$, ${f_3}^0$, ${f_3}^1$ und ${f_3}^=$ der Funktion $f_3 = f_1 \vee f_2$ nach (4.219)-(4.222) berechnet werden.*

$$ {f_3}^- = {f_1}^- \vee {f_2}^- \vee {f_1}^0 {f_2}^1 \vee {f_1}^1 {f_2}^0 \tag{4.219} $$

$$ {f_3}^0 = {f_1}^0 {f_2}^0 \vee {f_1}^0 {f_2}^= \vee {f_1}^= {f_2}^0 \tag{4.220} $$

$$ {f_3}^1 = {f_1}^1 {f_2}^1 \vee {f_1}^1 {f_2}^= \vee {f_1}^= {f_2}^1 \tag{4.221} $$

$$ {f_3}^= = {f_1}^= {f_2}^= \tag{4.222} $$

Beweisschritt 1:

Die Teilfunktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ der nach (4.162) dargestellten Funktion f_3 (wobei $x = x_i$) sind mit den Funktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ der Dekomposition (4.27) identisch. Wie bereits im Satz 4.7 bewiesen, ist die Berechnung der Teilfunktionen nach (4.71)-(4.73) möglich. Werden für die Ausdrücke $\bar{f_1^-}\bar{f_1^0}\bar{f_1^1}$ und $\bar{f_2^-}\bar{f_2^0}\bar{f_2^1}$ in (4.72) und (4.73) die Teilfunktionen $f_1^=$ bzw. $f_2^=$ nach (4.188) eingesetzt, entstehen die Formeln (4.219)-(4.221).

Beweisschritt 2:

Zur Berechnung der Funktion ${f_3}^=$ sind die Funktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ nach (4.188) zu verwenden (4.223). Nach dem Einsetzen der Ausdrücke (4.219)-(4.221) für die Funktionen

${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ wird (4.224) erhalten.

$$f_3^= = \overline{f_3^- \vee f_3^0 \vee f_3^1} \tag{4.223}$$

$$\begin{aligned} f_3^= = \; & \overline{{f_1}^- \vee {f_2}^- \vee {f_1}^0 {f_2}^1 \vee {f_1}^1 {f_2}^0} \\ & \overline{\vee {f_1}^0 {f_2}^0 \vee {f_1}^0 f_2^= \vee f_1^= {f_2}^0 \vee {f_1}^1 {f_2}^1 \vee {f_1}^1 f_2^= \vee f_1^= {f_2}^1} \end{aligned} \tag{4.224}$$

Für die Funktionen f_1^-, f_1^0, f_1^1, $f_1^=$, f_2^-, f_2^0, f_2^1 und $f_2^=$ gelten nach dem Satz 4.17 die Aussagen (4.225) und (4.226).

$$1 = f_1^- \vee f_1^0 \vee f_1^1 \vee f_1^= \tag{4.225}$$

$$1 = f_2^- \vee f_2^0 \vee f_2^1 \vee f_2^= \tag{4.226}$$

Die Ausdrücke (4.225) und (4.226) können in (4.224) eingesetzt werden (4.227).

$$\begin{aligned} f_3^= = \; & \overline{{f_1}^-(f_2^- \vee f_2^0 \vee f_2^1 \vee f_2^=) \vee {f_2}^-(f_1^- \vee f_1^0 \vee f_1^1 \vee f_1^=) \vee {f_1}^0 {f_2}^1 \vee {f_1}^1 {f_2}^0} \\ & \overline{\vee {f_1}^0 {f_2}^0 \vee {f_1}^0 f_2^= \vee f_1^= {f_2}^0 \vee {f_1}^1 {f_2}^1 \vee {f_1}^1 f_2^= \vee f_1^= {f_2}^1} \end{aligned} \tag{4.227}$$

Nach dem Ausmultiplizieren und nachfolgendem Ausklammern von ${f_1}^-$, ${f_1}^0$, ${f_1}^1$, ${f_1}^=$ und ${f_2}^-$, ${f_2}^0$, ${f_2}^1$, ${f_2}^=$ entsteht (4.228). Die weiteren elementaren Umwandlungen (4.229) unter Berücksichtigung von (4.225) und (4.226) führen zu (4.230).

$$\begin{aligned} f_3^= = \; & \overline{(f_1^- \vee f_1^0 \vee f_1^1 \vee f_1^=)(f_2^- \vee f_2^0 \vee f_2^1)} \\ & \overline{\vee (f_2^- \vee f_2^0 \vee f_2^1 \vee f_2^=)(f_1^- \vee f_1^0 \vee f_1^1)} \end{aligned} \tag{4.228}$$

$$f_3^= = \overline{f_1^- \vee f_1^0 \vee f_1^1 \vee f_2^- \vee f_2^0 \vee f_2^1} \tag{4.229}$$

$$f_3^= = (\overline{f_1^- \vee f_1^0 \vee f_1^1})(\overline{f_2^- \vee f_2^0 \vee f_2^1}) \tag{4.230}$$

Unter Berücksichtigung von (4.188) ist eine Umwandlung des Ausdrucks (4.230) in (4.231) möglich.

$$f_3^= = f_1^= f_2^= \tag{4.231}$$

□

Satz 4.23. [7] *Die nach (4.219), (4.220), (4.221) und (4.222) berechneten Teilfunktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ (4.162) der Funktion $f_3 = f_1 \vee f_2$ sind zueinander paarweise disjunkt:*

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \textit{für} \quad \alpha, \beta \in \{-, 0, 1, =\} \quad \textit{und} \quad \alpha \neq \beta \tag{4.232}$$

Für die im Abschnitt 4.2.1 verwendeten Funktionen $f_1(x_1, x_2, x_3, x_4)$ (Formel 4.233) und $f_2(x_1, x_2, x_3, x_4)$ (Formel 4.235) mit den Teilfunktionen (4.234) und (4.236) werden die Teilfunktionen der Funktion $f_3(\underline{x}) = f_1(\underline{x}) \vee f_2(\underline{x})$ mit (4.237) beschrieben (siehe auch Abbildung 4.14).

$$f_1(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_3 x_4 \vee x_3 \bar{x}_4 \tag{4.233}$$

$$\begin{aligned} f_1^-(x_2, x_3, x_4) &= x_3 \bar{x}_4 \\ f_1^0(x_2, x_3, x_4) &= \bar{x}_2 \bar{x}_3 \\ f_1^1(x_2, x_3, x_4) &= x_3 x_4 \\ f_1^=(x_2, x_3, x_4) &= x_2 \bar{x}_3 \end{aligned} \tag{4.234}$$

[7] Beweis siehe Anhang B

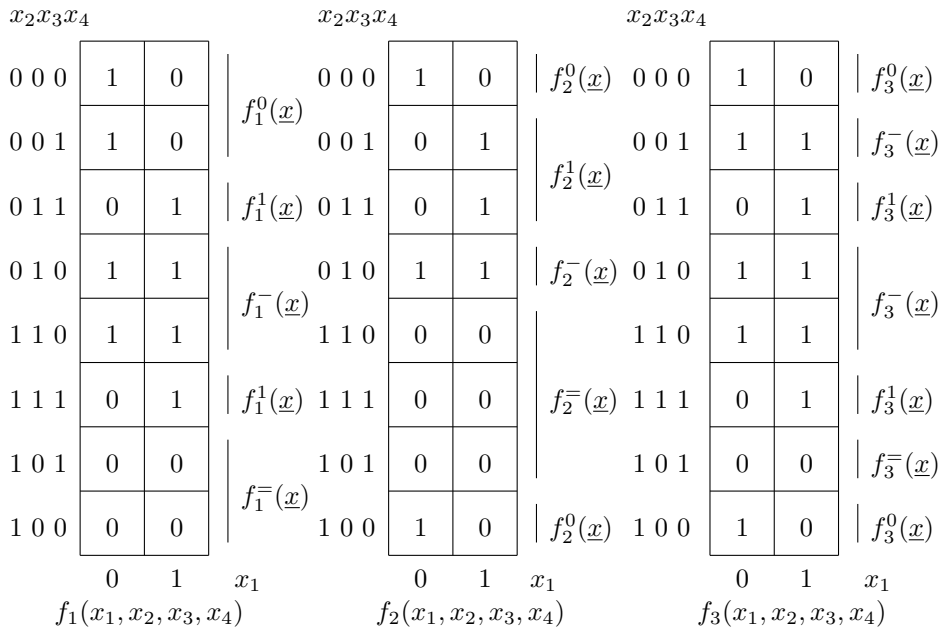


Abbildung 4.14: Funktion $f_3(x_1, x_2, x_3, x_4) = f_1(x_1, x_2, x_3, x_4) \vee f_2(x_1, x_2, x_3, x_4)$

$$f_2(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_3\bar{x}_4 \vee x_1\bar{x}_2x_4 \vee \bar{x}_2x_3\bar{x}_4 \tag{4.235}$$

$$\begin{aligned} f_2^-(x_2, x_3, x_4) &= \bar{x}_2x_3\bar{x}_4 \\ f_2^0(x_2, x_3, x_4) &= \bar{x}_3\bar{x}_4 \\ f_2^1(x_2, x_3, x_4) &= \bar{x}_2x_4 \\ f_2^-(x_2, x_3, x_4) &= x_2x_3 \vee x_2\bar{x}_3x_4 \end{aligned} \tag{4.236}$$

$$\begin{aligned} f_3^-(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3x_4 \vee x_3\bar{x}_4 \\ f_3^0(x_2, x_3, x_4) &= \bar{x}_3\bar{x}_4 \\ f_3^1(x_2, x_3, x_4) &= x_3x_4 \\ f_3^-(x_2, x_3, x_4) &= x_2\bar{x}_3x_4 \end{aligned} \tag{4.237}$$

Die Abbildung 4.14 zeigt, dass die Funktion $f_3^-(\underline{x})$ aus der Disjunktion der Funktionen $f_1^-(\underline{x})$ und $f_2^-(\underline{x})$ und der Konjunktion $\bar{x}_2\bar{x}_3x_4$ entsteht, die aus der konjunktiven Verknüpfung $f_1^0(\underline{x})f_2^1(\underline{x})$ gebildet wird. Die in der Funktion $f_3^0(\underline{x})$ enthaltenen Konjunktionen werden durch die Verknüpfungen $f_1^0(\underline{x})f_2^0(\underline{x})$ (die Konjunktion $\bar{x}_2\bar{x}_3\bar{x}_4$) und $f_2^0(\underline{x})f_1^=(\underline{x})$ (die Konjunktion $x_2\bar{x}_3\bar{x}_4$) gebildet. Die Konjunktion $\bar{x}_2x_3x_4$ der Funktion $f_3^1(\underline{x})$ entstand durch die Verknüpfung $f_1^1(\underline{x})f_2^1(\underline{x})$. Die weitere Konjunktion der Funktion $f_3^1(\underline{x})$ ist das Ergebnis der Verknüpfung $f_1^1(\underline{x})f_2^=(\underline{x})$. Die Funktion $f_3^=(\underline{x})$ ergibt sich aus der konjunktiven Verknüpfung $f_2^=(\underline{x})f_1^=(\underline{x})$.

Konjunktion

Satz 4.24. *Für die gegebenen Funktionen f_1 und f_2, die nach (4.162) durch Teilfunktionen $f_1{}^-$, $f_1{}^0$, $f_1{}^1$, $f_1{}^=$ und $f_2{}^-$, $f_2{}^0$, $f_2{}^1$, $f_2{}^=$ dargestellt sind, können die Teilfunktionen $f_3{}^-$,*

${f_3}^0$, ${f_3}^1$, ${f_3}^=$ der Funktion $f_3 = f_1 \cdot f_2$ nach (4.238)-(4.241) berechnet werden.

$$ {f_3}^- = {f_1}^- {f_2}^- \tag{4.238} $$

$$ {f_3}^0 = {f_1}^- {f_2}^0 \vee {f_1}^0 {f_2}^- \vee {f_1}^0 {f_2}^0 \tag{4.239} $$

$$ {f_3}^1 = {f_1}^- {f_2}^1 \vee {f_1}^1 {f_2}^- \vee {f_1}^1 {f_2}^1 \tag{4.240} $$

$$ {f_3}^= = {f_1}^1 {f_2}^0 \vee {f_1}^0 {f_2}^1 \vee {f_1}^= \vee {f_2}^= \tag{4.241} $$

Beweisschritt 1:

Die Teilfunktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ der nach (4.162) dargestellten Funktion f_3 (wobei $x = x_i$) sind mit den Funktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ der Zerlegung (4.27) identisch. Sie können, wie bereits im Satz 4.9 bewiesen, nach (4.101)-(4.103) berechnet werden.

Beweisschritt 2:

Zur Berechnung der Funktion ${f_3}^=$ werden die Funktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ nach (4.188) verwendet (4.242). Nach dem Einsetzen der Ausdrücke (4.238)-(4.240) für die Funktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ wird (4.243) erhalten.

$$ {f_3}^= = \overline{{f_3}^- \vee {f_3}^0 \vee {f_3}^1} \tag{4.242} $$

$$ {f_3}^= = \overline{{f_1}^- {f_2}^- \vee {f_1}^- {f_2}^0 \vee {f_1}^0 {f_2}^- \vee {f_1}^0 {f_2}^0 \vee {f_1}^- {f_2}^1 \vee {f_1}^1 {f_2}^- \vee {f_1}^1 {f_2}^1} \tag{4.243} $$

Weitere elementare Umformungen führen zu (4.245).

$$ {f_3}^= = \overline{({f_1}^- \vee {f_1}^0)({f_2}^- \vee {f_2}^0) \vee ({f_1}^- \vee {f_1}^1)({f_2}^- \vee {f_2}^1)} \tag{4.244} $$

$$ {f_3}^= = (\bar{{f_1}^-}\bar{{f_1}^0} \vee \bar{{f_2}^-}\bar{{f_2}^0})(\bar{{f_1}^-}\bar{{f_1}^1} \vee \bar{{f_2}^-}\bar{{f_2}^1}) \tag{4.245} $$

Die negierten Funktionen in den Ausdrücken $\bar{{f_1}^-}\bar{{f_1}^0}$, $\bar{{f_2}^-}\bar{{f_2}^1}$, $\bar{{f_1}^-}\bar{{f_1}^1}$ und $\bar{{f_2}^-}\bar{{f_2}^0}$ können durch die nicht negierten nach (4.196) ersetzt werden.

$$ {f_3}^= = ({f_1}^1 \vee {f_1}^= \vee {f_2}^1 \vee {f_2}^=)({f_1}^0 \vee {f_1}^= \vee {f_2}^0 \vee {f_2}^=) \tag{4.246} $$

Nach dem Ausmultiplizieren entsteht (4.247), wobei einige Konjunktionen wegen Orthogonalität der Funktionen ${f_1}^0$, ${f_1}^1$ und ${f_1}^=$ und der Funktionen ${f_2}^0$, ${f_2}^1$ und ${f_2}^=$ entfallen bzw. von den Funktionen ${f_1}^=$ und ${f_2}^=$ absorbiert werden.

$$ {f_3}^= = {f_1}^= \vee {f_1}^1 {f_2}^0 \vee {f_1}^0 {f_2}^1 \vee {f_2}^= \tag{4.247} $$

□

Satz 4.25. [8] *Die nach (4.238), (4.239), (4.240) und (4.241) berechneten Teilfunktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ (4.162) der Funktion $f_3 = f_1 \cdot f_2$ sind zueinander paarweise disjunkt:*

$$ f_3^\alpha \cdot f_3^\beta = 0, \quad \textit{für} \quad \alpha, \beta \in \{-, 0, 1, =\} \quad \textit{und} \quad \alpha \neq \beta \tag{4.248} $$

[8] Beweis siehe Anhang B

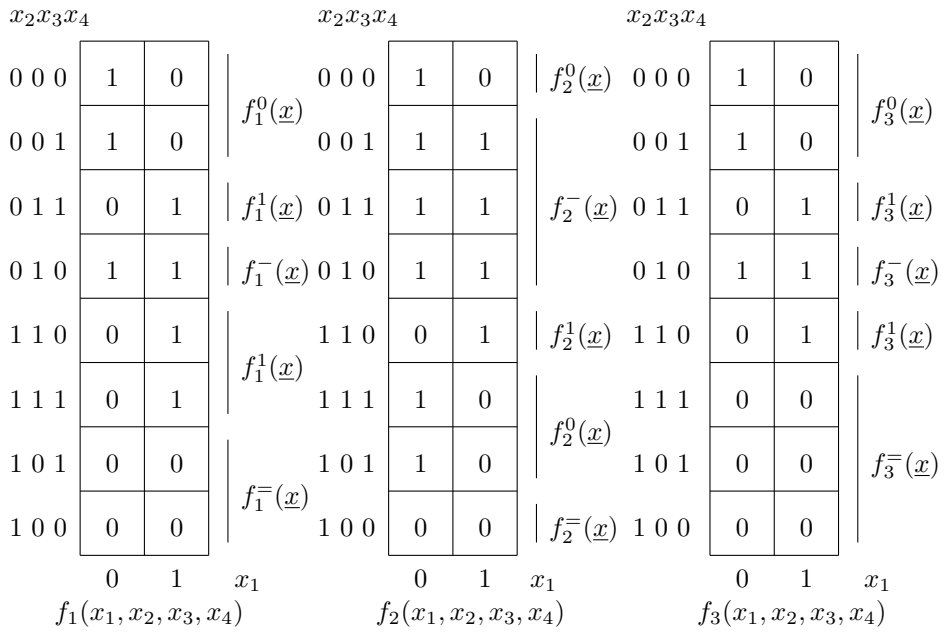


Abbildung 4.15: Funktion $f_3(x_1, x_2, x_3, x_4) = f_1(x_1, x_2, x_3, x_4) \cdot f_2(x_1, x_2, x_3, x_4)$

Für die Beispielfunktionen (4.249) und (4.251) mit den Teilfunktionen (4.250) und (4.252) werden die Teilfunktionen der Funktion $f_3(\underline{x}) = f_1(\underline{x}) \cdot f_2(\underline{x})$ mit (4.253) beschrieben.

$$f_1(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1x_3 \vee \bar{x}_1\bar{x}_2x_3\bar{x}_4 \tag{4.249}$$

$$\begin{aligned} f_1^-(x_2, x_3, x_4) &= \bar{x}_2x_3\bar{x}_4 \\ f_1^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3 \\ f_1^1(x_2, x_3, x_4) &= \bar{x}_2x_3x_4 \vee x_2x_3 \\ f_1^=(x_2, x_3, x_4) &= x_2\bar{x}_3 \end{aligned} \tag{4.250}$$

$$f_2(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2 \vee x_1\bar{x}_2x_4 \vee x_1x_3\bar{x}_4 \vee \bar{x}_1x_2x_4 \tag{4.251}$$

$$\begin{aligned} f_2^-(x_2, x_3, x_4) &= \bar{x}_2x_4 \vee \bar{x}_2x_3\bar{x}_4 \\ f_2^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3\bar{x}_4 \vee x_2x_4 \\ f_2^1(x_2, x_3, x_4) &= x_2x_3\bar{x}_4 \\ f_2^=(x_2, x_3, x_4) &= x_2\bar{x}_3\bar{x}_4 \end{aligned} \tag{4.252}$$

$$\begin{aligned} f_3^-(x_2, x_3, x_4) &= \bar{x}_2x_3\bar{x}_4 \\ f_3^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3 \\ f_3^1(x_2, x_3, x_4) &= \bar{x}_2x_3x_4 \vee x_2x_3\bar{x}_4 \\ f_3^=(x_2, x_3, x_4) &= x_2x_3x_4 \vee x_2\bar{x}_3 \end{aligned} \tag{4.253}$$

In der Abbildung 4.15 sind die oben beschiebenen Funktionen $f_1(x_1, x_2, x_3, x_4)$, $f_2(x_1, x_2, x_3, x_4)$ und $f_3(x_1, x_2, x_3, x_4)$ und ihre Teilfunktionen mittels eines Karnaughplans dargestellt. Die neue Funktion $f_3^-(\underline{x})$ entsteht durch konjunktive Verknüpfung der Funktionen $f_1^-(\underline{x})$ und $f_2^-(\underline{x})$. Die Konjunktionen der Funktion $f_3^0(\underline{x})$ werden durch Verknüpfungen

$f_1^0(\underline{x})f_2^0(\underline{x}) = \bar{x}_2\bar{x}_3\bar{x}_4$ und $f_1^0(\underline{x})f_2^-(\underline{x}) = \bar{x}_2\bar{x}_3x_4$ gebildet. Weiterhin entstehen die Konjunktionen der Funktion $f_3^1(\underline{x})$ durch Verknüpfungen $f_1^1(\underline{x})f_2^-(\underline{x})$ und $f_1^1(\underline{x})f_2^1(\underline{x})$. Die Konjunktion $x_2x_3x_4$ der Funktion $f_3^=(\underline{x})$ wird durch die konjunktive Verknüpfung $f_1^1(\underline{x})f_2^0(\underline{x})$ gebildet. Alle weiteren Konjunktionen der Funktion $f_3^=(\underline{x})$ sind entweder in der Funktion $f_1^=(\underline{x})$ oder in den beiden Funktionen $f_1^=(\underline{x})$ und $f_2^=(\underline{x})$ enthalten.

Antivalenz

Satz 4.26. *Für die gegebenen Funktionen f_1 und f_2 mit jeweils vier Teilfunktionen ${f_1}^-$, ${f_1}^0$, ${f_1}^1$, ${f_1}^=$ und ${f_2}^-$, ${f_2}^0$, ${f_2}^1$, ${f_2}^=$ (4.162) können die Teilfunktionen ${f_3}^-$, ${f_3}^0$, ${f_3}^1$ und ${f_3}^=$ der Funktion $f_3 = f_1 \oplus f_2$ nach (4.254)-(4.257) berechnet werden.*

$$ {f_3}^- = {f_1}^0{f_2}^1 \vee {f_1}^1{f_2}^0 \vee {f_1}^-{f_2}^= \vee {f_1}^={f_2}^- \tag{4.254} $$

$$ {f_3}^0 = {f_1}^1{f_2}^- \vee {f_1}^-{f_2}^1 \vee {f_1}^0{f_2}^= \vee {f_1}^={f_2}^0 \tag{4.255} $$

$$ {f_3}^1 = {f_1}^0{f_2}^- \vee {f_1}^-{f_2}^0 \vee {f_1}^1{f_2}^= \vee {f_1}^={f_2}^1 \tag{4.256} $$

$$ {f_3}^= = {f_1}^-{f_2}^- \vee {f_1}^0{f_2}^0 \vee {f_1}^1{f_2}^1 \vee {f_1}^={f_2}^= \tag{4.257} $$

Beweisschritt 1:

Die Teilfunktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ der nach (4.162) dargestellten Funktion f_3 (wobei $x = x_i$) sind mit den Funktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ der Drei-Tupel-Dekomposition (4.27) identisch und können, wie schon im Satz 4.11 bewiesen, nach (4.115)-(4.117) berechnet werden. Werden in (4.115), (4.116) und (4.117) die Ausdrücke $\bar{f_1}^-\bar{f_1}^0\bar{f_1}^1$ und $\bar{f_2}^-\bar{f_2}^0\bar{f_2}^1$ durch die Teilfunktion $f_1^=$ bzw. $f_2^=$ nach (4.188) ersetzt, ergeben sich die Ausdrücke (4.254)-(4.256).

Beweisschritt 2:

Die Funktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ können für die Berechnung der Funktion ${f_3}^=$ genutzt werden (siehe (4.188)). Nach der Substitution der Funktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ durch Ausdrücke (4.254)-(4.256) wird (4.259) erhalten. Beim Ausklammern der Funktionen ${f_1}^-$, ${f_1}^0$ ${f_1}^1$ und ${f_1}^=$ entsteht der Ausdruck (4.260).

$$ f_3^= = \overline{f_3^- \vee f_3^0 \vee f_3^1} \tag{4.258} $$

$$ \begin{aligned} f_3 = {} & \overline{{f_1}^0{f_2}^1 \vee {f_1}^1{f_2}^0 \vee {f_1}^-{f_2}^= \vee {f_1}^={f_2}^- \vee {f_1}^1{f_2}^- \vee {f_1}^-{f_2}^1} \\ & \overline{\vee {f_1}^0{f_2}^= \vee {f_1}^={f_2}^0 \vee {f_1}^0{f_2}^- \vee {f_1}^-{f_2}^0 \vee {f_1}^1{f_2}^= \vee {f_1}^={f_2}^1} \end{aligned} \tag{4.259} $$

$$ \begin{aligned} f_3^= = {} & \overline{{f_1}^0({f_2}^1 \vee {f_2}^= \vee {f_2}^-) \vee {f_1}^1({f_2}^0 \vee {f_2}^- \vee {f_2}^=)} \\ & \overline{\vee {f_1}^-({f_2}^= \vee {f_2}^1 \vee {f_2}^0) \vee {f_1}^=({f_2}^- \vee {f_2}^0 \vee {f_2}^1)} \end{aligned} \tag{4.260} $$

Die Disjunktionen im Ausdruck (4.260) lassen sich nach (4.195) durch negierte Funktionen $\bar{f_2}^0$, $\bar{f_2}^1$, $\bar{f_2}^-$ und $\bar{f_2}^=$ ersetzen (siehe Formel (4.261)).

$$ f_3^= = \overline{{f_1}^0\bar{f_2}^0 \vee {f_1}^1\bar{f_2}^1 \vee {f_1}^-\bar{f_2}^- \vee {f_1}^=\bar{f_2}^=} \tag{4.261} $$

$$ f_3^= = (\bar{f_1}^0 \vee {f_2}^0)(\bar{f_1}^1 \vee {f_2}^1)(\bar{f_1}^- \vee {f_2}^-)(\bar{f_1}^= \vee {f_2}^=) \tag{4.262} $$

Der Ausdruck (4.263) entsteht beim Ausmultiplizieren der Disjunktionen in (4.262). Einige der entstandenen Konjunktionen entfallen auf Grund der Orthogonalität der Funktionen

f_2^-, f_2^0, f_2^1 und $f_2^=$. Ein Ersatz der negierten Funktionen im Ausdruck (4.263) durch nicht negierte Funktionen ist nach (4.197) möglich (siehe die Formel (4.264)).

$$\begin{aligned} f_3^= &= \bar{f_1}^0 \bar{f_1}^1 \bar{f_1}^- f_2^= \vee \bar{f_1}^0 \bar{f_1}^1 \bar{f_1}^= f_2^- \\ &\quad \vee \bar{f_1}^0 \bar{f_1}^- \bar{f_1}^= f_2^1 \vee \bar{f_1}^1 \bar{f_1}^- \bar{f_1}^= f_2^0 \qquad (4.263) \\ f_3^= &= f_1^= f_2^= \vee f_1^- f_2^- \vee f_1^1 f_2^1 \vee f_1^0 f_2^0 \qquad (4.264) \end{aligned}$$

□

Satz 4.27. [9] *Die nach (4.254), (4.255), (4.256) und (4.257) berechneten Teilfunktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ (4.162) der Funktion $f_3 = f_1 \oplus f_2$ sind zueinander paarweise disjunkt:*

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \textit{für} \quad \alpha, \beta \in \{-, 0, 1, =\} \quad \textit{und} \quad \alpha \neq \beta \qquad (4.265)$$

Im folgenden Beispiel werden die Funktionen (4.266) und (4.268) über die Teilfunktionen (4.267) und (4.269) antivalent verknüpft. Die entstandenen Teilfunktionen der Ergebnisfunktion $f_3(\underline{x}) = f_1(\underline{x}) \oplus f_2(\underline{x})$ werden mit (4.270) beschrieben.

$$f_1(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_3 x_4 \vee x_3 \bar{x}_4 \qquad (4.266)$$

$$\begin{aligned} f_1^-(x_2, x_3, x_4) &= x_3 \bar{x}_4 \\ f_1^0(x_2, x_3, x_4) &= \bar{x}_2 \bar{x}_3 \\ f_1^1(x_2, x_3, x_4) &= x_3 x_4 \\ f_1^=(x_2, x_3, x_4) &= x_2 \bar{x}_3 \end{aligned} \qquad (4.267)$$

$$f_2(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \qquad (4.268)$$

$$\begin{aligned} f_2^-(x_2, x_3, x_4) &= \bar{x}_2 \bar{x}_3 \bar{x}_4 \\ f_2^0(x_2, x_3, x_4) &= x_2 \bar{x}_3 \bar{x}_4 \\ f_2^1(x_2, x_3, x_4) &= \bar{x}_2 x_4 \vee \bar{x}_2 x_3 \bar{x}_4 \\ f_2^=(x_2, x_3, x_4) &= x_2 x_3 \vee x_2 \bar{x}_3 x_4 \end{aligned} \qquad (4.269)$$

$$\begin{aligned} f_3^-(x_2, x_3, x_4) &= \bar{x}_2 \bar{x}_3 x_4 \vee x_2 x_3 \bar{x}_4 \\ f_3^0(x_2, x_3, x_4) &= \bar{x}_2 x_3 \bar{x}_4 \vee x_2 \bar{x}_3 \bar{x}_4 \\ f_3^1(x_2, x_3, x_4) &= \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_2 x_3 x_4 \\ f_3^=(x_2, x_3, x_4) &= \bar{x}_2 x_3 x_4 \vee x_2 \bar{x}_3 x_4 \end{aligned} \qquad (4.270)$$

In der Abbildung 4.16 sind die Funktionen (4.266) und (4.268) und die Funktion $f_3(\underline{x})$ mit ihren Teilfunktionen dargestellt. Es ist ersichtlich, dass die Konjunktionen der Funktion $f_3^-(\underline{x})$ das Ergebnis der Verknüpfungen $f_1^0(\underline{x}) f_2^1(\underline{x})$ und $f_1^-(\underline{x}) f_2^=(\underline{x})$ sind und die Konjunktionen der Funktion $f_3^0(\underline{x})$ durch die Verknüpfungen $f_1^-(\underline{x}) f_2^1(\underline{x})$ und $f_1^=(\underline{x}) f_2^0(\underline{x})$ entstehen. Weiterhin werden die Konjunktionen der Funktion $f_3^1(\underline{x})$ als $f_1^0(\underline{x}) f_2^-(\underline{x})$ und $f_1^1(\underline{x}) f_2^=(\underline{x})$ berechnet und die Konjunktionen der Funktion $f_3^=(\underline{x})$ durch die Verknüpfungen $f_1^1(\underline{x}) f_2^1(\underline{x})$ und $f_1^=(\underline{x}) f_2^=(\underline{x})$ gebildet.

[9] Beweis siehe Anhang B

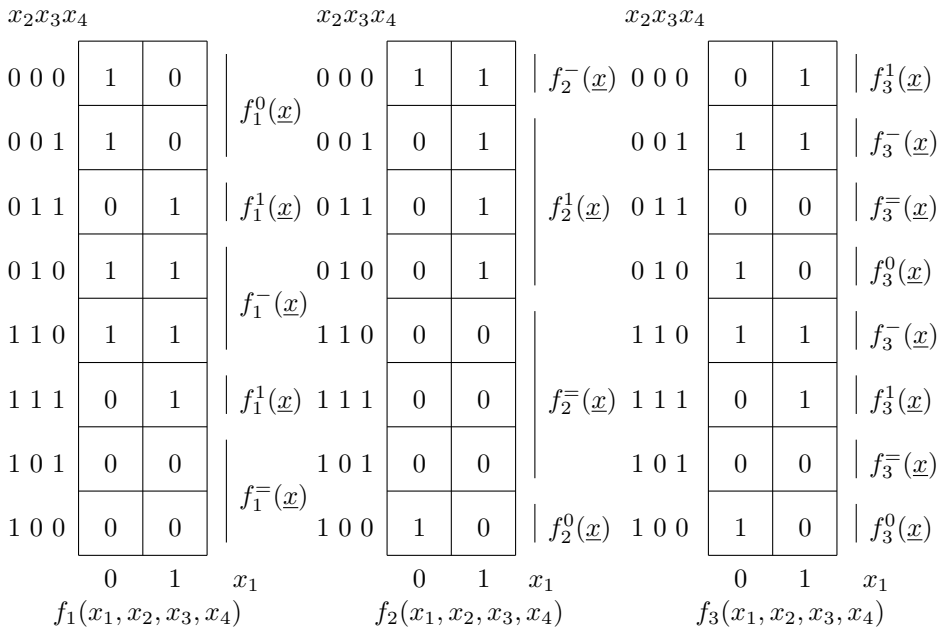


Abbildung 4.16: Funktion $f_3(x_1, x_2, x_3, x_4) = f_1(x_1, x_2, x_3, x_4) \oplus f_2(x_1, x_2, x_3, x_4)$

Äquivalenz

Satz 4.28. *Für die gegebenen Funktionen f_1 und f_2 mit jeweils vier Teilfunktionen ${f_1}^-$, ${f_1}^0$, ${f_1}^1$, ${f_1}^=$ und ${f_2}^-$, ${f_2}^0$, ${f_2}^1$, ${f_2}^=$ (4.162) können die Teilfunktionen ${f_3}^-$, ${f_3}^0$, ${f_3}^1$ und ${f_3}^=$ der Funktion $f_3 = f_1 \odot f_2$ nach (4.271)-(4.274) berechnet werden.*

$$ {f_3}^- = {f_1}^- {f_2}^- \vee {f_1}^0 {f_2}^0 \vee {f_1}^1 {f_2}^1 \vee {f_1}^= {f_1}^= \tag{4.271} $$

$$ {f_3}^0 = {f_1}^- {f_2}^0 \vee {f_1}^0 {f_2}^- \vee {f_1}^1 {f_2}^= \vee {f_1}^= {f_2}^1 \tag{4.272} $$

$$ {f_3}^1 = {f_1}^- {f_2}^1 \vee {f_1}^1 {f_2}^- \vee {f_1}^0 {f_2}^= \vee {f_1}^= {f_2}^0 \tag{4.273} $$

$$ {f_3}^= = {f_1}^0 {f_2}^1 \vee {f_1}^1 {f_2}^0 \vee {f_1}^- {f_2}^= \vee {f_1}^= {f_2}^- \tag{4.274} $$

Beweisschritt 1:

Die Teilfunktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ der nach (4.162) dargestellten Funktion f_3 (wobei $x = x_i$), sind mit den Funktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ der Zerlegung (4.27) identisch. Wie bereits im Satz 4.13 bewiesen, ist eine Berechnung der Teilfunktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ nach (4.133)-(4.135) möglich. Durch Substitution der Ausdrücke $\bar{f_1}^- \bar{f_1}^0 \bar{f_1}^1$ und $\bar{f_2}^- \bar{f_2}^0 \bar{f_2}^1$ in (4.133), (4.134) und (4.135) durch die Teilfunktion ${f_1}^=$ bzw. ${f_2}^=$ nach (4.188) werden die Formeln (4.271)-(4.273) erhalten.

