Universität Bremen

AGRA

Technische Informatik 1

Prof. Dr. Rolf Drechsler

Christina Plump

Überblick

Teil 1: Der Rechneraufbau (Kapitel 2-5)	Teil 2: Der Funktionalitätsaufbau (Kapitel 6-12)

• Rechner im Überblick
• Pipelining
• Speicher
• Parallelverarbeitung
	• Kodierung
	• Grundbegriffe, Boolesche Funktionen
	• **Darstellungsmöglichkeiten**
	  • Zweistufige Logiksynthese
	  • **Binäre Entscheidungsdiagramme**
	• Schaltkreise, Synthese, spezielle Schaltkreise

Kapitel 9: Decision Diagrams

Binary Decision Diagrams (BDDs)

Weitere DD-Typen

Lernziele

- Kronecker Functional Decision Diagrams (KFDD) als ausdrucksfähigere Form der Decision Diagrams kennenlernen
- Syntax und Semantik von KFDDs kennenlernen und verstehen
- Zerlegungsverfahren nach Davio kennen und anwenden können
- Kriterien für ein reduziertes KFDDs kennen und anwenden können
- Eindeutigkeit von reduzierten geordneten KFDDs verstehen
- Word-Level DDs kennen und verstehen
- Komplexitätsunterschiede in der Synthese und Expressivität verstehen

Decision Diagrams – Die Familie

Decision Diagrams

Binary
Decision Diagrams
(Bryant 1986)

Kronecker Functional
Decision Diagrams
(Drechsler et al., 1994)

Kronecker Multiplicative Binary Moment
Decision Diagrams
(Drechsler et al., 1996)

Multiplicative Binary Moment
Decision Diagrams
(Bryant et al., 1996)

und noch viele andere …

Zerlegungstypen - Erinnerung

Definition (Shannon-Zerlegung):
Sei $f \in \mathcal{B}_n$ eine Boolesche Funktion in n Variablen. Für $i \in \{1, \dots, n\}$ heißt $f = \left(x_i \cdot f_{x_i=1}\right) + \left(\overline{x_i} \cdot f_{x_i=0}\right)$ Shannon-Zerlegung von f.

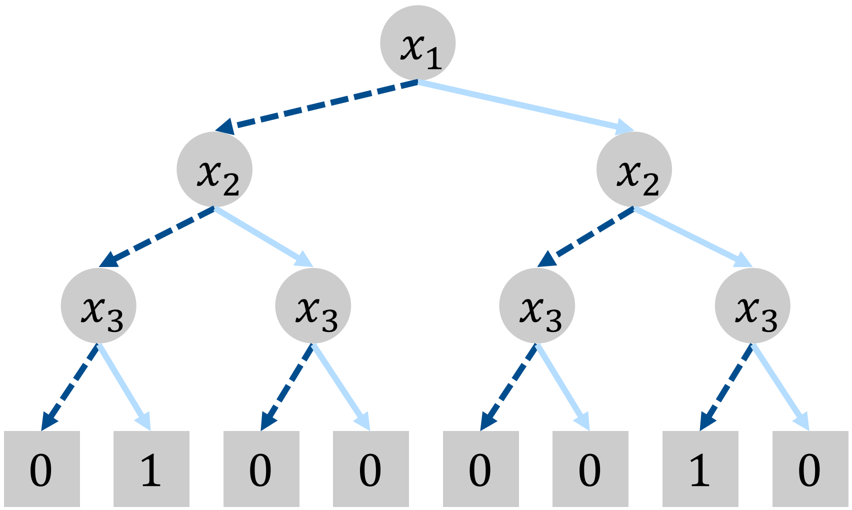
Definition (positive Davio-Zerlegung):
Sei $f \in \mathcal{B}_n$ eine Boolesche Funktion in n Variablen. Für $i \in \{1, \dots, n\}$ heißt $f = f_{x_i=0} \oplus x_i \cdot (f_{x_i=0} \oplus f_{x_i=1})$ positive Davio-Zerlegung von f.

Definition (negative Davio-Zerlegung):
Sei $f \in \mathcal{B}_n$ eine Boolesche Funktion in n Variablen. Für $i \in \{1, \dots, n\}$ heißt $f = f_{x_i=1} \oplus \overline{x_i} \cdot (f_{x_i=0} \oplus f_{x_i=1})$ positive Davio-Zerlegung von f.

