Rechnerstrukturen WS 2012/13

- Repräsentation von Daten
  - Repräsentation von Texten (Wiederholung)
  - Repräsentation ganzer Zahlen (Wiederholung)
  - Repräsentation rationaler Zahlen (Wiederholung)
  - Repräsentation anderer Daten
- Boolesche Funktionen und Schaltnetze
  - Einleitung
  - Boolesche Algebra
  - Repräsentationen boolescher Funktionen
  - Normalformen boolescher Funktionen
  - Repräsentation boolescher Funktionen mit OBDDs

Hinweis: *Folien teilweise a. d. Basis von Materialien von Thomas Jansen*

17. Oktober 2012

Repräsentation von Texten: Unicode

Prinzip: Standardisierte numerische Kodierung von Zeichen

aktueller Standard Unicode 6.2 (26. September 2012)

- verwaltet vom Unicode-Konsortium (`http://www.unicode.org`)
- unterstützt verschiedene Codierungsformate (Unicode Transformation Format): UTF-8, UTF-16, UTF-32 mit 8, 16, 32 Bits
- längere Formate erweitern kürzere Formate
- vereinbart auch weitere Informationen (z. B. Schreibrichtung, Kombination von Zeichen (Codepoints))

Repräsentation ganzer Zahlen: Überblick

feste Länge $l = 5$, $(2^l = 32,\ 2^{l-1} = 16,\ 2^{l-1} - 1 = 15)$

z	VZ-Betrag	Bias $b = 16$	Bias $b = 15$	1er-K.	2er-K.
1	00001	10001	10000	00001	00001
-1	10001	01111	01110	11110	11111
0	00000 10000	10000	01111	00000 11111	00000
15	01111	11111	11110	01111	01111
-15	11111	00001	00000	10000	10001
16	—	—	11111	—	—
-16	—	00000	—	—	10000

Zweierkomplement bei weitem gebräuchlichste Darstellung

Repräsentation rationaler Zahlen

Wunsch rationale Zahlen $q \in \mathbb{Q}$ repräsentieren können

Vereinbarung feste Repräsentationslänge l Bits

Feste Position des Kommas ⇒ **Festkommazahlen**

Beispiel

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
$= 16$	$= 8$	$= 4$	$= 2$	$= 1$	$= 0{,}5$	$= 0{,}25$	$= 0{,}125$
1	0	1	1	0,	1	0	1
16		$+4$	$+2$		$+0{,}5$		$+0{,}125$
$= 22{,}625$							

allgemein bei v Vorkomma- und m Nachkommastellen

$$z = \sum_{i=-m}^{v-1} z_i \cdot 2^i$$

Gleitkommazahlen: IEEE-754 1985

$z = (-1)^s \cdot m \cdot 2^e$
mit $s \in \{0, 1\}$, $e \in \mathbb{Z}$, $m \in \mathbb{Q}$ und $1 \leq m < 2$

Festlegungen

- führende 1 der „Mantisse“ wird nicht mit abgespeichert (heißt „implizite Eins“)
- Mantisse in Binärcodierung (Festkommazahlen mit Ziffern ausschließlich hinter dem Komma)
- Exponent in Exzessdarstellung mit Bias $b = 2^{l_e - 1} - 1$

Gesamtlänge	Vorzeichen	Exponent	Mantisse	*Precision*
32	1	8	23	*single*
64	1	11	52	*double*

IEEE 754-1985: ein Beispiel

$l = 32,\ l_s = 1,\ l_e = 8,\ l_m = 23$
$b = 2^7 - 1 = 127,\ e_{\text{min}} = -b + 1 = -126,\ e_{\text{max}} = 2^8 - b - 2 = 127$

Wir wollen -3 darstellen.

negativ, also Vorzeichenbit 1

Darstellung als Summe von Zweierpotenzen
$3 = 2 + 1 = 2^1 + 2^0 = \left(2^0 + 2^{-1}\right) \cdot 2^1$

Exponent 1 Exzessdarstellung $1 + b = 128$ darstellen

$128 = (1000\,0000)_2$

Mantisse 1,1, implizite Eins entfällt, also $100\,0000\,0000 \cdots$

Ergebnis $\underbrace{1}_{\text{VZ}}\ \underbrace{1000\,0000}_{\text{Exponent}}\ \underbrace{100\,0000\,0000\,0000\,0000\,0000}_{\text{Mantisse}}$