Beweisschritt 2:

Die Berechnung der Funktion ${f_3}^=$ ist mit Hilfe der Funktionen ${f_3}^-$, ${f_3}^0$ und ${f_3}^1$ möglich (siehe (4.188)), wobei nach dem Ersetzen dieser Funktionen durch Ausdrücke (4.271)-(4.273) der Ausdruck (4.276) erhalten wird. Nach dem Ausklammern der Funktionen ${f_1}^-$, ${f_1}^0$ ${f_1}^1$

und $f_1{}^{=}$ entsteht (4.277).

$$f_3^{=} = \overline{f_3^{-} \vee f_3^{0} \vee f_3^{1}} \tag{4.275}$$

$$\begin{aligned} f_3^{=} = \; & \overline{f_1{}^{-} f_2{}^{-} \vee f_1{}^{0} f_2{}^{0} \vee f_1{}^{1} f_2{}^{1} \vee f_1{}^{=} f_1{}^{=} \vee f_1{}^{-} f_2{}^{0} \vee f_1{}^{0} f_2{}^{-}} \\ & \overline{\vee f_1{}^{1} f_2{}^{=} \vee f_1{}^{=} f_2{}^{1} \vee f_1{}^{-} f_2{}^{1} \vee f_1{}^{1} f_2{}^{-} \vee f_1{}^{0} f_2{}^{=} \vee f_1{}^{=} f_2{}^{0}} \end{aligned} \tag{4.276}$$

$$\begin{aligned} f_3^{=} = \; & \overline{f_1{}^{-}(f_2{}^{-} \vee f_2{}^{0} \vee f_2{}^{1}) \vee f_1{}^{0}(f_2{}^{0} \vee f_2{}^{-} \vee f_2{}^{=})} \\ & \overline{\vee f_1{}^{1}(f_2{}^{1} \vee f_2{}^{=} \vee f_2{}^{-}) \vee f_1{}^{=}(f_2{}^{=} \vee f_2{}^{1} \vee f_2{}^{0})} \end{aligned} \tag{4.277}$$

Die Disjunktionen im Ausdruck (4.277) können nach (4.195) durch negierte Funktionen $\bar{f}_2{}^{0}$, $\bar{f}_2{}^{1}$, $\bar{f}_2{}^{-}$ und $\bar{f}_2{}^{=}$ ersetzt werden. Dabei entsteht der Ausdruck (4.278).

$$f_3^{=} = \overline{f_1{}^{-} \bar{f}_2{}^{=} \vee f_1{}^{0} \bar{f}_2{}^{1} \vee f_1{}^{1} \bar{f}_2{}^{0} \vee f_1{}^{=} \bar{f}_2{}^{-}} \tag{4.278}$$

$$f_3^{=} = (\bar{f}_1{}^{-} \vee f_2{}^{=})(\bar{f}_1{}^{0} \vee f_2{}^{1})(\bar{f}_1{}^{1} \vee f_2{}^{0})(\bar{f}_1{}^{=} \vee f_2{}^{-}) \tag{4.279}$$

Durch Ausmultiplizieren der Disjunktionen in (4.279) erhält man den Ausdruck (4.280). Einige der entstandenen Konjunktionen können dabei wegen Orthogonalität der Funktionen f_2^{-}, f_2^{0}, f_2^{1} und $f_2^{=}$ entfallen. Ein Ersatz der negierten Funktionen im Ausdruck (4.280) durch nicht negierte Funktionen ist nach (4.197) möglich und führt zu (4.281).

$$\begin{aligned} f_3^{=} = \; & \bar{f}_1{}^{-} \bar{f}_1{}^{0} \bar{f}_1{}^{1} f_2{}^{-} \vee \bar{f}_1{}^{-} \bar{f}_1{}^{0} f_2{}^{0} \bar{f}_1{}^{=} \\ & \vee \bar{f}_1{}^{-} f_2{}^{1} \bar{f}_1{}^{1} \bar{f}_1{}^{=} \vee f_2{}^{=} \bar{f}_1{}^{0} \bar{f}_1{}^{1} \bar{f}_1{}^{=} \end{aligned} \tag{4.280}$$

$$f_3^{=} = f_1{}^{=} f_2{}^{-} \vee f_1{}^{1} f_2{}^{0} \vee f_1{}^{0} f_2{}^{1} \vee f_1{}^{-} f_2{}^{=} \tag{4.281}$$

□

Satz 4.29. [10] *Die nach (4.271), (4.272), (4.273) und (4.274) berechneten Teilfunktionen f_3^{-}, f_3^{0}, f_3^{1} und $f_3^{=}$ (4.162) der Funktion $f_3 = f_1 \odot f_2$ sind zueinander paarweise disjunkt:*

$$f_3^{\alpha} \cdot f_3^{\beta} = 0, \quad \textit{für} \quad \alpha, \beta \in \{-, 0, 1, =\} \quad \textit{und} \quad \alpha \neq \beta \tag{4.282}$$

In der Abbildung 4.17 sind die Funktionen $f_1(\underline{x})$ (4.283) und $f_2(\underline{x})$ (4.285) mit ihren Teilfunktionen (4.284) und (4.286) sowie die Funktion $f_3(\underline{x}) = f_1(\underline{x}) \odot f_2(\underline{x})$ einschließlich ihrer Teilfunktionen (4.287) dargestellt.

$$f_1(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_3 x_4 \vee x_3 \bar{x}_4 \tag{4.283}$$

$$\begin{aligned} f_1^{-}(x_2, x_3, x_4) &= x_3 \bar{x}_4 \\ f_1^{0}(x_2, x_3, x_4) &= \bar{x}_2 \bar{x}_3 \\ f_1^{1}(x_2, x_3, x_4) &= x_3 x_4 \\ f_1^{=}(x_2, x_3, x_4) &= x_2 \bar{x}_3 \end{aligned} \tag{4.284}$$

$$f_2(x_1, x_2, x_3, x_4) = \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \tag{4.285}$$

$$\begin{aligned} f_2^{-}(x_2, x_3, x_4) &= \bar{x}_2 \bar{x}_3 \bar{x}_4 \\ f_2^{0}(x_2, x_3, x_4) &= \bar{x}_2 x_4 \vee x_2 \bar{x}_3 \bar{x}_4 \\ f_2^{1}(x_2, x_3, x_4) &= \bar{x}_2 x_3 \bar{x}_4 \\ f_2^{=}(x_2, x_3, x_4) &= x_2 x_3 \vee x_2 \bar{x}_3 x_4 \end{aligned} \tag{4.286}$$

[10] Beweis siehe Anhang B

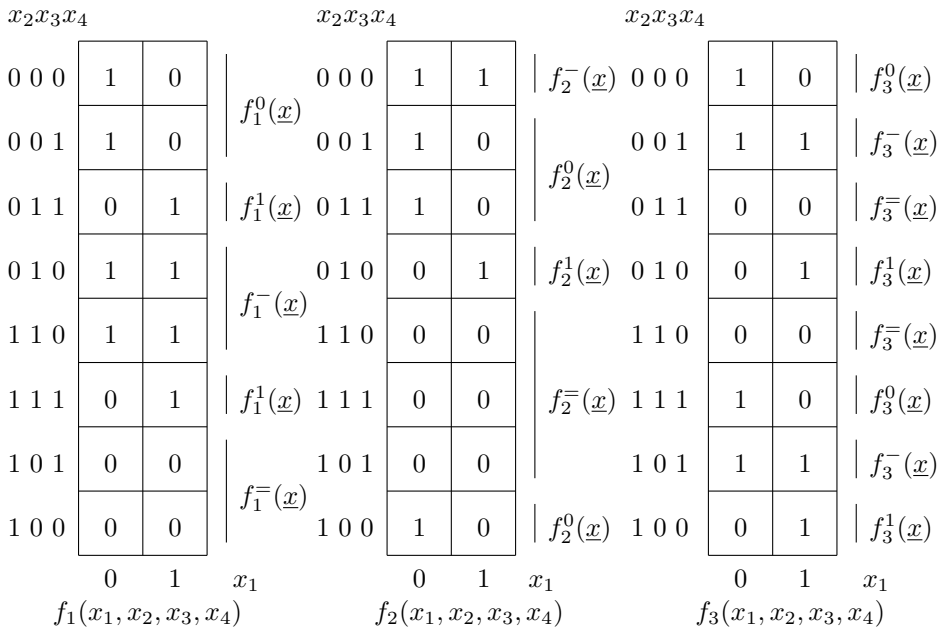


Abbildung 4.17: Funktion $f_3(x_1, x_2, x_3, x_4) = f_1(x_1, x_2, x_3, x_4) \odot f_2(x_1, x_2, x_3, x_4)$

$$\begin{aligned} f_3^-(x_2, x_3, x_4) &= \bar{x}_3 x_4 \\ f_3^0(x_2, x_3, x_4) &= \bar{x}_2\bar{x}_3\bar{x}_4 \vee x_2x_3x_4 \\ f_3^1(x_2, x_3, x_4) &= \bar{x}_2x_3\bar{x}_4 \vee x_2\bar{x}_3\bar{x}_4 \\ f_3^=(x_2, x_3, x_4) &= \bar{x}_2x_3x_4 \vee x_2x_3\bar{x}_4 \end{aligned} \tag{4.287}$$

Die Teilfunktion $f_3^-(\underline{x})$ besteht aus zwei Konjunktionen, die durch die Verknüpfungen $f_1^0(\underline{x})f_2^0(\underline{x})$ und $f_1^=(\underline{x})f_2^=(\underline{x})$ gebildet werden. Die Konjunktionen der Teilfunktion $f_3^0(\underline{x})$ lassen sich wie folgt berechnen: $\bar{x}_2\bar{x}_3\bar{x}_4 = f_1^0(\underline{x})f_2^-(\underline{x})$ und $x_2x_3x_4 = f_1^1(\underline{x})f_2^=(\underline{x})$. Die Konjunktionen der Teilfunktion $f_3^1(\underline{x})$ sind das Ergebnis der Verknüpfungen $f_1^-(\underline{x})f_2^1(\underline{x}) = \bar{x}_2x_3\bar{x}_4$ und $f_1^=(\underline{x})f_2^0(\underline{x}) = x_2\bar{x}_3\bar{x}_4$. Die Teilfunktion $f_3^=(\underline{x})$ der Funktion $f_3(x_1, \underline{x})$ besteht aus zwei Konjunktionen, die sich aus Verknüpfungen $f_1^1(\underline{x})f_2^0(\underline{x})$ und $f_1^-(\underline{x})f_2^=(\underline{x})$ ergeben.

4.2.3 Analyse

Um die Vor- und Nachteile der Zerlegung einer Funktion in vier Teile zu analysieren, werden hier die bereits im Abschnitt 4.1.3 diskutierten Kriterien herangezogen. Es handelt sich um:

- den Speicherplatz, der zur Abspeicherung der Booleschen Funktion benötigt wird,
- die Zeit, die für das Ausführen der Grundoperationen zwischen zwei Booleschen Funktionen notwendig ist.

Der Speicherplatzbedarf zur Dastellung der vier Teilfunktionen kann in worst case nicht den Speicherplatzbedarf zur Dastellung einer nicht zerlegten Funktion mit n Variablen oder drei Teilfunktionen überschreiten, wobei die im Abschnitt 4.1.3 genannten Gründe anzugeben sind: Aus der Orthogonalität der vier Teilfunktionen folgt, dass die Größe der vier Funktionen in der Summe 2^{n-1} Ternärvektoren nicht überschreitet. Dieser Wert entspricht

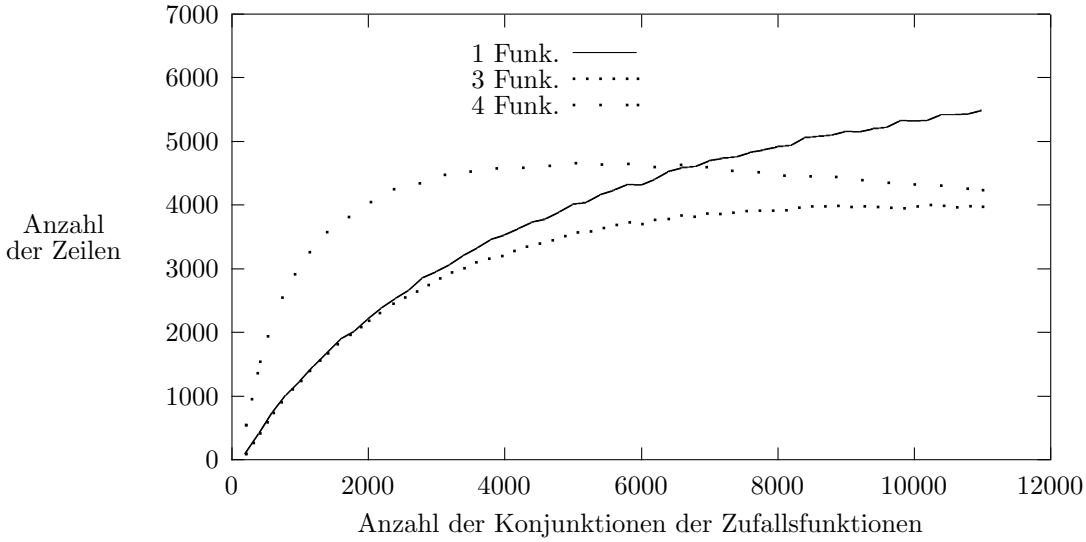


Abbildung 4.18: Anzahl der Konjunktionen der Funktionen vor und nach Dekomposition

der maximal möglichen Zahl unterschiedlicher Binärvektoren in einem Booleschen Raum mit $n-1$ Variablen.

Nach Abschnitt 4.1.3 verfügt eine TVL der nicht zerlegten Funktion in average case über 2^{n-2} Zeilen. Dabei wird unterstellt, dass in einer TVL im Mittel nur die Hälfte der maximal möglichen Zeilen enthalten ist. Die Zeilensumme über drei bzw. vier Teilfunktionen beträgt $3 \cdot 2^{n-3}$ bzw. $4 \cdot 2^{n-3}$. Das heißt, zur Darstellung einer Booleschen Funktion durch vier Funktionen werden mehr Ternärvektoren gebraucht als zur Darstellung durch drei Teilfunktionen.

In der Abbildung 4.18 sind die Ergebnisse der Untersuchungen am Beispiel zufällig erzeugter Funktionen mit 14 Variablen dargestellt. Die Anzahl der Konjunktionen (oder Ternärvektoren) der vier Teilfunktionen ist in der Summe größer als bei der Zerlegung in drei Teilfunktionen. Bei kleineren Funktionen überschreitet dieser Wert auch die Zeilenzahl der TVL der nichtzerlegten Funktionen. Zwei Gründe sind als Ursache zu nennen:

- Das Teilen der Funktion erschwert das Zusammenfassen von mehreren Ternärvektoren zu einem TV. Bei kleineren Funktionen ist diese Auswirkung besonders groß.
- Der zusätzliche Speicherplatz wird für die Abbildung der vierten Funktion benötigt, wobei es sich um die in der ursprünglichen Funktion nicht enthaltenen TV handelt.

Die Grundoperationen für die in vier Teile zerlegten Funktionen erfordern spezielle Algorithmen. Um die vier Teilfunktionen der Ergebnisfunktion zu berechnen, werden wie im Fall der Drei-Tupel-Dekomposition mehrere Verknüpfungen von Teilfunktionen durchgeführt.

Der Aufwand für die Operationen Konjunktion, Negation und Disjunktion kann wie folgt abgeschätzt werden:

- Konjunktion

  Ein Vergleich der Formeln (4.101)-(4.103) und (4.238)-(4.240) ergibt, dass die Teilfunktionen f^-, f^0 und f^1 auf gleichen Berechnungsvorschriften aufbauen. Aus diesem

Grund kann der Zeitaufwand, der für die Berechnug der Teilfunktionen f^-, f^0 und f^1 durch die Konjunktion zweier in vier Teile zerlegten Funktionen benötigt wird, in worst und average case nach den Formeln (4.150) und (4.152) berechnet werden. Für die zusätzlich erforderliche Berechnung der Teilfunktion $f^=$ sind zwei konjunktive und eine disjunktive Verknüpfungen mit jeweils zwei Ternärvektorlisten mit $N_A/3$ und $N_B/3$ Zeilen in worst case und mit $N_A/6$ und $N_B/6$ Zeilen in average case auszuführen. Zwei weitere disjunktive Verknüpfungen können in konstanter Zeit durch das Zusammenfügen von Ternärvektorlisten realisiert werden (siehe Formel (4.241)). Damit ergibt sich der Gesamtaufwand und die Komplexitätsordnung für den worst case (4.288) und für den average case nach (4.289).

$$f_{wc} = 10/9 \times N_A \times N_B + 6c = O_{wc}(N_A \times N_B) \tag{4.288}$$

$$f_{ac} = 10/36 \times N_A \times N_B + 6c = O_{ac}(N_A \times N_B) \tag{4.289}$$

- Negation

  Die Teilfunktionen der Ergebnisfunktion werden nach (4.210)-(4.213) berechnet. Da die Teilfunktionen der Ergebnisfunktion sich unmittelbar aus den Teilfunktionen der zu negierenden Funktion ergeben, können die Berechnungen in konstanter Zeit durchgeführt werden (4.290).

$$f_{wc} = f_{ac} = 4c = O_{wc}(c) = O_{ac}(c) \tag{4.290}$$

- Disjunktion

  Aus den für die Berechnung der Teilfunktionen f^-, f^0, f^1 und $f^=$ genutzten Formeln (4.219)-(4.222) folgt, dass neun konjunktive und eine disjunktive Verknüpfungen mit jeweils zwei Ternärvektorlisten mit $N_A/3$ und $N_B/3$ Zeilen in worst case und mit $N_A/6$ und $N_B/6$ Zeilen in average case durchzuführen sind. Weitere 6 disjunktive Verknüpfungen werden in konstanter Zeit durchgeführt. Eine Berechnung des Gesamtaufwands und der Komplexitätsordnung für worst und average case ist damit nach (4.288) und (4.289) möglich.

In die Abbildungen 4.19 und 4.20 wurden die für die Durchführung der Konjunktion und Disjunktion aus zwei Zufallsfunktionen benötigten Zeiten aufgenommen. Die Ergebnisse der Konjunktion zweier in 4 Teile zerlegten Funktionen verschlechtern sich geringfügig gegenüber den Ergebnissen der Konjunktion der in 3 Teile zerlegten Funktionen. Im Vergleich zur Konjunktion nicht zerlegter Funktionen konnten aber insbesondere bei größeren Funktionen auch dann noch bessere Ergebnisse erzielt werden. Eine wesentliche Verbesserung wurde bei Disjunktion nachgewiesen, wenn die Zerlegung von Funktion nicht in 3 sondern in 4 Teile erfolgte. Wie mit Untersuchungen von Zufallsfunktionen für die Operation Negation belegt werden konnte, war der Rechenzeitbedarf erwartungsgemäß für zerlegte Funktionen praktisch gleich Null.

4.3 Analyse der rekursiven Anwendung

Die rekursive Zerlegung nach (4.27) bzw. nach (4.162) bedeutet, dass jede der Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ bzw. bei der 4-Zerlegung auch die Funktion $f^=(\underline{x_0})$) in weitere

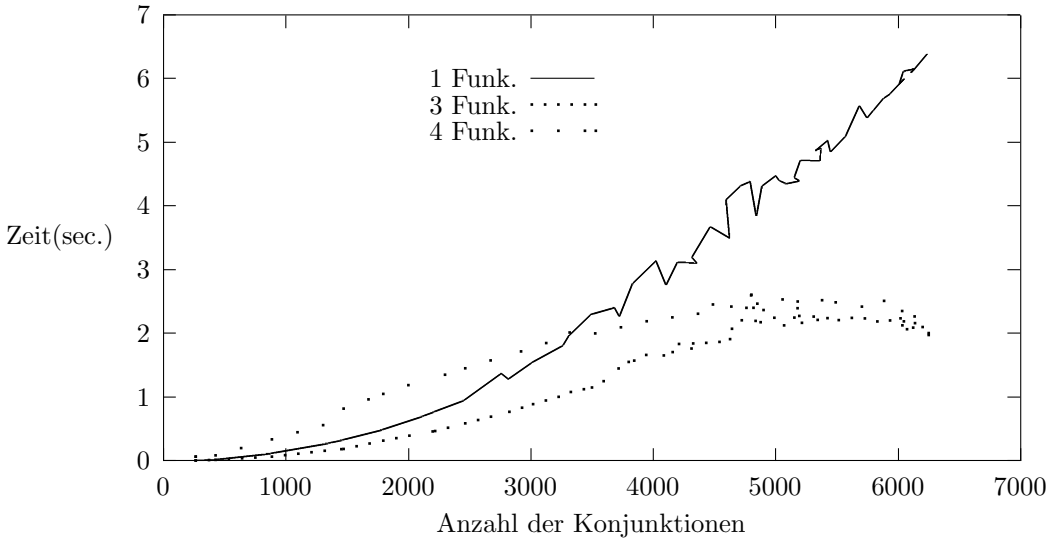


Abbildung 4.19: Berechnungszeit für Operation Konjunktion

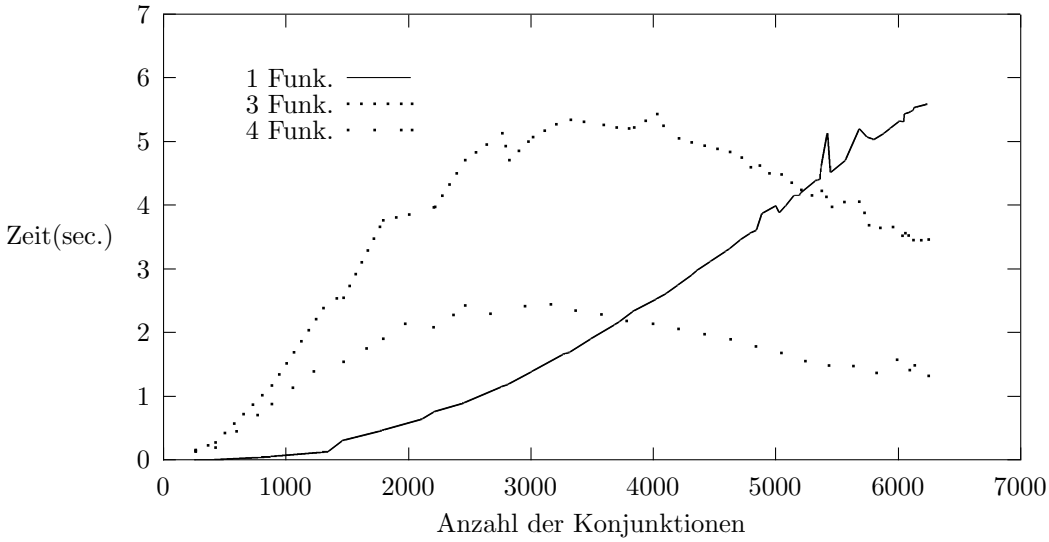


Abbildung 4.20: Berechnungszeit für Operation Disjunktion

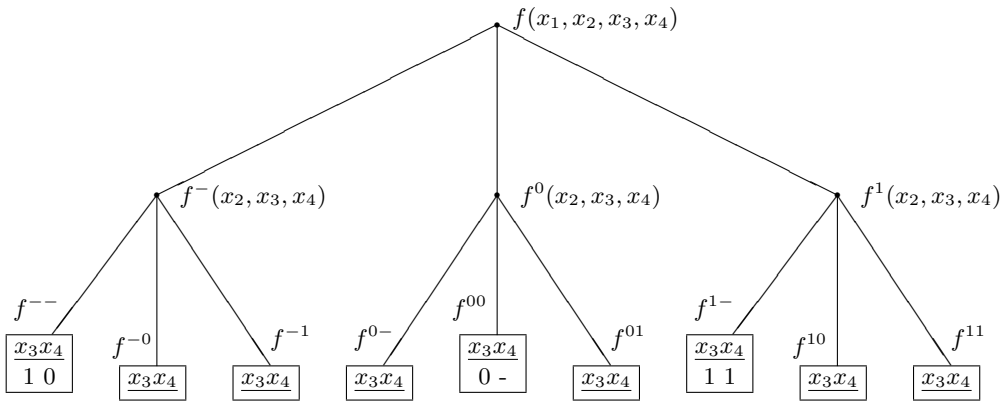


Abbildung 4.21: Rekursive 3-Zerlegung der Funktion $f(x_1, x_2, x_3, x_4)$ in Teilfunktionen

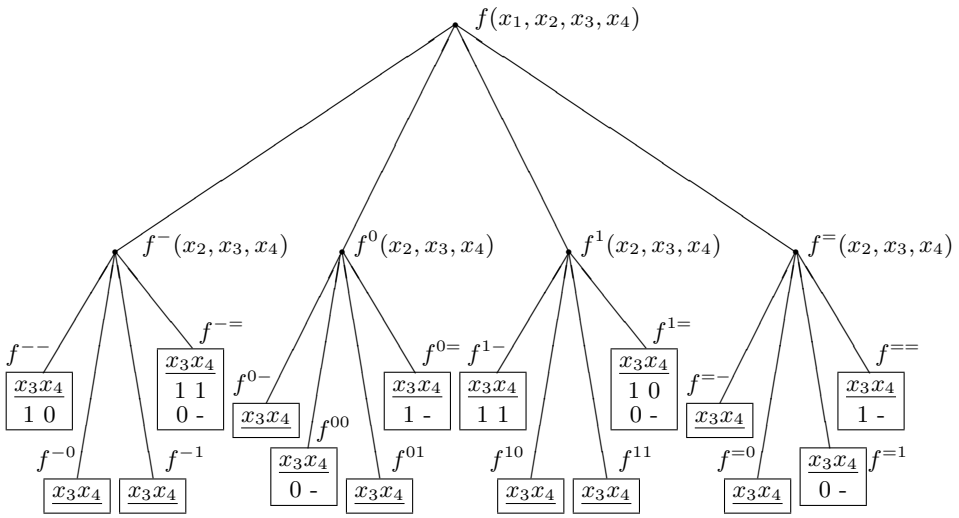


Abbildung 4.22: Rekursive 4-Zerlegung der Funktion $f(x_1, x_2, x_3, x_4)$ in Teilfunktionen

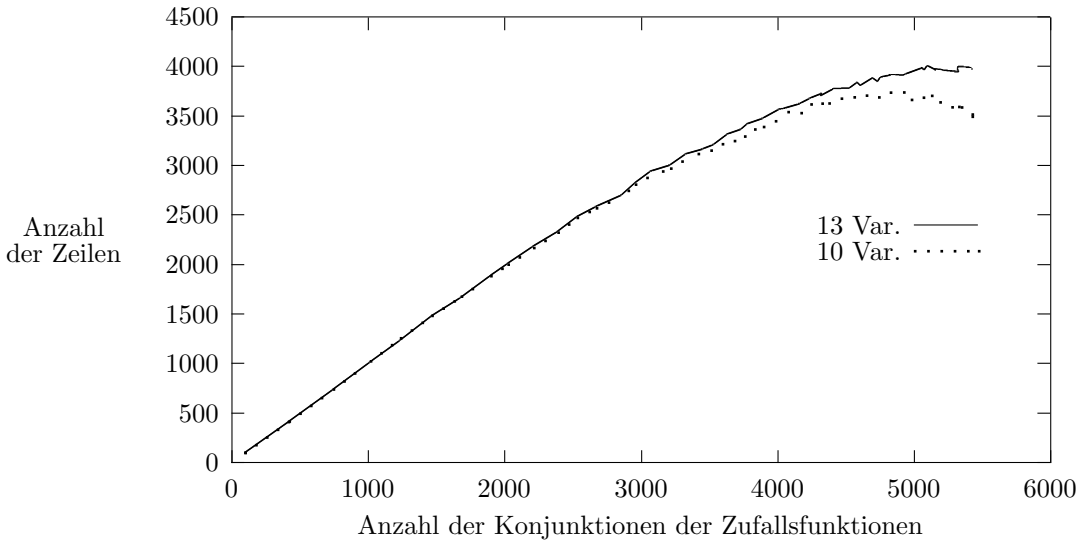


Abbildung 4.23: Anzahl der Konjunktionen der rekursiv 3-zerlegten Zufallsfunktionen.

3 oder 4 Teilfunktionen aufgespaltet wird. Dabei entsteht eine baumartige Struktur mit Knoten, die aus Verweisen auf die Funktionen der weiteren Zerlegungsebene und der Zerlegungsvariable bestehen. An den Blättern des Baums bleiben die nicht weiter zu zerlegenden Funktionen, die z. B. durch TVL dargestellt werden können. In den Abbildungen 4.21 und 4.22 sind die Ergebnisse der rekursiven Zerlegung in 3 (Abbildung 4.21) und 4 Teilfunktionen (Abbildung 4.22) der Funktion (4.48) nach den Variablen x_1 und x_2 dargestellt.

Nachfolgend wird die Beanspruchung der Ressourcen des Rechners im Fall der rekursiven Zerlegung analysiert, wobei der Speicherplatzbedarf und die für die Durchführung der Operationen benötigte Rechenzeit den Schwerpunkt der Untersuchung bilden.

Bezüglich des Speicherplatzbedarfs sind folgende Sachverhalte festzustellen:

- 3-Zerlegung:
  - Wie im Abschnitt 4.1.3 beschrieben, folgt aus der Orthogonalität der Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ die Tatsache, dass die drei als TVL dargestellten Funktionen in der Summe in worst case nicht über mehr Zeilen als die maximale Anzahl der Ternärvektoren der nicht zerlegten Funktion verfügen können. Auch bei einer rekursiven Zerlegung bleibt die Orthogonalität aller Teilfunktionen erhalten. D. h. auch für die rekursiv zerlegte Funktion kann die maximale Anzahl der Ternärvektoren aller Teilfunktionen den Wert 2^{n-1} nicht überschreiten.
  - In average case reduziert jede 3-Zerlegung einerseits die Anzahl der Zeilen jeder einzelnen Teilfunktion um das 2-fache, anderseits vergrößert sich die Anzahl der Teilfunktion mit jeder Zerlegung auf den 3-fachen Wert. Wird in Analogie zu 4.1.3 unterstellt, dass eine TVL im Mittel nur über die Hälfte der maximal möglichen Zeilen verfügt, werden für die TVL der nicht zerlegten Funktion

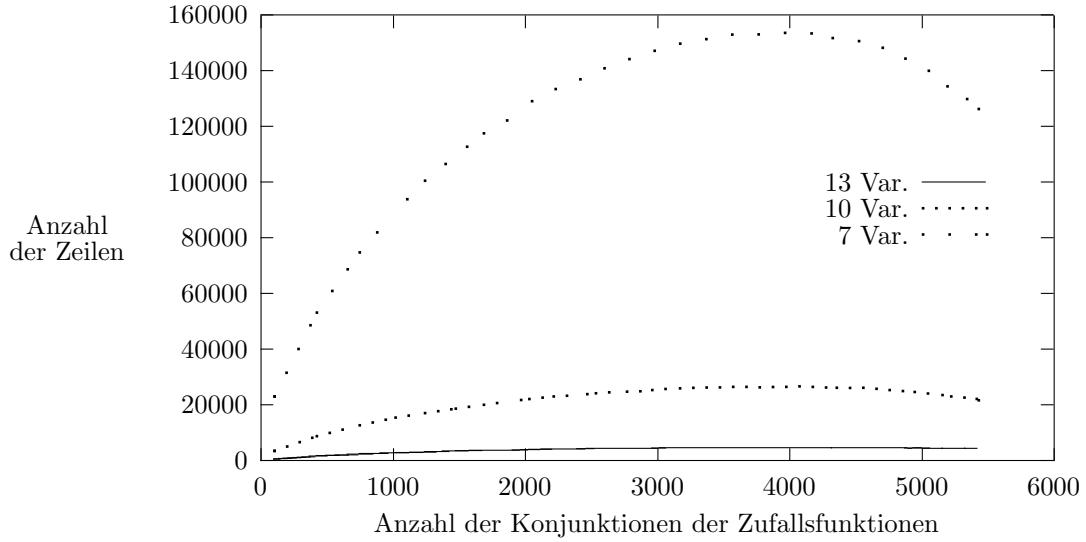


Abbildung 4.24: Anzahl der Konjunktionen der rekursiv 4-zerlegten Zufallsfunktionen.

2^{n-2} und für die Ternärvektorlisten der m-mal zerlegten Funktion $3^m \cdot 2^{n-2-m}$ Zeilen erhalten. Das bedeutet, zur Darstellung einer Boolschen Funktion durch Teilfunktionen werden unter Umständen mehr Ternärvektoren benötigt als zur Darstellung durch eine TVL.

– Praktische Untersuchungen mit Zufallsfunktionen bestätigen diese Überlegungen. Die schon im Abschnitt 4.1.3 beschriebenen, als Ternärvektorlisten dargestellten Zufallsfunktionen wurden hier weiter bis zu TVL mit 10 und 7 Variablen rekursiv zerlegt. Die Ergebnisse der Untersuchung sind in der Abbildung 4.23 dargestellt. Die Deckungsgleichheit der Kurven für TVL mit 7 bzw. 10 Variablen verhinderte die Darstellung beider Abhängigkeiten. Aus dem Vergleich der Abbildungen 4.7 und 4.23 ist ersichtlich, dass während die erste Zerlegung zu einer Reduzierung des Speicherplatzbedarfes führt, beeinflussen die weiteren Zerlegungen die Ergebnisse immer weniger, so dass die Ergebnisse für die bis 10 und 7 Variablen zerlegten TVL praktisch nicht zu unterscheiden sind.