KFDD: Formale Definition

Definition (Kronecker Functional Decision Diagram über X_n - Syntax):
Ein Kronecker Functional Decision Diagram (KFDD) ist ein Decision Diagram mit Konstanten 0 und 1 über X_n und einer Zerlegungstyp-Liste $(d_1, d_2, \ldots, d_n), d_i \in \{S, pD, nD\}$.

$$d = (nD, pD, S)$$



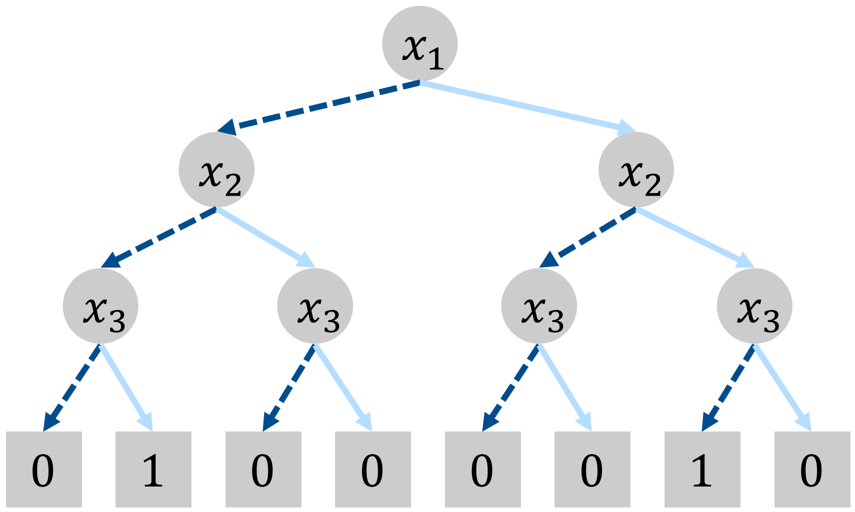
KFDD: Formale Definition

Definition (Kronecker Functional Decision Diagram über X_n - Semantik):
Die Funktion f_v an einem inneren Knoten $v \in V_n$ mit $l(v) = x_i \in X_n$ entspricht:

- bei $d_i = S$: $f_v = x_i \cdot f_{high(v)} + \overline{x_i} \cdot f_{low(v)}$
- bei $d_i = pD$: $f_v = f_{low(v)} \oplus x_i \cdot f_{high(v)}$
- bei $d_i = nD$: $f_v = f_{low(v)} \oplus \overline{x_i} \cdot f_{high(v)}$

Die Funktion f_v an einem Blattknoten $v \in V_t$ mit $l(v) = c \in \{0,1\}$ entspricht dem Label selbst, d.h. $f_v = c$.

$d = (nD, pD, S)$



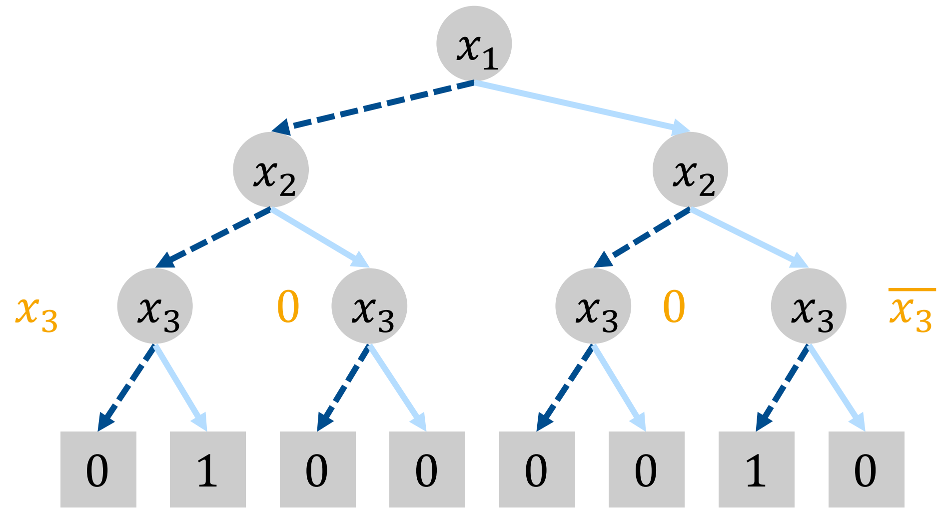
KFDD: Formale Definition

Definition (Kronecker Functional Decision Diagram über X_n - Semantik):
Die Funktion f_v an einem inneren Knoten $v \in V_n$ mit $l(v) = x_i \in X_n$ entspricht:

- bei $\boldsymbol{d_i = S}$: $\boldsymbol{f_v = x_i \cdot f_{high(v)} + \overline{x_i} \cdot f_{low(v)}}$
- bei $d_i = pD$: $f_v = f_{low(v)} \oplus x_i \cdot f_{high(v)}$
- bei $d_i = nD$: $f_v = f_{low(v)} \oplus \overline{x_i} \cdot f_{high(v)}$

Die Funktion f_v an einem Blattknoten $v \in V_t$ mit $l(v) = c \in \{0,1\}$ entspricht dem Label selbst, d.h. $f_v = c$.

$d = (nD, pD, \boldsymbol{S})$



Shannon

KFDD: Formale Definition

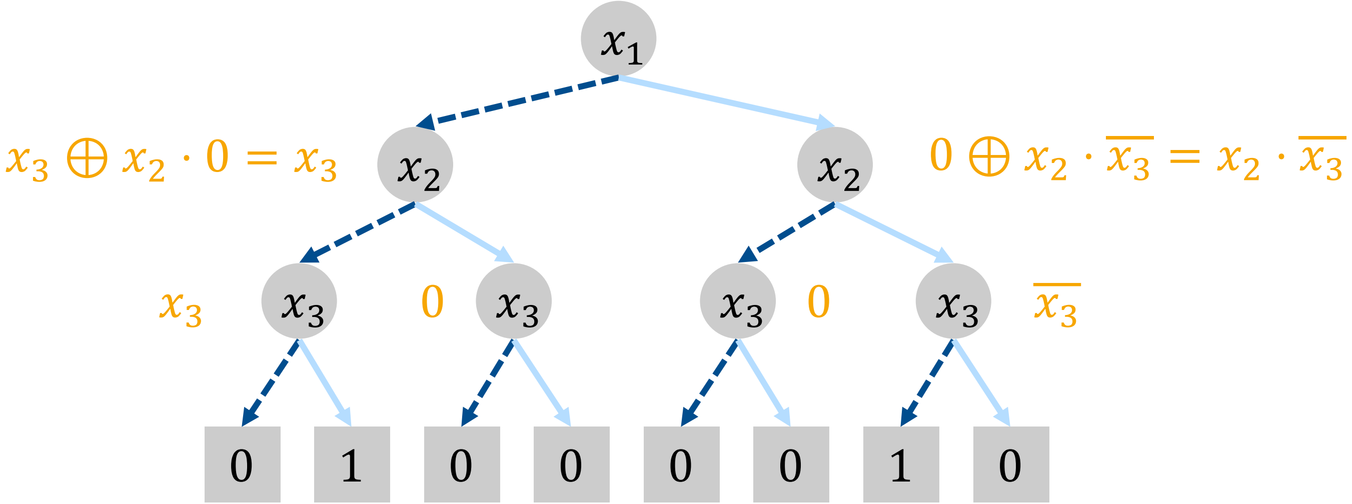
Definition (Kronecker Functional Decision Diagram über X_n - Semantik):

Die Funktion f_v an einem inneren Knoten $v \in V_n$ mit $l(v) = x_i \in X_n$ entspricht:

- bei $d_i = S$: $f_v = x_i \cdot f_{high(v)} + \overline{x_i} \cdot f_{low(v)}$
- bei $\boldsymbol{d_i = pD}$: $\boldsymbol{f_v = f_{low(v)} \oplus x_i \cdot f_{high(v)}}$
- bei $d_i = nD$: $f_v = f_{low(v)} \oplus \overline{x_i} \cdot f_{high(v)}$

Die Funktion f_v an einem Blattknoten $v \in V_t$ mit $l(v) = c \in \{0{,}1\}$ entspricht dem Label selbst, d.h. $f_v = c$.