Repräsentation anderer Daten

hier für uns interessant „primitive Daten"
(von Hardware direkt unterstützt)

zentral Programme
in der Regel Bitmuster fester Länge (z.B. 4 Byte): Befehl, Operand

Problem Was repräsentiert ein Byte im Speicher?

kaum verwendet: Typbits

Repräsentation von Datenfolgen

Speicher oft in Worten organisiert

Wort ja nach Rechner 2 Bytes, 4 Bytes, . . .

heterogene Daten hintereinander in den Speicher schreiben
dabei manchmal Wortgrenzen beachten
dann leere Zellen (Bytes) möglich

homogene Daten Arrays

Problem Wie erkennt man das Ende der Folge?

- feste Anzahl vereinbaren
- Länge am Anfang speichern
- spezielles Endezeichen verwenden

Boolesche Funktionen und Schaltnetze

Boolesche Funktionen

vielleicht schon bekannt Aussagenlogik

Satz ist Aussage mit eindeutigem Wahrheitswert

Wahrheitswerte wahr, falsch

neue zusammengesetzte Aussagen
durch Verknüpfung von Aussagen

Verknüpfungen

- Negation ($\neg$, „nicht“)
- Konjunktion ($\wedge$, „und“)
- Disjunktion ($\vee$, „oder“)

Definition der Verknüpfungen

Seien A, B zwei Aussagen.

Definition Negation

A	$\neg A$
falsch	wahr
wahr	falsch

Definition Konjunktion

A	B	$A \wedge B$
falsch	falsch	falsch
falsch	wahr	falsch
wahr	falsch	falsch
wahr	wahr	wahr

Definition Disjunktion

A	B	$A \vee B$
falsch	falsch	falsch
falsch	wahr	wahr
wahr	falsch	wahr
wahr	wahr	wahr

Boolesche Algebra

Definition 2

Wir nennen $(B, \cup, \cap, ^{-})$ mit $B = \{0, 1\}$ und $x \cup y = \max\{x, y\}$, $x \cap y = \min\{x, y\}$, $\overline{x} = 1 - x$ für alle $x, y \in B$ boolesche Algebra.

George Boole, englischer Mathematiker, 1815–1864

beobachte Entsprechungen:

falsch	$\Leftrightarrow$	0
wahr	$\Leftrightarrow$	1
$\wedge$	$\Leftrightarrow$	$\cap$
$\vee$	$\Leftrightarrow$	$\cup$
$\neg$	$\Leftrightarrow$	$^{-}$

Rechengesetze

Satz 3

In der booleschen Algebra $(B, \cup, \cap, ^{-})$ gilt für alle $x, y, z \in B$:

Kommutativität: $x \cup y = y \cup x$, $x \cap y = y \cap x$

Assoziativität: $(x \cup y) \cup z = x \cup (y \cup z)$,
$(x \cap y) \cap z = x \cap (y \cap z)$

Distributivität: $x \cap (y \cup z) = (x \cap y) \cup (x \cap z)$,
$x \cup (y \cap z) = (x \cup y) \cap (x \cup z)$

Neutralelemente: $x \cup 0 = x$, $x \cap 1 = x$

Nullelemente: $x \cup 1 = 1$, $x \cap 0 = 0$

Rechengesetze II

Satz 3 (cont.)

In der booleschen Algebra $(B, \cup, \cap, ^{-})$ gilt für alle $x, y, z \in B$ auch:

Idempotenz: $x = x \cup x = x \cap x$

Involution: $x = \overline{\overline{x}} = \neg\neg x$

Absorption: $(x \cup y) \cap x = x$, $(x \cap y) \cup x = x$

Resolution: $(x \cup y) \cap (\overline{x} \cup y) = y$, $(x \cap y) \cup (\overline{x} \cap y) = y$

Komplementarität: $x \cup (y \cap \overline{y}) = x$, $x \cap (y \cup \overline{y}) = x$

de Morgansche Regeln: $\overline{x \cup y} = \overline{x} \cap \overline{y}$, $\overline{x \cap y} = \overline{x} \cup \overline{y}$

Beweis Absorption

Absorption: $(x \cup y) \cap x = x$

x	y	$x \cup y$	linke Seite $(x \cup y) \cap x$	rechte Seite x	
0	0	0	0	0	$\surd$
0	1	1	0	0	$\surd$
1	0	1	1	1	$\surd$
1	1	1	1	1	$\surd$

Repräsentationen boolescher Funktionen

Definition 4

Seien $n, m \in \mathbb{N}$. Eine Funktion $f\colon B^n \to B^m$ heißt boolesche Funktion.