- 4-Zerlegung:

  – Im Abschnitt 4.2.3 wurde festgestellt, dass die Teilfunktionen der in 4 Teile zerlegten Funktion in der Summe nicht über mehr als 2^{n-1} Ternärvektoren verfügen können. Die vier Teilfunktionen beschreiben vollständig einen Booleschen Raum mit $n-1$ Variablen, wobei in einem Booleschen Raum mit $n-1$ Variablen maximal soviele verschiedene Binärvektoren möglich sind. Die Orthogonalität der Funktionen ist die Ursache, dass jeder Binärvektor nur in einer Funktion auftreten kann. Die folgende Zerlegung der Teilfunktionen in weitere vier Teilfunktionen erfordert die vierfache vollständige Beschreibung eines Booleschen Raums

mit $n-2$ Variablen. Damit bestehen die 16 Teilfunktionen in der Summe aus maximal $4 \cdot 2^{n-2}$ Zeilen, wie auch am Beispiel der Funktion (4.48) zu sehen ist. Wird diese Funktion nach den Variablen x_1 und x_2 zerlegt, so entstehen auf der unteren Zerlegungsebene die Funktionen $f^{-=}(x_3, x_4)$, $f^{0=}(x_3, x_4)$, $f^{1=}(x_3, x_4)$ und $f^{==}(x_3, x_4)$, die sich zum Teil überdecken (siehe Abbildung 4.22). Die gesamte Zeilenzahl der m-mal zerlegten Funktion ($m \geq 1$) kann in worst case mit dem Wert $4^{m-1} \cdot 2^{n-m} = 2^{n-2+m}$ abgeschätzt werden. D. h. durch jede weitere Zerlegung (außer der ersten) verdoppelt sich die gesamte Zeilenzahl.

- Für average case gelten die gleichen Überlegungen wie für worst case. Im Gegensatz zu worst case ist jedoch zu unterstellen, dass die Ternärvektorlisten nur über die Hälfte der maximal möglichen Zeilen verfügen. Damit ergibt sich eine Gesamtzeilenzahl der m-mal zerlegten Funktion von $4^{m-1} \cdot 2^{n-1-m} = 2^{n-3+m}$.
- Auch hier wurden die oben aufgeführten Überlegungen durch Untersuchungen an Zufallsfunktionen bestätigt (siehe Abbildung 4.24). Aus dem Vergleich der Abbildungen 4.18 und 4.24 folgt, dass die erste Zerlegung die erwartete unwesentliche Veränderung des Speicherplatzbedarfes verursacht und weitere Zerlegungen den Speicherplatzbedarf um ca. 2^{m-1} vergrößeren.

Die Analysen des Speicherplatzbedarfs haben gezeigt, dass für eine Untersuchung der rekursiven 4-Zerlegung auf Grund des außerordentlichen Anstiegs des Speicherplatzbedarfs keine praktische Notwendigkeit besteht. Die Reduzierung des Speicherplatzbedarfs fällt bei einer rekursiven 3-Zerlegung nur gering aus. Eine derartige Zerlegung würde sich damit nur in dem Fall lohnen, wenn die Durchführung der Operationen zwischen zwei rekursiv 3-zerlegten Funktionen zu besseren Ergebnissen als zwischen zwei einmalig 3-zerlegten Funktionen führt. Diese Überlegung begründet auch die nachfolgende Abschätzung der Rechenzeit für die Operation Konjunktion.

Aus den Formeln (4.101), (4.102) und (4.103) ist ersichtlich, dass für die Berechnung der Teilfunktionen $f^{-}(\underline{x_0})$, $f^{0}(\underline{x_0})$ und $f^{1}(\underline{x_0})$ 7 konjunktive Verknüpfungen benötigt werden. Werden die mit den Formeln (4.101) - (4.103) beschriebenen Algorithmen rekursiv m-mal verwendet, sind für die Konjunktion der zerlegten Funktionen 7^m Verknüpfungen notwendig. Die konjunktiven Verknüpfungen werden dabei mit jeweils zwei Ternärvektorlisten mit $\frac{N_A}{3^m}$ und $\frac{N_B}{3^m}$ Zeilen in worst case und mit $\frac{N_A}{2 \cdot 3^m}$ und $\frac{N_B}{2 \cdot 3^m}$ Zeilen in average case durchgeführt. Da disjunktive Verknüpfungen in konstanter Zeit durchgeführt werden, finden sie hier keine Berücksichtigung. Der gesamte Berechnungsaufwand kann damit für worst case nach (4.291) und für average case nach (4.292) berechnet werden.

$$\begin{aligned} f_{wc} &= 7^m \times \frac{N_A}{3^m} \times \frac{N_B}{3^m} \\ &= (7/9)^m \times N_A \times N_B \end{aligned} \tag{4.291}$$

$$\begin{aligned} f_{ac} &= 7^m \times \frac{N_A}{2 \cdot 3^m} \times \frac{N_B}{2 \cdot 3^m} \\ &= 1/4 \times (7/9)^m \times N_A \times N_B \end{aligned} \tag{4.292}$$

Der mit den Formeln (4.291) und (4.292) berechnete Aufwand bezieht sich nur auf die Durchführung der TVL-Operationen. Bei den Operationen mit rekursiv zerlegten Funktionen entsteht zusätzlicher Aufwand für den Zugriff auf die Funktionen einer Baumstruktur.

Aus den Formeln (4.291) und (4.292) geht hervor, dass sich der Berechnungsaufwand für die Konjunktion der rekursiv 3-zerlegten Funktionen mit jeder Zerlegung reduziert, obgleich mit der ersten 3-Zerlegung der größte Effekt erzielt wird. Aus der Analysen des Abschitts 4.1.3 ist bekannt, dass während für die Berechnung einer Konjunktion nur 7 konjunktive Verknüpfungen mit Teilfunktionen benötigt werden, sind zur Berechnung einer Disjunktion abgesehen von den Berechnungen der Ausdrücke $\bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1$ und $\bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1$ die 10 konjunktiven bzw. disjunktiven Verknüpfungen durchzuführen (siehe Formeln (4.71)-(4.73)). Der mit den Formeln (4.158) und (4.159) eingeschätzte Berechnungsaufwand ist vertretbar bei einer erstmaligen Zerlegung, bei der rekursiven Dekomposition jedoch würde die Zunahme der Anzahl der Verknüpfungen schneller erfolgen als die entgegenwirkende Verringerung der Größe der Teilfunktionen. Damit ist bei der rekursiven Disjunktion die umgekehrte Tendenz als bei der rekursiven Konjunktion zu beobachten: der Berechnungsaufwand steigt mit jeder weiteren Zerlegung, obgleich weniger als bei der ersten 3-Zerlegung.

Aus dem Vergleich der einmaligen 3- und 4-Zerlegungen ist bekannt, dass die Konjunktion zweier 4-zerlegten Funktionen mehr Zeit benötigt als die Konjunktion zweier 3-zerlegten Funktionen (siehe Formeln (4.150) und (4.152) und Formeln (4.288) und (4.289)). Als Ursache ist zu nennen, dass die Konjunktion der 4-zerlegten Funktionen zusätzliche Zeit für die Berechnung der 4. Funktion benötigt. Damit müssen neben den 7 konjunktiven Verknüpfungen für die Berechnung der Funktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$ und $f^1(\underline{x_0})$ zwei konjunktive Verknüpfungen und eine disjunktive Verknüpfung (zwischen $f_1^=$ und $f_2^=$ Funktionen) nach (4.241) für die Berechnung der Funktion $f^=(\underline{x_0})$ durchgeführt werden. Die weiteren disjunktiven Verknüpfungen bleiben unberücksichtigt, da sie in konstanter Zeit durchgeführt werden. Folglich ergeben sich für die rekursiv zerlegten Funktionen insgesamt 10^m Verknüpfungen. Die Verknüpfungen werden dabei mit jeweils zwei Ternärvektorlisten mit $\frac{N_A}{3^m}$ und $\frac{N_B}{3^m}$ Zeilen in worst case und mit $\frac{N_A}{2 \cdot 3^m}$ und $\frac{N_B}{2 \cdot 3^m}$ Zeilen in average case durchgeführt. Der gesamte Berechnungsaufwand kann für worst case nach (4.293) und für average case nach (4.294) berechnet werden.

$$\begin{aligned} f_{wc} &= 10^m \times \frac{N_A}{3^m} \times \frac{N_B}{3^m} \\ &= (10/9)^m \times N_A \times N_B \end{aligned} \tag{4.293}$$

$$\begin{aligned} f_{ac} &= 10^m \times \frac{N_A}{2 \cdot 3^m} \times \frac{N_B}{2 \cdot 3^m} \\ &= (10/9)^m \times 1/4 \times N_A \times N_B \end{aligned} \tag{4.294}$$

Eine Verbesserung des Berechnungsaufwands für die Negation einer rekursiv 4-zerlegten Funktion gegenüber einer einmalig 4-zerlegten Funktion ist nicht möglich, da die Negation in konstanter Zeit durchgeführt wird (siehe Formel (4.290)).

Aus den oben aufgeführten Analysen ist zusammenfassend herauszustellen:

- Die rekursive 3-Zerlegung bewirkt im Vergleich zu einer einmaligen 3-Zerlegung nur eine sehr geringe Reduzierung des Speicherplatzbedarfes bei der Darstellung einer Booleschen Funktion. Die Operationen mit rekursiv 3-zerlegen Funktionen sind im Vergleich zu Operationen mit einmalig 3-zerlegten Funktionen unwesentlich weniger zeitaufwendig. Insgesamt erweist sich damit die rekursive 3-Zerlegung als wenig sinnvoll.

- Die rekursive 4-Zerlegung führt im Vergleich zu einer einmaligen 4-Zerlegung zu einem extremen Anstieg des Speicherplatzbedarfes. Darüber hinaus sind die Operationen mit den rekursiv 4-zerlegten Funktionen zeitaufwendiger, so dass auch diese rekursive Zerlegung sich für die Darstellung einer Booleschen Funktion als ungeeignet erweist.

4.4 Optimierung

4.4.1 Berechnung der Teilfunktionen

Beim Zerlegen einer Booleschen Funktion in drei bzw. vier Teilfunktionen, müssen die Teilfunktionen $f^0(\underline{x_0})$, $f^1(\underline{x_0})$, $f^-(\underline{x_0})$ und $f^=(\underline{x_0})$ nach (4.21), (4.22), (4.20) und (4.161) entsprechend ihrer Definition berechnet werden. Für die Teilfunktionen $f^-(\underline{x_0})$ und $f^=(\underline{x_0})$ kann die Berechnung auch nach den Formeln (4.295) und (4.296) erfolgen, die die Operationen Minimum und Maximum statt der Operation Ableitung verwenden. Die Korrektheit der Formeln folgt unmittelbar aus (2.16).

$$f^-(\underline{x_0}) = f(x_i, \underline{x}) \overline{\frac{\partial f(x_1, \underline{x})}{\partial x_i}} = \min_{x_i} f(x_i, \underline{x_0}) \tag{4.295}$$

$$f^=(\underline{x_0}) = \overline{f(x_i, \underline{x})} \overline{\frac{\partial f(x_1, \underline{x})}{\partial x_i}} = \overline{f(x_i, \underline{x}) \vee \frac{\partial f(x_1, \underline{x})}{\partial x_i}} = \overline{\max_{x_i} f(x_i, \underline{x_0})} \tag{4.296}$$

Eine weitere Möglichkeit, die Berechnung der Teilfunktionen $f^0(\underline{x_0})$, $f^1(\underline{x_0})$ und $f^-(\underline{x_0})$ zu beschleunigen besteht in der Verwendung der Operationen Konjunktion und Differenz. Aus dem Vergleich der Teilfunktion der Tupel-Dekomposition mit den Funktionen der Shannon-Dekomposition (siehe Abbildung 4.11) ist zu sehen, dass die Teilfunktionen $f^0(\underline{x_0})$, $f^1(\underline{x_0})$ und $f^-(\underline{x_0})$ nach (4.297), (4.298) und (4.299) berechnet werden können. Die Operation Maximum ist nur zum Entfernen der Variable x_i aus den Teilfunktionen $f^0(\underline{x_0})$, $f^1(\underline{x_0})$ und $f^-(\underline{x_0})$ notwendig. Die Teilfunktion sind von der Variable x_i unabhängig.

$$f^-(\underline{x_0}) = \max_{x_i}(f(\underline{x_0}, x_i = 0) \cdot f(\underline{x_0}, x_i = 1)) \tag{4.297}$$

$$f^0(\underline{x_0}) = \max_{x_i}(f(\underline{x_0}, x_i = 0) \setminus f(\underline{x_0}, x_i = 1)) \tag{4.298}$$

$$f^1(\underline{x_0}) = \max_{x_i}(f(\underline{x_0}, x_i = 1) \setminus f(\underline{x_0}, x_i = 0)) \tag{4.299}$$

Mit der Zielstellung, die aus Sicht der Bearbeitungszeit günstigste Berechnungsvorschrift für die Teilfunktionen zu ermitteln, wurden die schon in den Abschnitten 4.1.3 und 4.2.3 verwendeten Zufallsfunktionen entsprechend der Formeln (4.295), (4.21) und (4.22), sowie (4.297), (4.298) und (4.299) zerlegt und der benötigte Zeitaufwand gemessen. Die Ergebnisse der Untersuchungen sind in der Abbildung 4.25 dargestellt. Die Teilfunktion $f^=(\underline{x_0})$ wurde in beiden Fällen nach (4.296) berechnet, so dass die Zeit für die Berechnung dieser Funktion nicht berücksichtigt werden musste. Der Abbildung ist zu entnehmen, dass durch die Anwendung der Operationen Konjunktion und Differenz an Stelle der Ableitungsoperationen die Berechnungszeiten reduziert werden können. Die durchschnittliche Reduzierung beträgt 33%.

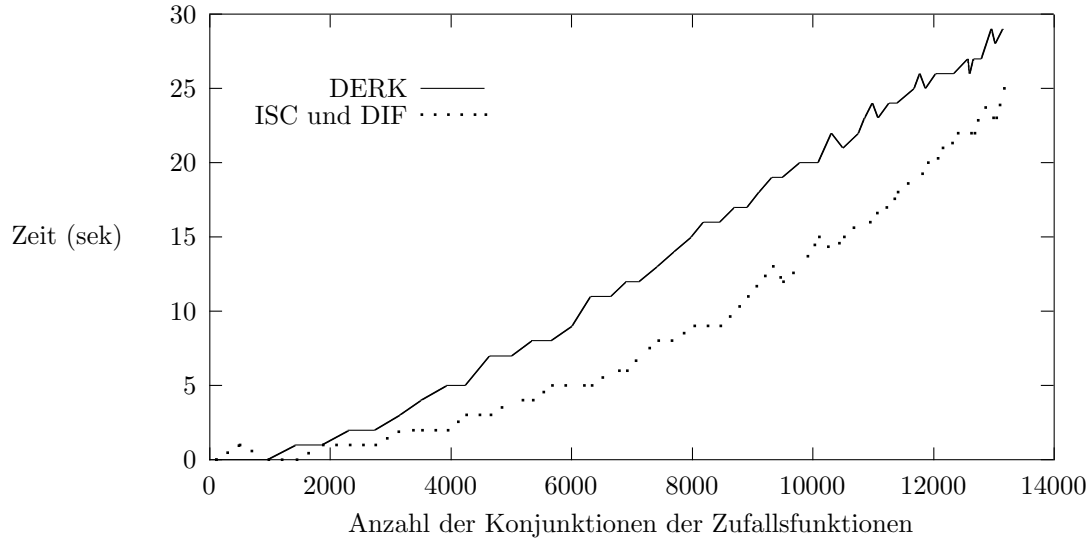


Abbildung 4.25: Verwendung von verschiedenen Algorithmen

4.4.2 Durchführung der Tupeloperationen

Wie aus den Abschnitten 4.1.2 und 4.2.2 bekannt, werden bei der Durchführung der Tupeloperationen Teilfunktionen des Ergebnisses durch mehrere unterschiedliche Verknüpfungen (Negationen, Konjunktionen und Disjunktionen) zwischen den Teilfunktionen der zu verknüpfenden Funktionen berechnet. Ein Teil der Negationen und Disjunktionen können durch andere, hinsichtlich Zeitaufwand günstigere, Operationen ersetzt werden. In den Teilberechnungen der Tupeloperationen Negation, Disjunktion, Antivalenz und Äquivalenz zwischen den in drei Teilfunktionen zerlegten Funktionen ist z. B. die Operation Differenz an Stelle der Negation möglich.

Die Operation Negation von F (NEG(F)) ist im XBOOLE-System intern als Differenz einer 1-Funktion und der Ternärvektorliste F realisiert [BoSt91]. Die 1-Funktion umfasst alle im gegebenen Booleschen Raum möglichen Binärvektoren und wird als TVL mit einem Ternärvektor, der nur Strichelemente enthält, dargestellt. Werden aus dieser Liste alle Binärvektoren der Ternärvektorliste F entfernt, so ergibt sich die negierte Ternärvektorliste. Für die negierten Teilfunktionen der zerlegten Funktion gilt auch, dass sie durch die Differenz zwischen 1-Funktion und den Teilfunktionen berechnet werden können.

Aus der Abbildung 4.26 ist ersichtlich, dass der in den Formeln für die Teilfunktionen der Tupeloperationen Negation, Disjunktion, Antivalenz und Äquivalenz (4.50), (4.72)-(4.73), (4.115)-(4.117) und (4.133)-(4.135) verwendete Ausdruck $\bar{f^-}\bar{f^0}\bar{f^1}$ durch die Differenz zwischen der 1-Funktion mit den Teilfunktionen f^-, f^0 und f^1 ersetzt werden kann (4.300).

$$\bar{f^-}\bar{f^0}\bar{f^1} = 1 \backslash f^- \backslash f^0 \backslash f^1 = \bar{f^-} \backslash f^0 \backslash f^1 \tag{4.300}$$

Diese Möglichkeit wurde bei allen im Abschnitt 4.1.3 beschriebenen Untersuchungen genutzt und in den Aufwandabschätzungen der Tupeloperationen (Formeln 4.155, 4.157, 4.158 und 4.159) berücksichtigt.

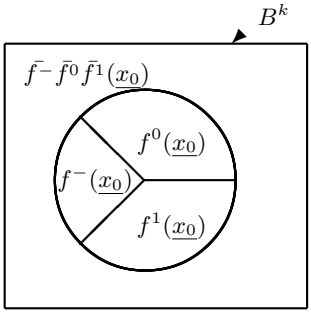


Abbildung 4.26: Verhältnis zwischen den Teilfunktionen der Tupel-Dekomposition und den negierten Teilfunktionen

Eine andere Möglichkeit die für die Durchführung der Tupeloperation benötigte Zeit zu verringern ergibt sich aus der Orthogonalität der Teilfunktionen. Für die Berechnung der Teilfunktionen des Ergebnisses der Tupeloperationen Negation, Disjunktion, Antivalenz und Äquivalenz zwischen den in drei (4.50)-(4.52), (4.71)-(4.73), (4.101)-(4.103), (4.115)-(4.117) und (4.133)-(4.135) bzw. in vier Teilfunktionen zerlegten Funtionen (4.210)-(4.213), (4.219)-(4.222), (4.238)-(4.241), (4.254)-(4.257) und (4.271)-(4.274) sind mehrere disjunktive Verknüpfungen von Konjunktionen der Teilfunktionen notwendig.

Zur Realisierung dieser disjunktiven Verknüpfungen kann die XBOOLE-Operation Disjunktion eingesetzt werden. Die XBOOLE-Operation Disjunktion besteht intern aus den Operationen Verketten und Orthogonalisieren. Bei der Operation Verketten werden die Ternärvektorlisten variablengerecht und ohne weiteren Umformungen zusammengefügt und bilden die neue Ternärvektorliste. Im allgemeinen geht bei dieser Operation die Orthogonalität verloren. Um die Orthogonalität des Ergebnisses beizubehalten wird nachfolgend die Operation Orthogonalisieren durchgeführt.

Die Disjunktionen in den Formeln (4.50)-(4.52), (4.71)-(4.73), (4.101)-(4.103), (4.115)-(4.117), (4.133)-(4.135), (4.210)-(4.213), (4.219)-(4.222), (4.238)-(4.241), (4.254)-(4.257) und (4.271)-(4.274) mit Ausnahme der Disjunktion $f_1^- \vee f_2^-$ in den Formeln (4.71) und (4.219) und der Disjunktion $f_1^= \vee f_2^=$ in der Formel (4.238) werden zwischen den orthogonalen Konjunktionen durchgeführt und müssen nicht nachträglich orthogonalisiert werden.

Die Operation Verketten wurde in den in Abschnitten 4.1.3 und 4.2.3 beschriebenen Untersuchungen an Stelle der Disjunktion verwendet und in den Aufwandabschätzungen für die Tupeloperationen berücksichtigt.

Kapitel 5

Verteilte Verarbeitung

5.1 Analyse

Im Kapitel 4 wurde beschrieben, wie eine Boolesche Funktion durch ein Tupel aus Booleschen Funktionen dargestellt werden kann. Die Elemente des Tupels sind wiederum Funktionen, die als Ergebnis einer Dekomposition dieser Funktion nach den in den Abschnitten 4.1.1 und 4.2.1 beschriebenen Prinzipien (siehe die Sätze 4.3 und 4.15) entstanden sind. Die Darstellung einer Booleschen Funktion durch einen Tupel aus mehreren Funktionen führt zu der Notwendigkeit, bei den Verknüpfungen der zerlegten Funktionen mit den Booleschen Operationen Negation, Konjunktion, Disjunktion, Antivalenz und Äquivalenz statt nur eine Verknüpfung mehrere Verknüpfungen auf die Elementen des Tupels anzuwenden. Im den Abschnitten 4.1.2 und 4.2.2 wurden die Algorithmen zur Durchführung der Operationen mit den nach Formeln (4.27) und (4.162) zerlegten Funktionen beschrieben. In den Abschitten 4.1.3 und 4.2.3 wurde der dazu benötigte Aufwand abgeschätzt.

Es wurde festgestellt, dass z. B. bei der Durchführung der Tupeloperationen Antivalenz und Äquivalenz der in vier Teilfunktionen zerlegten Funktionen 16 konjunktive und 12 disjunktive Verknüpfungen erforderlich sind (siehe Formeln (4.254)-(4.257) und 4.271-4.274). Nach Einschätzungen auf der Grundlage der in den Abschnitten 4.1.3 und 4.2.3 gewonnenen Erkenntnisse, ist der Hauptaufwand auf die Durchführung der Konjunktionen und einiger weniger Disjunktionen zurückzuführen. Zum Beispiel führt die Anwendung der Operation Disjunktion auf die in drei Teilfunktionen zerlegten Funktionen zu der Disjunktion $f_1{}^{-} \vee f_2{}^{-}$ und den Konjunktionen $f_1{}^{0} f_2{}^{1}$, $f_1{}^{1} f_2{}^{0}$, $f_1^0 f_2^0$, $f_1^0 \bar{f_2^-} \bar{f_2^0} f_2^1$, $f_2^0 \bar{f_1^-} \bar{f_1^0} f_1^1$, $f_1^1 f_2^1$, $f_1^1 \bar{f_2^-} \bar{f_2^0} f_2^1$ und $f_2^1 \bar{f_1^-} \bar{f_1^0} f_1^1$ (siehe Formeln (4.71)-4.73). Für die in analoger Form angewendete Operation Konjunktion sind das die Konjunktionen $f_1{}^{-} f_2{}^{-}$, $f_1{}^{-} f_2{}^{0}$, $f_1{}^{0} f_2{}^{-}$, $f_1{}^{0} f_2{}^{0}$, $f_1{}^{-} f_2{}^{1}$, $f_1{}^{1} f_2{}^{-}$ und $f_1{}^{1} f_2{}^{1}$ (siehe Formeln (4.101)-(4.103)). Die Durchführung von allen weiteren Disjunktionen in den Formeln (4.71)-(4.73) und (4.101)-(4.103) benötigt, wie in den Abschnitten 4.1.3 und 4.4.2 schon festgestellt wurde, konstante Zeiten. Es wurde nachgewiesen, dass auch bei allen anderen Tupeloperationen der größte Zeitaufwand für die Durchführung der konjunktiven Verknüpfungen erforderlich ist.

Bei den in den Abschitten 4.1.3 und 4.2.3 aufgeführten praktischen Untersuchungen werden die konjunktiven Verknüpfungen der Tupeloperationen sequentiell ausgeführt. Zum Beispiel wird bei der Tupeloperation Konjunktion zuerst die Teilfunktion $f_3{}^{-}$ der Ergebnisfunktion f_3 als $f_1{}^{-} f_2{}^{-}$ berechnet (siehe Abbildung 5.1). Anschließend werden für die Berechnung der Teilfunktion $f_3{}^{0}$ nacheinander die Konjunktionen $f_1{}^{-} f_2{}^{0}$, $f_1{}^{0} f_2{}^{-}$ und $f_1{}^{0} f_2{}^{0}$

$f_3{}^{-} = f_1{}^{-} f_2{}^{-}$
$h_1 = f_1{}^{-} f_2{}^{0}$
$h_2 = f_1{}^{0} f_2{}^{-}$
$h_3 = f_1{}^{0} f_2{}^{0}$
$f_3{}^{0} = h_1 \vee h_2 \vee h_3$
$h_4 = f_1{}^{-} f_2{}^{1}$
$h_5 = f_1{}^{1} f_2{}^{-}$
$h_6 = f_1{}^{1} f_2{}^{1}$
$f_3{}^{1} = h_4 \vee h_5 \vee h_6$

Abbildung 5.1: Sequentielle Ausführung der Tupeloperation Konjunktion

$f_3{}^{-} = f_1{}^{-} f_2{}^{-}$	$h_1 = f_1{}^{-} f_2{}^{0}$	$h_2 = f_1{}^{0} f_2{}^{-}$	$h_3 = f_1{}^{0} f_2{}^{0}$	$h_4 = f_1{}^{-} f_2{}^{1}$	$h_5 = f_1{}^{1} f_2{}^{-}$	$h_6 = f_1{}^{1} f_2{}^{1}$
$f_3{}^{0} = h_1 \vee h_2 \vee h_3$						
$f_3{}^{1} = h_4 \vee h_5 \vee h_6$						

Abbildung 5.2: Parallele Ausführung der Tupeloperation Konjunktion

ausgeführt und disjunktiv verknüpft. Zuletzt wird die Funktion ${f_3}^1$ berechnet, indem die drei Konjunktionen ${f_1}^- {f_2}^1$, ${f_1}^1 {f_2}^-$ und ${f_1}^1 {f_2}^1$ nacheinander berechnet und anschließend disjunktiv verknüpft werden.

Wesentlich ist die Erkenntnis, dass die für die Berechnung der Teilfunktionen benötigten Konjunktionen unabhängig voneinander bearbeitet werden. Somit ist die Möglichkeit gegeben, die Berechnungen parallel auszuführen. Folgerichtig kann die Berechnung der 7 Konjunktionen, die im Ergebnis der Tupeloperation Konjunktion erforderlich wurden, ebenfalls parallel bearbeitet werden (siehe Abbildung 5.2). Die letzten Schritte der Berechnungen der Funktionen ${f_3}^0$ und ${f_3}^1$, die die disjunktiven Verknüpfungen der zuvor berechneten Konjunktionen beinhalten (in den Abbildungen 5.1 und 5.2 sind das die Berechnungen ${f_3}^0 = h_1 \vee h_2 \vee h_3$ und ${f_3}^1 = h_4 \vee h_5 \vee h_6$), können auch unabhängig voneinander und damit parallel ausgeführt werden. Da diese Berechnungen unabhängig von der Größe der zu verknüpfenden Funktionen in konstanter Zeit durchgeführt werden (siehe Abschnitte 4.1.3 und 4.4.2) und die verteilte Verarbeitung mit den zusätzlichen Zeitaufwendungen z. B. für Datenübertragung verbunden ist, wäre die parallele Verarbeitung hier unbegründbar. Die gleichen Überlegungen gelten auch für die anderen Tupeloperationen, das heißt, diese Aussagen treffen sowohl für die in 3 Teilfunktionen wie auch in 4 Teilfunktionen zerlegten Funktionen zu.

5.2 Verteilungsszenarium

5.2.1 Konzept

Die im Abschnitt 5.1 durchgeführte Analyse führte zu der Schlussfolgerung, dass die in der Arbeit untersuchte Problemstellung eine parallele Bearbeitung unabhängiger Teilaufgaben zulässt. Unter Bezugnahme auf die unter 2.4.2 gewonnenen Aussagen ist eine Eignung des Client-Server Modells für eine rechentechnische Umsetzung gegeben. Wie bereits dargelegt, sieht das Modell die Realisierung von Aufgaben durch einen oder mehrere Server vor, wobei die Bearbeitung der Aufträge durch Clients ausgelöst wird. Im Regelfall werden die Clients und Server auf verteilter Technik laufen, da nur so die Vorzüge der verteilten Verarbeitung wie Verbesserung von Laufzeiten oder Zugriff auf verteilte Ressourcen genutzt werden können. Den Vorteilen steht ein erhöhter Implementierungsaufwand entgegen. Insbesondere sind über klar definierte Schnittstellen Daten zwischen den Clients und Servern auszutauschen. Die benötigten Übertragungszeiten mindern die Zeiteinsparung, die durch die parallele Auftragsbearbeitung erreicht wird. Ein verbessertes Zeitverhalten wird folglich nur bei verarbeitungsintensiven Problemstellungen zu erwarten sein.

Die Zeitmessungen im Abschnitt 5.3.2 belegen, dass diese Voraussetzung für den konkreten Anwendungsfall zutrifft. An anderer Stelle durchgeführte Messungen von Übertragungszeiten (Abschnitt 5.3.6) lassen die Schlussfolgerung zu, dass unter Berücksichtigung des im konkreten Anwendungsfall zu übertragenden Datenvolumens ein spürbar verbessertes Zeitverhalten zu erwarten ist.

Die Datenübertragung zwischen Client und Server wurde im Anwendungsfall auf der Grundlage des Socket Interface des BSD UNIX [CoSt93] realisiert. Als Übertragungsprotokoll wurde das Transmissions Control Protocol (TCP) genutzt. Da hierbei eine Synchronisation des Datenaustauschs zwischen Sender und Empfänger erfolgt, handelt es sich um einen blockierenden Nachrichtenaustausch. Die Art der ausgetauschten Informationen erfüllen die Kriterien einer auftragsorientierten Kommunikation (siehe Abschnitt 2.4.2).

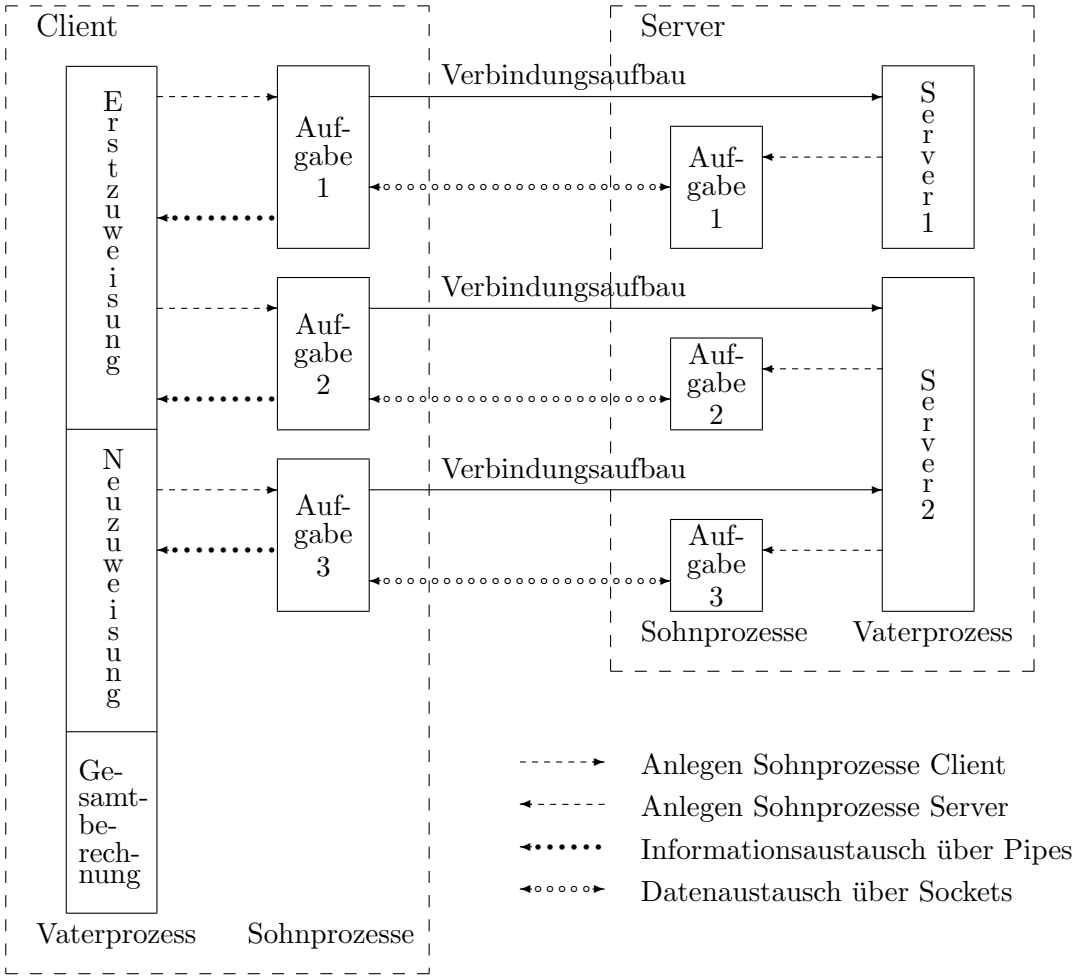


Abbildung 5.3: Vereinfachte Darstellung des Steuerflusses

Für den Einsatz von Sockets spricht im Vergleich zur Anwendung von Middelware, mit der die Implementierung verteilter Lösungen unterstützt wird, insbesondere die Verfügbarkeit auf unterschiedlichen Plattformen. Die den Compilern zur Verfügung stehenden Bibliotheken, die eine verteilte Verarbeitung auf der Basis von Sockets zulassen, sind auf allen gängigen UNIX- sowie Windowsystemen vorhanden. Eine Portierung der Lösung auf andere Plattformen ist daher ohne zusätzliche Aufwendungen problemlos möglich.

TCP ist ein verbindungsorientierter Transportdienst, der eine zuverlässige Datenübertragung garantiert. Folgerichtig werden die Daten mit TCP verlustfrei, ohne zusätzliche Duplikate und in der gesendeten Folge beim Empfänger bereitgestellt. TCP nimmt eventuell notwendige Fehlerkorrekturen selbständig vor und leitet die erneute Anforderung verfälschter oder verloren gegangener Daten ein. Das Protokoll greift dabei auf das Sliding-Window-Verfahren [Tan96] zurück. Der korrekte Übertragungsverlauf wird im Verfahren unter Bezugnahme auf Sequenz- und Quittungsnummern beurteilt.