$d = (nD, \boldsymbol{pD}, S)$



pos. Davio

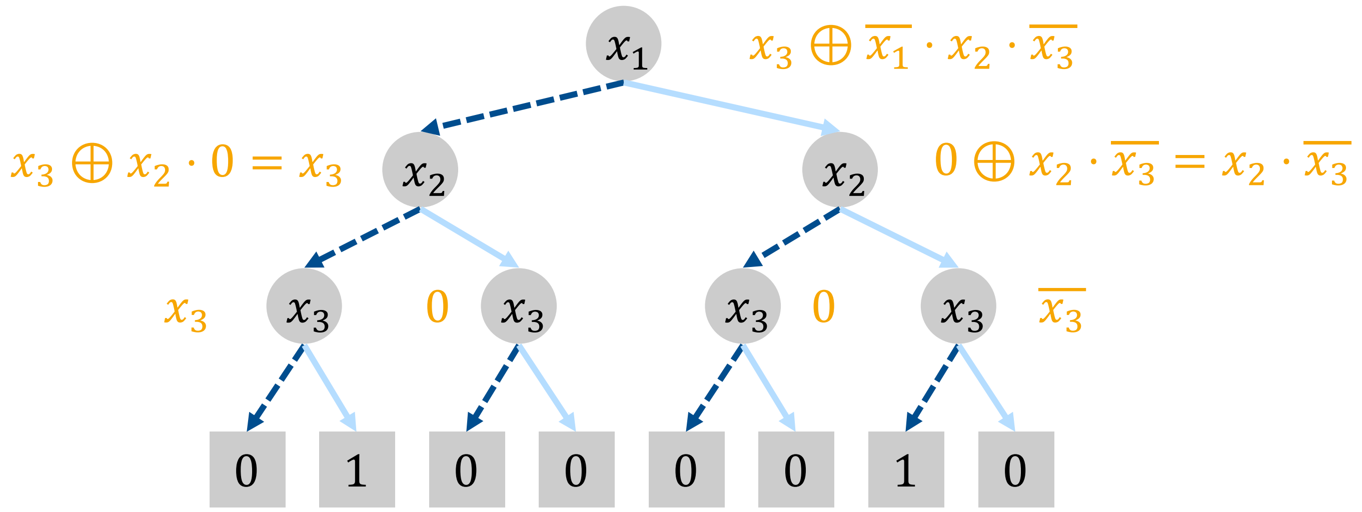
KFDD: Formale Definition

Definition (Kronecker Functional Decision Diagram über X_n - Semantik):
Die Funktion f_v an einem inneren Knoten $v \in V_n$ mit $l(v) = x_i \in X_n$ entspricht:

- bei $d_i = S$: $f_v = x_i \cdot f_{high(v)} + \overline{x_i} \cdot f_{low(v)}$
- bei $d_i = pD$: $f_v = f_{low(v)} \oplus x_i \cdot f_{high(v)}$
- bei $\boldsymbol{d_i = nD}$: $\boldsymbol{f_v = f_{low(v)} \oplus \overline{x_i} \cdot f_{high(v)}}$

Die Funktion f_v an einem Blattknoten $v \in V_t$ mit $l(v) = c \in \{0,1\}$ entspricht dem Label selbst, d.h. $f_v = c$.

$d = (\boldsymbol{nD}, pD, S)$



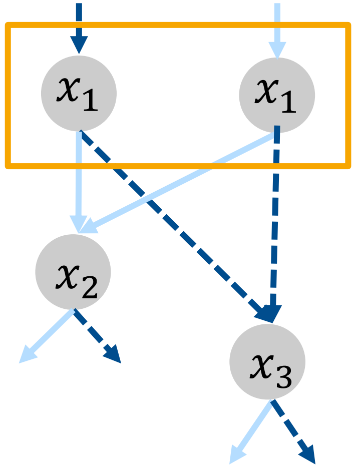
neg. Davio

KFDD ist geordnet,
aber nicht reduziert!