Notation $B^n =$ Menge aller n-stelligen Tupel über B
Beispiel $B^2 = \{(0,0), (0,1), (1,0), (1,1)\}$

Anzahl boolescher Funktionen

boolesche Funktion $f\colon B^n \to B^m$ als Wertetabelle darstellbar
mit $|B^n| = 2^n$ Zeilen
und $|B^m| = 2^m$ Möglichkeiten je Zeile

$\Rightarrow 2^{m^{2^n}} = 2^{m \cdot 2^n}$ boolesche Funktionen $f\colon B^n \to B^m$

Beispiel $2^{1\cdot 2^2} = 2^4 = 16$ boolesche Funktionen $f\colon B^2 \to B$

Alle booleschen Funktionen $f: B^2 \to B$

x	0	0	1	1				x	0	0	1	1		
y	0	1	0	1				y	0	1	0	1		
f_1	0	0	0	0	Nullfkt.	0		f_9	1	0	0	0	NOR	
f_2	0	0	0	1	AND	$\wedge$		f_{10}	1	0	0	1	Äquiv.	$\Leftrightarrow$
f_3	0	0	1	0				f_{11}	1	0	1	0	Negation	$\neg y$
f_4	0	0	1	1	Proj.	x		f_{12}	1	0	1	1		
f_5	0	1	0	0				f_{13}	1	1	0	0	Negation	$\neg x$
f_6	0	1	0	1	Proj.	y		f_{14}	1	1	0	1	Impl.	$\Rightarrow$
f_7	0	1	1	0	XOR	$\oplus$		f_{15}	1	1	1	0	NAND	
f_8	0	1	1	1	OR	$\vee$		f_{16}	1	1	1	1	Einsfkt.	1

Verwenden im Weiteren $\wedge$ (Konjunktion), $\vee$ (Disjunktion), $\bar{\ }$ (Negation)

Darstellung boolescher Funktionen

gerade gesehen Wertetabelle
(Orientierung meistens wie hier)

x	y	f_7
0	0	0
0	1	1
1	0	1
1	1	0

bei fester Reihenfolge Wertevektor

$f_7 : (0, 1, 1, 0)$

Index und Minterm

Index	x	y	f_7	
0	0	0	0	nicht einschlägig
1	0	1	1	einschlägig
2	1	0	1	einschlägig
3	1	1	0	nicht einschlägig

Definition

Die boolesche Funktion, für die nur der Index i einschlägig ist, heißt Minterm zum Index i.

Ein Minterm ist nur mit Negationen und Konjunktionen darstellbar:

$$x_j = \begin{cases} 0 & \rightsquigarrow \overline{x_j} \\ 1 & \rightsquigarrow x_j \end{cases}$$

und dann Konjunktion all dieser Literale ($\widehat{=}$ [negierte] Variable)

Beispiel zu Index und Minterm

Index	x_1	x_2	f_7	
0	0	0	0	nicht einschlägig
1	0	1	1	einschlägig
2	1	0	1	einschlägig
3	1	1	0	nicht einschlägig

Beispiel Minterm zum Index $2 = (1\,0)_2$, also $m_2(x_1, x_2) = x_1 \wedge \overline{x_2}$

Index	x_1	x_2	$x_1\,\overline{x_2}$
0	0	0	0
1	0	1	0
2	1	0	1
3	1	1	0

Hinweis: In der Regel abkürzende Notation der Konjunktion, z.B.:

$x_1 \wedge \overline{x_2} \quad \rightsquigarrow \quad x_1\,\overline{x_2}$

Normalformen

Für wie viele Eingaben liefert ein Minterm 1?

klar für genau 1

Folgerungen

- Disjunktion aller Minterme zu einschlägigen Indizes einer booleschen Funktion f ist wieder f
- XOR-Verknüpfung aller Minterme zu einschlägigen Indizes einer booleschen Funktion f ist wieder f

Normalformen

Definition 8

- Die Darstellung von f als Disjunktion all ihrer Minterme zu einschlägigen Indizes heißt disjunktive Normalform (DNF).
- Die Darstellung von f als XOR-Verknüpfung all ihrer Minterme zu einschlägigen Indizes heißt Ringsummen-Normalform (RNF).

