Kapitel 9

Decision Diagrams

9.1 Binary Decision Diagrams

9.2 Weitere DD-Typen

Übersicht

Decision Diagrams als effiziente Darstellung Boolescher Funktionen

- erstes Beispiel „mehrstufiger" Schaltkreise
- wichtig als Datenstruktur
  - für die Logiksynthese
  - für die formale Verifikation

Begriffe

- Decision Diagrams (DDs)
  - Binary Decision Diagrams (BDDs)
  - Kronecker Functional Decision Diagrams (KFDDs)
  - Multiplicative Binary Moment Diagrams (*BMDs)
  - Kronecker Multiplicative Binary Moment Diagrams (K*BMDs)

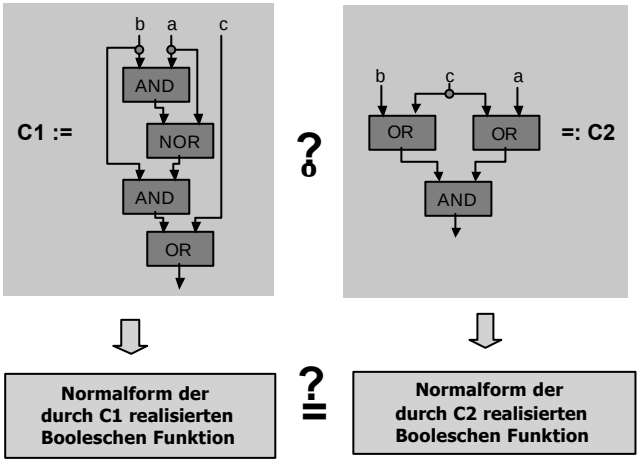
Ziele

- **Gelungener Kompromiss zwischen**

  kompakter Darstellung <--> effizienter Manipulation

- **Kanonische Darstellung**
  - Äquivalenzüberprüfung "einfach", nämlich Gleichheitstest
  - insbesondere auch der Tautologietest

Illustration dieser Idee



Normalformen Boolescher Funktionen

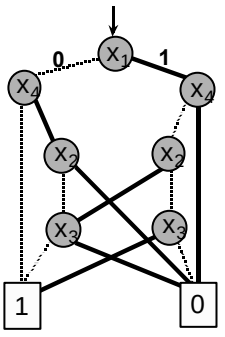
- Kan. Disjunktive Normalform
  $(a' \wedge b' \wedge c) \vee (a' \wedge b \wedge c) \vee (a \wedge b' \wedge c) \vee (a \wedge b \wedge c') \vee (a \wedge b \wedge c)$
- Kan. Konjunktive Normalform
  $(a \vee b \vee c) \wedge (a \vee b' \vee c) \wedge (a' \vee b \vee c)$
- Ring-Summen Expansion
  $c \oplus (a \wedge b) \oplus (a \wedge b \wedge c)$

... sind aufgrund ihrer sehr grossen Kosten in der Praxis nur für Boolesche Funktionen mit wenigen Eingängen einsetzbar

Decision Diagrams (DDs) als Alternative

Definition

- azyklischer Graph mit einer Wurzel
- **Innere Knoten**
  - sind mit einer Variablen markiert
  - haben zwei Nachfolger,
    - das **low-Kind** und
    - das **high-Kind**
- **Blätter**
  - sind mit einer Konstanten markiert

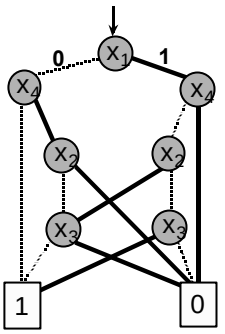


9.1 Binary Decision Diagrams

Definition [BDD über X_n]

- DD, in dem die Blätter alle mit 0 oder 1 markiert sind.
- Interpretation f_v eines Knotens v:
  - Ist v ein mit c markiertes Blatt, so beschreibt das Blatt die konstante Funktion, die jedes Argument auf c abbildet
  - Ist v ein innerer Knoten, so ist er mit einem $x_j \in X_n$ markiert, es gilt dann