5.2.2 Struktur des Client-Server-Systems

Die unabhängig voneinander durchführbaren Teilberechnungen sind durch einen unterschiedlich hohen Rechenaufwand gekennzeichnet und führen zu Teilergebnissen, die nachfolgend für die Ermittlung des Gesamtresultats benötigt werden. Bezüglich der Bearbeitungsreihenfolge der Teilberechnungen bestehen keine Einschränkungen, so dass neben einer sequentiellen Bearbeitung in beliebiger Folge auch eine vollständige oder eingeschränkte Parallelisierung möglich ist. Genau der zuletzt genannte Umstand ist zutreffend, wenn die Zahl der parallel ausführbaren Teilaufgaben die Anzahl verfügbarer Server übersteigt.

Eine Anforderung an die allgemeingültige Verarbeitungsstruktur des Systems besteht in der Anpassung des Verarbeitungsablaufs an die aktuelle Aufgabenstruktur und die verfügbare Rechentechnik. Das bedeutet, beim Clientstart müssen neben den Daten der unabhängig voneinander lösbaren Teilaufgaben die verfügbaren Server bekannt sein. In der vorliegenden Implementierung wurde das durch Argumentübergabe beim Clientaufruf realisiert.

Die Abbildung 5.3 gibt einen näheren Überblick des Zusammenwirkens eines Clientes und der genutzten Server. Um ein verbessertes Zeitverhalten zu erreichen, wird die Problemstellung nach dem Start eines Clientes die parallele Inanspruchnahme mehrere Server erfordern. Folglich wird der Client ein nebenläufiges Abarbeitungsverhalten aufweisen müssen, wobei jeweils innerhalb eines Sohnprozesses des Clientes die Kommunikation zu genau einem Server gesichert wird. Der Lebenszyklus der Sohnprozesse erstreckt sich auf die Zeitdauer der Datenübertragung einschließlich der Bearbeitungszeit des Servers. Erst nach der Rückübertragung der Ergebnisdaten vom Server zum Client kann der jeweilige Sohnprozess terminiert werden.

Für die konkrete Implementierung bedeutet das, nach dem Clientstart werden die Teilaufgaben in der Reihenfolge ihrer Bereitstellung in der Argumentliste den verfügbaren Servern zugeteilt. Das Ansteuern der Server erfolgt innerhalb der Sohnprozesse. Der sequentielle Verarbeitungsanteil bleibt daher auf das Anlegen der Prozesskopien beschränkt.

Im Regelfall übersteigt die Zahl der Teilaufgaben die verfügbare Serverzahl. Ein vollständiges Zuweisen der Aufgaben zu Servern ist dann im ersten Verarbeitungsschritt nicht möglich. Im weiteren Verarbeitungsablauf wird über ein Steuerungspipe von den Sohnprozessen an den wartenden Vaterprozess dann eine Nachricht übermittelt, wenn über die Socket-Verbindung vom Server die Ergebnisdaten der betreffenden Teilaufgabe eingegangen sind. Dieses Hergehen sichert, dass unterschiedlich große Bearbeitungszeiten der Server nicht zu Verzögerungen führen. Vielmehr werden unmittelbar nach dem Eingang der Fertigstellungsnachricht über ein prozessspezifisches Datenpipe die Teilergebnisse zum Vaterprozess übertragen. Liegen noch unbearbeitete Teilaufgaben vor, erfolgt in Analogie zum beschriebenen Vorgehen die Übertragung der nächsten Aufgabe an diesen Server, indem innerhalb eines weiteren Sohnprozesses die Serveransteuerung und der Datenaustausch erfolgt.

In einer anderen Clientvariante wurde die Rückübertragung der Daten vom Server und das Auslösen der nächsten Aufgabe ebenfalls parallelisiert, d. h. es werden bereits während des Empfangs der Ergebnisdaten Eingangsdaten zum gleichen Server übermittelt. Da die Übertragungszeiten der Daten im Vergleich zum Rechenaufwand nur wenig ins Gewicht fallen, war jedoch durch dieses Vorgehen keinen spürbare Verbesserung des Gesamtzeitverhaltens nachweisbar.

Die Berechnung des Gesamtergebnisses ist erst nach dem erfolgreichen Abschluss des letzten Teilauftrags möglich. Bereits nach der Beendigung jedes Teilauftrags wurden die Resultate dem Vaterprozess zur Verfügung gestellt, so dass die Voraussetzungen für die

Ermittlung des Gesamtergebnisses zu diesem Zeitpunkt gegeben sind.

Die Kommunikation zwischen Vater- und Sohnprozessen über den gemeinsamen Steuerungspipe und die Datenübertragung durch prozessspezifische Datenpipes sichert die geordnete Datenbereitstellung in der Folge ihres Eingangs von den Servern.

Serverseitig wurde zunächst eine iterative Arbeitsweise vorgesehen. Als Konsequenz ergab sich, dass ein Server genau einen Auftrag in der Zeiteinheit bearbeiten konnte. Bei mehrfacher Ansteuerung hatte dieses Herangehen ein Einordnen weiterer Aufträge in eine Warteschlage zur Folge, was schließlich zu einer sequentiellen Auftragsbearbeitung führte. Die Optimierungsbestrebungen bei der Datenübergabe aber auch ein erweitertes Versuchsszenarium erforderte serverseitige Maßnahmen zur Parallelisierung, das heißt auch hier war eine Nebenläufigkeit erforderlich. Die einzuleitenden Schritte gestalteten sich einfach, da bei jedem Auftragseingang nur eine Prozesskopie angelegt werden musste. Ohne Rückkopplung zum Vaterprozess wird innerhalb dieser Kopie die Bereitstellung der Eingangsdaten, die Auftragsbearbeitung und das Senden der Resultate realisiert. Ein Terminieren des Sohnprozesses erfolgt nach dem Abschluss des Versendens der Resultatdaten.

5.2.3 Versuchsdurchführung

Mit der Zielstellung die Bearbeitungszeiten zu minimieren wurden Aufgaben unterschiedlicher Komplexität unter veränderten Versuchsbedingungen getestet. Restriktionen waren insbesondere durch die Leistungsfähigkeit und die Zahl der verfügbaren Rechner gegeben.

Für einen Vergleich der Bearbeitungszeiten wurden Aufräge wiederholt bearbeitet, wobei zunächst unterschiedliche Serverzahlen herangezogen wurden. Bei allen Tests bestand die Einschränkung, dass die Anzahl der Aufträge die Serverzahl überschritt. Ein weiterer Aspekt bezog sich auf die mehrfache Bereitstellung eines Servers beim Clientstart, was eine parallele Ansteuerung eines Servers zur Folge hatte. Damit wurde die Möglichkeit getestet, bereits im ersten Verarbeitungsschritt die Bearbeitung aller Teilaufgaben einzuleiten, wobei jedoch eine paralle Bearbeitung mehrerer Teilaufgaben auf einem Rechner in Kauf genommen werden musste. Aus den Messergebnissen sind Schlussfolgerungen für die Vertretbarkeit dieses Herangehens zu gewinnen.

Mit dem Umstand der begrenzten Verfügbarkeit leistungsfähiger Technik ist zu begründen, dass das Szenarium einen Start von Client und Server auf dem gleichen Rechner vorsieht.

Im Sinne einer Optimierung der Auftragsbearbeitung ist die gezielte Zuweisung von Aufgaben zu speziellen Servern und eine Einflussnahme auf die Bearbeitsreihenfolge zu verstehen. Eine Möglichkeit ist durch die Gestaltung der Argumente beim Clientstart gegeben, da die Folge der Serveradressen und Eingangsdaten für die Ansteuerung der Server herangezogen wird. Prinzipiell erweist es sich als vorteilhaft, wenn die verfügbaren Server möglichst gleich ausgelastet sind. Die Wartezeit auf noch ausstehende Teilberechnungen wird dadurch minimiert und damit die Gesamtbearbeitungszeit reduziert. Nach einer Einschätzung zu erwartender Bearbeitungszeiten auf Basis des Eingangsdatenvolumens, sollten die zeitintensiven Berechnungen vorrangig bearbeitet werden. Die Unabhängigkeit der Teilaufgaben liefert für dieses Vorgehen die Grundlage.

5.3 Ergebnisse praktischer Untersuchungen

5.3.1 Versuchsbedingungen

Mit der Zielstellung eines Vergleichs der Verarbeitungszeiten bei veränderten Bedingungen wurden praktische Messungen in einem Subnetz mit 4 UNIX-Workstation unterschiedlicher Leistungsfähigkeit durchgeführt.

Hostname	Typ	Anz. Prozessoren
Luna	Ultra 10	1
Space	Sparc Station 20	2
Ninive	Ultra 1	1
Assur	Sparc Station 20	2

Die Workstation wurden unter dem Betriebssystem Solaris betrieben, wobei die durch den Einsatz gleichartiger Rechner erreichte Homogenität bewusst angestrebt wurde. Da die Problemstellung eine Verteilung von Teilberechnungen einer Gesamtoperation zum Gegenstand hat, besteht somit eine verbesserte Vergleichbarkeit. Weiterhin wurden erhöhte Aufwendungen bei der Implementierung der sehr speziellen Aufgabenstellung vermieden. Schließlich kann unterstellt werden, dass im praktischen Einsatz in der Regel mehrere Rechner gleichen Typs verfügbar sind.

Als Eingangsdaten wurden für die Untersuchung Zufallsfunktionen mit 20 Variablen herangezogen, die unterschiedliche Anzahl der Konjunktionen (oder in der TVL-Darstellung: unterschiedliche Anzahl der Zeilen) enthalten.

5.3.2 Verteilte Verarbeitung auf zwei Servern

Für die ersten Untersuchungen wurden nur zwei Server genutzt, die auf den Rechner Luna und Space installiert waren. Die unterschiedliche Leistungsfähigkeit der Rechner war Anlass, zwei Versuchsreihen mit veränderter Clientzuordnung durchzuführen:

- Client auf Luna, je ein Server auf Luna und Space
- Client auf Space, je ein Server auf Luna und Space

Zum Vergleich wurde die Operation Tupelkonjuktion auch sequentiell auf beiden Rechnern durchgeführt und damit der Mittelwert der Verarbeitungszeiten ohne Verteilung der Teilaufgaben gewonnen.

Die Tabelle 5.1 und die Abbildung 5.4 enthalten die Ergebnisse der Messungen. Es ist zu beachten, dass in die Auswertungen grundsätzlich die Mittelwerte beider Versuchsreihen aufgenommen wurden.

In der Abbildung 5.4 sind die Zeiten, die für die Durchführung der Operation Konjunktion der je zwei Drei-Tupel-Funktionen benötigt werden, in Abhängigkeit von der mittleren Zeilenzahl beider zu verknüpfenden Funktionen dargestellt. Die Tabelle 5.1 enthält die absolut gemessenen Gesamtbearbeitungszeiten in Abhängigkeit zur Zeilenzahl. Durch die Gegenüberstellung der sequentiellen und verteilten Verarbeitung (Spalten 3 und 4) konnte die erreichte Zeitreduzierung in Prozent (Spalte 5) ausgewiesen werden.

Die Ergebnisse bestätigen die bereits unter 5.2.1 getroffene Annahme, dass eine Verteilung erst bei größeren Funktionen zu Zeiteinsparungen führt. Die mittlere Berechnungszeit bei einer Verteilung sinkt für die untersuchten Funktionen bis auf 54% im Vergleich zur

Tabelle 5.1: Ergebnisse der verteilten Verarbeitung mit zwei Servern und der sequentiellen Verarbeitung

Zeilenzahl		sequent.	verteilt		sortiert		Verbesserung
1.Funk.	2.Funk	sek	sek	%	sek	%	%
6844	6966	5	10	190	9	170	20
16503	16798	22	27	120	25	111	9
30372	30341	109	97	89	85	78	11
72580	73526	562	388	69	360	64	5
121663	121390	1282	808	63	779	61	2
171161	171637	2113	1321	63	1286	61	2
221095	221306	2952	1737	59	1771	60	-1
267380	268208	3726	2197	59	2198	59	0
311722	313228	4438	2527	57	2367	53	4
352719	354462	5031	2893	58	2775	55	3
391298	391227	5493	3252	59	3067	56	3
425420	424962	5830	3159	54	3181	55	-1

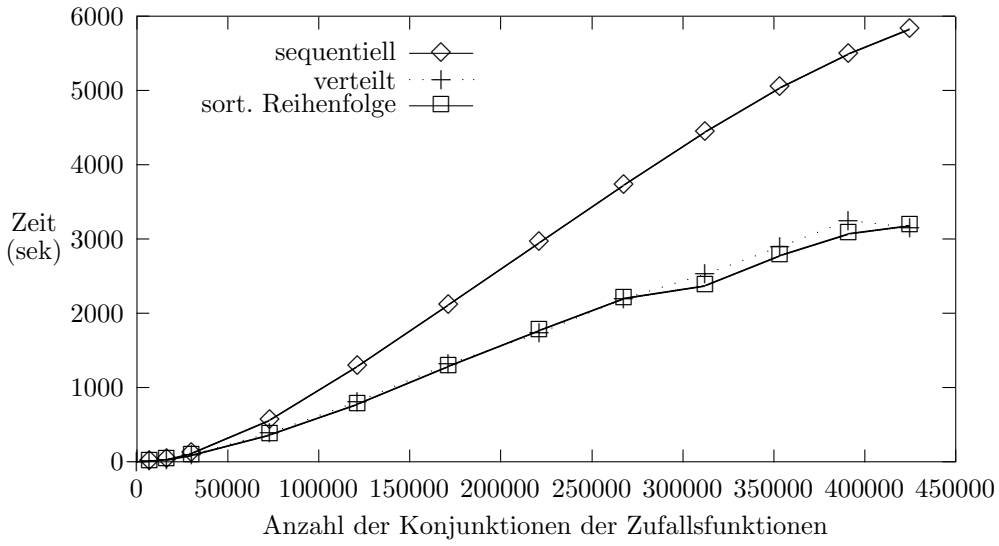


Abbildung 5.4: Vergleich der verteilten Verarbeitung mit zwei Servern und der sequentiellen Verarbeitung

Bearbeitungszeit bei einer sequentiellen Verarbeitung. Abbildung und Tabelle lassen die Tendenz erkennen, dass mit einer steigenden Funktionsgröße die Verteilung eine weitere Verbesserung der Bearbeitungszeiten im Vergleich zur sequentiellen Verarbeitung erwarten lässt.

Der Verbesserung der Verarbeitungszeiten bei einer verteilten Verarbeitung steht der vergrößerte Overhead entgegen, der mit dem Verteilungsvorgang verbunden ist. Insbesondere sind folgende Einflussgrößen zu nennen:

- Datenübertragung zwischen Client und Server über Sockets
- Informationsaustausch zwischen den einzelnen Prozessen des Clientes bzw. der Server
- Erhöhter interner Verwaltungsaufwand durch Anlegen weiterer Prozesse und den damit verbundenen Kontextwechsel

Unter Berücksichtigung dieser Einflussfaktoren und der Tatsache, dass eine Parallelisierung aller Verarbeitungsschritte auf Grund der Programmstruktur nicht möglich ist, wird die theoretisch erreichbare Halbierung der Verarbeitungszeit nicht möglich sein.

5.3.3 Verteilte Verarbeitung auf zwei Servern mit optimierter Auftragsfolge

Nachdem im Abschnitt 5.2.2 beschriebenen Szenarium werden die Aufträge ohne Berücksichtigung ihrer Größe an die verfügbaren Server verteilt, wobei von der Restriktion Anzahl Aufträge > Anzahl verfügbarer Server ausgegangen wird. Im allgemeinen spielt die Reihenfolge der Auftragsbearbeitung keine Rolle, da der Verarbeitungsalgorithmus die Bearbeitung mehrerer kleinerer Aufträge auf einem Server zulässt, während auf einem anderen Server ein großer Auftrag bearbeitet wird.

Eine ungünstige Verteilung von Aufträgen führt jedoch dann zu einer erhöhten Gesamtbearbeitungszeit, wenn der größte Auftrag zuletzt bearbeitet wird. Die damit verbundene ungleichmäßige Auslastung der Server führt zwangsläufig zu einer erhöhten Gesamtbearbeitungszeit.

Voraussetzung für die Optimierung der Bearbeitungsfolge ist eine Beurteilung der Auftragsgröße. Als geeignetes Bewertungskriterium kann der Speicherplatz herangezogen werden, der zur Darstellung der Funktion benötigt wird.

Da die Aufträge jeweils die Bearbeitung von zwei Teilfunktionen beinhalten, ist das Produkt des Speicherplatzbedarfs beider Funktionen zu berücksichtigen. Die Optimierung des Verarbeitungsablaufs besteht nun in einer Auftragsbearbeitung in der absteigenden Folge des Speicherplatzbedarfs. Wie bereits dargelegt, kann in der genutzten Implementierung über eine gezielte Ordnung der Argumentliste beim Programmstart auf die Bearbeitungsreihenfolge Einfluss genommen werden.

In der Abbildung 5.4 und in der Tabelle 5.1 wurden die Ergebnisse einer optimierten verteilten Verarbeitung aufgenommen (Spalte 5 und 6). Die erreichten Einsparungen gegenüber einer nicht optimierten verteilten Verarbeitung sind als relativ unerheblich einzuschätzen (Spalte 8). Teilweise Verschlechterungen können mit Fremdeinflüssen (Netz- bzw. anderweitige Rechnerbelastung) zurückgeführt werden.

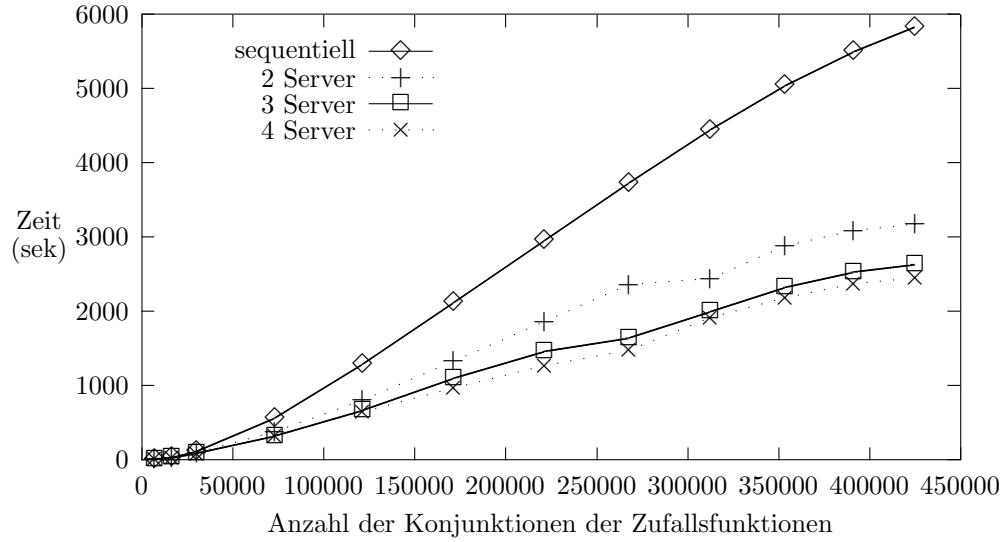


Abbildung 5.5: Vergleich der verteilten Verarbeitung mit zwei, drei und vier Servern und der sequentiellen Verarbeitung

5.3.4 Verteilte Verarbeitung mit erhöhter Serverzahl

Den Messungen unter Einsatz einer vergrößerten Serverzahl sollen einige grundlegende Überlegungen vorangestellt werden. Eine ideale Konstellation ist gegeben, wenn die Anzahl der Server der Auftragszahl entspricht, von einer gleichen Leistungsfähigkeit der verfügbaren Technik ausgegangen werden kann und Aufträge gleichen Umfangs vorliegen. Unter dieser Voraussetzung ist eine quasi gleichzeitige Fertigstellung der Teilaufträge zu erwarten, was letztlich eine Minimierung der Gesamtbearbeitungszeit bedeutet.

In der Regel kann von derartigen Idealbedingungen nicht ausgegangen werden. Unter der Annahme Auftragszahl = Serverzahl wird die längste Bearbeitungszahl eines Auftrags die Gesamtbearbeitungszeit bestimmen. Eine manuelle Zuordnung der Aufträge zu Rechnern entsprechend der Leistungsfähigkeit ist mit Unsicherheiten behaftet und wird vielfach deshalb zu keinen optimalen Ergebnissen führen. Unterschreitet die Serverzahl die Auftragszahl ist wie bereits oben beschrieben eine Optimierung möglich, wenn die Aufträge absteigend geordnet nach ihrem Umfang bereitgestellt werden. Nach der Fertigstellung eines Auftrags wird der frei werdende Server mit dem jeweils größten noch nicht bearbeiteten Auftrag betraut. Dadurch wird der kleinste Auftrag zuletzt bearbeitet, was bei einer ungleichmäßigen Auslastung ein Minimieren der "Überhangzeit" zu Folge hat.

Die praktischen Messungen wurden mit zwei, drei und vier Servern durchgeführt, wobei für die Server die Rechner Space und Luna sowie Ninive (3. Server) und Assur (4. Server) eingesetzt wurden (siehe auch Abschnitt 5.3.1). Als Hostrechner für den Client wurde grundsätzlich der Rechner Space genutzt. Folgerichtig wurden auch für die Vergleichsmessung ohne Verteilung der Rechner Space herangezogen.

Die Ergebnisse der Messung können der Abbildung 5.5 und der Tabelle 5.2 entnommen werden. Geringfügige Abweichungen zur Tabelle 5.1 sind mit der Versuchsdurchführung zu erklären. Tabelle 5.1 enthält im Gegensatz zu den hier durchgeführten Messungen Mittel-

Tabelle 5.2: Ergebnisse der verteilten Verarbeitung mit zwei, drei und vier Servern im Vergleich zur sequentiellen Verarbeitung

Zeilenzahl		sequent.	2 Server		3 Server		4 Server	
1.Funk.	2.Funk	sek	sek	%	sek	%	sek	%
6844	6966	5	10	200	10	200	10	200
16503	16798	22	28	127	28	127	24	109
30372	30341	109	92	85	88	81	74	68
72580	73526	562	378	67	321	57	327	58
121663	121390	1282	805	63	666	52	646	50
171161	171637	2113	1327	63	1097	52	966	46
221095	221306	2952	1858	63	1453	49	1269	43
267380	268208	3726	2365	63	1640	44	1470	39
311722	313228	4438	2440	55	1994	45	1917	43
352719	354462	5031	2878	57	2323	46	2178	43
391298	391227	5493	3087	56	2527	46	2368	43
425420	424962	5830	3172	54	2625	45	2452	42
Summe		31562	18440	58	14772	47	13701	43

werte, die sich aus Messreihen mit unterschiedlicher Clientzuordnung ergaben.

Der Einsatz weiterer Server führt zu einer zusätzlichen Reduzierung der Berechnungszeiten. Im Vergleich zur sequentiellen Verarbeitung sinkt der durchschnittliche Zeitaufwand von 58% (zwei Server) auf 47% (drei Server) bzw. 43% (vier Server). Die erzielten zusätzlichen Einsparungen von Verarbeitungszeiten nehmen mit steigender Serverzahl ab, was für eine sinkende Effizienz der Serverauslastung spricht.

Als erste Schlussfolgerung ist herauszustellen, dass der Einsatz von zwei Servern zu spürbaren Verbesserungen führt und durch die Verteilung auf eine größere Serverzahl nur noch geringere Verbesserungen erreichbar sind. Besteht die Notwendigkeit einer Minimierung der Bearbeitungszeit, sollte jedoch die Serverzahl der Auftragszahl entsprechen.

5.3.5 Nebenläufige Verarbeitung an Multiprozessor-Workstation

Verfügt eine Workstation über mehrere Prozessoren, wird bei einem Multiprozessor-Timesharing-Betriebssystem wie Solaris die parallele Ansteuerung der Prozessoren durch das Betriebssystem vorgenommen, so dass ein echter Parallelbetrieb möglich ist. Ein Einfluss des Nutzers auf die Prozesszuordnung ist dabei nicht gegeben.

Für die vorliegende Problemstellung bietet es sich an, diese Möglichkeiten der Parallelverarbeitung zu nutzen, indem einem Rechner mit n Prozessoren auch n Aufträge zugeordnet werden. Im konkreten Fall wurden dem Rechner Space, der über zwei Prozessoren verfügt, zeitgleich zwei Aufträge übergeben. In der vorliegende Implementierung war dazu der Hostname des Rechners doppelt in der Argumentliste beim Clientstart aufzuführen.

Die Ergebnisse der Messungen im Vergleich zur sequentiellen Verarbeitung können der Abbildung 5.6 und der Tabelle 5.3 entnommen werden. Der Vorteil einer derartigen Herangehensweise zeigt sich speziell bei großen Aufträgen. Die Reduzierung der Berechnungszeiten der Tupeloperation Konjunktion beträgt für die untersuchten Funktionen bis zu 38%. Wie

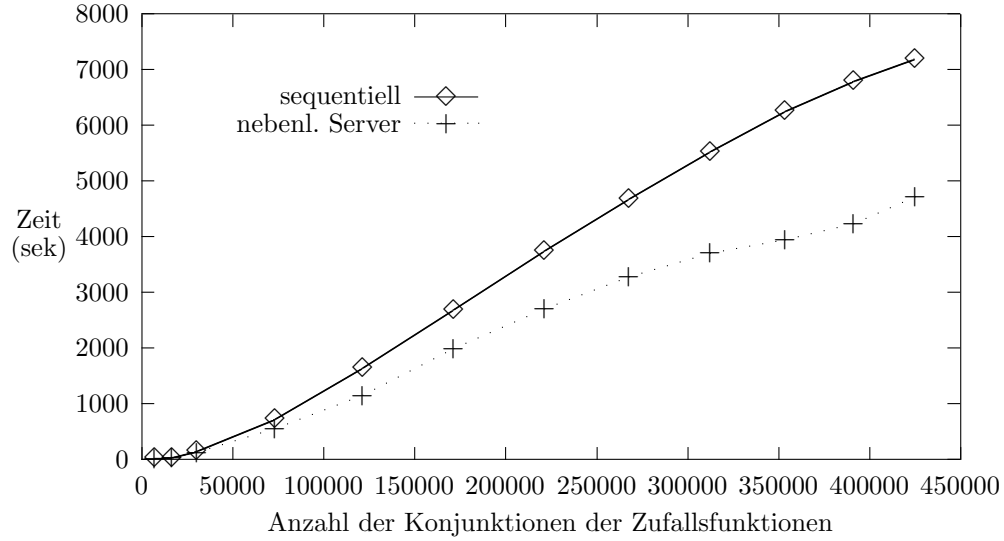


Abbildung 5.6: Vergleich der nebenläufigen Verarbeitung an 2-Prozessor-Workstation mit der sequentiellen Verarbeitung

Tabelle 5.3: Nebenläufige Verarbeitung an Multiprozessor Workstation

1.Funk.	2.Funk.	sequent.	parallel	%
6844	6966	7	13	186
16503	16798	28	36	129
30372	30341	138	122	88
72580	73526	716	551	77
121663	121390	1631	1149	70
171161	171637	2678	1984	74
221095	221306	3731	2703	72
267380	268208	4667	3280	70
311722	313228	5516	3703	67
352719	354462	6240	3939	63
391298	391227	6787	4230	62
425420	424962	7189	4708	65

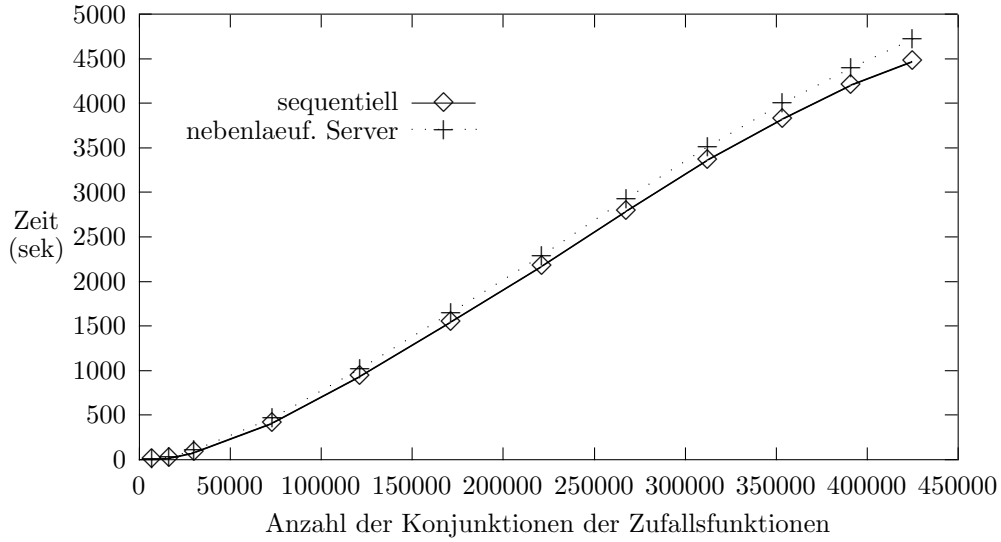


Abbildung 5.7: Vergleich der nebenläufigen Verarbeitung an 1-Prozessor-Workstation mit der sequentiellen Verarbeitung

zu erwarten, fällt im Vergleich zur Verteilung der Aufträge auf unterschiedliche Rechner die Reduzierung geringer aus. Schlussfolgernd ist festzuhalten, dass es durchaus sinnvoll ist, an Mehrprozessorrechner mehrere Aufträge parallel zu vergeben, wenn die verfügbare Rechnerzahl die Auftragszahl unterschreitet.

Schließlich wurde eine analoge Messung an einer Workstation mit einem Prozessor durchgeführt (Abbildung 5.7). Die Bearbeitung der Tupeloperation führte bei Parallelverarbeitung im Vergleich zu sequentiellen Abarbeitung sogar zu einem erhöhten Zeitaufwand. Diese Aussage entsprach den vorangestellten Überlegungen, da der notwendige Aufwand für das Anlegen weiterer Prozesse, der Datenaustausch und zusätzliche Kontextwechsel zu eine weitere Belastung des Prozessors mit sich brachte.

5.3.6 Analyse zusätzlicher Aufwendungen

Wie bereits oben erwähnt, erfordert die Problemlösung auf der Grundlage einer verteilten Verarbeitung einen Informationsaustausch zwischen dem Client und den Servern sowie zwischen den parallel ablaufenden Prozessen. Durch eine Analyse der benötigten zeitlichen Aufwendungen soll Aussagen zur Wirtschaftlichkeit des Herangehens gewonnen werden.

Zunächst ist festzuhalten, dass mit jedem Serveraufruf clientseitig ein neuer Prozess kreiert wird. Da die Anzahl der Serveraufrufe der Anzahl der Aufträge entspricht, sind maximal sieben Prozesse anzulegen. Messungen ergaben einen Zeitwand kleiner 1 s pro erzeugten Prozess, so dass im Vergleich zum Gesamtaufwand dieser Einflussfaktor zu vernachlässigen ist.

Der Datenaustausch zwischen dem Vaterprozess und den Sohnprozessen erfolgt über Pipes, wobei von allen Prozessen ein gemeinsames Steuerpipe und ein prozessspezifisches Datenpipe genutzt wird. Da beim Erzeugen eines Prozesses von der Gesamtheit des Vaterprozesses (einschließlich der Daten) eine Kopie angelegt wird, ist ein Datenaustausch nur

Tabelle 5.4: Übertragungzeiten der verteilten Verarbeitung mit zwei Servern

1.Funk.	2.Funk.	ges. Zeit	Übertr.-Zeit	%
6844	6966	7	0	0,00
16503	16798	22	0	0,00
30372	30341	79	0	0,00
72580	73526	344	1	0,29
121663	121390	796	4	0,50
171161	171637	1325	4	0,30
221095	221306	1761	5	0,28
267380	268208	2067	6	0,29
311722	313228	2421	7	0,29
352719	354462	2607	6	0,23
391298	391227	3140	7	0,22
425420	424962	3340	11	0,33
Summe		17909	51	0,28

in Richtung Sohn- zu Vaterprozess über Pipes erforderlich. Eine Optimierung des Ablaufs bestand nun darin, bereits nach dem Eingang der Fertigmeldung über das Steuerpipe, einen neuen Auftrag an den Server zu übergeben. Die Datenrückgabe erfolgte dadurch parallel zu Bearbeitung des Folgeauftrags und führt damit zu keinen zusätzlichen Aufwendungen. Eine Messung der Übertragungszeiten zwischen Prozessen konnte folglich entfallen. Der Übertragungsaufwand, der beim letzten Auftrag entstehen kann, ist zu vernachlässigen.

Der Datenaustausch zwischen dem Client und den Servern erfolgt in der vorliegenden Implementierung über Sockets, wobei in beiden Richtungen Informationen auszutauschen sind. Es ist zu berücksichtigen, dass nur dann die nachfolgend gemessenen Übertragungszeiten voll in das Gesamtergebnis eingehen, wenn keine zeitliche Überlappung der von einem Server bearbeiteten Aufträge vorliegt. Unter 5.2.2 wurde bereits erwähnt, dass die Parallelisierung des nächsten Serveraufrufs und der Datenrückübertragung zu keinen spürbaren Verbesserungen führte.

Die Ergebnisse der Messung von Übertragungszeiten mittels Sockets in Abhängigkeit zum Datenvolumen können der Tabelle 5.4 entnommen werden. Für das Experiment wurde der Client auf dem Rechner Luna und die Serverprogramme auf den Rechnern Luna und Space gestartet.

Die Spalten 1 und 2 der Tabelle enthalten Angaben zur Größe der Funktionen (Anzahl der Konjunktionen oder Zeilenzahl). In der Spalten 3 wurde die für die Ausführung der Tupeloperationen benötigte Gesamtzeit und in der Spalte 4 die reine Übertragungszeit aufgenommen. Aus der Spalte 5 kann der prozentuale Anteil der Übertragungszeit an der Gesamtzeit entnommen werden. Die Messungen belegen, dass die Übertragungszeit nur unwesentlich die Gesamtbearbeitungszeit beeinflusst. Der Durchschnittswert von 0,28% kann als repräsentativ angesehen werden, da keine Veränderungen des prozentualen Anteils in Abhängigkeit zum Datenvolumen nachweisbar sind.