Anmerkung Normalformen sind eindeutig.

Beispiel

Index	x	y	f_7	Minterm
0	0	0	0	$\overline{x}\,\overline{y}$
1	0	1	1	$\overline{x}\,y$
2	1	0	1	$x\,\overline{y}$
3	1	1	0	$x\,y$

DNF von f_7 $\overline{x}\,y \vee x\,\overline{y}$

RNF von f_7 $\overline{x}\,y \oplus x\,\overline{y}$

Funktionale Vollständigkeit

Beobachtung jede boolesche Funktion $f\colon B^n \to B$ nur mittels Konjunktion, Disjunktion und Negation darstellbar (z. B. durch ihre DNF)

Definition 5

Eine Menge $\mathcal{F}$ von booleschen Funktionen heißt funktional vollständig, wenn sich jede boolesche Funktion durch Einsetzen und Komposition von Funktionen aus $\mathcal{F}$ darstellen lässt.

Satz 6

$\{\wedge, \vee, \neg\}$ ist funktional vollständig.

Funktionale Vollständigkeit

Gibt es kleinere funktional vollständige Mengen?

Behauptung $\{\vee, \neg\}$ und $\{\wedge, \neg\}$ sind beide funktional vollständig.

Beobachtung Zum Beweis genügt es zu zeigen, dass $\{\wedge, \vee, \neg\}$ darstellbar ist.

Beweis.

Anwendung der de Morgan-Regeln (Satz 3)

$$x \vee y = \overline{\overline{x} \wedge \overline{y}} \quad \Rightarrow \quad \{\vee\} \text{ mit } \{\wedge, \neg\} \text{ darstellbar}$$

$$x \wedge y = \overline{\overline{x} \vee \overline{y}} \quad \Rightarrow \quad \{\wedge\} \text{ mit } \{\vee, \neg\} \text{ darstellbar}$$

□

Kleinste funktional vollständige Mengen

Wie viele Funktionen für funktionale Vollständigkeit mindestens?

Satz 7

$\{\text{NAND}\}$ ist funktional vollständig.

Beweis.

Es genügt, $\{\neg, \vee\}$ mit NAND darzustellen.

$\neg x = \text{NAND}(x, x)$

x	$\neg x$	$\text{NAND}(x, x)$
0	1	1
1	0	0

$x \vee y = \text{NAND}(\text{NAND}(x, x), \text{NAND}(y, y))$

x	y	$x \vee y$	$\neg x$	$\neg y$	$\text{NAND}(\text{NAND}(x, x), \text{NAND}(y, y))$
0	0	0	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	1	0	0	1

Vorsicht, Notation!

Anmerkung $\overline{x\,y} \neq \overline{x}\,\overline{y}$

$$\overline{x\,y} = \neg(x\,y)$$
$$\overline{x}\,\overline{y} = (\neg x) \wedge (\neg y)$$

x	y	$\overline{x\,y}$	$\overline{x}\,\overline{y}$
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0

Maxterme

Minterm-Darstellung betont Funktionswert 1.

Definition

Die boolesche Funktion, für die nur der Index i nicht einschlägig ist, heißt Maxterm zum Index i.

Beobachtung Definition Maxterm unterscheidet sich nur in „nicht“ von Definition Minterm

Beobachtung m_i Minterm zum Index i, M_i Maxterm zum Index i
$\Rightarrow M_i = \neg m_i$

Beobachtung Konjunktion aller Maxterme zu nicht einschlägigen Indizes einer booleschen Funktion f ist wieder f

Normalformen

Fortsetzung von Definition 8

- Die Darstellung von f als Konjunktion all ihrer Maxterme zu nicht einschlägigen Indizes heißt konjunktive Normalform (KNF).

Beispiel

Index	x	y	z	f_{bsp}	$x \vee y \vee \overline{z}$	$x \vee \overline{y} \vee z$	$x \vee \overline{y} \vee \overline{z}$
0	0	0	0	1	1	1	1
1	0	0	1	0	0	1	1
2	0	1	0	0	1	0	1
3	0	1	1	0	1	1	0
4	1	0	0	1	1	1	1
5	1	0	1	1	1	1	1
6	1	1	0	1	1	1	1
7	1	1	1	1	1	1	1

Darstellungen boolescher Funktionen

Wozu stellt man boolesche Funktionen dar?