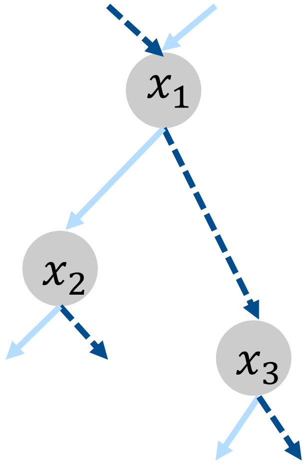
Reduktionsregeln bei KFDDs – Isomorphe Knoten

Definition (isomorphe Knoten im BDD):
Zwei verschiedene Knoten v, w mit $l(v) = l(w) = x_i$ in einem Binary Decision Diagram heißen isomorph, wenn $low(v) = low(w)$ und $high(v) = high(w)$, das heißt, beide Kinder jeweils identisch sind.

Definition (isomorphe Knoten im KFDD):
Zwei verschiedene Knoten v, w mit $l(v) = l(w) = x_i$ in einem Kronecker Functional Decision Diagram heißen isomorph, wenn $low(v) = low(w)$ und $high(v) = high(w)$, das heißt, beide Kinder jeweils identisch sind.



$$low(a_1) = a_4 = low(a_2)$$
$$high(a_1) = a_3 = high(a_2)$$



Gilt also unabhängig vom Zerlegungstyp

Reduktionsregeln bei KFDDs – Redundante Knoten

Definition (redundanter Knoten im BDD):
Ein nicht-terminaler Knoten v mit $l(v) = x_i$ in einem Binary Decision Diagram heißt redundant, wenn $(f_v)_{x_i=0} = (f_v)_{x_i=1}$ gilt, das heißt beide Kofaktoren identisch sind.

Definition (redundanter Knoten im KFDD):
Ein nicht-terminaler Knoten v mit $l(v) = x_i$ in einem Kronecker Functional Decision Diagram heißt redundant, wenn $(f_v)_{x_i=0} = (f_v)_{x_i=1}$ gilt, das heißt beide Kofaktoren identisch sind.

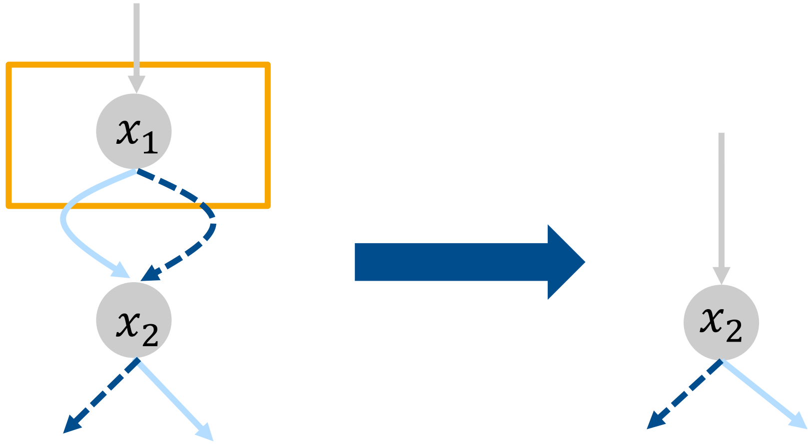
Identische Kofaktoren werden je nach
Zerlegungstyp unterschiedlich dargestellt

Reduktionsregeln bei KFDDs – Redundante Knoten

Definition (redundanter Knoten im KFDD):
Ein nicht-terminaler Knoten v mit $l(v) = x_i$ in einem Kronecker Functional Decision Diagram heißt redundant, wenn $(f_v)_{x_i=0} = (f_v)_{x_i=1}$ gilt, das heißt beide Kofaktoren identisch sind.