5.3.7 Zusammenfassung und mögliche Weiterführung

Bereits mit der Auswertung der Messergebnisse waren erste Schlussfolgerungen verbunden. Es konnte nachgewiesen werden, dass die Problemstellung für eine teilweise parallele Bearbeitung gut geeignet ist. Die in der Implementation vorgenommene Abgrenzung der unabhängigen Teilaufgaben wird als vorteilhaft eingeschätzt. Die gewählte Größe der Teilaufgaben war einerseits problembedingt, gleichzeitig konnte dabei der verteilungsbedingte Aufwand im Vergleich zum Rechenaufwand gering gehalten werden. Auf eine ungleichmäßige Serverauslastung zurückzuführende "Überhangzeiten" sind speziell bei einer optimierten Auftragsbearbeitung als vertretbar einzuschätzen.

Es ist erkennbar, dass bereits der Einsatz von zwei Rechnern sich als sehr vorteilhaft erweist. Die gleichzeitige Zuordnung eines Servers und des Clientes zu einem Rechner ist auf Grund der zeitlichen Verschiebung rechenintensiver Phasen des Clientes und des Servers vertretbar. Einsparungen beim Einsatz weiterer Server zeigen einen degressiven Verlauf, so dass im Regelfall die Verteilung auf zwei Rechner als ausreichend einzuschätzen ist. Schließlich wurde noch nachgewiesen, dass die Abarbeitung mehrerer Serverprozesse auf einem Rechner nur dann sinnvoll ist, wenn das Betriebssystem auf mehrere Prozessoren zugreifen kann.

Den Ausgangspunkt für die konkrete Implementation bildete ein Prototyp, der wesentliche Elemente der Lösung enthält. Es kann unterstellt werden, dass der allgemeingültige Lösungansatz des Prototyps die Grundlage für die Realisierung ähnlich gelagerter Problemstellungen sein könnte. Eine Übertragbarkeit der Programmstruktur ist damit gegeben.

Nachfolgend werden Vorschläge unterbreitet, die zu einer Optimierung des Verarbeitungsablaufs und damit zu einer weiteren Verminderung der Verarbeitungszeiten führen. Kernstück ist ein sogenannter Trader, der zur Verwaltung verfügbarer Server herangezogen wird. Der Einsatz von Tradern zur optimalen Bereitstellung von Diensten wird in [RuRI96, RuRI97] beschrieben. Danach meldet sich jeder Server beim Start am Trader an. Clients wenden sich dann mit ihren Dienstwünschen an den Trader, bekommen einen oder mehrere Serverschnittstellen übemittelt, um schließlich eine Verbindung zu den ausgewählten Servern herzustellen.

Für die hier untersuchte Problemstellung entfällt damit das manuelle Zuweisen der Server, indem auf die am Trader registrierten Dienstangebote zurückgegriffen wird. Vorteilhaft wird dieses Herangehen dann sein, wenn die Leistungen des Traders ein optimales Zuweisen der Aufträge zu den Servern einschließen. Über das Datenvolumen der Teilaufträge kann auf die Größe des Auftrags geschlossen werden. Neben einer Vorgabe der Leistungsfähigkeit der einzelnen Server, sollte auf der Grundlage einer Auswertung der Verarbeitungszeiten der Teilaufträge eine ständige Aktualisierung der vom Trader gespeicherten Leistungsparameter erfolgen. Die Optimierung der Verarbeitung besteht nun in der automatischen Zuweisung der Server unter Berücksichtigung der Auftragsgröße und der Leistungsparameter der Server in Analogie zur oben beschriebenen manuellen Zuweisung, wobei jedoch subjektive Faktoren weitgehend ausgeschlossen werden.

Eine weitere Möglichkeit die Berechnungszeiten zu reduzieren besteht in der Verwendung der leichgewichtigen Prozesse.

Kapitel 6

Einsatz von Tupeloperationen für die Bi-Decomposition

6.1 Kriterien der Bi-Decomposition

Aus den mehrfach in der Literatur beschriebenen Analysen kombinatorischer Schaltnetzwerke (z. B. [BoSt91]) ist bekannt, dass jeder Gatteranschluss einer kombinatorischen Schaltung Träger einer binären Funktion ist. Die Funktion am Gatterausgang wird eindeutig durch die im Gatter verwirklichte logische Verknüpfung der binären Funktionen an den Gattereingängen gebildet. Das dekombinatorische Syntheseverfahren für kombinatorische Schaltnetzwerke besteht damit in der Zerlegung einer binären Funktion in zwei oder mehrere einfachere Funktionen, so dass diese in Verbindung mit einem verfügbaren Gatter die Ausgangsfunktion erzeugen. Die Dekompositionsstrategie bekannt als Bi-Decomposition [SaBu97, MSP01] oder Gruppierung [Le89, BoSt91, BDS91, Dres92] wird im weiteren genauer betrachtet.

Unter Bi-Decomposition wird die Zerlegung einer Booleschen Funktion $f(\underline{x_a}, \underline{x_b}, \underline{x_c})$ in zwei Funktionen $g(\underline{x_a}, \underline{x_c})$ und $h(\underline{x_b}, \underline{x_c})$ mit nicht leeren, disjunkten Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ entsprechend der Formel (6.1) verstanden, wobei $\circ \in \{\vee, \cdot, \oplus\}$.

$$f(\underline{x_a}, \underline{x_b}, \underline{x_c}) = g(\underline{x_a}, \underline{x_c}) \circ h(\underline{x_b}, \underline{x_c}) \tag{6.1}$$

In Abhängigkeit vom Operationszeichen in der Formel (6.1) handelt es sich um eine OR-, AND- oder EXOR-Bi-Decomposition (bzw. disjunktive, konjunktive oder antivalente Gruppierung), die sich schaltungtechnisch mit einem OR-, AND- oder EXOR-Gatter realisieren lässt (siehe Abbildung 6.1). Die größte Vereinfachung einer mit der Funktion $f(\underline{x_a}, \underline{x_b}, \underline{x_c})$ beschriebenen Schaltung ergibt sich, wenn alle Variablen von $f(\underline{x_a}, \underline{x_b}, \underline{x_c})$ in disjunkte (möglichst gleichmächtige) Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ aufgeteilt werden können und die Variablenmenge $\underline{x_c}$ keine Variablen enthält. Es gibt Boolesche Funktionen, für die keine Bi-Decomposition (6.1) existiert. In diesem Fall ist nach [Le89] die Weak-Bi-Decomposition (Schwachgruppierung) einer Booleschen Funktion $f(\underline{x}) = f(\underline{x_a}, \underline{x_c})$ in zwei Funktionen $g(\underline{x_a}, \underline{x_c})$ und $h(\underline{x_c})$ bezüglich $\circ \in \{\vee, \cdot\}$ und der Variablenmenge $\underline{x_a}$ entsprechend (6.2) möglich (siehe Abbildung 6.2).

$$f(\underline{x_a}, \underline{x_c}) = g(\underline{x_a}, \underline{x_c}) \circ h(\underline{x_c}) \tag{6.2}$$

Weist eine Boolesche Funktion sowohl die OR- wie auch die AND-Weak-Bi-Decomposition nicht auf, so ist die EXOR-Bi-Decomposition möglich.

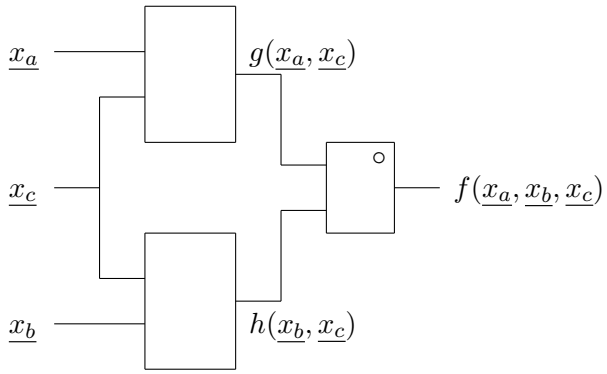


Abbildung 6.1: Schaltungsstruktur für OR-, AND- bzw. EXOR-Bi-Decomposition

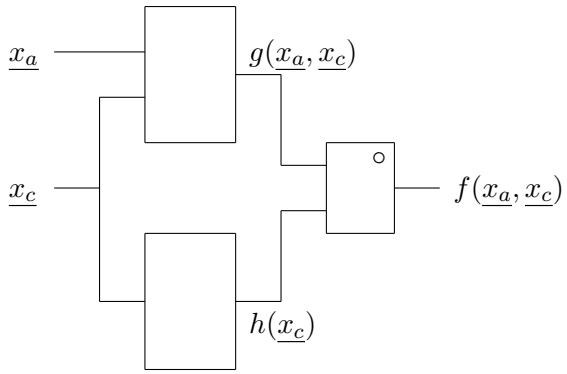


Abbildung 6.2: Schaltungsstruktur für OR-, AND- bzw. EXOR-Weak-Bi-Decomposition

Wird an Stelle einer Funktion von einem Funktionsverband ausgegangen, erhöht sich die Wahrscheinlichkeit, dass eine Funktion mit der Eigenschaft Bi-Decomposition in diesem Funktionsverband enthalten ist. Das Kriterium der Bi-Decomposition wird für den Funktionsverband wie folgt formuliert: Eine Bi-Decomposition einer Booleschen Funktion $f(\underline{x}) = f(\underline{x_a}, \underline{x_b}, \underline{x_c})$ ist bezüglich der Variablen $\underline{x_a}$, $\underline{x_b}$ und der Operation $\circ \in \{\vee, \cdot, \oplus\}$ unter der Nebenbedingung $\varphi(\underline{x})$ möglich, wenn eine Funktion $f^*(\underline{x})$ aus dem Funktionsverband mit der oberen Grenze $p(\underline{x})$ (6.3) und der unteren Grenze $q(\underline{x})$ (6.4) existiert, für die (6.5) gilt.

$$p(\underline{x}) = f(\underline{x}) \vee \varphi(\underline{x}) \tag{6.3}$$

$$q(\underline{x}) = f(\underline{x}) \cdot \overline{\varphi(\underline{x})} \tag{6.4}$$

$$f^*(\underline{x}) = g(\underline{x_a}, \underline{x_c}) \circ h(\underline{x_b}, \underline{x_c}) \tag{6.5}$$

Die Bedingungen für die OR- und AND-Bi-Decompositionen lassen sich mit Hilfe der Verbandskennfunktionen $q(\underline{x})$ und $r(\underline{x})$ wie folgt formulieren:

- die OR-Bi-Decomposition der Funktion $f(\underline{x})$ ist möglich, wenn die Gleichung (6.6) erfüllt ist,
$$q(\underline{x}) \cdot \max_{\underline{x_a}}{}^k r(\underline{x}) \cdot \max_{\underline{x_b}}{}^k r(\underline{x}) = 0 \tag{6.6}$$

- die AND-Bi-Decomposition der Funktion $f(\underline{x})$ ist möglich, wenn die Gleichung (6.7) erfüllt ist.
$$r(\underline{x}) \cdot \max_{\underline{x_a}}{}^k q(\underline{x}) \cdot \max_{\underline{x_b}}{}^k q(\underline{x}) = 0 \tag{6.7}$$

Die Bedingung der EXOR-Bi-Decomposition wurde in [BoSt91, Le89, Dres92, StWe95] formuliert. An dieser Stelle soll die EXOR-Bi-Decomposition nicht näher betrachtet werden.

Das Kriterium für die OR-Bi-Decomposition besagt, dass sich keine Einsbelegung ($q(\underline{x})$) in den Nullprojektionen bezüglich beider Gruppierungsrichtungen ($\max_{\underline{x_a}}{}^k r(\underline{x})$ und $\max_{\underline{x_b}}{}^k r(\underline{x})$) befinden darf. Die AND-Bi-Decomposition schließt aus, dass eine Nullbelegung ($r(\underline{x})$) in den Einsprojektionen bezüglich beider Gruppierungsrichtungen ($\max_{\underline{x_a}}{}^k q(\underline{x})$ und $\max_{\underline{x_b}}{}^k q(\underline{x})$) vorkommt. Weiterhin ist aus den Formeln (6.6) und (6.7) erkennbar, dass sich die Sperrfunktion $\varphi(\underline{x})$ auf die Bi-Decomposition positiv auswirkt. Die Sperrfunktion tritt zwar in den Formeln (6.6) und (6.7) nicht auf, verkleinert aber die Kennfunktionen $q(\underline{x})$ und $r(\underline{x})$.

6.2 Suche der Variablenmengen der Bi-Decomposition

6.2.1 Analyse

Eine wesentliche Aufgabe bei der Durchführung einer Bi-Decomposition besteht in der Suche nach den geeigneten Variablenmengen $\underline{x_a}$ und $\underline{x_b}$. Die Lösung ist nach [SSS93, StSt94, MSP01] in zwei Schritten möglich:

- Zunächst werden zwei einzelne Variablen in der Variablenmenge $\underline{x}$ gesucht, die die Dekomposition ermöglichen und damit die Initialisierungsmengen $\underline{x_a}$ und $\underline{x_b}$ bilden (siehe Algorithmus in der Abbildung 6.3).

```
1     gefunden=false
2     solange gefunden==false und für alle x ∈ x
2.1       x_a = {x}
2.2       x = x − {x}
2.3       solange nicht gefunden und für alle y ∈ x
2.3.1         x_b = {y}
2.3.2         if (q(x) · max_{x_a}^k r(x) · max_{x_b}^k r(x) == 0) gefunden=true
```

Abbildung 6.3: Algorithmus der Suche der Initialisierungsvariablenmengen für die OR-Bi-Decomposition

```
1       für alle z ∈ x − (x_a ∪ x_b)
1.1         if Anzahl der Variablen in x_a kleiner als in x_b
1.1.1           x_a = x_a ∪ z
1.1.2           if (q(x) · max_{x_a}^k r(x) · max_{x_b}^k r(x) ≠ 0)
1.1.2.1             x_a = x_a − {z}
1.1.2.2             x_b = x_b ∪ z
1.1.2.3             if (q(x) · max_{x_a}^k r(x) · max_{x_b}^k r(x) ≠ 0) x_b = x_b − {z}
1.2         sonst
1.2.1           x_b = x_b ∪ z
1.2.2           if (q(x) · max_{x_a}^k r(x) · max_{x_b}^k r(x) ≠ 0)
1.2.2.1             x_b = x_b − {z}
1.2.2.2             x_a = x_a ∪ z
1.2.2.3             if (q(x) · max_{x_a}^k r(x) · max_{x_b}^k r(x) ≠ 0) x_a = x_a − {z}
```

Abbildung 6.4: Algorithmus der Suche der optimalen Variablenmengen für die OR-Bi-Decomposition

- Im zweiten Schritt werden ausgehend von den Initialisierungsmengen die noch nicht untersuchten Variablen aus der Variablenmenge $\underline{x}$ der Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ nacheinander hinzugefügt. Die neu entstandenen Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ sind danach auf die Durchführbarkeit der Dekomposition zu testen (siehe Algorithmus in der Abbildung 6.4). Gesucht werden dabei die optimalen Variablenmengen $\underline{x_a}$ und $\underline{x_b}$, d. h. die Variablenmengen, die sich in der Anzahl der Variablen möglichst wenig unterscheiden.

Die Algorithmen der Abbildungen 6.3 und 6.4 ermöglichen die Durchführbarkeit der OR-Bi-Decomposition zu untersuchen. Bei der Beurteilung der Variablenmengen bezüglich der Durchführbarkeit der AND-Bi-Decomposition wird nur die Berechnung $q(\underline{x}) \cdot \max\limits_{\underline{x_a}}^k r(\underline{x}) \cdot \max\limits_{\underline{x_b}}^k r(\underline{x})$ durch $r(\underline{x}) \cdot \max\limits_{\underline{x_a}}^k q(\underline{x}) \cdot \max\limits_{\underline{x_b}}^k q(\underline{x})$ ersetzt, sonst sind die Algorithmen für OR- und AND-Bi-Decomposition identisch.

Die konjunktiven Verknüpfungen $q(\underline{x}) \cdot \max_{\underline{x_a}}{}^k r(\underline{x}) \cdot \max_{\underline{x_b}}{}^k r(\underline{x})$ werden innerhalb der Algorithmen 6.3 und 6.4 wiederholt für die in Abhängigkeit von den Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ geänderten Funktionen $\max_{\underline{x_a}}{}^k r(\underline{x})$ und $\max_{\underline{x_b}}{}^k r(\underline{x})$ durchgeführt. Um eine Funktion mit n Variablen auf die OR-Bi-Decomposition zu untersuchen, werden in worst case bei der Suche nach der Initialisierungsvariablenmengen $\sum_{i=1}^{n-1}(n-i) = n\frac{n-1}{2}$ solche Berechnungen benötigt. Da im zweiten Schritt der Suche nach vollständigen Mengen $\underline{x_a}$ und $\underline{x_b}$ jede der $n-2$ noch nicht in $\underline{x_a}$ und $\underline{x_b}$ enthaltenen Variablen in worst case durch das Hinzufügen der Variable zu den Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ zweimal getestet werden muss, ergeben sich hier maximal $2(n-2)$ weitere Berechnungen. Damit sind bei der Untersuchung einer Booleschen Funktion mit n Variablen auf die OR-Bi-Decomposition in worst case insgesamt $2n\frac{n-1}{2} + 4(n-2)$ konjunktive Verknüpfungen notwendig.

Die Analysen im Kapitel 4 (Abschnitt 4.1.3) haben ergeben, dass die Tupelkonjunktion der in drei Teilfunktionen zerlegten Funktionen wesentlich schneller als die Konjunktion der gleichen nicht zerlegten Funktionen abläuft. Aus dem Diagramm in der Abbildung 4.8 ist die Reduzierung des Zeitbedarfs im Ergebnis der Zerlegung bis auf etwa ein Drittel ersichtlich. Damit kann unterstellt werden, dass die Suche der Gruppierungsvariablenmengen $\underline{x_a}$ und $\underline{x_b}$ der OR- bzw. AND-Bi-Decomposition unter Verwendung der Tupelkonjunktion effektiver abläuft.

Aus den weiteren im Kapitel 5 beschriebenen Untersuchungen folgt, dass für große Funktionen die Verteilung der Teilberechnungen der Tupelkonjunktion auf unterschiedliche Rechner sinnvoll ist. Nach der Tabelle 5.1 erwies sich eine Verteilung auf zwei Rechner für die Booleschen Funktionen mit mehr als 30000 Konjunktion (bei der Variablenanzahl gleich 20) gegenüber der sequentiellen Verarbeitung als vorteilhafter. Der Einsatz weiterer Rechner führte zu einer zusätzlichen Reduzierung der Verarbeitungszeit, wobei aber mit steigender Rechnerzahl die erzielte Effekte abnahmen (siehe Tabelle 5.2). Der degressive Verlauf ist damit zu erklären, dass auf Grund der beschränkten Anzahl der Teilaufgaben (in der Tupelkonjunktion sind das 7 Teilaufgaben) die geringere Leistungfähigkeit eines langsameren Servers nur teilweise durch einen leistingsfähigeren Server ausgeglichen werden kann. Im Kapitel 5 wurde das prinzipielle Herangehen beim Zuordnen der Teilaufgaben zu den verfügbaren Server beschrieben. Danach wird allen Servern beim Start einer Aufgabe die Bearbeitung genau eines Auftrags übertragen. Nach der Fertigstellung eines Auftrags wird der frei werdende Server mit einem noch nicht bearbeiteten Auftrag betraut. Sind keine weiteren Aufträge zu vergeben, bleiben die frei gewordenen Server unausgelastet bis die Gesamtbearbeitung mit der Fertigstellung des letzten Auftrags beendet ist. Diese "Überhangzeit" beeinflusst bei einer geringen Anzahl von Aufträgen deutlich die Gesamtverarbeitungszeit. Bei der Suche der Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ der Bi-Decomposition ist die Anzahl der zu bearbeitenden Aufträge bedeutend größer und die Auswirkung der "Überhangzeit" im Vergleich zu Gesamtverarbeitungszeit demzufolge wesentlich geringer.

Wie bereits erwähnt, sind bei der Suche der Initialisierungsvariablenmengen der Bi-Decomposition einer Booleschen Funktion mit n Variablen bis $2n\frac{n-1}{2}$ Konjunktionen bzw. bei der Verwendung einer Tupeldarstellung Tupelkonjunktionen zu berechnen. Für die Suche der vollständigen Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ werden weitere maximal $4(n-2)$ Tupelkonjunktionen benötigt. Die Analysen des Abschitts 5.1 haben gezeigt, dass die 7 bei der Durchführung einer Tupelkonjunktion benötigten Teilberechnungen parallel bearbeitet werden können (siehe Abbildung 5.2). Damit ergeben sich bei der Suche der Initialisierungsvariablenmengen bis $14n\frac{n-1}{2}$ Teilaufgaben, die theoretisch parallel berechnet werden

könnten. Praktisch wird die Anzahl parallel laufender Berechnungen einerseits durch die Anzahl der zur Verfügung stehenden Server begrenzt. Um unnötige Berechnungen zu vermeiden soll andererseits nach dem Auffinden der ersten geeigneten Variablen die Suche abgebrochen werden. Damit ist es sinnvoll, für jede Variable x (Kandidat der Variablenmenge $\underline{x_a}$) nur zwei Variablen y (Kandidaten der Variablenmenge $\underline{x_b}$) (die Schritte 2.3.1 und 2.3.2 des Algorithmus 6.3) gleichzeitig zu testen. Bei der anschließenden Suche der vollständigen Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ kann die Beurteilung jeder noch nicht untersuchten Variablen parallel für beide Variablenmengen (die Schritte 1.1.1-1.1.2.3 und die Schritte 1.2.1-1.2.2.3) erfolgen (siehe Abbildung 6.4).

6.2.2 Verwendung der Tupel-Dekomposition

Jede Boolesche Funktion kann durch Tupel aus drei Booleschen Funktionen abgebildet werden (siehe Abschitt 4.1.1). Diese Aussage gilt auch für die Verbandsfunktionen $q(\underline{x})$ und $r(\underline{x})$, die nach dem im Abschnitt 4.1.1 formulierten Satz 4.3 wie folgt dargestellt werden:

$$q(x_i, \underline{x_0}) = q^-(\underline{x_0}) \vee \overline{x_i} q^0(\underline{x_0}) \vee x_i q^1(\underline{x_0}) \tag{6.8}$$

$$r(x_i, \underline{x_0}) = r^-(\underline{x_0}) \vee \overline{x_i} r^0(\underline{x_0}) \vee x_i r^1(\underline{x_0}) \tag{6.9}$$

Die Suche nach den geeigneten Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ kann unter Verwendung dieser Darstellung nach den Algorithmen in den Abbildungen 6.3 und 6.4 erfolgen.

Werden die Funktionen $q(\underline{x})$ und $r(\underline{x})$ nach Satz 4.3 zerlegt, so müssen für die Berechnung $q(\underline{x}) \cdot \max\limits_{\underline{x_a}}^k r(\underline{x}) \cdot \max\limits_{\underline{x_b}}^k r(\underline{x})$ die in den Algorithmen der Bi-Decomposition benötigten Operationen k-faches Maximum und Konjunktion durch Tupeloperationen ersetzt werden. Das Ergebnis der Tupelkonjunktion wird dabei wie im Abschnitt 4.9 beschrieben nach den Formeln (4.101)-(4.103) ermittelt und kann in den Algorithmen der Bi-Decomposition durch einen einfachen Aufruf der Tupelkonjunktion realisiert werden. Das k-fache Maximum $g(\underline{x_0}, x)$ einer in 3 Teile zerlegten Booleschen Funktion $f(\underline{x_0}, x, \underline{x_i})$ lässt sich nach der Formel (6.13) berechnen.

Die Richtigkeit der Formel (6.13) wird wie folgt bewiesen:
Nach dem Einsetzen von (4.27) für die nach der Variablen x zerlegte Funktion $f(\underline{x_0}, x, \underline{x_i})$ in (6.10) entsteht (6.11).

$$g(\underline{x_0}, x) = \max_{\underline{x_i}}^k f(\underline{x_0}, x, \underline{x_i}) \tag{6.10}$$

$$g(\underline{x_0}, x) = \max_{\underline{x_i}}^k (f^-(\underline{x_0}, \underline{x_i}) \vee \overline{x} f^0(\underline{x_0}, \underline{x_i}) \vee x f^1(\underline{x_0}, \underline{x_i})) \tag{6.11}$$

Die Aussage (6.12) folgt unmittelbar aus der Eigenschaft (2.25) des k-fachen Maximums (siehe Abschnitt 2.1.2).

$$g(\underline{x_0}, x) = \max_{\underline{x_i}}^k f^-(\underline{x_0}, \underline{x_i}) \vee \max_{\underline{x_i}}^k (\overline{x} f^0(\underline{x_0}, \underline{x_i})) \vee \max_{\underline{x_i}}^k (x f^1(\underline{x_0}, \underline{x_i})) \tag{6.12}$$

Da die Funktionen $f^0(\underline{x_0}, \underline{x_i})$ und $f^1(\underline{x_0}, \underline{x_i})$ von der Variable x unabhängig sind, folgt aus (6.12) unter Berücksichtigung von (2.22) die Formel (6.13).

$$g(\underline{x_0}, x) = \max_{\underline{x_i}}{}^k f^-(\underline{x_0}, \underline{x_i}) \vee \overline{x} \cdot \max_{\underline{x_i}}{}^k f^0(\underline{x_0}, \underline{x_i}) \vee x \cdot \max_{\underline{x_i}}{}^k f^1(\underline{x_0}, \underline{x_i}) \tag{6.13}$$

□

Werden für die Funktionen der Gleichung (6.13) die Bezeichnungen (6.14) eingeführt, kann der Ausdruck (6.13) in (6.15) überführt werden.

$$\begin{aligned} g'^-(\underline{x_0}) &= \max_{\underline{x_i}}{}^k f^-(\underline{x_0}, \underline{x_i}) \\ g'^0(\underline{x_0}) &= \max_{\underline{x_i}}{}^k f^0(\underline{x_0}, \underline{x_i}) \\ g'^1(\underline{x_0}) &= \max_{\underline{x_i}}{}^k f^1(\underline{x_0}, \underline{x_i}) \end{aligned} \tag{6.14}$$

$$g(\underline{x_0}, x) = g'^-(\underline{x_0}) \vee \overline{x} \cdot g'^0(\underline{x_0}) \vee x \cdot g'^1(\underline{x_0}) \tag{6.15}$$

Die Funktionen $g'^-(\underline{x_0})$, $g'^0(\underline{x_0})$ und $g'^1(\underline{x_0})$ sind keine disjunkte Funktionen. Werden die disjunkten Teile der Funktionen $g'^-(\underline{x_0})$, $g'^0(\underline{x_0})$ und $g'^1(\underline{x_0})$ entsprechend als $g^{--}(\underline{x_0})$, $g^{00}(\underline{x_0})$ und $g^{11}(\underline{x_0})$ bezeichnet und die gemeinsamen Teile als $g^{-0}(\underline{x_0})$, $g^{-1}(\underline{x_0})$ und $g^{01}(\underline{x_0})$, so gilt (6.16).

$$\begin{aligned} g'^-(\underline{x_0}) &= g^{--}(\underline{x_0}) \vee g^{-0}(\underline{x_0}) \vee g^{-1}(\underline{x_0}) \\ g'^0(\underline{x_0}) &= g^{00}(\underline{x_0}) \vee g^{-0}(\underline{x_0}) \vee g^{01}(\underline{x_0}) \\ g'^1(\underline{x_0}) &= g^{11}(\underline{x_0}) \vee g^{-1}(\underline{x_0}) \vee g^{01}(\underline{x_0}) \end{aligned} \tag{6.16}$$

Durch das Einsetzen der Funktionen (6.16) in den Ausdruck (6.15) wird (6.17) erhalten.

$$\begin{aligned} g(\underline{x_0}, x) = {} & g^{--}(\underline{x_0}) \vee g^{-0}(\underline{x_0}) \vee g^{-1}(\underline{x_0}) \vee \\ & \vee \overline{x} \cdot (g^{00}(\underline{x_0}) \vee g^{-0}(\underline{x_0}) \vee g^{01}(\underline{x_0})) \\ & \vee x \cdot (g^{11}(\underline{x_0}) \vee g^{-1}(\underline{x_0}) \vee g^{01}(\underline{x_0})) \end{aligned} \tag{6.17}$$

$$\begin{aligned} g(\underline{x_0}, x) = {} & g^{--}(\underline{x_0}) \vee g^{-0}(\underline{x_0}) \vee g^{-1}(\underline{x_0}) \vee (g^{01}(\underline{x_0})) \cdot (\overline{x} \vee x) \\ & \vee \overline{x} \cdot g^{00}(\underline{x_0}) \vee \overline{x} \cdot g^{-0}(\underline{x_0}) \\ & \vee x \cdot g^{11}(\underline{x_0}) \vee x \cdot g^{-1}(\underline{x_0}) \end{aligned} \tag{6.18}$$

Nach den elementaren Umformungen entsteht (6.19).

$$\begin{aligned} g(\underline{x_0}, x) = {} & g^{--}(\underline{x_0}) \vee g^{-0}(\underline{x_0}) \vee g^{-1}(\underline{x_0}) \vee g^{01}(\underline{x_0}) \\ & \overline{x} \cdot g^{00}(\underline{x_0}) \\ & \vee x \cdot g^{11}(\underline{x_0}) \end{aligned} \tag{6.19}$$

Damit ergeben sich für die Berechnung der disjunkten Teilfunktionen $g^-(\underline{x_0})$, $g^0(\underline{x_0})$ und $g^1(\underline{x_0})$ die Vorschriften (6.20).

$$\begin{aligned} g^-(\underline{x_0}) &= g'^-(\underline{x_0}) \vee g'^0(\underline{x_0}) \cdot g'^1(\underline{x_0}) \\ g^0(\underline{x_0}) &= g'^0(\underline{x_0}) \setminus g^-(\underline{x_0}) \\ g^1(\underline{x_0}) &= g'^1(\underline{x_0}) \setminus g^-(\underline{x_0}) \end{aligned} \tag{6.20}$$

In Algorithmen 6.3 und 6.4 werden die Ergebnisse der Operation Maximum als Operanden einer Tupelkonjunktion verwendet. Die Tupelkonjunktion kann auch mit den durch (6.14) dargestellten Funktionen ausgeführt werden. Die Teilfunktionen $g'^{-}(\underline{x_0})$, $g'^{0}(\underline{x_0})$ und $g'^{1}(\underline{x_0})$ der Ergebnisfunktion sind in diesem Fall ebenfalls keine disjunkte Funktionen, können aber nach (6.20) in die disjunkten Teilfunktion $g^{-}(\underline{x_0})$, $g^{0}(\underline{x_0})$ und $g^{1}(\underline{x_0})$ überführt werden. Bei der Suche der Dekompositionsmengen $\underline{x_a}$ und $\underline{x_b}$ ist die Aussage wesentlich, ob es sich bei der Ergebnisfunktion um eine 0-Funktion handelt. Bei der Darstellung einer 0-Funktion durch ein TVL-Tupel sind alle Elemente des Tupels die leeren TVL. Somit kann auf die Umrechnung nicht orthogonaler Teilfunktionen in orthogonale Teilfunktionen verzichtet werden. Es muss nur geprüft werden, ob die Funktionen $g'^{-}(\underline{x_0})$, $g'^{0}(\underline{x_0})$ und $g'^{1}(\underline{x_0})$ 0-Funktionen sind.

6.2.3 Verwendung der verteilten Verarbeitung

Die analytischen Betrachtungen im Abschnitt 6.2.1 haben ergeben, dass die Suche der Gruppierungsvariablenmengen $\underline{x_a}$ und $\underline{x_b}$ auf der Basis einer verteilten Lösung effektiv sein kann. Die bei der Berechnung von $q(\underline{x}) \cdot \max\limits_{\underline{x_a}}{}^{k} r(\underline{x}) \cdot \max\limits_{\underline{x_b}}{}^{k} r(\underline{x})$ auszuführenden konjunktiven Verknüpfungen werden dabei nicht durch jeweils zwei Aufrufe der im Kapitel 5 beschriebenen verteilten Tupelkonjunktion realisiert. Vielmehr werden alle für die Suche der Gruppierungsvariablenmengen $\underline{x_a}$ und $\underline{x_b}$ benötigten konjunktiven Verknüpfungen in einem Unterprogramm ausgeführt.

Die parallele Suche der Gruppierungsvariablenmengen $\underline{x_a}$ und $\underline{x_b}$ in einem Unterprogramm kann auf der Basis der Algorithmen in Abbildungen 6.3 und 6.4 realisiert werden, führt jedoch zum zusätzlichen algorithmischen Aufwand:

- Die Überprüfung des Kriteriums (6.6) erfordert die Berechnung von zwei konjunktiven Verknüpfungen: $K_1 = q(\underline{x}) \cdot \max\limits_{\underline{x_a}}{}^{k} r(\underline{x})$ und $K_2 = K_1 \cdot \max\limits_{\underline{x_b}}{}^{k} r(\underline{x})$. Im Algorithmus zur parallelen Suche der Mengen $\underline{x_a}$ und $\underline{x_b}$ ist zu berücksichtigen, dass die Bearbeitung der Konjunktion K_2 die Bereitstellung aller Teilergebnisse der Konjunktion K_1 voraussetzt. D. h. nach der Vergabe aller Teilaufträge für K_1 kann die Vergabe der Teilaufträge für K_2 nicht unmittelbar, sondern erst nach dem Bereitstellen aller Teilergebnisse der ersten Konjunktion erfolgen. Um eine durchgehende Auslastung der verfügbaren Server zu sichern, soll der Vergabe der Teilaufträge der Konjunktion K_1 die Vergabe der Teilaufträge für K_1 der nächsten Berechnung mit geänderten Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ folgen. Im Algorithmus für die Suche der Gruppierungsvariablenmengen im Parallelverfahren werden deswegen zwei aufeinanderfolgende Berechnungen betrachtet.