    $f_v=(x_j \wedge f_{high(v)}) \vee (x_j' \wedge f_{low(v)})$



Zerlegungstyp

Kofaktoren und Shannon-Zerlegung

- Zu gegebener Boolescher Funktion f : $\mathbf{B}^n \to \mathbf{B}$ heißt
  $f_{x_{i=1}}$: $\mathbf{B}^n \to \mathbf{B}$ definiert durch
  $f_{x_{i=1}}(\alpha_1,,\ldots,\alpha_{i-1},\alpha_i,\alpha_{i+1},\ldots,\alpha_n)= f(\alpha_1,,\ldots,\alpha_{i-1},1,\alpha_{i+1},\ldots,\alpha_n)$
  für alle $\alpha \in \{0,1\}^n$
  der Kofaktor von f nach $x_i = 1$

  Entsprechend heißt $f_{x_{i=0}}$ der Kofaktor von f nach $x_i = 0$
- Es gilt dann für beliebiges f : $f = (x_i \wedge f_{x_{i=1}}) \vee (x_i' \wedge f_{x_{i=0}})$
- $f = (x_i \wedge f_{x_{i=1}}) \vee (x_i' \wedge f_{x_{i=0}})$
  heißt Shannon-Zerlegung von f

Weitere Zerlegungstypen

neben der Shannon-Zerlegung $f = (x \wedge f_{x=1}) \vee (x' \wedge f_{x=0})$

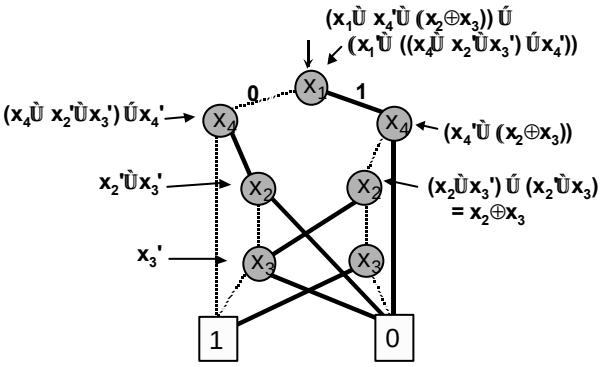
- $f = f_{x=0} \oplus x \wedge (f_{x=0} \oplus f_{x=1})$