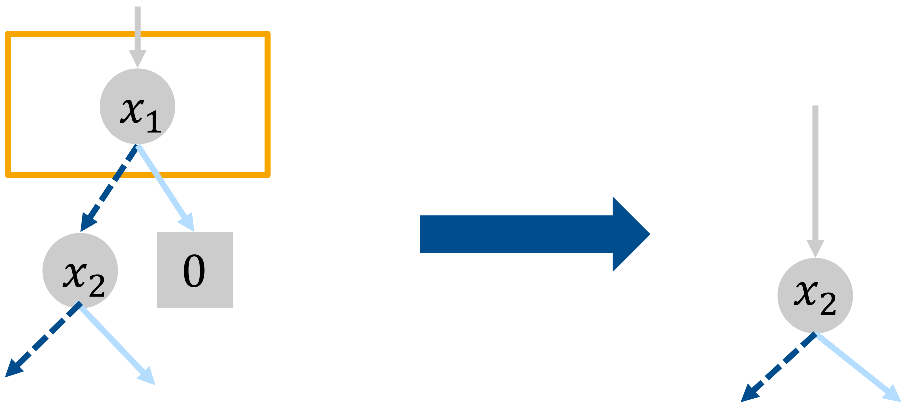
$$(f_{a_0})_{x_1=0} = f_{a_1} = (f_{a_0})_{x_1=1}$$

$$high(a_0) = low(a_0)$$



Shannon-Zerlegung

$$high(a_0) = (f_{a_0})_{x_1=0} \oplus (f_{a_0})_{x_1=1} = 0$$



Davio-Zerlegung

Kanonizität der KFDDs

Satz:
Für eine beliebige feste Variablenordnung und eine beliebige feste Zerlegungstyp-Liste sind reduzierte KFDDs kanonische Darstellungen Boolescher Funktionen.

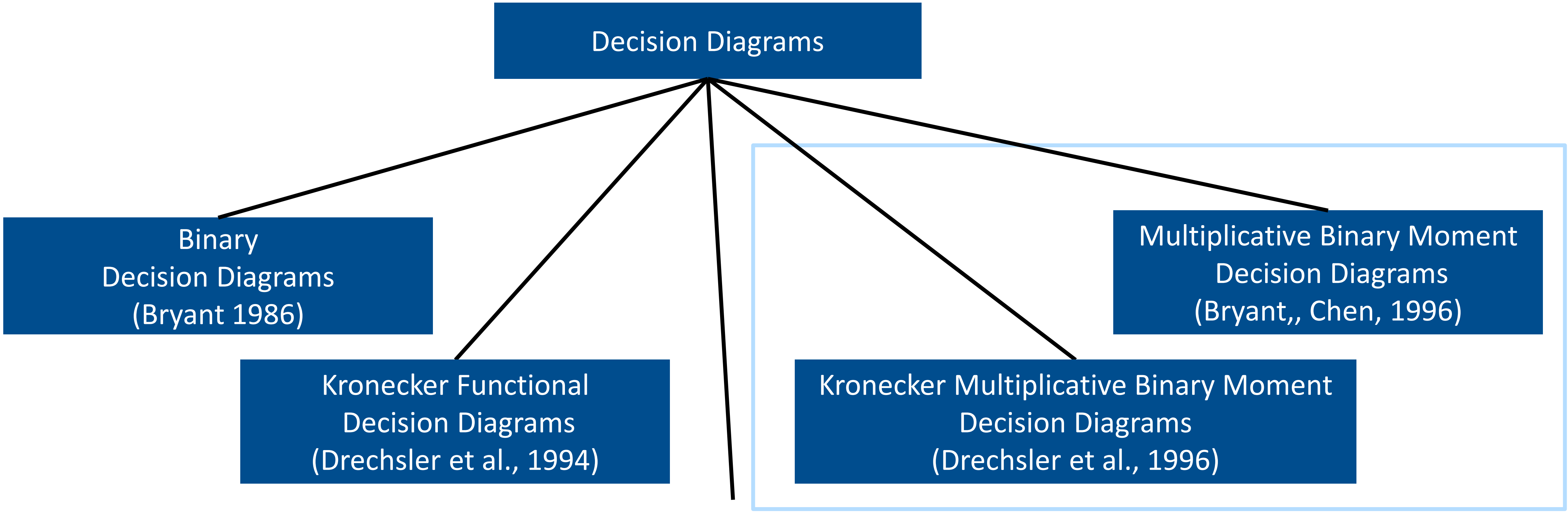
Beweis: Analog zum Satz von Bryant.

Vergleich KFDDs versus BDDs

- KFDDs sind mächtiger als BDDs! (BDDs sind spezielle KFDDs.)
- Ausnutzung des zusätzlichen Potentials oft schwierig, da Berechnung notwendig von
  - Variablenordnung und
  - Zerlegungstyp-Liste
- Operationen können bei KFDDs teuer sein
  - XOR polynomiell
  - AND, OR exponentiell

Erinnerung:
Bei BDDs ist die Laufzeit von Syntheseoperationen polynomiell beschränkt.