- Die Änderung der Variablenmengen wird Unterschiede aufweisen, die im Algorithmus ebenfalls zu berücksichtigen sind, wobei die Suche nach Initialisierungsmengen bzw. vollständigen Variablenmengen zu unterscheiden ist. Bei der Suche nach Initialisierungsmengen muss zunächst die Eignung des nächsten Variablenpaars als Initialisierungsmenge $\underline{x_a}$ und $\underline{x_b}$ überprüft werden. Bei der Suche der vollständigen Variablenmengen sind die noch nicht untersuchten Variablen der Variablenmenge $\underline{x}$ abwechselnd den Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ hinzuzufügen um optimale Variablenmengen zu ermitteln. Das wird durch die Zuordnung der soeben betrachteten Variablen zu

```
1       Auswählen 1 Variable für x_a und 2 Variablen für x_b
2       Berechnen max^k_{x_a} r(x) und max^k_{x_b} r(x) für beide x_b-Mengen
3       Erteilen der Aufträge entsprechend Serverzahl
4       solange weitere Aufträge vorhanden
4.1         Lesen des Teilergebnises des fertigen Auftrags
4.2         falls weitere Aufträge vorhanden: Auftragswiedervergabe
4.3         falls Gesamtergebnis K_1 und Teilergebnisse K_2 vorhanden
4.3.1           Gesamtberechnung K_2
4.3.2           falls Bi-Decomposition nicht möglich
4.3.2.1             falls Initialisierungsmengen gesucht
                        Auswahl der nächsten Variable für x_b
                        bzw. für x_a und x_b
                        Berechnen von max^k_{x_a} r(x) und max^k_{x_b} r(x)
4.3.2.2             sonst
                        Entfernen der untersuchten Varable aus x_a oder x_b
                        Hinzufügen der nächsten Variable zu x_a oder x_b
                        Berechnen von max^k_{x_a} r(x) oder max^k_{x_b} r(x)
4.3.3           sonst
4.3.3.1             falls Initialisierungsmengen gesucht
                        Hinzufügen der nächsten Variablen zu x_a und x_b
                        Berechnen von max^k_{x_a} r(x) für die 1.Berechnung
                        Berechnen von max^k_{x_b} r(x) für die 2.Berechnung
4.3.3.2             sonst
                        Hinzufügen der nächsten Variable zu x_a oder x_b
                        Berechnen von max^k_{x_a} r(x) oder max^k_{x_b} r(x)
4.4         sonst
4.4.1           falls Teilergebnisse K_1 vorhanden
                    Gesamtberechnung K_1
```

Abbildung 6.5: Algorithmus der Suche der Gruppierungsvariablenmengen mit verteilter Verarbeitung

der anderen Gruppierungsvariablenmenge bzw. durch die Untersuchung der nächsten Variablen realisiert.

Der Algorithmus für die Suche der Gruppierungsvariablenmengen im Parallelverfahren ist in der Abbildung 6.5 in Übersichtsform dargestellt. Wie dem Algorithmus zu entnehmen ist, werden nach dem Bereitstellen der Funktionen $\max^k_{\underline{x_a}} r(\underline{x})$ und $\max^k_{\underline{x_b}} r(\underline{x})$ für die zwei ersten Berechnungen des Kriteriums (6.6) (Schritte 1 und 2) die Aufträge an alle zur Verfügung stehende Server erteilt (Schritt 3). Ein Auftrag entspricht dabei in Analogie zu den Untersuchungen im Kapitel 5 der Teilberechnung einer Tupelkonjunktion. Der weitere Programmablauf besteht in der wiederholten Entgegennahme des Ergebnisses des Auftrags (Schritt 4.1), der Berechnung der Ergebnisfunktion $q(\underline{x}) \cdot \max^k_{\underline{x_a}} r(\underline{x}) \cdot \max^k_{\underline{x_b}} r(\underline{x})$ und dem Bereitstellen der Funktionen $\max^k_{\underline{x_a}} r(\underline{x})$ und $\max^k_{\underline{x_b}} r(\underline{x})$ für weitere Berechnungen.

Wie bereits erwähnt, sind bei der Berechnung des Kriteriums (6.6) zwei Konjunktionen

K_1 und K_2 notwendig. Das Gesamtergebnis der Konjunktion K_1 kann nach der erfolgreichen Bereitstellung aller Zwischenergebnisse von unterschiedlichen Servern berechnet werden (Schritt 4.4.1). Die Kontrolle des Vorhandenseins aller Zwischenergebnisse ist nach dem Eingang jedes Teilergebnisses auszuführen, um die Berechnung des Gesamtergebnisses ohne Verzögerung einleiten zu können. Der Berechnung der Konjunktion K_2 wird zusätzlich das Vorhandensein des Gesamtergebnisses der ersten Konjunktion vorausgesetzt. Auch diese Überprüfung findet nach dem Eingang jedes Teilergebnisses statt (Schritt 4.3).

Bei der darauf folgenden Änderung der Variablenmengen für die weiteren Untersuchungen wird berücksichtigt, ob die OR-Bi-Decomposition mit den eben untersuchten Variablenmengen $\underline{x_a}$ und $\underline{x_b}$ nicht möglich (K_2 ist eine 0-Funktion) oder möglich ist (Schritte 4.3.2 und Schritt 4.3.3). Innerhalb dieser Unterscheidung wird bei der Änderung der Variablenmengen eine weitere Unterteilung in die Suche nach Initialisierungsmengen (Schritte 4.3.2.1 und 4.3.3.1) und vollständigen Variablenmengen (Schritte 4.3.2.2 und 4.3.3.2) vorgenommen. In jedem dieser vier Zweige werden für die geänderten Variablenmengen die Funktionen $\max^k_{\underline{x_a}} r(\underline{x})$ und $\max^k_{\underline{x_b}} r(\underline{x})$ sequentiell berechnet.

6.2.4 Ergebnisse

Die Analysen des Abschnitts 6.2.1 wurden durch praktische Untersuchungen mit den Algorithmen der Abschnitte 6.2.1-6.2.3 überprüft. Als zu untersuchenden Daten wurden Boolesche Funktionen mit 18, 19 und 20 Variablen verwendet. Die Größe der Funktionen wurde bewusst gewählt, da aus den Ausführungen im Kapitel 4 bekannt ist, dass die Darstellung Boolescher Funktionen durch einen Tupel aus 3 Teilfunktionen erst für Funktionen einer bestimmten Größe sinnvoll ist. Weiterhin geht aus den Untersuchungen des Kapitels 5 hervor, dass die verteilte Verarbeitung bei kleineren Funktionen nicht gerechtfertigt ist. Diese schon bewiesenen Tatsachen wurden bei der Auswahl der zu untersuchenden Funktionen berücksichtigt.

Die Gruppierungsvariablenmengen der OR-Bi-Decomposition der Booleschen Funktionen $f(x_1, x_2, ..., x_n)$ eines Funktionsverbands, der durch die Kennfunktionen $q(x_1, x_2, ..., x_n)$ und $r(x_1, x_2, ..., x_n)$ representiert wird, werden in drei Untersuchungsreihen wie folgt bereitgestellt:

1. Die Funktionen $q(x_1, x_2, ..., x_n)$ und $r(x_1, x_2, ..., x_n)$ werden als nicht zerlegte Funktionen jeweils durch eine TVL dargestellt. Für die Suche der Gruppierungsvariablenmengen werden die Algorithmen 6.3 und 6.4 herangezogen.

2. Die Suche der Gruppierungsvariablenmengen erfolgt ebenfalls nach den Algorithmen 6.3 und 6.4. Die Funktionen $q(x_1, x_2, ..., x_n)$ und $r(x_1, x_2, ..., x_n)$ werden als Tupel aus 3 Teilfunktionen $q^-(x_2, ..., x_n)$, $q^0(x_2, ..., x_n)$, $q^1(x_2, ..., x_n)$, $r^-(x_2, ..., x_n)$, $r^0(x_2, ..., x_n)$ und $r^1(x_2, ..., x_n)$ in TVL-Form dargestellt.

3. Die Funktionen $q(x_1, x_2, ..., x_n)$ und $r(x_1, x_2, ..., x_n)$ werden in Analogie zur zweiten Untersuchngsreihe als Tupel aus 3 Teilfunktionen $q^-(x_2, ..., x_n)$, $q^0(x_2, ..., x_n)$, $q^1(x_2, ..., x_n)$, $r^-(x_2, ..., x_n)$, $r^0(x_2, ..., x_n)$ und $r^1(x_2, ..., x_n)$ in TVL-Form dargestellt. Für die Suche der Gruppierungsvariablenmengen wird der Algorithmus 6.5 unter Einsatz einer unterschiedlichen Serveranzahl verwendet.

Die Ergebnisse der ersten und der zweiten Untersuchungsreihe sind in der Tabelle 6.1 zusammengefasst. Wie erwartet, wird die für die Suche der Variablenmengen der Bi-Decomposition

Tabelle 6.1: Berechnungszeiten bei der Suche der Gruppierungsvariablenmengen mit Tupeloperationen

Varablen-	Zeilenzahl		Berechnungszeit (sek.)		Reduzierung
zahl	$q(\underline{x})$	$r(\underline{x})$	nicht zerlegt	Tupel	(%)
18	51 091	43 185	24 240	6 483	73,3
19	94 836	96 394	123 886	31 202	74,8
20	64 749	91 169	56 685	21 938	61,3

Tabelle 6.2: Berechnungszeiten bei der Suche der Gruppierungsvariablenmengen unter Einbeziehung mehrerer Server

Zeilenzahl		Berechnungszeit (sek.)				Reduzierung (%)		
$q(\underline{x})$	$r(\underline{x})$	seq.	2 Server	3 Server	4 Server	2 Server	3 Server	4 Server
51 091	43 185	6 483	6 075	4 852	3 457	6,3	25,2	46,7
94 836	96 394	31 202	24 146	19 004	13 772	22,6	39,1	55,9
64 749	91 169	21 938	16 643	11 937	9 888	24,1	45,6	54,9

notwendige Zeit durch die Verwendung der Tupelkonjunktion wesentlich reduziert. Es kann unterstellt werden, dass das verbesserte Zeitverhalten bei der Suche der Variablenmengen auf den Einsatz sowohl der Tupeloperation Konjunktion wie auch des Tupelmaximums zurückzuführen ist. Die Operation k-faches Maximum einer Booleschen Funktion in TVL-Darstellung nach der Variablenmenge $\underline{x}$ beinhaltet das Streichen der Spalten aller in $\underline{x}$ enthaltenen Variablen und das anschließende Orthogonalisieren der Funktions-TVL (siehe [BoSt91]). Das Streichen der Spalten geschieht unabhängig von der Anzahl der Konjunktionen einer Funktion in konstanter Zeit. Bei der Durchführung des Orthogonalisierens werden alle Zeilen einer Funktions-TVL miteinander verglichen (siehe Orthogonalitätssatz (2.4)). Für einer TVL mit N Zeilen sind folglich $\frac{N^2}{2} - \frac{N}{2}$ Vergleiche erforderlich. Bei dem Durchführen des Tupelmaximums wird die Maximumoperation an 3 TVL mit jeweils $N/2$ Zeilen angewandt. Damit ergeben sich für das Tupelmaximum $3 \cdot (\frac{(N/2)^2}{2} - \frac{N/2}{2})$ Vergleiche. Die Anzahl der Vergleiche für das Tupelmaximum ist kleiner als die Anzahl der Vergleiche des einfachen Maximums, wobei das Verhältnis im Grenzfall 3/4 beträgt (6.21).

$$\lim_{N\to\infty} \frac{3 \cdot (\frac{(N/2)^2}{2} - \frac{N/2}{2})}{\frac{N^2}{2} - \frac{N}{2}} = \lim_{N\to\infty} \frac{3}{4} \cdot \frac{N^2 - 2N}{N^2 - N} = \frac{3}{4} \tag{6.21}$$

In den Tabellen 6.1 und 6.2 sind Ergebnisse enthalten, die bei der Auswertung von Untersuchungen der gleichen Funktion gewonnen wurden. Im Gegensatz zu Tabelle 6.1 entstanden die Ergebnisse der Tabelle 6.2 durch eine Verteilung der Teilaufgaben auf mehrere Server. Die Tabelle enthält Bearbeitungszeiten, die beim Einsatz von 2, 3 und 4 Servern gemessen wurden. Aus der Gegenüberstellung mit der sequentiellen Verarbeitung ist ein spürbar verbessertes Zeitverhalten ersichtlich. Unter der Voraussetzung einer großen Zeilenzahl konnte für jeden weiteren genutzten Server eine weitere Verbesserung der Verarbeitungszahl erzielt

werden. Auf die Ursachen für die geringfügige Reduzierung der Bearbeitungszeit (6,3%) bei kleinen Aufgaben wurde bereits im Kapitel 5 eingegangen. Beim Unterschreiten einer kritischen Mindestgröße der Aufgabe wirken sich die zusätzlichen Aufwendungen für die Verteilung verstärkt aus, wobei die Größe dieses Aufwands von der Zahl der Aufträge und nicht von der Serverzahl abhängt. Diese Aussage begründet auch die oben gemessene Reduzierung der Bearbeitungszeit für jeden weiteren Server.

Die durch eine verteilte Lösung erzielten Zeiteinsparungen bei der Suche der Gruppierungsmengen $\underline{x_a}$ und $\underline{x_b}$ fallen geringer aus als bei der Ausführung einer Tupelkonjunktion (siehe Kapitel 5). Dieses Verhalten ist mit dem hohen Anteil nicht verteilter Berechnungen im Algorithmus 6.5 zu begründen. So werden bei der Berechnung der Tupeloperation nur abschließend die Funktionen ${f_3}^0$ und ${f_3}^1$ (siehe Abbildung 5.2) sequentiell berechnet. Dagegen erfolgt bei der Suche der Gruppierungsmengen die Berechnung der Funktionen $\max^k_{\underline{x_a}} r(\underline{x})$ und $\max^k_{\underline{x_b}} r(\underline{x})$ auch sequentiell. Schließlich besteht ein zusätzlicher Zeitbedarf für die Eingangskontrolle der von den Servern bereitgestellten Teilergebnisse.

Zusammenfassend ist herauszustellen, dass bei der Bearbeitung der Tupeloperationen Konjunktion und Maximum deutlich verbesserte Bearbeitungszeiten durch eine Verteilung von Teilaufgaben auf mehrere Server erzielt werden können. Vorausgesetzt wird eine Mindestgröße der Aufgaben, da nur in diesem Fall der zusätzliche Verteilungsaufwand im Vergleich zur unmittelbaren Berechnungszeit gering ausfällt. Es ist zu berücksichtigen, dass mit steigender Serverzahl ein erhöhter Speicherplatzbedarf für die am Client eingehenden Zwischenergebnisse besteht. Durch eine annähernd gleiche Leistungsfähigkeit der Server wird gesichert, dass die Bereitstellung der verteilt berechneten Teilaufgaben in einem vertretbaren Zeitraum erfolgt, so dass der für die Zwischenspeicherung benötigte Speicherplatz schneller freigegeben werden kann.

Kapitel 7

Schlussfolgerung und Zusammenfassung

Das Ziel dieser Arbeit bestand darin, durch die Dekomposition einer in TVL-Darstellung vorliegenden Booleschen Funktion effizientere Datenstrukturen und Algorithmen zu entwerfen. Angestrebt wurde dabei die Reduzierung der Komplexität der Darstellung einer Booleschen Funktion und ihre vereinfachte Handhabung, deren praktische Auswirkung unter folgenden Aspekten zu sehen ist:

- Aus dem Speicherplatzbedarf einer Booleschen Funktion kann auf die Effizienz der Datenstruktur geschlossen werden.
- Die Zeit, die für das Ausführen der Grundoperationen zwischen zwei Booleschen Funktionen notwendig ist, ist als Bewertungsgröße für die Effizienz der Datenstruktur und den Algorithmen der Operationen geeignet.

Unter diesen Gesichtspunkten wurde die Datenstruktur TVL-Tupel betrachtet. Die Datenstruktur TVL-Tupel entsteht durch Dekomposition einer Booleschen Funktion in TVL-Darstellung in mehrere Teilfunktionen. Im Kapitel 3 traten als Tupelelemente lokale Phasenlisten auf, die sich als geeignetere Darstellungsform gegenüber der globalen Phasenliste erwiesen. Die praktischen Untersuchungen mit den Benchmark-Schaltungen LGSynth91 bestätigten weiterhin, dass der Tupel aus lokalen Phasenlisten optimierter Größe, die durch begrenzte Komposition von lokalen Phasenlisten entstehen, eine bezüglich des Speicherplatzbedarfs noch günstigere Darstellungsform ist. Als eine weitere Erkenntnis ist herauszustellen, dass die Dekomposition einer Booleschen Funktion das Speichern von einzelnen Tupelelementen in verschiedenen Booleschen Räumen erlaubt und damit zur weiteren Reduzierung des Speicherplatzbedarfs führt.

Im Kapitel 4 enstanden die Elemente des Tupels im Ergebnis der Zerlegung einer Booleschen Funktion $f(x_i, \underline{x_0})$ in drei bzw. vier Teilfunktionen nach (7.1) und (7.2).

$$f(x_i, \underline{x_0}) = f^{-}(\underline{x_0}) \vee \overline{x_i} f^{0}(\underline{x_0}) \vee x_i f^{1}(\underline{x_0}) \tag{7.1}$$

$$f(x_i, \underline{x_0}) = 1 \cdot f^{-}(\underline{x_0}) \vee \overline{x_i} \cdot f^{0}(\underline{x_0}) \vee x_i \cdot f^{1}(\underline{x_0}) \vee 0 \cdot f^{=}(\underline{x_0}) \tag{7.2}$$

Die Elemente des Tupels sind somit die Funktionen $f^{-}(\underline{x_0})$, $f^{0}(\underline{x_0})$, $f^{1}(\underline{x_0})$ und $f^{=}(\underline{x_0})$, die mit den Formeln (4.20)-(4.22) und (4.161) definiert werden. Die Beziehungen zwischen

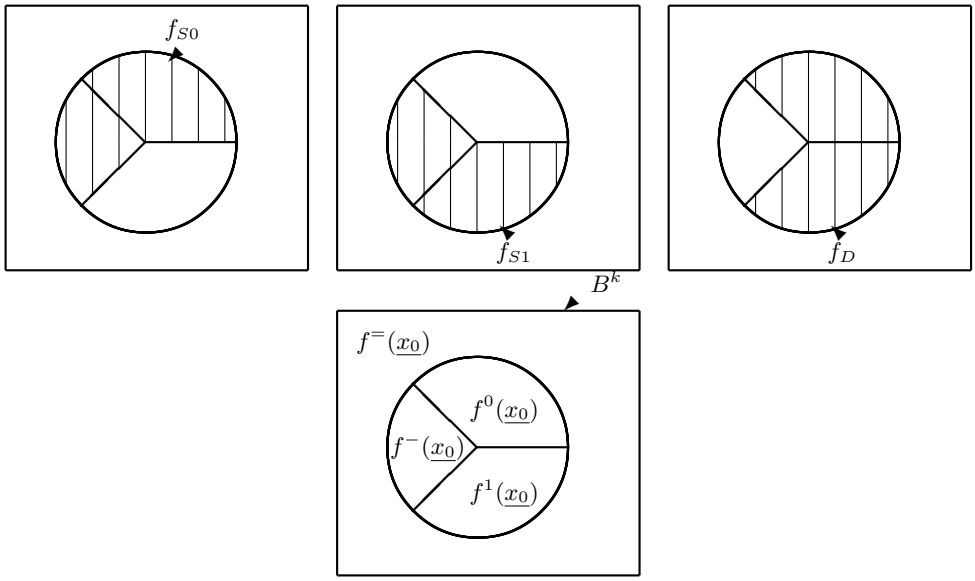


Abbildung 7.1: Verhältnis zwischen den Teilfunktionen der Tupel-Dekomposition und den Funktionen der Shanon- und Davio-Dekomposition

den Teilfunktionen $f^-(\underline{x_0})$, $f^0(\underline{x_0})$, $f^1(\underline{x_0})$ und den Funktionen der Shannon- und Davio-Dekomposition können mit (7.3) beschrieben werden (siehe auch Abbildung 7.1).

$$\begin{aligned} f_{S0} &= f^0 \vee f^- \\ f_{S1} &= f^1 \vee f^- \\ f_D &= f^0 \vee f^1 \end{aligned} \tag{7.3}$$

Die Tupel-Dekompositionen und die bekannten Dekomposition (1.1)-(1.5) (siehe Kapitel 1) lassen sich allgemein mit (7.4) beschreiben, wobei $\circ \in \{\vee, \oplus\}$ ist.

$$f = 1 \cdot f_1 \circ \overline{x}\, f_2 \circ x f_3 \circ 0 \cdot f_4 \tag{7.4}$$

Unter Berücksichtigung der Beziehungen (7.3) zwischen den Teilfunktionen f^0, f^1 und f^- und den Funktionen der Shannon- und Davio-Dekomposition ergeben sich für die Funktionen f_1, f_2, f_3 und f_4 die in der Tabelle 7.1 zusammengefassten Zusammenhänge.

Die Tabelle gibt weiterhin einen Überblick, welche Dekompositionsarten für die bekannten Datenstrukturen (siehe Kapitel 1) verwendet werden. Als ein Vorteil der Funktionen der Drei- bzw. Vier-Tupel-Dekomposition gegenüber der Funktionen aller anderen Dekompositionen ist ihre Disjunktheit zu nennen.

Die Algorithmen der Operationen mit nach (7.1) und (7.2) zerlegten Booleschen Funktionen sind ein weiteres Ergebnis der Arbeit. Die Korrektheit der Zerlegungsprinzipien und der erstellten Operationsalgorithmen wurde im Kapitels 4 bewiesen.

[1]siehe [Sas93, SaFu96, Sas97]
[2]siehe Abschitt 4.1.1
[3]siehe Abschitt 4.2.1

Datenstruktur	f_1	f_2	f_3	f_4	$\circ$
BDD	0	f_{S0}	f_{S1}	0	$\vee$
FDD	f_{S0}	0	f_D	0	$\oplus$
FDD	f_{S1}	f_D	0	0	$\oplus$
AND-OR-TDD	f^-	f_{S0}	f_{S1}	0	$\vee$
ETDD	f_D	f_{S0}	f_{S1}	0	$\oplus^1$
3-Tupel	f^-	f^0	f^1	0	$\vee^2$
4-Tupel	f^-	f^0	f^1	$f^=$	$\vee^3$

Tabelle 7.1: Zusammenhänge zwischen Funktionen verschiedener Dekompositionsarten

Im Ergebnis der Untersuchungen mit zerlegten Funktionen ist der Vorteil der Dekomposition von Funktionen mit einer hohen Anzahl von Konjunktionen zu nennen, der im geringeren Speicherplatzbedarf der in drei Teilfunktionen zerlegten Funktionen besteht. Die Analyse der Komplexität der Tupeloperationen zeigt, dass die Tupeloperationen die gleiche Komplexitätsordnung aufweisen wie die Operationen mit nicht zerlegten Funktionen. Der detaillierte Vergleich der Berechnungszeiten für die Operationen Konjunktionen und Negation erbrachte den Nachweis, dass eine Verringerung des Zeitbedarfs als Folge der Zerlegung einer Funktion in drei Teilfunktionen zu erwarten ist. Die in der Arbeit durchgeführten Abschätzungen des Speicherplatzbedarfs sowie der Bearbeitungszeiten in Abhängigkeit zur Art der Operation und der Tiefe der Zerlegung beruhten auf praktischen Untersuchungen auf der Basis von Zufallsfunktionen. Die Untersuchungen für die Operation Negation mit in vier Teilfunktionen zerlegten Funktionen belegen die Voraussage, dass die Negation die konstante Komplexitätsordnung besitzt. Der Rechenzeitbedarf war erwartungsgemäß für zerlegte Funktionen praktisch gleich Null.

In der Arbeit konnte weiterhin nachgewiesen werden, dass die Tupeloperationen für eine teilweise parallele Bearbeitung gut geeignet sind (siehe Kapitel 5). Die in der Implementation vorgenommene Abgrenzung der unabhängigen Teilaufgaben kann als vorteilhaft eingeschätzt werden. Die gewählte Größe der Teilaufgaben war einerseits problembedingt, gleichzeitig konnte dadurch der verteilungsbedingte Aufwand im Vergleich zum Rechenaufwand gering gehalten werden. Grundsätzlich war beim Überschreiten einer kritischen Mindestgröße der Aufgabe ein günstigeres Verarbeitungsverhalten nach der Verteilung nachzuweisen. Den Ausgangspunkt für die konkrete Implementation bildete ein Prototyp, der wesentliche Elemente der Lösung enthält. Es kann unterstellt werden, dass der allgemeingültige Lösungsansatz des Prototyps als Grundlage für die Realisierung ähnlich gelagerter Problemstellungen geeignet ist.

Die Erkentnisse der Untersuchungen wurden für die Suche der Gruppierungsmengen der OR-Bi-Decomposition verwendet. Bereits der Einzatz der Tupelkonjunktion für die Überprüfung des Kriteriums der OR-Bi-Decomposition (siehe Formel (6.6)), die eine große Anzahl konjunktiver Verknüpfungen erfordert, erwies sich als sehr vorteilhaft. Ein weitere spürbare Verbesserung der Suchzeiten war durch parallele Bearbeitung möglich.

Literaturverzeichnis

[Ake59] S. B. Akers: On a Theory of Boolean functions, Proc. Ind. Appl. Math., 7, 1959, pp. 487-498

[Ake78] S. B. Akers: Binary Dicision Diagrams, IEEE Trans. Comp., vol. C-27, June 1978, pp. 509-516

[ATPG] Automatic Test Pattern Generation for Combinational Circuits (ATPG), Version 1.0, 4/29/86, Author: Ruey-Sing Wei and Tony Ma, mcnc.mcnc.org

[BDS91] D. Bochmann, F. Dresig, B. Steinbach: A new decomposition method for multilevel circuit design, Proc. of Euro-DAC, 1991, pp. 374-377

[BDT93] B. Becker, R. Drechsler, M. Theobald: On the Implementation of a Package for Efficient Representation and Manipulation of Functional Decision Diagrams, IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design, Hamburg, 1993

[BDT95] B. Becker, R. Drechsler, M. Theobald: OKFDDs versus OBDDs and OFDDs, in ICALP, LNCS 944, 1995, pp. 475-486

[BGMS94] J. Bern, J. Gergov, C. Meinel, A. Slobodova: Boolean Manipulation with Free BDDs - First Experimental Results, Proceedings of European Conference on Design Automation, 1994

[Boch75] D. Bochmann: Einführung in die strukturelle Automatentheorie, Verlag Technik, Berlin, Hanser Verlag, München, 1975

[BoPo81] D. Bochmann, Ch. Posthoff: Binäre dynamische Systeme, Akademie-Verlag, Berlin, 1981

[BoSt91] D. Bochmann, B. Steinbach: Logikentwurf mit XBOOLE, Verlag Technik, Berlin, 1991

[BRB90] K. S. Brace, R. L. Rudell, R. E. Bryant: Efficient Implementation of a BDD Package, 27th ACM/ IEEE Design Automation Conference, 1990, pp. 40-45

[Bry86] R. E. Bryant: Graph-Based Algorithm for Boolean Function Manipulation, IEEE Trans. Comp., vol C-35, Aug. 1886, pp. 677-691

[Bry92] R. E. Bryant: Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams, Computing Surveys, Vol. 24, No.3, September 1992, pp. 293-318

[CoSt93] D. E. Comer, D. L. Stevens: Internetworking with TCP/IP, Vol.III, Client-Server Programming and Application, Englewood Cliffs, Prentice Hall, 1993

[deMi94] G. de Micheli: Synthesis and Optimization of Digital Circuits, McGraw-Hill,1994

[DKSW92] F. Dresig, N. Kümmling, B. Steinbach, J. Wazel: Programmieren mit XBOOLE, Wissenschaftliche Schriftenreihe Technische Universität Chemnitz, Chemnitz, 1992

[DrBe95] R. Drechsler, B. Becker: Dynamic minimization of OKFDDs, in Int'l Conf. on Comp. Design, 1995, pp. 602-607

[DrBe98] R. Drechsler, B. Becker: Graphenbasierte Funktionsdarstellungen, B. G. Teubner Stuttgart, 1998

[Dres92] F. Dresig: Gruppierung - Theorie und Anwendung in der Logiksynthese, Dissertation zur Erlangung des akademischen Grades Doktor-Ingenieur, Chemnitz, 1992

[DSTBP94] R. Drechsler, A. Sarabi, M. Theobald, B. Becker, M. A. Perkowski: Efficient representation and manipulation of switching functions based on Ordered Kronecker Functional Decision Diagrams, in Design Automation Conf., 1994, pp.415-419

[GeMe96] J. Gergov, C. Meinel: Mod2-OBDDs: A Data Structure that generalizes EXOR-sum-of-products and Ordered Binary Decision Diagrams, in Formal Methods in System Design 8, Kluwer, 1996, pp.273-282

[HaSo96] G. Hachtel, F. Somenzi: Logic Synthesis and Verification Algorithms, Kluwer Academic Publishers, 1996

[Hess94] K. Hesse: Bearbeitung hochdimensionaler Boolescher Probleme auf Transputernetzen, Proc. Workshop Boolean Problems, Freiberg, 1994, pp. 9-15

[Hess98] K. Hesse: On the Numeric Complexity of Some Basic Algorithms for Ternary Vectors, Proc. Workshop Boolean Problems, Freiberg, 1998, pp. 31-37

[Hess99] K. Hesse: Ein Beitrag zur Lösung hochdimensionaler Boolescher Probleme mit parallelen Algorithmen, Dissertation zur Erlangung des akademischen Grades Doktor-Ingenieur, Chemnitz, 1999

[KeLa98] G. Kempe, Ch. Lang: Efficient Representation of Boolean Functions by Three- and Four-Function Decomposition, Proc. Workshop Boolean Problems, Freiberg, 1998, pp. 39-46

[Kem96] G. Kempe: Eignung eines Phasenlisten-Tupels als Datenmodell für eine Boolesche Funktion, Proc. Workshop Boolean Problems, Freiberg, 1996, pp. 39-46

[KeRo93] U. Kebschull, W. Rosenstiel: Efficient Graph-Based Computation and Manipulation of Functional Decision Diagrams, Proc. EDAC, pp.278-282, Paris, 1993

[KeSt94] G. Kempe, B. Steinbach: Vergleich der Darstellungen einer Booleschen Funktion als TVL und ROBDD, Tagungsunterlagen des Workshops "Boolesche Probleme", Freiberg, 1994, S. 25-32

[KSR92] U. Kebschull, E. Schubert, W. Rosenstiel: Multilevel Logic Synthesis Based on Functional Decision Diagrams, Proc. EDAC, 1992, pp. 43-47

[LaSch94] H. Langendörfer, B. Schnor: Verteilte Systeme, Carl Hanser Verlag München Wien, 1994

[Le89] Trung Quoc Le: Testbarkeit kombinatorischer Schaltungen - Theorie und Entwurf, Dissertation zur Erlangung des akademischen Grades Doktor-Ingenieur, Karl-Marx-Stadt, 1989

[LDB00] P. Lindgren, R. Drechsler, B. Becker: Minimization of ordered pseudo Kronecker decision diagrams, Proceedings International Conference on Computer Design, 2000, pp. 504-510

[MaSa01] M. Matsuura, T. Sasao: Representation of Incompletely Specified Switching Functions Using Pseudo-Kronecker Decision Diagrams, Proceedings of the 5th International Workshop on Applications of the Reed-Muller Expansion in Circuit Design, August 10-11, 2001, pp. 27-33

[MeSa99] C. Meinel, H. Sack: Algorithmic Considerations of $\oplus$-OBDD Reordering, in Proc. of the 4th International Workshop on Applications of the Reed-Muller Expansion in Circuit Design (Victoria, BC, Canada), 1999, pp. 197-184

[MeSa01.1] C. Meinel, H. Sack: Heuristics for $\oplus$-OBDD Minimization, the 10th International Workshop on Logic and Synthesis, Granlibakken, Canada, June 12-15, 2001, pp. 304-308

[MeSa01.2] C. Meinel, H. Sack: Improving XOR-Node Placement for $\oplus$-OBDD, Proc. of the 5th International Workshop on Applications of the Reed-Muller Expansion in Circuit Design, Aug. 10-11, 2001, pp. 51-56

[MeTh98] C. Meinel, T. Theobald: Algorithms and Data Structures in VLSI Design: OBDD - Foundations and Applications, Springer, Heidelberg, 1998

[MoSc99] P. Molitor, Ch. Scholl: Datenstrukturen und effiziente Algorithmen für die Logiksynthese kombinatorischer Schaltungen, B. G. Teubner, Stuttgart, Leipzig, 1999

[MSP01] A. Mishchenko, B. Steinbach, M. Perkowski: An Algorithm for Bi-Decomposition of Logic Functions, Proceedings of the 38th Design Automation Conference 2001, June 18-22, 2001, Las Vegas (Nevada) USA, pp. 103-108

[Moo65] G. E. Moore: Cramming more components onto integrated circuits, Electronics, Volume 38, Number 8, April 19, 1965

[Mul54] D. E. Muller: Application of Boolean Algebra to Switching Circuit Design and Error Detection, in IRE Trans. on Electronic Computing EC-3, 1954, pp. 6-12

[PoBo86] Ch. Posthoff, D. Bochmann, K. Haubold: Diskrete Mathematik, BSB B. G. Teubner Verlagsgesellschaft, Leipzig, 1986

[PoSt79.1] Ch. Posthoff, B. Steinbach: Binäre Gleichungen - Algorithmen und Programme. Wissenschaftliche Schriftenreihe der Technischen Hochschule, Karl-Marx-Stadt, Heft 1, 1979

[PoSt79.8] Ch. Posthoff, B. Steinbach: Binäre dynamische Systeme - Algorithmen und Programme. Wissenschaftliche Schriftenreihe der Technischen Hochschule, Karl-Marx-Stadt, Heft 8, 1979

[Ree54] L. S. Reed: A Class of Multiple Error-Correcting Codes and their Decoding Scheme, in IRE Trans. on Information Theory 4, 1954, pp. 38-42

[RoBa94] R. Rohde, R. Barthel: Zur Effizienz der Behandlung Boolescher Probleme mit Ternärbäumen, Workshop Boolesche Probleme, Freiberg, 1994

[RuRI96] St. Rudolf, K. Richter, K. Irmscher: Ein Traderdienst zur Auswahl von Servern und Mediatoren in verteilten Systemen; in: C. Cap (Hrsg.), Workstations und ihre Anwendungen, SIWORK'96, Proceedings der Fachtagung SIWORK'96, Universität Zürich, 14./15.5.96, Hochschulverlag AG an der ETH Zürich, Zürich, 1996, S. 349-360

[RuRI97] St. Rudolf, K. Richter, K. Irmscher: Temporäre Einbindung mobiler Clienten und Optimierung der Diensteauswahl in einem verteilten System; in: M. Zitterbart (Hrsg.), Kommunikation in verteilten Systemen, KiVS'97, GI/ITG-Fachtagung, Braunschweig, 19.-21.Febr.1997, Springer Verlag, Berlin 1997, S. 344-358

[SaBu97] T. Sasao, J. Butler: On bi-decomposition of logic functions, Proc. of IWLS, 1997

[SaFu96] T. Sasao, M. Fujita: Representations of Discrete Functions, Kluwer Academic Publishers, Boston-London-Dordrecht, 1996

[Sas93] T. Sasao: Ternary Decision Diagrams and Their Application, Proceedings of International Workshop on Logic Synthesis, Lake Tahoe, May 1993, pages 6c, 1-11

[Sas97] T. Sasao: Ternary Decision Diagrams and Their Application, Proc. 27th Int. Symp. on Multiple-Valued Logic, May 1997, 241-250

[SSS93] B. Steinbach, F. Schumann, M. Stöckert: Functional Decomposition of Speed Optimized Circuits. Power and Timing Modelling for Performance of Integrated Circuits, IT Press Verlag, Bruchsal, 1993, pages 65-77

[StDo99] B. Steinbach, Ch. Dorotska: Fast Boolean Calculations Using Ordered Lists of Ternary Vectors, in: Proceedings of the Third International Conference on Computer - Aided Design of Discrete Devices (CAD DD'99), Volume 1, Minsk, Belarus, pp. 20-27

[StDo00] B. Steinbach, Ch. Dorotska: Orthogonal Blockbuilding Using Ordered Lists of Ternary Vectors, Proceedings of the 4th International Workshops on Boolean Problems, 21. - 22. September 2000, Freiberg University of Mining and Technology, Freiberg, 2000, pp. 125-134

[StDo01] B. Steinbach, Ch. Dorotska: Calculation of Set Operations Using Ordered Lists of Ternary Vectors, in: Proceedings of the Fourth International Conference on Computer - Aided Design of Discrete Devices (CAD DD 2001), Volume 1, 2001, Minsk, Belarus, pp. 61-68

[StDob00] B. Steinbach, T. Dobrev: A Concurent and Distributed Model for Complex Boolean Calculation, Proceedings of the 4th International Workshops on Boolean Problems, 21. - 22. September 2000, Freiberg University of Mining and Technology, Freiberg, 2000, pp. 183-189

[SDD01] B. Steinbach, Ch. Dorotska, T. Dobrev: Distributed Software for Complex Boolean Tasks in Circuit Design, in: The Experience of Designing and Application of CAD Systems in Microelectronics - Proceedings of the VI-th International Conference CADSM 2001, Lviv Slavsko, Ukraine, 2001, pp. 187-190

[Ste84] B. Steinbach: Theorie, Algorithmen und Programme für den rechnergestützten logischen Entwurf digitaler Systeme, Dissertation B, Karl-Marx-Stadt, Techn. Hochschule, 1984

[Ste95] B. Steinbach: Lists of Ternary Vectors - Concepts - Properties - Comparison with BDDs, Dagstuhl-Seminar 9507 "Computer Aided Design and Test", Internationales Begegnungs- und Forschungszentrum für Informatik, Schloß Dagstuhl, gehalten am 13.02.1995

[StSa98] R. S. Stankovic, T. Sasao: Decision diagrams for representation of discrete functions: uniform interpretation and classification, Proc. ASP-DAC'98, Yokohama, Japan, February 13-17, 1998

[StSt94] B. Steinbach, M. Stöckert: Design of Fully Testable Circuits by Functional Decomposition and Implicit Test Pattern Generation, Proc. of VLSI Test 1994, pp. 22-27

[StWe95] B. Steinbach, A. Wereszczynski: Synthesis of Multi-Level Circuits Using EXOR-Gates, Proc. of IFIP WG 10.5 - Workshop on Applications of the Applications of the Reed-Muller Expansion in Circuits Design, Japan, 1995, pp. 161-168

[Tan96] A. Tanenbaum: Computer Networks, 3rd Edition, Upper Saddle River, Prentice Hall, 1996

[Web98] M. Weber: Verteilte Systeme, Spektrum Akademischer Verlag GmbH, Heidelberg Berlin, 1998

[Weg87] I. Wegener: The Complexity of Boolean Functions. John Wiley & Sons Ltd., and B. G. Teubner, Stuttgart (Wiley-Teubner Series in Computer Science), 1987

[Yas95] K. Yasuoka: Ternary Decision Diagrams to represent Ringsum-of-Products Forms, IFIP WG. 10.5 Workshop on Applications of the Reed-Muller Expansions in Circuit Design, Aug. 1995

Anhang A

Funktionale Komposition - Algorithmen

A.1 Aufbau der Datenstruktur für die funktionale Komposition

A.1.1 Vorbemerkungen

Für jede Boolesche Funktion einer Benchmark-Schaltung LGSynth91 wird unabhängig von den anderen Funktionen eine Liste aus Knoten erstellt, die entweder Ternärvektorlisten lokaler Phasenlisten oder unabhängige Variablen sind. Die Daten werden dabei den .phl-Dateien entnommen.