  heißt positive Davio-Zerlegung von f

- $f = f_{x=1} \oplus x' \wedge (f_{x=0} \oplus f_{x=1})$

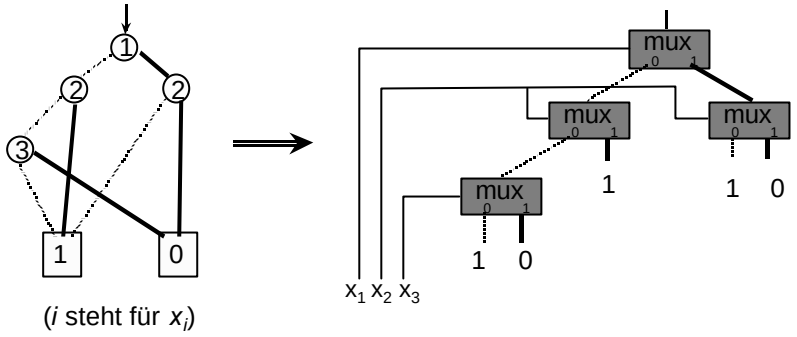
  heißt negative Davio-Zerlegung von f

Binary Decision Diagrams ff



$f_v = (x_j \wedge f_{high(v)}) \vee (x_j' \wedge f_{low(v)})$

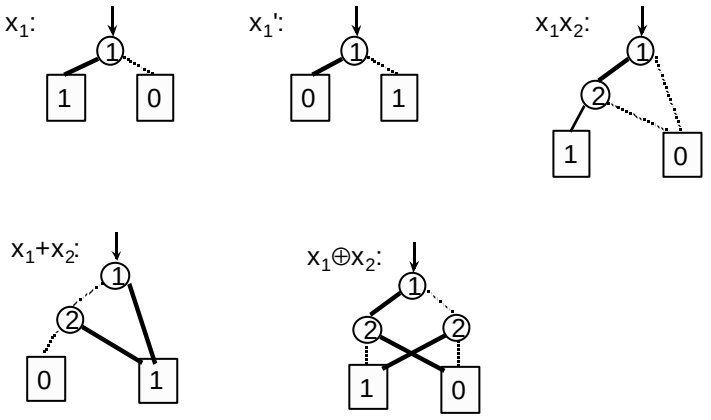
Interpretation von BDDs als mehrstufiger Schaltkreis



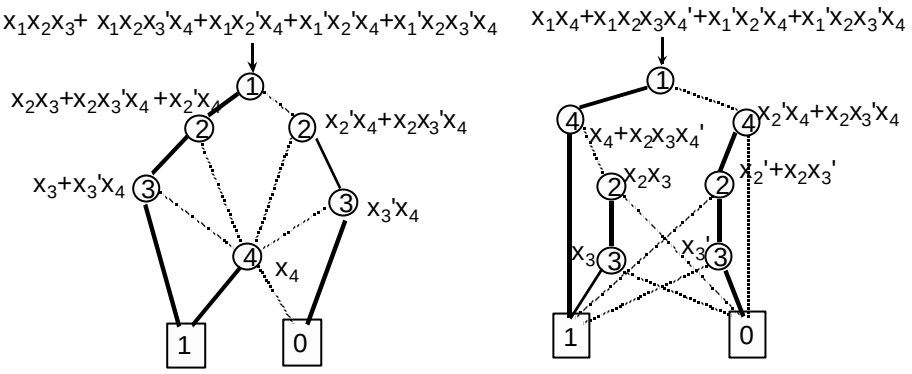
(*i* steht für x_i)

... beschreibt die Boolesche Funktion $x_1'x_2'x_3'+x_1'x_2+x_1x_2'$

BDDs von typischen Funktionen



Weitere Beispiele für BDDs



... beschreiben beide die Boolesche Funktion $x_1x_2x_3+x_2'x_4+x_3'x_4$

... doch keine kanonische Darstellungen !!?

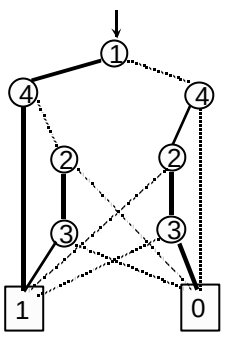
Geordnete DDs

Definition

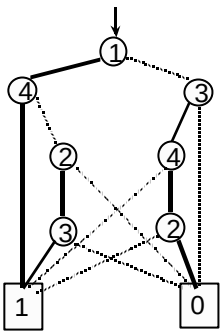
Ein **DD** heißt **frei**, wenn auf jedem Pfad von der Wurzel zu einem Blatt jede Variable höchstens einmal als Markierung vorkommt.

Ein **DD** heißt **geordnet**, wenn auf jedem Pfad von der Wurzel zu einem Blatt die Variablen in der gleichen Reihenfolge abgefragt werden.

Beispiele (nicht) geordneter freier DDs



geordneter freier DD



ungeordneter freier DD

Reduzierte BDDs

Idee

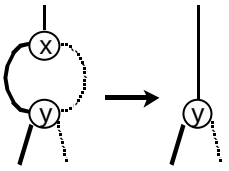
Stelle Information möglichst kompakt dar, „isomorphe Teil-BDDs" werden identifiziert