- Realisierung
- Verifikation
- Fehleranalyse
- Synthese
- ...

Wo stellt man boolesche Funktionen dar?

- auf dem Papier
- im Computer

Probleme

- Wertetabelle, Wertevektor immer groß
- Normalformen oft groß
- Normalformen unterstützen gewünschte Operationen kaum

Wunsch andere Repräsentation

Eine Datenstruktur für boolesche Funktionen

Ziel $f\colon B^n \rightarrow B$ darstellen

Wünsche

- zu einer Belegung $x_1, x_2, \ldots, x_n$ schnell den Funktionswert $f(x_1, x_2, \ldots, x_n)$ ausrechnen können
- Funktionen schnell auf Gleichheit testen können
- Funktionen schnell manipulieren (z. B. eine Variable konstant setzen) können
- schnell eine Null-Eingabe/eine Eins-Eingabe finden können
- Funktionen möglichst klein repräsentieren
- ...

Ordered Binary Decision Diagrams

OBDDs

erster Schritt Festlegen einer Variablenordnung π
(z. B. $\pi = (x_3, x_1, x_2, x_4)$)

dann Baue πOBDD aus Knoten x_4 oder $\boxed{1}$

und Kanten $x_3 \xrightarrow{1} x_2$ nach folgenden Regeln:

- Knoten mit Variablen, 0 oder 1 markiert
- Kanten mit 0 oder 1 markiert
- Variablen-Knoten mit je einer ausgehenden 0- und 1-Kante
- Konstanten-Knoten ohne ausgehende Kante
- genau ein Knoten ohne eingehende Kante
- Kanten zwischen Variablenknoten beachten π