Das Anlegen der Datenstruktur "Knotenliste" erfolgt für jede Boolesche Funktion (Ausgangsvariable) im Unterprogramm `ausgang()`. Durch den Aufruf des rekursiven Unterprogramms `next()` von `ausgang()` wird der Listenaufbau fortgesetzt. Neben einem Kennzeichen zur Unterscheidung von PHL und Variablen enthält die Datenstruktur "Knotenliste" die Identifikationsnummer der PHL bzw. den Index der Variable. Das Unterprogramm `next()` wird weiterhin für den Aufbau der Liste der Vorknoten jedes Knotens genutzt.

Beim Listenaufbau ist zu sichern, dass jeder Knoten nur einmal abgespeichert wird. Über den Rückgabewert des Unterprogramms `pruefen()` ist die entsprechende Aussage möglich.

Die Liste der Nachknoten wird anschließend für alle Knoten einer Funktion erstellt und in die fertige Liste eingetragen. In einem weiteren Schritt wird die Liste um die Anzahl der Vor- und Nachknoten ergänzt.

Um zwischen Ausgangsvariablen sowie abhängigen und unabhängigen Eingangsvariablen unterscheiden zu können, werden alle Variablen einer Benchmark-Schaltung in zwei XBOOLE-Objekten verwaltet. Das eine Objekt enthält für jede Variable den Index und den Variablenname. Das andere Objekt enthält für die Variable entsprechend ihres Index die Kodierung zur Unterscheidung der Variablenart (Ausgangsvariable, abhängige bwz. unabhängige Eingangsvariable).

A.1.2 Unterprogramme

Unterprogramm `ausgang()`

Parameter:

dateiname - Name der .phl-Datei,
oa1 - XBOOLE-Objekt Variablenliste,
oa2 - XBOOLE-Objekt Kodierung der Variablen.

Algorithmus:

```
solange in oa2 noch Ausgangsvariablen vorhanden
      Ausgangsvariable in oa2 suchen und ihr Index index ermitteln;
      den ersten Knoten node der Struktur erstellen;
      prenode:=NULL;
      nextnode:=NULL;
      Aufruf des rekursiven Unterprogramms next(index,node,prenode,nextnode)
      Nachknoten eintragen;
      Anzahl Vorknoten und Nachknoten eintragen;
      PHLs aus der .phl-Datei einlesen;
```

Unterprogramm `next()`

Parameter:

dateiname - Name der .phl-Datei,
index - Index der Variable,
oa1 - XBOOLE-Objekt Variablenliste,
oa2 - XBOOLE-Objekt Kodierung der Variablen,
node - der aktuelle Knoten,
prenode - der Vorknoten,
nextnode - der nächste Knoten in der Struktur,
begin - der erste Knoten.

Algorithmus:

```
aus dem gegebenen Index den Variablenname ermitteln;
phl-Datei öffnen;
in der Datei die PHL mit der Ausgangsvariable suchen, die PHL-Nummer speichern;
wenn noch kein Knoten mit PHL-Nummer vorhanden (Unterprogramm pruefen())
      v_o_l:=0;
      list_nr:=PHL-Nummer;
      prenode in die Liste der Vorknoten schreiben;
      für jede Eingangsvariable der PHL
           Name der Variable aus phl-Datei lesen;
```

```
          Index und Variablenkodierung ermitteln;
          nextnode erstellen mit list_nr:=0;
          nextnode als nächster Knoten des Knotens node speichern;
          node:=nextnode;
          wenn variable=primärer Eingang
               wenn Knoten mit der Variable noch nicht existiert
                    v_o_l:=0;
                    list_nr:=Index der Variable;
                    prenode in die Liste der Vorknoten schreiben;
          sonst
              Aufruf next(dateiname,index,ao1,ao2,node,prenode,nextnode,begin);
phl-Datei schliessen
```

Unterprogramm `pruefen()`

Parameter:

nr - Nummer der PHL bzw. Index der Eingangsvariable,
param - ist gleich 1 bei PHL und 0 bei Variablen,
prenode - der Vorknoten,
begin - der erste Knoten,
end - der letzte in der Struktur abgespeicherte Knoten.

Algorithmus:

```
wenn kein prenode existiert return 1;
sonst
     gefunden:=0;
     Knoten:=begin;
     solange Knoten nicht gleich end
          wenn bei dem Knoten list_nr=nr und v_o_l=param
               in der Liste der Vorknoten prenode suchen;
               wenn prenode nicht gefunden
                    prenode in die Liste der Vorknoten schreiben;
               gefunden:=1;
          Knoten:= nächster Knoten;
     wenn gefunden return 0;
     sonst return 1;
```

A.2 Globale Phasenliste

Algorithmus zum Entfernen von Variablen bei der Durchschnittsbildung

```
Erstellen der Objekte oa9 (alle Variablen der Funktion) und oa10 (Anzahl
der Verwendungen dieser Variablen in TVL der gesamten Funktion);
```

```
Erstellen der Objekte oa11 (Variablen der zu verknüpfenden TVL) und
oa12 (Anzahl der Verwendungen dieser Variablen in den zu verknüpfenden TVL);

wenn die Ausgangsvariable nur noch einmal in den zu verknüpfenden
TVL verwendet wird,
   wenn die Ausgangsvariable nur noch einmal in der Funktion verwendet wird,
      Maximum nach diese Variable bilden;
   sonst
      Anzahl der Verwendungen der Variable in oa10 auf 0 setzen
      (d.h. die Variable wird in den anderen TVL auftreten und
      kann deswegen nicht gestrichen werden);
sonst
   die Anzahl der Verwendungen der Variable in oa10 und
   oa12 wird um 1 reduziert.
```

Anhang B

Tupel-Dekomposition: Beweise

B.1 Satz 4.6

Satz 4.6: Die nach (4.50), (4.51) und (4.52) berechneten Teilfunktionen f_3^-, f_3^0 und f_3^1 (4.27) der Funktion $f_3 = \bar{f}_1$ sind zueinander paarweise disjunkt:

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \text{für} \quad \alpha, \beta \in \{-, 0, 1\} \quad \text{und} \quad \alpha \neq \beta$$

Beweis:

1.
$$f_3^- \cdot f_3^0 = \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1 \cdot f_1^1 = 0 \tag{B.1}$$

2.
$$f_3^- \cdot f_3^1 = \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1 \cdot f_1^0 = 0 \tag{B.2}$$

3.
$$f_3^1 \cdot f_3^0 = f_1^0 \cdot f_1^1 = 0 \tag{B.3}$$

Die Aussage (B.3) folgt aus der Orthogonalität der Funktionen f_1^0 und f_1^1. □

B.2 Satz 4.8

Satz 4.8: Die nach (4.71), (4.72) und (4.73) berechneten Teilfunktionen f_3^-, f_3^0 und f_3^1 (4.27) der Funktion $f_3 = f_1 \vee f_2$ sind zueinander paarweise disjunkt:

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \text{für} \quad \alpha, \beta \in \{-, 0, 1\} \quad \text{und} \quad \alpha \neq \beta$$

Beweis:

1.

$$\begin{aligned} f_3^- \cdot f_3^0 &= 0 && \text{(B.4)}\\ (f_1{}^- \vee f_2{}^- \vee f_1{}^0 f_2{}^1 \vee f_1{}^1 f_2{}^0) & && \\ \cdot(f_1^0 f_2^0 \vee f_1^0 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee f_2^0 \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1) &= 0 && \text{(B.5)}\\ 0 &= 0 && \text{(B.6)} \end{aligned}$$

2.

$$\begin{aligned} f_3^- \cdot f_3^1 &= 0 && \text{(B.7)}\\ (f_1{}^- \vee f_2{}^- \vee f_1{}^0 f_2{}^1 \vee f_1{}^1 f_2{}^0) & && \\ \cdot(f_1^1 f_2^1 \vee f_1^1 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee f_2^1 \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1) &= 0 && \text{(B.8)}\\ 0 &= 0 && \text{(B.9)} \end{aligned}$$

3.

$$\begin{aligned} f_3^0 \cdot f_3^1 &= 0 && \text{(B.10)}\\ (f_1^0 f_2^0 \vee f_1^0 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee f_2^0 \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1) \cdot & && \\ (f_1^1 f_2^1 \vee f_1^1 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee f_2^1 \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1) &= 0 && \text{(B.11)}\\ 0 &= 0 && \text{(B.12)} \end{aligned}$$

Nach dem Einsetzen von (4.71), (4.72) und (4.73) für die Funktionen f_3^-, f_3^0 und f_3^1 in (B.4), (B.7) und (B.10) entstehen (B.5), (B.8) und (B.11). Durch elementare Umformungen der Gleichungen (B.5), (B.8) und (B.11) unter Berücksichtigung der Orthonalität der Funktionen f_1^-, f_1^0 und f_1^1 sowie der Funktionen f_2^-, f_2^0 und f_2^1 werden (B.6), (B.9) und (B.12) erhalten. □

B.3 Satz 4.10

Satz 4.10: Die nach (4.101), (4.102) und (4.103) berechneten Teilfunktionen f_3^-, f_3^0 und f_3^1 (4.27) der Funktion $f_3 = f_1 \cdot f_2$ sind zueinander paarweise disjunkt:

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \text{für} \quad \alpha, \beta \in \{-, 0, 1\} \quad \text{und} \quad \alpha \neq \beta$$

Beweis:

Die Gleichungen (B.14), (B.17) und (B.20) entstehen nach dem Einsetzen von (4.101), (4.102) und (4.103) für die Funktionen f_3^-, f_3^0 und f_3^1 in (B.13), (B.16) und (B.19). Die elementaren Umformungen der Gleichungen (B.14), (B.17) und (B.20) unter Berücksichtigung der Orthogonalität der Funktionen f_1^-, f_1^0 und f_1^1 sowie der Funktionen f_2^-, f_2^0 und f_2^1 führen zu (B.15), (B.18) und (B.21).

1.

$$\begin{aligned} f_3^- \cdot f_3^0 &= 0 && \text{(B.13)}\\ f_1{}^- f_2{}^- \cdot (f_1^- f_2^0 \vee f_1^0 f_2^- \vee f_1^0 f_2^0) &= 0 && \text{(B.14)}\\ 0 &= 0 && \text{(B.15)} \end{aligned}$$

2.

$$\begin{aligned} f_3^- \cdot f_3^1 &= 0 && \text{(B.16)}\\ f_1{}^- f_2{}^- \cdot (f_1^- f_2^1 \vee f_1^1 f_2^- \vee f_1^1 f_2^1) &= 0 && \text{(B.17)}\\ 0 &= 0 && \text{(B.18)} \end{aligned}$$

3.

$$\begin{aligned} f_3^0 \cdot f_3^1 &= 0 && \text{(B.19)}\\ (f_1^- f_2^0 \vee f_1^0 f_2^- \vee f_1^0 f_2^0) \cdot (f_1^- f_2^1 \vee f_1^1 f_2^- \vee f_1^1 f_2^1) &= 0 && \text{(B.20)}\\ 0 &= 0 && \text{(B.21)} \end{aligned}$$

□

B.4 Satz 4.12

Satz 4.12: Die nach (4.115), (4.116) und (4.117) berechneten Teilfunktionen f_3^-, f_3^0 und f_3^1 (4.27) der Funktion $f_3 = f_1 \oplus f_2$ sind zueinander paarweise disjunkt:

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \text{für} \quad \alpha, \beta \in \{-, 0, 1\} \quad \text{und} \quad \alpha \neq \beta$$

Beweis:

Nach dem Einsetzen von (4.115), (4.116) und (4.117) für die Funktionen f_3^-, f_3^0 und f_3^1 in (B.22), (B.25) und (B.28) entstehen die Gleichungen (B.23), (B.26) und (B.29). Diese Gleichungen führen durch elementare Umformungen unter Berücksichtigung der Tatsache, dass die Funktionen f_1^-, f_1^0 und f_1^1 und die Funktionen f_2^-, f_2^0 und f_2^1 zueinander disjunkt sind, zu (B.24), (B.27) und (B.30).

1.

$$\begin{aligned} f_3^- \cdot f_3^0 &= 0 && \text{(B.22)}\\ (f_1{}^0 f_2{}^1 \vee f_1^1 f_2^0 \vee f_1^- \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee f_2^- \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1) & \\ \cdot (f_1^1 f_2^- \vee f_1^- f_2^1 \vee f_1^0 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee f_2^0 \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1) &= 0 && \text{(B.23)}\\ 0 &= 0 && \text{(B.24)} \end{aligned}$$

2.

$$\begin{aligned} f_3^- \cdot f_3^1 &= 0 && \text{(B.25)}\\ (f_1{}^0 f_2{}^1 \vee f_1^1 f_2^0 \vee f_1^- \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee f_2^- \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1) & \\ \cdot (f_1^0 f_2^- \vee f_1^- f_2^0 \vee f_1^1 \bar{f}_2^- \bar{f}_2^0 \bar{f}_2^1 \vee f_2^1 \bar{f}_1^- \bar{f}_1^0 \bar{f}_1^1) &= 0 && \text{(B.26)}\\ 0 &= 0 && \text{(B.27)} \end{aligned}$$

3.

$$
\begin{aligned}
f_3^0 \cdot f_3^1 &= 0 && \text{(B.28)}\\
(f_1^1 f_2^- \vee f_1^- f_2^1 \vee f_1^0 \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee f_2^0 \bar{f_1^-} \bar{f_1^0} \bar{f_1^1}) & \\
\cdot(f_1^0 f_2^- \vee f_1^- f_2^0 \vee f_1^1 \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee f_2^1 \bar{f_1^-} \bar{f_1^0} \bar{f_1^1}) &= 0 && \text{(B.29)}\\
0 &= 0 && \text{(B.30)}
\end{aligned}
$$

□

B.5 Satz 4.14

Satz 4.14: Die nach (4.133), (4.134) und (4.135) berechneten Teilfunktionen f_3^-, f_3^0 und f_3^1 (4.27) der Funktion $f_3 = f_1 \odot f_2$ sind zueinander paarweise disjunkt:

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \text{für} \quad \alpha, \beta \in \{-, 0, 1\} \quad \text{und} \quad \alpha \neq \beta$$

Beweis:

Die Gleichungen (B.32), (B.35) und (B.38) entstehen durch das Einsetzen von (4.133), (4.134) und (4.135) für die Funktionen f_3^-, f_3^0 und f_3^1 in (B.31), (B.34) und (B.37). Da die Funktionen f_1^-, f_1^0 und f_1^1 und die Funktionen f_2^-, f_2^0 und f_2^1 zueinander disjunkt sind, können die Gleichungen (B.32), (B.35) und (B.38) durch elementare Umformungen in (B.33), (B.36) und (B.39) überführt werden.

1.

$$
\begin{aligned}
f_3^- \cdot f_3^0 &= 0 && \text{(B.31)}\\
(f_1{}^- f_2^- \vee f_1{}^0 f_2{}^0 \vee f_1{}^1 f_2{}^1 \vee \bar{f_1^-} \bar{f_1^0} \bar{f_1^1} \bar{f_2^-} \bar{f_2^0} \bar{f_2^1}) & \\
\cdot(f_1{}^- f_2{}^0 \vee f_1{}^0 f_2^- \vee f_1{}^1 \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee f_2{}^1 \bar{f_1^-} \bar{f_1^0} \bar{f_1^1}) &= 0 && \text{(B.32)}\\
0 &= 0 && \text{(B.33)}
\end{aligned}
$$

2.

$$
\begin{aligned}
f_3^- \cdot f_3^1 &= 0 && \text{(B.34)}\\
(f_1{}^- f_2^- \vee f_1{}^0 f_2{}^0 \vee f_1{}^1 f_2{}^1 \vee \bar{f_1^-} \bar{f_1^0} \bar{f_1^1} \bar{f_2^-} \bar{f_2^0} \bar{f_2^1}) & \\
\cdot(f_1{}^- f_2{}^1 \vee f_1{}^1 f_2^- \vee f_1{}^0 \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee f_2{}^0 \bar{f_1^-} \bar{f_1^0} \bar{f_1^1}) &= 0 && \text{(B.35)}\\
0 &= 0 && \text{(B.36)}
\end{aligned}
$$

3.

$$
\begin{aligned}
f_3^0 \cdot f_3^1 &= 0 && \text{(B.37)}\\
(f_1{}^- f_2{}^0 \vee f_1{}^0 f_2^- \vee f_1{}^1 \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee f_2{}^1 \bar{f_1^-} \bar{f_1^0} \bar{f_1^1}) & \\
\cdot(f_1{}^- f_2{}^1 \vee f_1{}^1 f_2^- \vee f_1{}^0 \bar{f_2^-} \bar{f_2^0} \bar{f_2^1} \vee f_2{}^0 \bar{f_1^-} \bar{f_1^0} \bar{f_1^1}) &= 0 && \text{(B.38)}\\
0 &= 0 && \text{(B.39)}
\end{aligned}
$$

□

B.6 Satz 4.21

Satz 4.21: Die nach (4.210), (4.211), (4.212) und (4.213) berechneten Teilfunktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ (4.162) der Funktion $f_3 = \bar{f}_1$ sind zueinander paarweise disjunkt:

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \text{für} \quad \alpha, \beta \in \{-, 0, 1, =\} \quad \text{und} \quad \alpha \neq \beta$$

Beweis:

Nach dem Einsetzen der Ausdrücke (4.210)-(4.213) für die Funktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ in die Formeln (B.40), (B.42), (B.44), (B.46), (B.48) und (B.50) entstehen die Ausdrücke (B.41), (B.43), (B.45), (B.47), (B.49) und (B.51).

1.

$$f_3^- \cdot f_3^0 = 0 \tag{B.40}$$
$$f_1^= \cdot f_1^1 = 0 \tag{B.41}$$

2.

$$f_3^- \cdot f_3^1 = 0 \tag{B.42}$$
$$f_1^= \cdot f_1^0 = 0 \tag{B.43}$$

3.

$$f_3^- \cdot f_3^= = 0 \tag{B.44}$$
$$f_1^= \cdot f_1^- = 0 \tag{B.45}$$

4.

$$f_3^0 \cdot f_3^1 = 0 \tag{B.46}$$
$$f_1^1 \cdot f_1^0 = 0 \tag{B.47}$$

5.

$$f_3^0 \cdot f_3^= = 0 \tag{B.48}$$
$$f_1^1 \cdot f_1^- = 0 \tag{B.49}$$

6.

$$f_3^1 \cdot f_3^= = 0 \tag{B.50}$$
$$f_1^0 \cdot f_1^- = 0 \tag{B.51}$$

Die Richtigkeit der Aussagen (B.41), (B.43), (B.45), (B.47), (B.49) und (B.51) folgt aus der Orthogonalität der Funktionen f_1^-, f_1^0, f_1^1 und $f_1^=$ (siehe Sätze 4.4 und 4.16). □

B.7 Satz 4.23

Satz 4.23: Die nach (4.219), (4.220), (4.221) und (4.222) berechneten Teilfunktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ (4.162) der Funktion $f_3 = f_1 \vee f_2$ sind zueinander paarweise disjunkt:

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \text{für} \quad \alpha, \beta \in \{-, 0, 1, =\} \quad \text{und} \quad \alpha \neq \beta$$

Beweis:

Nach dem Einsetzen der Ausdrücke (4.219)-(4.222) für die Funktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ in die Formeln (B.52), (B.55), (B.58), (B.61), (B.64) und (B.67) entstehen die Ausdrücke (B.53), (B.56), (B.59), (B.62), (B.65) und (B.68).

1.

$$\begin{aligned} f_3^- \cdot f_3^0 &= 0 && \text{(B.52)}\\ (f_1{}^- \vee f_2{}^- \vee f_1{}^0 f_2{}^1 \vee f_1{}^1 f_2{}^0) & && \\ \cdot(f_1{}^0 f_2{}^0 \vee f_1{}^0 f_2{}^= \vee f_1{}^= f_2{}^0) &= 0 && \text{(B.53)}\\ 0 &= 0 && \text{(B.54)} \end{aligned}$$

2.

$$\begin{aligned} f_3^- \cdot f_3^1 &= 0 && \text{(B.55)}\\ (f_1{}^- \vee f_2{}^- \vee f_1{}^0 f_2{}^1 \vee f_1{}^1 f_2{}^0) & && \\ \cdot(f_1{}^1 f_2{}^1 \vee f_1{}^1 f_2{}^= \vee f_1{}^= f_2{}^1) &= 0 && \text{(B.56)}\\ 0 &= 0 && \text{(B.57)} \end{aligned}$$

3.

$$\begin{aligned} f_3^- \cdot f_3^= &= 0 && \text{(B.58)}\\ (f_1{}^- \vee f_2{}^- \vee f_1{}^0 f_2{}^1 \vee f_1{}^1 f_2{}^0) \cdot (f_1{}^= f_2{}^=) &= 0 && \text{(B.59)}\\ 0 &= 0 && \text{(B.60)} \end{aligned}$$

4.

$$\begin{aligned} f_3^0 \cdot f_3^1 &= 0 && \text{(B.61)}\\ (f_1{}^0 f_2{}^0 \vee f_1{}^0 f_2{}^= \vee f_1{}^= f_2{}^0) & && \\ \cdot(f_1{}^1 f_2{}^1 \vee f_1{}^1 f_2{}^= \vee f_1{}^= f_2{}^1) &= 0 && \text{(B.62)}\\ 0 &= 0 && \text{(B.63)} \end{aligned}$$

5.

$$\begin{aligned} f_3^0 \cdot f_3^= &= 0 && \text{(B.64)}\\ (f_1{}^0 f_2{}^0 \vee f_1{}^0 f_2{}^= \vee f_1{}^= f_2{}^0) \cdot (f_1{}^= f_2{}^=) &= 0 && \text{(B.65)}\\ 0 &= 0 && \text{(B.66)} \end{aligned}$$

6.

$$\begin{aligned} f_3^1 \cdot f_3^= &= 0 && \text{(B.67)}\\ (f_1{}^1 f_2{}^1 \vee f_1{}^1 f_2{}^= \vee f_1{}^= f_2{}^1) \cdot (f_1{}^= f_2{}^=) &= 0 && \text{(B.68)}\\ 0 &= 0 && \text{(B.69)} \end{aligned}$$

Die Richtigkeit der Aussagen (B.54), (B.57), (B.60), (B.63), (B.66) und (B.69) folgt aus der Orthogonalität der Funktionen f_1^-, f_1^0, f_1^1 und $f_1^=$ und der Funktionen f_2^-, f_2^0, f_2^1 und $f_2^=$ (siehe Sätze 4.4 und 4.16). □

B.8 Satz 4.25

Satz 4.25: Die nach (4.238), (4.239), (4.240) und (4.241) berechneten Teilfunktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ (4.162) der Funktion $f_3 = f_1 \cdot f_2$ sind zueinander paarweise disjunkt:

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \text{für} \quad \alpha, \beta \in \{-, 0, 1, =\} \quad \text{und} \quad \alpha \neq \beta$$

Beweis:

Nach dem Einsetzen der Ausdrücke (4.238)-(4.241) für die Funktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ in die Formeln (B.70), (B.73), (B.76), (B.79), (B.82) und (B.85) entstehen die Ausdrücke (B.71), (B.74), (B.77), (B.80), (B.83) und (B.86). Die Richtigkeit der Aussagen (B.72), (B.75), (B.78), (B.81), (B.84) und (B.87) folgt aus der Orthogonalität der Funktionen f_1^-, f_1^0, f_1^1 und $f_1^=$ und der Funktionen f_2^-, f_2^0, f_2^1 und $f_2^=$ (siehe Sätze 4.4 und 4.16).

1.

$$\begin{aligned} f_3^- \cdot f_3^0 &= 0 && \text{(B.70)}\\ (f_1{}^- f_2{}^-) \cdot (f_1{}^- f_2{}^0 \vee f_1{}^0 f_2{}^- \vee f_1{}^0 f_2{}^0) &= 0 && \text{(B.71)}\\ 0 &= 0 && \text{(B.72)} \end{aligned}$$

2.

$$\begin{aligned} f_3^- \cdot f_3^1 &= 0 && \text{(B.73)}\\ (f_1{}^- f_2{}^-) \cdot (f_1{}^- f_2{}^1 \vee f_1{}^1 f_2{}^- \vee f_1{}^1 f_2{}^1) &= 0 && \text{(B.74)}\\ 0 &= 0 && \text{(B.75)} \end{aligned}$$

3.

$$\begin{aligned} f_3^- \cdot f_3^= &= 0 && \text{(B.76)}\\ (f_1{}^- f_2{}^-) \cdot (f_1{}^1 f_2{}^0 \vee f_1{}^0 f_2{}^1 \vee f_1{}^= \vee f_2{}^=) &= 0 && \text{(B.77)}\\ 0 &= 0 && \text{(B.78)} \end{aligned}$$

4.

$$\begin{aligned} f_3^0 \cdot f_3^1 &= 0 && \text{(B.79)}\\ (f_1{}^- f_2{}^0 \vee f_1{}^0 f_2{}^- \vee f_1{}^0 f_2{}^0) & && \\ \cdot (f_1{}^- f_2{}^1 \vee f_1{}^1 f_2{}^- \vee f_1{}^1 f_2{}^1) &= 0 && \text{(B.80)}\\ 0 &= 0 && \text{(B.81)} \end{aligned}$$

5.

$$\begin{aligned} f_3^0 \cdot f_3^= &= 0 && \text{(B.82)}\\ (f_1{}^- f_2{}^0 \vee f_1{}^0 f_2{}^- \vee f_1{}^0 f_2{}^0) & && \\ \cdot(f_1{}^1 f_2{}^0 \vee f_1{}^0 f_2{}^1 \vee f_1{}^= \vee f_2{}^=) &= 0 && \text{(B.83)}\\ 0 &= 0 && \text{(B.84)} \end{aligned}$$

6.

$$\begin{aligned} f_3^1 \cdot f_3^= &= 0 && \text{(B.85)}\\ (f_1{}^- f_2{}^1 \vee f_1{}^1 f_2{}^- \vee f_1{}^1 f_2{}^1) & && \\ \cdot(f_1{}^1 f_2{}^0 \vee f_1{}^0 f_2{}^1 \vee f_1{}^= \vee f_2{}^=) &= 0 && \text{(B.86)}\\ 0 &= 0 && \text{(B.87)} \end{aligned}$$

□

B.9 Satz 4.27

Satz 4.27: Die nach (4.254), (4.255), (4.256) und (4.257) berechneten Teilfunktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ (4.162) der Funktion $f_3 = f_1 \oplus f_2$ sind zueinander paarweise disjunkt:

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \text{für} \quad \alpha, \beta \in \{-, 0, 1, =\} \quad \text{und} \quad \alpha \neq \beta$$

Beweis:

Nach dem Einsetzen der Ausdrücke (4.254)-(4.257) für die Funktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ in die Formeln (B.88), (B.91), (B.94), (B.97), (B.100) und (B.103) entstehen die Ausdrücke (B.89), (B.92), (B.95), (B.98), (B.101) und (B.104). Die Richtigkeit der Aussagen (B.90), (B.93), (B.96), (B.99), (B.102) und (B.105) folgt aus der Orthogonalität der Funktionen f_1^-, f_1^0, f_1^1 und $f_1^=$ und der Funktionen f_2^-, f_2^0, f_2^1 und $f_2^=$ (siehe Sätze 4.4 und 4.16).