Decision Diagrams – Die Familie



und noch viele andere …

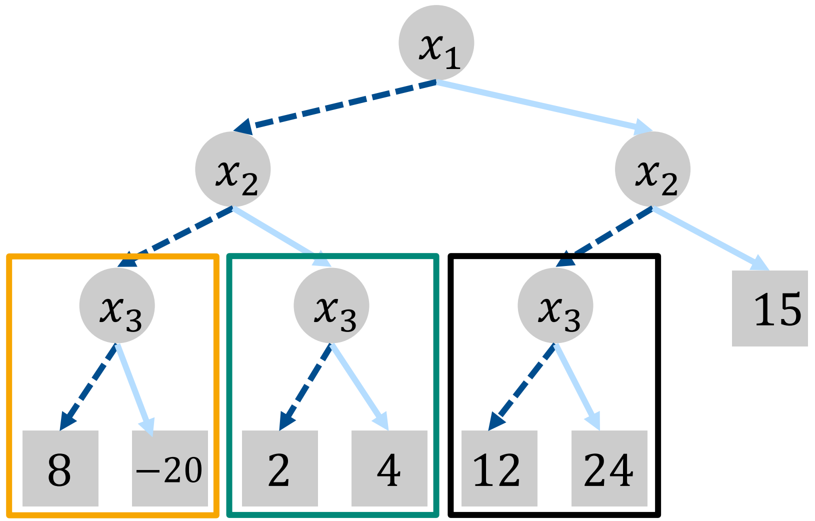
Word-Level DDs

- stellen Pseudo-Boolesche Funktionen $f\colon\{0,1\}^n \to \mathbb{Z}$ (bzw. $f\colon\{0,1\}^n \to \mathbb{Q}$) dar.
- Blätter sind mit ganzen Zahlen (rationalen Zahlen) markiert
- Andere Zerlegungstypen, beispielsweise $f = f_{x_i=0} + x_i \cdot (f_{x_i=1} - f_{x_i=0})$

$+,\cdot,-$ sind hier arithmetische Operationen

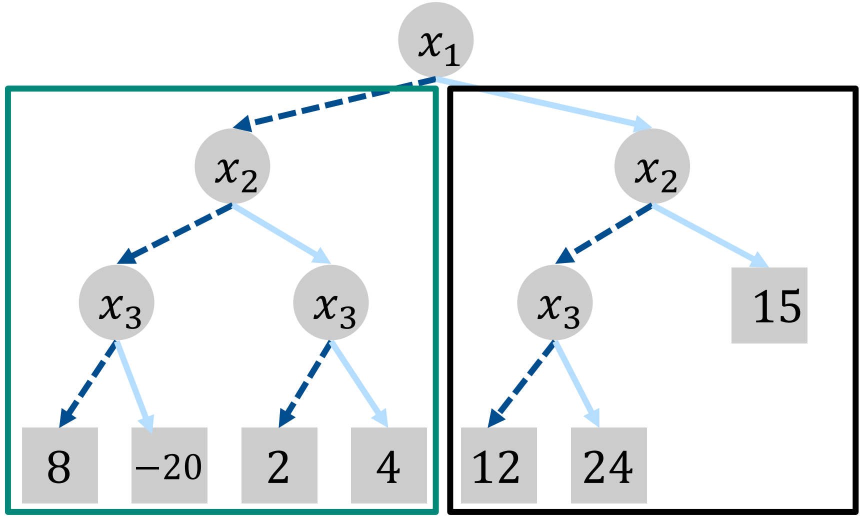
Beispiel

$$
\begin{aligned}
f(x,y,z) &= 8 - 20x_3 + 2x_2 + 4x_2x_3 + 12x_1 + 24x_1x_3 + 15x_1x_2 \\
&= (8 - 20x_3 + 2x_2 + 4x_2x_3) + x_1(12 + 24x_3 + 15x_2) \\
&= \big((8 - 20x_3) + x_2(2 + 4x_3)\big) + x_1\big((12 + 24x_3) + 15x_2\big)
\end{aligned}
$$