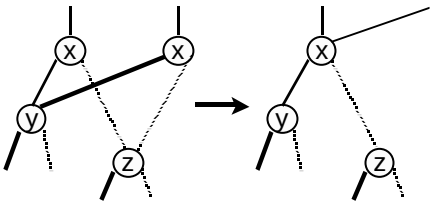
Definition

Ein **BDD** heisst **reduziert**, wenn es

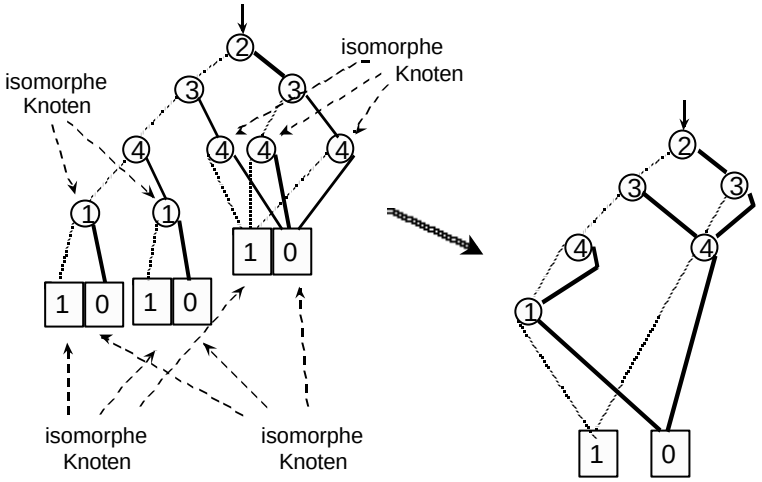
- keinen nichtterminalen Knoten v gibt mit Index x und $(f_v)_x=(f_v)_{x'}$
- keine verschiedenen Knoten v und w gibt, die gleich markiert sind und deren low- und high-Kind (falls existent) jeweils identisch sind

Reduktionsregeln

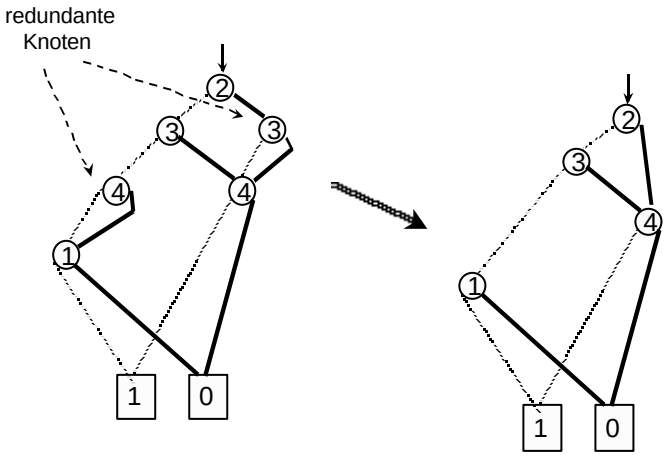




Beispiel für reduzierte und nicht reduzierte BDDs



Beispiel ff



Satz von Bryant´86

Geordnete reduzierte BDDs sind kanonische Darstellungen Boolescher Funktionen.

Schreibkonvention

BDD = reduzierter geordneter BDD

Satz von Bryant: der Beweis

Sei f in n Variablen über X_n gegeben.
Die Ordnung der Variablen sei oBdA $x_1,...,x_n$.

Wir zeigen, dass der BDD eindeutig bestimmt ist durch Induktion nach der Anzahl der Variablen, von denen f abhängt.

Zeige zuerst die Eindeutigkeit der BDD-Darstellung der konstanten Funktion **0**:

Sei also G ein BDD der konstanten Booleschen Funktion **0**.
Dann gilt:

Satz von Bryant: der Beweis ff

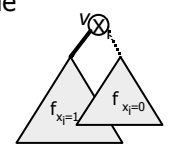
- Alle Pfade, die an der Wurzel von G starten, enden in dem mit 0 markierten Blatt.
- G enthält nur Blätter, die mit 0 markiert sind.
- Da G reduziert ist, enthält G nur genau ein mit 0 markiertes Blatt.
- Gäbe es in G einen inneren Knoten,
  - so gibt es auch einen inneren Knoten v, dessen beide Nachfolger Blätter sind, da G ein endlicher azyklischer Graph ist;
  - da G aber nur ein Blatt enthält, muss dieser Knoten v redundant sein.