1.

$$\begin{aligned} f_3^- \cdot f_3^0 &= 0 && \text{(B.88)}\\ (f_1{}^0 f_2{}^1 \vee f_1{}^1 f_2{}^0 \vee f_1{}^- f_2{}^= \vee f_1{}^= f_2{}^-) & && \\ \cdot(f_1{}^1 f_2{}^- \vee f_1{}^- f_2{}^1 \vee f_1{}^0 f_2{}^= \vee f_1{}^= f_2{}^0) &= 0 && \text{(B.89)}\\ 0 &= 0 && \text{(B.90)} \end{aligned}$$

2.

$$\begin{aligned} f_3^- \cdot f_3^1 &= 0 && \text{(B.91)}\\ (f_1{}^0 f_2{}^1 \vee f_1{}^1 f_2{}^0 \vee f_1{}^- f_2{}^= \vee f_1{}^= f_2{}^-) & && \\ \cdot(f_1{}^0 f_2{}^- \vee f_1{}^- f_2{}^0 \vee f_1{}^1 f_2{}^= \vee f_1{}^= f_2{}^1) &= 0 && \text{(B.92)}\\ 0 &= 0 && \text{(B.93)} \end{aligned}$$

3.

$$
\begin{aligned}
f_3^- \cdot f_3^= &= 0 && \text{(B.94)}\\
(f_1{}^0 f_2{}^1 \vee f_1{}^1 f_2{}^0 \vee f_1{}^- f_2{}^= \vee f_1{}^= f_2{}^-) & && \\
\cdot(f_1{}^- f_2{}^- \vee f_1{}^0 f_2{}^0 \vee f_1{}^1 f_2{}^1 \vee f_1{}^= f_2{}^=) &= 0 && \text{(B.95)}\\
0 &= 0 && \text{(B.96)}
\end{aligned}
$$

4.

$$
\begin{aligned}
f_3^0 \cdot f_3^1 &= 0 && \text{(B.97)}\\
(f_1{}^1 f_2{}^- \vee f_1{}^- f_2{}^1 \vee f_1{}^0 f_2{}^= \vee f_1{}^= f_2{}^0) & && \\
\cdot(f_1{}^0 f_2{}^- \vee f_1{}^- f_2{}^0 \vee f_1{}^1 f_2{}^= \vee f_1{}^= f_2{}^1) &= 0 && \text{(B.98)}\\
0 &= 0 && \text{(B.99)}
\end{aligned}
$$

5.

$$
\begin{aligned}
f_3^0 \cdot f_3^= &= 0 && \text{(B.100)}\\
(f_1{}^1 f_2{}^- \vee f_1{}^- f_2{}^1 \vee f_1{}^0 f_2{}^= \vee f_1{}^= f_2{}^0) & && \\
\cdot(f_1{}^- f_2{}^- \vee f_1{}^0 f_2{}^0 \vee f_1{}^1 f_2{}^1 \vee f_1{}^= f_2{}^=) &= 0 && \text{(B.101)}\\
0 &= 0 && \text{(B.102)}
\end{aligned}
$$

6.

$$
\begin{aligned}
f_3^1 \cdot f_3^= &= 0 && \text{(B.103)}\\
(f_1{}^0 f_2{}^- \vee f_1{}^- f_2{}^0 \vee f_1{}^1 f_2{}^= \vee f_1{}^= f_2{}^1) & && \\
\cdot(f_1{}^- f_2{}^- \vee f_1{}^0 f_2{}^0 \vee f_1{}^1 f_2{}^1 \vee f_1{}^= f_2{}^=) &= 0 && \text{(B.104)}\\
0 &= 0 && \text{(B.105)}
\end{aligned}
$$

□

B.10 Satz 4.29

Satz 4.29: Die nach (4.271), (4.272), (4.273) und (4.274) berechneten Teilfunktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ (4.162) der Funktion $f_3 = f_1 \odot f_2$ sind zueinander paarweise disjunkt:

$$f_3^\alpha \cdot f_3^\beta = 0, \quad \text{für} \quad \alpha, \beta \in \{-, 0, 1, =\} \quad \text{und} \quad \alpha \neq \beta$$

Beweis:

Nach dem Einsetzen der Ausdrücke (4.271)-(4.274) für die Funktionen f_3^-, f_3^0, f_3^1 und $f_3^=$ in die Formeln (B.106), (B.109), (B.112), (B.115), (B.118) und (B.121) entstehen die Ausdrücke (B.107), (B.110), (B.113), (B.116), (B.119) und (B.122). Wegen der Orthogonalität der Funktionen f_1^-, f_1^0, f_1^1 und $f_1^=$ und der Funktionen f_2^-, f_2^0, f_2^1 und $f_2^=$ (siehe Sätze 4.4 und 4.16) werden nach dem Ausmultiplizieren der Disjunktionen in den Ausdrücken (B.107), (B.110), (B.113), (B.116), (B.119) und (B.122) die Ausdrücke (B.108), (B.111), (B.114), (B.117), (B.120) und (B.123) erhalten.

1.

$$
\begin{aligned}
f_3^- \cdot f_3^0 &= 0 && \text{(B.106)}\\
(f_1{}^- f_2{}^- \vee f_1{}^0 f_2{}^0 \vee f_1{}^1 f_2{}^1 \vee f_1{}^= f_1{}^=)& && \\
\cdot(f_1{}^- f_2{}^0 \vee f_1{}^0 f_2{}^- \vee f_1{}^1 f_2{}^= \vee f_1{}^= f_2{}^1) &= 0 && \text{(B.107)}\\
0 &= 0 && \text{(B.108)}
\end{aligned}
$$

2.

$$
\begin{aligned}
f_3^- \cdot f_3^1 &= 0 && \text{(B.109)}\\
(f_1{}^- f_2{}^- \vee f_1{}^0 f_2{}^0 \vee f_1{}^1 f_2{}^1 \vee f_1{}^= f_1{}^=)& && \\
\cdot(f_1{}^- f_2{}^1 \vee f_1{}^1 f_2{}^- \vee f_1{}^0 f_2{}^= \vee f_1{}^= f_2{}^0) &= 0 && \text{(B.110)}\\
0 &= 0 && \text{(B.111)}
\end{aligned}
$$

3.

$$
\begin{aligned}
f_3^- \cdot f_3^= &= 0 && \text{(B.112)}\\
(f_1{}^- f_2{}^- \vee f_1{}^0 f_2{}^0 \vee f_1{}^1 f_2{}^1 \vee f_1{}^= f_1{}^=)& && \\
\cdot(f_1{}^0 f_2{}^1 \vee f_1{}^1 f_2{}^0 \vee f_1{}^- f_2{}^= \vee f_1{}^= f_2{}^-) &= 0 && \text{(B.113)}\\
0 &= 0 && \text{(B.114)}
\end{aligned}
$$

4.

$$
\begin{aligned}
f_3^0 \cdot f_3^1 &= 0 && \text{(B.115)}\\
(f_1{}^- f_2{}^0 \vee f_1{}^0 f_2{}^- \vee f_1{}^1 f_2{}^= \vee f_1{}^= f_2{}^1)& && \\
\cdot(f_1{}^- f_2{}^1 \vee f_1{}^1 f_2{}^- \vee f_1{}^0 f_2{}^= \vee f_1{}^= f_2{}^0) &= 0 && \text{(B.116)}\\
0 &= 0 && \text{(B.117)}
\end{aligned}
$$

5.

$$
\begin{aligned}
f_3^0 \cdot f_3^= &= 0 && \text{(B.118)}\\
(f_1{}^- f_2{}^0 \vee f_1{}^0 f_2{}^- \vee f_1{}^1 f_2{}^= \vee f_1{}^= f_2{}^1)& && \\
\cdot(f_1{}^0 f_2{}^1 \vee f_1{}^1 f_2{}^0 \vee f_1{}^- f_2{}^= \vee f_1{}^= f_2{}^-) &= 0 && \text{(B.119)}\\
0 &= 0 && \text{(B.120)}
\end{aligned}
$$

6.

$$
\begin{aligned}
f_3^1 \cdot f_3^= &= 0 && \text{(B.121)}\\
(f_1{}^- f_2{}^1 \vee f_1{}^1 f_2{}^- \vee f_1{}^0 f_2{}^= \vee f_1{}^= f_2{}^0)& && \\
\cdot(f_1{}^0 f_2{}^1 \vee f_1{}^1 f_2{}^0 \vee f_1{}^- f_2{}^= \vee f_1{}^= f_2{}^-) &= 0 && \text{(B.122)}\\
0 &= 0 && \text{(B.123)}
\end{aligned}
$$

□

Anhang C

Tupel-Dekomposition: Primäre Daten

C.1 Allgemeine Erläuterungen

Die Daten der folgenden Abschnitte des Anhangs sind im Ergebnis der Messungen mit einer UNIX-Workstation (Sparc Station 20) unter dem Betriebssystem Solaris entstanden. Als Eingangsdaten wurden für die Untersuchung Zufallsfunktionen mit 14 Variablen herangezogen, die über eine unterschiedliche Anzahl Konjunktionen (oder in der TVL-Darstellung: unterschiedliche Anzahl der Zeilen) verfügen. Alle Zeitangaben erfolgen in Sekunden.

C.2 Anzahl der Konjunktionen der Funktionen vor nach Dekomposition

In den Spalten der nachfolgenden Tabelle sind von links nach rechts dargestellt:

- die Anzahl der Konjunktionen der nicht optimierten Funktionen,
- die Anzahl der Zeilen der Funktionen in der TVL-Darstellung,
- die Anzahl der Zeilen der Funktion $f^{-}(\underline{x})$, $f^{0}(\underline{x})$, $f^{1}(\underline{x})$ und $f^{=}(\underline{x})$ in der TVL-Darstellung,
- die gesamte Anzahl der Zeilen der durch Tupel aus drei Teilfunktionen dargestellten Funktionen in der TVL-Darstellung,
- die gesamte Anzahl der Zeilen der durch Tupel aus vier Teilfunktionen dargestellten Funktionen in der TVL-Darstellung.

	$f(\underline{x})$	$f^{-}(\underline{x})$	$f^{0}(\underline{x})$	$f^{1}(\underline{x})$	$f^{=}(\underline{x})$	3-Tupel	4-Tupel
200	95	0	51	44	447	95	542
400	416	9	195	212	1120	416	1536
600	741	22	334	383	1497	739	2236
800	1012	56	450	506	1701	1012	2713
1000	1233	100	557	578	1833	1235	3068
1200	1465	138	667	675	1892	1480	3372

Fortsetzung der Tabelle:

	$f(\underline{x})$	$f^{-}(\underline{x})$	$f^{0}(\underline{x})$	$f^{1}(\underline{x})$	$f^{=}(\underline{x})$	3-Tupel	4-Tupel
1400	1679	204	743	719	1922	1666	3588
1600	1896	255	834	796	1856	1885	3741
1800	2024	304	816	889	1875	2009	3884
2000	2209	371	939	874	1847	2184	4031
2200	2386	404	896	1036	1797	2336	4133
2400	2530	469	1038	977	1787	2484	4271
2600	2670	579	937	1070	1720	2586	4306
2800	2855	576	1015	1109	1650	2700	4350
3000	2956	689	1038	1098	1597	2825	4422
3200	3070	733	1049	1157	1581	2939	4520
3400	3212	793	1056	1157	1492	3006	4498
3600	3326	845	1192	1080	1451	3117	4568
3800	3455	892	1184	1093	1407	3169	4576
4000	3534	952	1081	1180	1363	3213	4576
4200	3632	987	1108	1222	1272	3317	4589
4400	3735	1061	1210	1095	1216	3366	4582
4600	3778	1127	1216	1075	1195	3418	4613
4800	3894	1158	1216	1103	1138	3477	4615
5000	4013	1244	1112	1213	1093	3569	4662
5200	4043	1277	1100	1197	1035	3574	4609
5400	4166	1320	1085	1224	1007	3629	4636
5600	4237	1351	1115	1215	943	3681	4624
5800	4324	1434	1186	1113	912	3733	4645
6000	4316	1487	1154	1062	904	3703	4607
6200	4410	1495	1132	1149	817	3776	4593
6400	4526	1581	1043	1158	819	3782	4601
6600	4583	1651	1135	1056	794	3842	4636
6800	4606	1634	1158	1019	757	3811	4568
7000	4694	1719	1004	1160	727	3883	4610
7200	4733	1775	972	1101	675	3848	4523
7400	4757	1763	1120	1011	648	3894	4542
7600	4830	1797	1125	995	596	3917	4513
7800	4866	1828	1097	990	602	3915	4517
8000	4921	1841	1079	991	539	3911	4450
8200	4941	1910	969	1043	568	3922	4490
8400	5063	2009	937	1042	492	3988	4480
8600	5076	1952	949	1060	448	3961	4409
8800	5097	2025	1020	961	449	4006	4455
9000	5161	2080	907	978	428	3965	4393
9200	5152	2081	887	1010	416	3978	4394
9400	5198	2118	882	972	393	3972	4365
9600	5217	2146	844	974	386	3964	4350
9800	5330	2170	935	840	354	3945	4299
10000	5320	2210	844	921	354	3975	4329

Fortsetzung der Tabelle:

	$f(\underline{x})$	$f^{-}(\underline{x})$	$f^{0}(\underline{x})$	$f^{1}(\underline{x})$	$f^{=}(\underline{x})$	3-Tupel	4-Tupel
10200	5326	2225	823	955	323	4003	4326
10400	5425	2270	795	926	313	3991	4304
10600	5422	2321	891	755	304	3967	4271
10800	5435	2333	888	758	274	3979	4253
11000	5495	2345	774	858	254	3977	4231

C.3 Berechnungszeiten für Operation Konjunktion

In der folgenden Tabelle sind dargestellt:

- die Zeilenzahl der TVL der zu verknüpfenden Funktionen,
- die für die Durchführung der Operation Konjunktion benötigten Zeiten bei der Darstellung der Funktionen durch eine TVL und durch Tupel aus 3 und 4 TVL.

1.F.	2.F.	1 TVL	3-Tupel	4-Tupel	1. F.	2. F.	1 TVL	3-Tupel	4-Tupel
95	416	0,00	0,00	0,06	4757	4830	4,39	2,60	2,45
96	740	0,01	0,01	0,08	4795	4899	3,84	1,98	2,46
741	1012	0,11	0,06	0,33	4866	4921	4,31	2,36	2,43
1233	1465	0,27	0,16	0,58	4941	5063	4,48	2,21	2,49
1249	1674	0,32	0,18	0,81	5016	5049	4,40	2,11	2,52
1679	1896	0,48	0,31	1,05	5076	5097	4,34	2,13	2,54
2024	2209	0,69	0,43	1,27	5146	5250	4,40	2,34	2,47
2024	2402	0,76	0,46	1,30	5161	5152	4,45	2,42	2,50
2386	2530	0,95	0,58	1,45	5198	5217	4,72	2,16	2,48
2670	2855	1,37	0,73	1,63	5315	5409	4,71	2,32	2,55
2670	2965	1,28	0,76	1,64	5326	5425	4,91	2,33	2,50
2956	3070	1,54	0,89	1,79	5330	5320	4,87	2,18	2,46
3212	3326	1,81	1,03	1,92	5422	5435	5,03	2,24	2,47
3225	3416	1,97	1,08	2,03	5432	5470	4,85	2,18	2,49
3455	3534	2,29	1,15	1,98	5576	5573	5,09	2,23	2,45
3632	3735	2,40	1,35	2,09	5682	5689	5,57	2,25	2,42
3639	3812	2,26	1,52	2,09	5706	5790	5,38	2,21	2,37
3778	3894	2,79	1,57	2,12	5789	5820	5,50	2,17	2,36
4013	4043	3,14	1,74	2,24	5851	5924	5,69	2,22	2,52
4046	4175	2,76	1,60	2,22	5922	5958	5,76	2,19	2,57
4166	4237	3,12	1,83	2,28	6047	6058	5,99	2,10	2,27
4292	4429	3,10	1,72	2,30	6050	5978	5,91	2,23	2,37
4324	4316	3,19	1,84	2,34	6058	6034	6,11	2,07	2,12
4410	4526	3,67	1,85	2,45	6097	6170	6,16	2,02	2,03
4572	4682	3,49	1,88	2,40	6098	6112	6,09	2,12	2,11
4583	4606	4,09	2,00	2,42	6125	6126	6,13	2,26	2,16
4694	4733	4,31	2,20	2,40	6239	6242	6,39	1,97	2,00

C.4 Berechnungszeiten für Operation Negation

Die Spalten der folgenden Tabelle beziehen sich auf:

- die Zeilenzahl der TVL der zu negierenden Funktion,
- die für die Durchführung der Operation Negation benötigten Zeiten bei der Dastellung der Funktionen durch eine TVL und durch Tupel aus 3 und 4 TVL.

Funk.	1	3	4	Funk.	1	3	4	Funk.	1	3	4
95	0,01	0,01	0,00	4013	3,26	1,40	0,00	5315	4,49	1,30	0,00
96	0,00	0,01	0,00	4043	3,19	1,37	0,00	5320	4,17	1,30	0,00
416	0,08	0,07	0,00	4046	3,23	1,38	0,00	5326	4,15	1,31	0,00
740	0,24	0,16	0,00	4166	3,32	1,40	0,00	5330	4,10	1,30	0,00
741	0,21	0,17	0,00	4175	3,40	1,40	0,00	5409	4,50	1,31	0,00
1012	0,37	0,27	0,00	4237	3,52	1,46	0,01	5422	5,42	1,30	0,00
1233	0,53	0,37	0,00	4292	3,57	1,38	0,00	5425	4,31	1,30	0,00
1249	0,64	0,40	0,00	4316	3,53	1,40	0,01	5432	4,45	1,35	0,00
1465	0,72	0,49	0,00	4324	3,50	1,43	0,01	5470	4,39	1,25	0,00
1674	0,96	0,59	0,00	4410	3,70	1,39	0,00	5573	4,69	1,35	0,01
1679	0,91	0,59	0,00	4429	3,82	1,37	0,01	5576	4,38	1,27	0,00
1896	1,17	0,69	0,00	4526	3,75	1,58	0,00	5682	5,03	1,30	0,00
2024	1,24	0,76	0,00	4572	3,83	1,40	0,01	5689	5,06	1,32	0,01
2024	1,34	0,76	0,00	4583	3,67	1,41	0,01	5706	4,94	1,29	0,00
2209	1,45	1,00	0,00	4606	3,67	1,37	0,00	5789	4,74	1,24	0,00
2386	1,64	1,00	0,00	4682	3,77	1,38	0,00	5790	5,04	1,22	0,00
2402	1,67	0,91	0,00	4694	3,73	1,38	0,00	5820	4,59	1,16	0,00
2530	1,73	1,09	0,00	4733	3,75	1,40	0,00	5851	4,69	1,27	0,00
2670	1,86	1,06	0,01	4757	3,80	1,35	0,01	5922	4,50	1,25	0,00
2670	1,89	1,05	0,00	4795	3,95	1,43	0,00	5924	4,68	1,28	0,00
2855	2,08	1,08	0,00	4830	3,87	1,38	0,01	5958	5,11	1,19	0,00
2956	2,14	1,17	0,00	4866	3,87	1,44	0,00	5978	4,41	1,13	0,01
2965	2,14	1,12	0,01	4899	4,13	1,47	0,01	6034	4,80	1,09	0,00
3070	2,31	1,20	0,00	4921	4,00	1,35	0,01	6047	4,97	1,14	0,01
3212	2,39	1,27	0,00	4941	3,89	1,41	0,00	6050	5,01	1,19	0,00
3225	2,41	1,25	0,00	5016	4,66	1,49	0,00	6058	4,93	1,13	0,01
3326	2,49	1,33	0,00	5049	4,58	1,42	0,01	6058	4,68	1,19	0,00
3416	2,60	1,29	0,00	5063	4,15	1,47	0,01	6097	4,63	1,10	0,00
3455	2,70	1,30	0,00	5076	4,37	1,39	0,00	6098	4,94	1,11	0,00
3534	2,81	1,30	0,00	5097	4,34	1,38	0,01	6112	4,75	1,10	0,00
3632	2,91	1,41	0,00	5146	4,74	1,38	0,00	6125	4,64	1,13	0,00
3639	2,85	1,33	0,00	5152	4,07	1,35	0,01	6126	5,30	1,13	0,00
3735	3,02	1,40	0,01	5161	4,06	1,54	0,00	6170	4,67	1,09	0,00
3778	3,00	1,34	0,00	5198	4,12	1,32	0,01	6239	4,59	1,10	0,00
3812	3,00	1,36	0,00	5217	4,07	1,34	0,00				
3894	3,19	1,34	0,00	5250	4,38	1,31	0,00				

C.5 Berechnungszeiten für Operation Disjunktion

In der folgenden Tabelle sind dargestellt:

- die Zeilenzahl der TVL der zu verknüpfenden Funktionen,
- die für die Durchführung der Operation Disjunktion benötigten Zeiten bei der Darstellung der Funktionen durch eine TVL und durch als Tupel aus 3 und 4 TVL.

1.F.	2.F.	1 TVL	3-Tupel	4-Tupel	1. F.	2. F.	1 TVL	3-Tupel	4-Tupel
95	416	0,00	0,15	0,13	4757	4830	3,56	4,59	1,78
96	740	0,00	0,27	0,19	4795	4899	3,61	4,68	1,78
741	1012	0,05	1,17	0,88	4866	4921	3,87	4,53	1,76
1233	1465	0,13	2,52	1,58	4941	5063	3,99	4,47	1,68
1249	1674	0,30	2,54	1,54	5016	5049	3,88	4,50	1,68
1679	1896	0,46	3,76	1,90	5076	5097	3,99	4,39	1,66
2024	2209	0,64	3,90	2,33	5161	5152	4,15	4,32	1,60
2024	2402	0,75	3,97	2,04	5146	5250	4,15	4,29	1,60
2386	2530	0,90	4,71	2,42	5198	5217	4,19	4,19	1,56
2670	2855	1,15	5,13	2,23	5330	5320	4,39	4,13	1,52
2670	2965	1,19	4,71	2,31	5315	5409	4,41	4,17	1,56
2956	3070	1,39	5,07	2,46	5326	5425	4,61	4,27	1,52
3212	3326	1,66	5,34	2,42	5422	5435	5,13	4,04	1,48
3225	3416	1,69	5,34	2,34	5432	5470	4,52	3,97	1,52
3455	3534	1,90	5,29	2,34	5576	5573	4,71	4,06	1,54
3632	3735	2,12	5,23	2,19	5682	5689	5,20	4,05	1,40
3639	3812	2,17	5,18	2,20	5706	5790	5,07	3,69	1,38
3778	3894	2,33	5,22	2,15	5789	5820	5,03	3,61	1,33
4013	4043	2,52	5,43	2,13	5851	5924	5,13	3,67	1,55
4046	4175	2,61	5,13	2,13	5922	5958	5,21	3,67	1,56
4166	4237	2,75	5,05	2,05	6050	5978	5,32	3,58	1,58
4324	4316	2,91	4,98	2,01	6058	6034	5,31	3,44	1,54
4292	4429	2,98	4,95	1,99	6047	6058	5,43	3,64	1,61
4410	4526	3,12	4,92	1,95	6098	6112	5,47	3,39	1,30
4583	4606	3,28	4,83	1,91	6125	6126	5,49	3,50	1,54
4572	4682	3,32	4,84	1,89	6097	6170	5,53	3,44	1,33
4694	4733	3,46	4,75	1,82	6239	6242	5,59	3,46	1,32

C.6 Anzahl der Konjunktionen der Funktionen nach der Anwendung der rekursive Dekomposition

In drei Versuchsreihen wurden die Booleschen Funktionen mit 14 Variablen (Zeilenzahl in der linken Spalte) rekursiv in 3 und 4 Teilfunktionen zerlegt, so dass die Teilfunktionen 13, 10 und 7 Variablen enthalten. Die gesamten Zeilenzahlen über alle Teilfunktionen sind in der folgenden Tabelle dargestellt:

	3-Dekomposition			4-Dekomposition		
14 Var.	13 Var.	10 Var.	7 Var.	13 Var.	10 Var.	7 Var.
95	95	95	95	542	3380	22936
416	416	416	415	1536	8751	53020
741	739	740	739	2236	12566	74627
1012	1012	1023	1023	2713	15366	89533
1233	1235	1255	1257	3068	17013	100348
1465	1480	1480	1481	3372	18518	109313
1679	1666	1677	1674	3588	19972	117392
1896	1885	1880	1881	3741	21313	123264
2024	2009	1999	1999	3884	22122	128616
2209	2184	2167	2173	4031	22971	133144
2386	2336	2322	2324	4133	23461	136323
2530	2484	2469	2469	4271	24033	139677
2670	2586	2564	2567	4306	24600	142107
2855	2700	2688	2697	4350	24877	145448
2956	2825	2809	2812	4422	25390	146927
3070	2939	2894	2909	4520	25771	148168
3212	3006	2966	2971	4498	25949	150324
3326	3117	3064	3086	4568	26182	151127
3455	3169	3133	3126	4576	26309	151904
3534	3213	3152	3157	4576	26333	153011
3632	3317	3249	3241	4589	26374	152367
3735	3366	3242	3252	4582	26381	152947
3778	3418	3344	3337	4613	26320	153045
3894	3477	3387	3396	4615	26455	153729
4013	3569	3465	3485	4662	26528	153227
4043	3574	3545	3531	4609	26558	153603
4166	3629	3526	3564	4636	26421	153291
4237	3681	3613	3618	4624	26319	152261
4324	3733	3639	3645	4645	26292	152815
4316	3703	3581	3578	4607	26204	151624
4410	3776	3650	3650	4593	25919	150868
4526	3782	3697	3691	4601	26092	150496
4583	3842	3677	3699	4636	25904	150010
4606	3811	3706	3718	4568	25662	148403
4694	3883	3706	3694	4610	25687	148607
4733	3848	3681	3691	4523	25255	146029
4757	3894	3716	3736	4542	25222	145564

Fortsetzung der Tabelle:

	3-Dekomposition			4-Dekomposition		
14 Var.	13 Var.	10 Var.	7 Var.	13 Var.	10 Var.	7 Var.
4830	3917	3738	3725	4513	24954	144951
4866	3915	3736	3723	4517	24911	144301
4921	3911	3736	3744	4450	24602	142970
4941	3922	3662	3683	4490	24662	142100
5063	3988	3663	3666	4480	24012	139615
5076	3961	3739	3740	4409	24022	139330
5097	4006	3733	3744	4455	23776	138354
5161	3965	3668	3685	4393	23514	136297
5152	3978	3645	3651	4394	23379	135660
5198	3972	3636	3635	4365	22938	134039
5217	3964	3632	3624	4350	22995	133445
5330	3945	3545	3559	4299	22580	131138
5320	3975	3648	3626	4329	22624	130906
5326	4003	3603	3633	4326	22383	129800
5425	3991	3524	3514	4304	22032	128778
5422	3967	3494	3506	4271	21631	126138

C.7 Optimierung der Dekomposition

In den zwei Versuchsreihen wurden in den Algorithmen der 3- und 4-Dekomposition verwendet:

- die Ableitungsoperation (Abl.),
- die Operationen Konjunktion und Differenz (K./D.).

In der folgenden Tabelle sind die Zeiten zusammengefasst, die zur Dekomposition der Funktionen mit in der linken Spalte dargestellen Zeilenzahl benötigt werden.

Zeilen-	3-Dekomp.		4-Dekomp.		Zeilen-	3-Dekomp.		4-Dekomp.	
zahl	Abl.	K./D.	Abl.	K./D.	zahl	Abl.	K./D.	Abl.	K./D.
100	0	0	0	0	9328	19	13	28	22
492	0	1	0	1	9501	19	12	29	21
969	0	0	2	1	9790	20	13	30	23
1433	1	0	2	2	9953	20	14	30	23
1869	1	1	3	3	10097	20	15	30	24
2314	2	1	4	4	10317	22	14	31	25
2740	2	1	6	5	10512	21	15	32	25
3168	3	2	7	6	10766	22	16	32	26
3512	4	2	9	7	10864	23	16	33	27
3938	5	2	10	7	10996	24	16	34	26
4246	5	3	11	7	11092	23	17	34	28
4648	7	3	11	8	11275	24	17	34	28
5007	7	4	12	9	11396	24	18	34	28
5346	8	4	14	10	11683	25	19	36	30
5671	8	5	14	11	11780	26	19	36	31
6015	9	5	17	11	11882	25	20	36	30
6323	11	5	17	12	12043	26	20	36	31
6652	11	6	19	13	12101	26	21	37	32
6908	12	6	19	14	12283	26	21	38	32
7122	12	7	20	15	12339	26	22	37	33
7435	13	8	21	15	12573	27	22	37	33
7706	14	8	22	16	12607	26	22	39	34
7997	15	9	22	17	12677	27	22	38	34
8190	16	9	24	18	12799	27	24	39	35
8461	16	9	25	18	12966	29	23	38	35
8702	17	10	26	19	13039	28	23	39	35
8921	17	11	27	20	13163	29	25	39	36
9121	18	12	27	22					

Anhang D

Verteilte Verarbeitung: Primäre Daten

D.1 Allgemeine Erläuterungen

Die Daten der folgenden Abschnitte des Anhangs sind im Ergebnis der Messungen in einem Subnetz mit 4 UNIX-Workstation (LUNA, SPACE, NINIVE und ASSUR) unterschiedlicher Leistungsfähigkeit unter dem Betriebssystem Solaris entstanden (siehe die Tabelle im Abschnitt 5.3.1). Als Eingangsdaten wurden für die Untersuchung Zufallsfunktionen mit 20 Variablen herangezogen, die über eine unterschiedliche Anzahl Konjunktionen (oder in der TVL-Darstellung: unterschiedliche Anzahl der Zeilen) verfügen. Alle Zeitangaben erfolgen in Sekunden.

D.2 Verteilte Verarbeitung mit zwei Servern

1.Funk.	2.Funk.	seq,sp	sp+lu,sp	seq,lu	lu+sp,lu	sp+lu,sp,s.	lu+sp,lu,s.
6844	6966	7	11	3	8	10	7
16503	16798	28	29	16	24	28	21
30372	30341	138	114	79	80	92	77
72580	73526	716	388	408	388	378	342
121663	121390	1631	885	933	731	805	753
171161	171637	2678	1350	1548	1292	1327	1245
221095	221306	3731	1791	2173	1683	1858	1683
267380	268208	4667	2361	2784	2032	2365	2030
311722	313228	5516	2465	3360	2589	2440	2293
352719	354462	6240	2915	3822	2871	2878	2671
391298	391227	6787	3452	4199	3051	3087	3047
425420	424962	7189	3153	4470	3165	3172	3190

Erläuterungen zu den Abkürzungen der Tabelle:

1.Funk. - die Anzahl der Konjunktionen der 1. Funktion.

2.Funk. - die Anzahl der Konjunktionen der 2. Funktion.

seq,sp - die Zeit der sequentiellen Verarbeitung auf dem Rechner SPACE in Sekunden.

sp+lu,sp - die Zeit der verteilten Verarbeitung in Sekunden mit den Serverprogrammen auf SPACE und LUNA und dem Clientprogramm auf SPACE.

seq,lu - die Zeit der sequentiellen Verarbeitung auf dem Rechner LUNA in Sekunden.

lu+sp,lu - die Zeit der verteilten Verarbeitung in Sekunden mit den Serverprogrammen auf LUNA und SPACE und dem Clientprogramm auf LUNA.

sp+lu,sp,s. - die Zeit der verteilten Verarbeitung in Sekunden mit den Serverprogrammen auf SPACE und LUNA und dem Clientprogramm auf SPACE. Die Aufträge werden absteigend sortiert abgearbeitet.

lu+sp,lu,s. - die Zeit der verteilten Verarbeitung in Sekunden mit den Serverprogrammen auf LUNA und SPACE und dem Clientprogramm auf LUNA. Die Aufträge werden absteigend sortiert abgearbeitet.

D.3 Verteilte Verarbeitung mit 3 und 4 Servern und nebenläufige Verarbeitung auf einer Workstation

1.Funk.	2.Funk.	lu+sp+ni,sp	lu+sp+ni+as,sp	lu+lu,lu	sp+sp,sp
6844	6966	10	10	9	13
16503	16798	28	24	26	36
30372	30341	88	74	105	122
72580	73526	321	327	466	551
121663	121390	666	646	1021	1149
171161	171637	1097	966	1655	1984
221095	221306	1453	1269	2293	2703
267380	268208	1640	1470	2931	3280
311722	313228	1994	1917	3516	3703
352719	354462	2323	2178	4005	3939
391298	391227	2527	2368	4396	4230
425420	424962	2625	2452	4719	4708

Erläuterung zu den Abkürzungen der Tabelle:

1.Funk. - die Anzahl der Konjunktionen der 1. Funktion.

2.Funk. - die Anzahl der Konjunktionen der 2. Funktion.

lu+sp+ni,sp - die Zeit der verteilten Verarbeitung in Sekunden mit den Serverprogrammen auf LUNA, SPACE und NINIVE und dem Clientprogramm auf SPACE. Die Aufträge werden absteigend sortiert abgearbeitet.

lu+sp+ni+as,sp - die Zeit der verteilten Verarbeitung in Sekunden mit den Serverprogrammen auf LUNA, SPACE, NINIVE und ASSUR und dem Clientprogramm auf SPACE. Die Aufträge werden absteigend sortiert abgearbeitet.

lu+lu,lu - die Zeit der verteilten Verarbeitung in Sekunden mit dem Serverprogramm auf LUNA, wobei der Server als nebenläufiger Server verwendet wird. Das Clientprogramm läuft ebenfalls auf LUNA. Die Aufträge werden absteigend sortiert abgearbeitet.

sp+sp,sp - die Zeit der verteilten Verarbeitung in Sekunden mit dem Serverprogramm auf SPACE, wobei der Server auf SPACE als nebenläufiger Server verwendet wird. Das Clientprogramm läuft ebenfalls auf SPACE. Die Aufträge werden absteigend sortiert abgearbeitet.

D.4 Zusätzliche Aufwendungen

1.Funk.	2.Funk.	ges. Zeit	Übertr.-Zeit	ges. Zeit	Übertr.-Zeit
6844	6966	7	0	7	0
16503	16798	22	0	21	1
30372	30341	79	0	78	2
72580	73526	344	1	349	2
121663	121390	796	4	791	4
171161	171637	1325	4	1319	2
221095	221306	1761	5	1740	5
267380	268208	2067	6	2094	3
311722	313228	2421	7	2452	11
352719	354462	2607	6	2604	7
391298	391227	3140	7	2932	8
425420	424962	3340	11	3367	7

Erläuterung zu den Abkürzungen der Tabelle: Verteilte Verarbeitung erfolgt mit den Serverprogrammen auf LUNA und SPACE und dem Clientprogramm auf LUNA. Die Spalten 3 und 5 beziehen sich auf die gesamte Verarbeitungszeit. Die Werte der Spalten 4 und 6 sind die Zeiten, die für die Datenübertragungen über Socket gebraucht wurden. Die in den Spalten 3 und 4 angegebenen Zeiten beziehen sich auf die Versuche, bei denen die Übertragungen der Daten für den nächsten Auftrag erst nach der Übertragung der Ergebnisse des fertigen Auftrags erfolgten. Die in den Spalten 5 und 6 angegebenen Zeiten beziehen sich auf die Versuche mit der gleichzeitigen Übertragung der Ergebnisse des fertigen Auftrags und der Daten für den nächsten Auftrag.

Index

Abbildungsverzeichnis

Tabellenverzeichnis

Danksagung

Die vorliegende Dissertation entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Informatik der Technischen Universität Bergakademie Freiberg.

Mein besonderer Dank gilt Herrn Prof. Bernd Steinbach, der nicht nur die interessante Themenstellung vergab, sondern auch stets die Zeit für gute Ratschläge und fachliche Hinweise fand. Prof. Steinbach war es auch, der offensichtlich nie den Glauben an einen erfolgreichen Abschluss der Arbeit verlor und mich damit in moralischer Hinsicht unterstützte.

Bedanken möchte ich mich bei meinen Gutachtern, Herrn Prof. J. Beister und Herrn Prof. M. Gössel, die mir Gelegenheit gaben, bereits im Vorfeld aus meiner Arbeit vorzutragen und mit ihren Hinweisen zum Gelingen beitrugen.

Schließlich gilt der Dank meinen Freunden und Kollegen. Ihr Interesse an den Fortschritten der Arbeit war sicherlich auch ein Ansporn, einen erfolgreichen Abschluss anzustreben. Hervorheben möchte ich Frau Schüttauf, die sich der Mühe unterzog, sehr gewissenhaft das Script durchzusehen und mich auf Mängel aufmerksam zu machen.

Freiberg, im Juni 2003 Galina Kempe