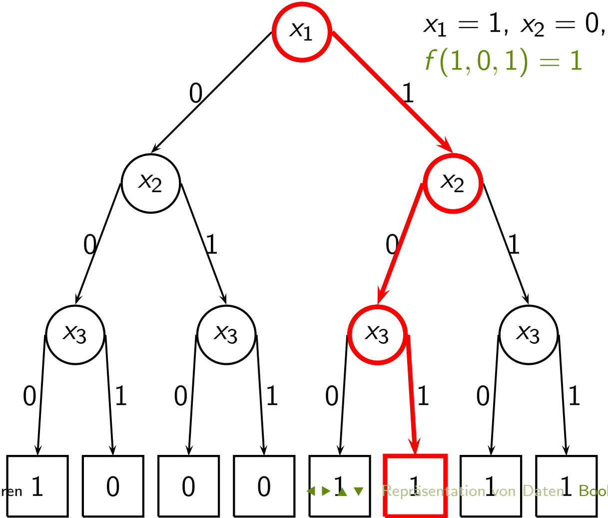
πOBDD – Ein Beispiel

Variablenordnung $\pi = (x_1, x_2, x_3)$

Beispiel Auswertung $f(1,0,1)$

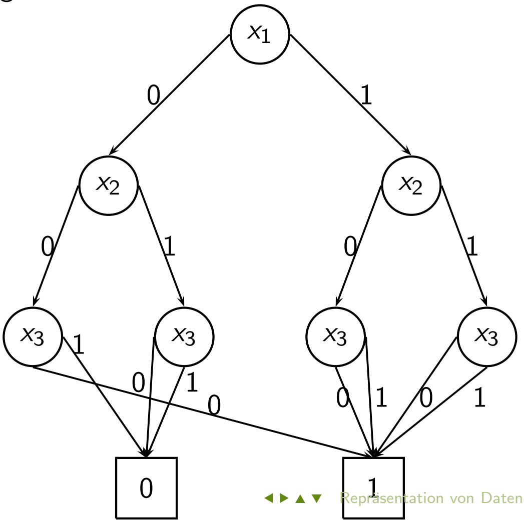
$x_1 = 1,\ x_2 = 0,\ x_3 = 1$

$f(1,0,1) = 1$



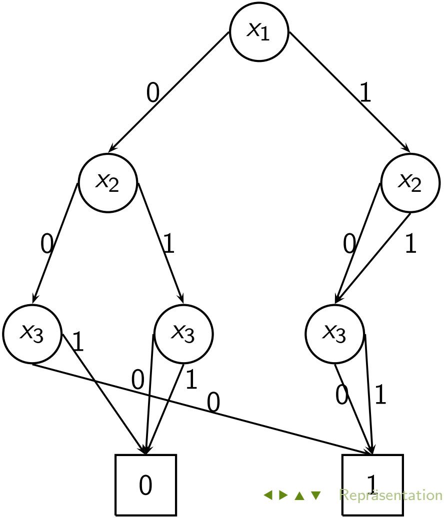
πOBDD-Größe

gleichartige Senken verschmelzen



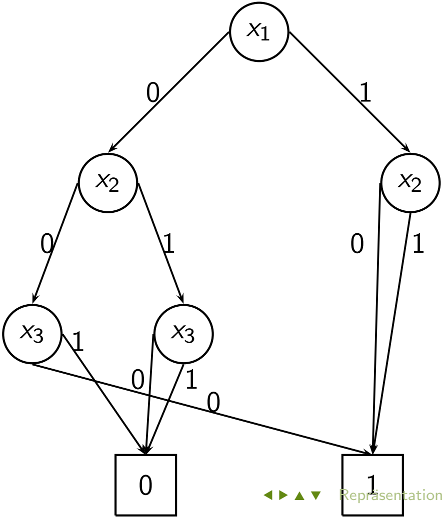
πOBDD-Größe

gleichartige Knoten verschmelzen



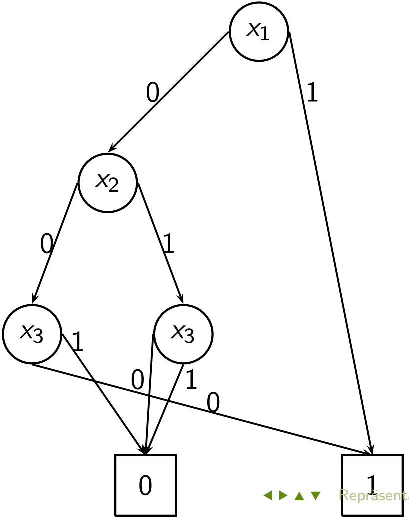
πOBDD-Größe

Knoten ohne Einfluss eliminieren



πOBDD-Größe

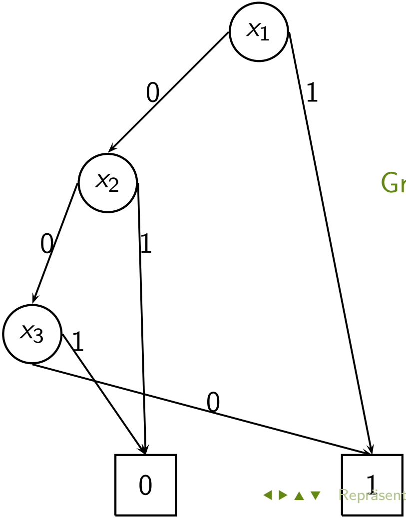
Knoten ohne Einfluss eliminieren



technische universität
dortmund

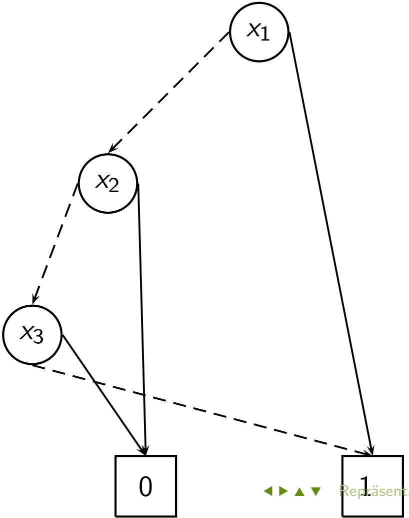
πOBDD-Größe

Knoten ohne Einfluss eliminieren



Größe minimal

Alternative Darstellung eines πOBDDs



OBDD-Reduzierung

Satz 9

Die erschöpfende Anwendung der

- Verschmelzungsregel „Knoten mit gleicher Markierung und gleichen Nachfolgern können verschmolzen werden" und
- Eliminationsregel „Ein Knoten mit gleichem Null- und Einsnachfolger kann entfernt werden"

in beliebiger Reihenfolge führt zum reduzierten πOBDD.

reduziert = minimale Größe und eindeutig