- G besteht genau aus dem mit 0 markierten Blatt und ist somit eindeutig

Satz von Bryant: der Beweis ff

Sei nun f eine Boolesche Funktion, die wenigstens von einer Variablen abhängt.

- Seien G_1 und G_2 zwei BDDs von f mit gleicher Variablenordnung.
- Die Wurzel von G_1 bzw. G_2 sei mit der Variablen x_i bzw. x_j markiert, d.h. es gilt $f = x_i f_{x_i=1} + x_i' f_{x_i=0}$ und $f = x_j f_{x_j=1} + x_j' f_{x_j=0}$.
- Nehme an, dass $x_i \neq x_j$ und dass O.B.d.A. $x_i < x_j$ ist, dann gilt:
  - f hängt von x_i ab, ansonsten wäre die Wurzel von G_1 redundant
  - Kein Knoten von G_2 ist mit x_i markiert. G_2 beschreibt also eine Boolesche Funktion, die unabhängig von x_i ist. Widerspruch.
- Also gilt $x_i = x_j$.
- Da die beiden Kofaktoren $f_{x_i=0}$ und $f_{x_i=1}$ Boolesche Funktionen sind, die von einer Variablen weniger abhängen als f, sind die dazugehörigen BDDs eindeutig bestimmt.
- Hiermit folgt auch schon die Aussage



Konstruktion eines BDDs über die Kofaktoren

Konstruiere BDD bdd_f von $f = x_1x_2x_3+x_2'x_4+x_3'x_4$ bezgl. der Variablenordnung x_1, x_2, x_3, x_4

- Generiere sukzessive die BDDs für alle Kofaktoren und reduziere

Konstruktion eines BDDs über Boolesche Operationen

Konstruiere BDD bdd_f von $f = x_1x_2x_3+x_2'x_4+x_3'x_4$

- Generiere bdd_{x1}, bdd_{x2}, bdd_{x3}, bdd_{x4} ;
- Berechne $bdd_{x1x2} := \underline{AND}(bdd_{x1}, bdd_{x2})$;
- Berechne $bdd_{x1x2x3} := \underline{AND}(bdd_{x1x2}, bdd_{x3})$;
- Berechne $bdd_{x2'} := \underline{NOT}(bdd_{x2})$;
- Berechne $bdd_{x2'x4} := \underline{AND}(bdd_{x2'}, bdd_{x4})$;
- Berechne $bdd_{x3'} := \underline{NOT}(bdd_{x3})$;
- Berechne $bdd_{x3'x4} := \underline{AND}(bdd_{x3'}, bdd_{x4})$;
- Berechne $bdd_{x1x2x3+x2'x4} := \underline{OR}(bdd_{x1x2x3}, bdd_{x2'x4})$;
- Berechne $bdd_f := \underline{OR}(bdd_{x1x2x3+x2'x4}, bdd_{x3'x4})$;

Manipulation von BDDs: Ideen

Zu realisieren sind also die binären Booleschen Operatoren!

Beobachtung

Alle binären Booleschen Operatoren können auf den ternären Operator

ITE(F,G,H) := FG+F'H

zurückgeführt werden.

Beispiel

AND(F,G) = ITE(F,G,0)
OR(F,G) = ITE(F,1,G)
NOT(F) = ITE(F,0,1)

Rückführung binärer Operatoren auf ITE

Funktion	Name	Ausdruck	ITE-Ausdruck
0000	0	0	0
0001	AND(F,G)	FG	ITE(F,G,0)
0010	F>G	FG‘	ITE(F,G‘,0)
0011	F	F	F
0100	F<G	F'G	ITE(F,0,G)
0101	G	G	G
0110	XOR(F,G)	F⊕G	ITE(F,G‘,G)
0111	OR(F,G)	F+G	ITE(F,1,G)
1000	NOR(F,G)	F'G‘	ITE(F,0,G')
1001	XNOR(F,G)	(F⊕G)‘	ITE(F,G,G')
1010	NOT(G)	G‘	ITE(G,0,1)
1011	F≥G	F+G‘	ITE(F,1,G')
1100	NOT(F)	F‘	ITE(F,0,1)
1101	F≤G	F'+G	ITE(F,G,1)
1110	NAND(F,G)	F'+G‘	ITE(F,G‘,1)
1111	1	1	1

Idee zur Realisierung von ITE

Es gilt

$$\begin{aligned} ITE(F,G,H) &= FG+F'H \\ &= x(FG+F'H)_{x=1}+x'(FG+F'H)_{x=0} \\ &= x(F_{x=1}G_{x=1}+(F_{x=1})'\,H_{x=1})+x'(F_{x=0}G_{x=0}+(F_{x=0})'\,H_{x=0}) \\ &= x\cdot ITE(F_{x=1},G_{x=1},H_{x=1})+x'\cdot ITE(F_{x=0},G_{x=0},H_{x=0}) \end{aligned}$$

⇒ **Vorgehensweise**

- Berechne rekursiv $ITE(F_{x=1},G_{x=1},H_{x=1})$ und $ITE(F_{x=0},G_{x=0},H_{x=0})$
- Überprüfe, ob ein Knoten v mit index(v)=x, $f_{low(v)}= ITE(F_{x=0},G_{x=0},H_{x=0})$ und $f_{high(v)}= ITE(Fx=1,Gx=1,Hx=1)$ schon generiert wurde
- Falls nein, füge einen solchen Knoten ein.
- Zeitkomplexität: $O(|F|\cdot|G|\cdot|H|)$ (ohne Beweis)

Repräsentationsgrösse

Kernproblem: gute Variablenordnung

Zur Erinnerung

Für eine beliebige, aber feste Variablenordnung sind BDDs kanonische Darstellungen Boolescher Funktionen

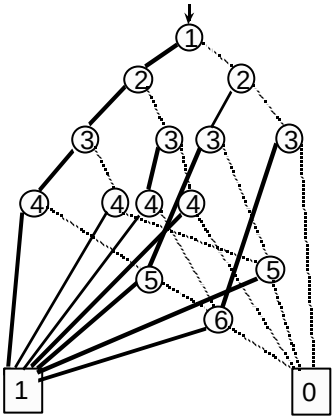
Beobachtung

Die Größe der BDDs hängt sehr von der gewählten Variablenordnung ab !

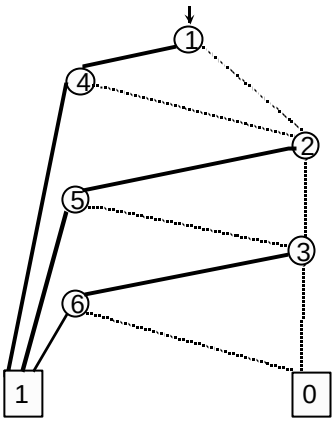
Beispiel $f^{(n)}(x_1,\ldots,x_{2n}) = x_1x_{n+1}+x_2x_{n+2}+\ldots+x_nx_{2n}$

Einfluss der Variablenordnungen: Beispiel

$f^{(n)}(x_1,\ldots,x_{2n}) = x_1x_{n+1}+x_2x_{n+2}+\ldots+x_nx_{2n}$



exponentielle Grösse



lineare Grösse

BDD-Minimierung

- **Komplexitätstheoretisch schwierig**
  - NP-schwer
  - Es gibt (nur laufzeitintensive) exakte Algorithmen !
- **Heuristiken zur BDD-Minimierung**
  - **Initiale Verfahren**
    bauen ausgehend von einer anderen Darstellung einen ersten BDD auf
  - **Umordnungsverfahren**
    versuchen eine bestehende Variablenordnung zu verbessern.
- Die Multiplikation ist nur mit einem exponentiellen Aufwand in der Anzahl der Bitbreite durch BDDs darstellbar, unabhängig von der gewählten Ordnung.