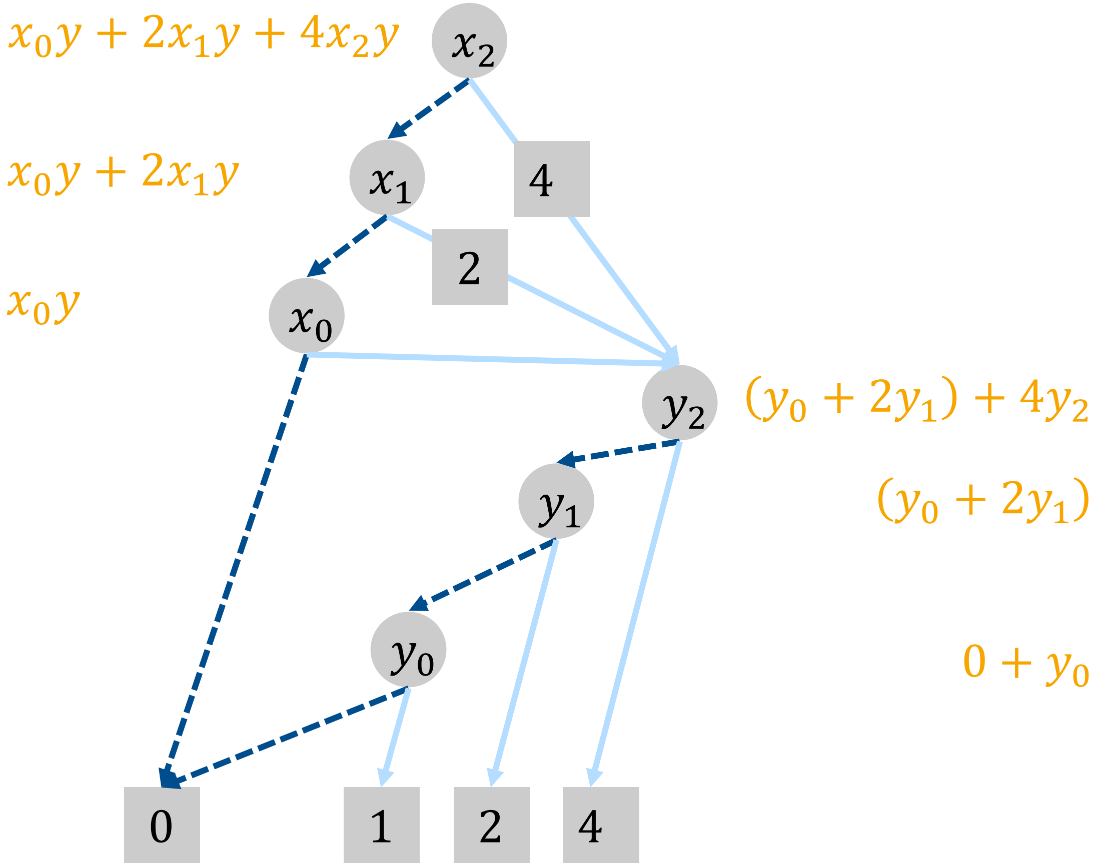
Beispiel

$$
\begin{aligned}
f(x,y,z) &= 8 - 20x_3 + 2x_2 + 4x_2x_3 + 12x_1 + 24x_1x_3 + 15x_1x_2 \\
&= (8 - 20x_3 + 2x_2 + 4x_2x_3) + x_1(12 + 24x_3 + 15x_2) \\
&= \big(\underline{(8 - 20x_3) + x_2(2 + 4x_3)}\big) + x_1\big(\underline{(12 + 24x_3) + 15x_2}\big)
\end{aligned}
$$



Word-Level DD der Multiplikation

- Multiplikation lässt sich durch ein Word-Level DD mit $2n$ inneren Knoten beschreiben
- Zerlegungsvorschrift (bei $l(v) = x_i$):
  $c_{low(v)} \cdot f_{low(v)} + x_i \cdot c_{high(v)} \cdot f_{high(v)}$
- Beispiel stellt die 3-bit Multiplikation dar



Ablauf des Verfikationsprozesses

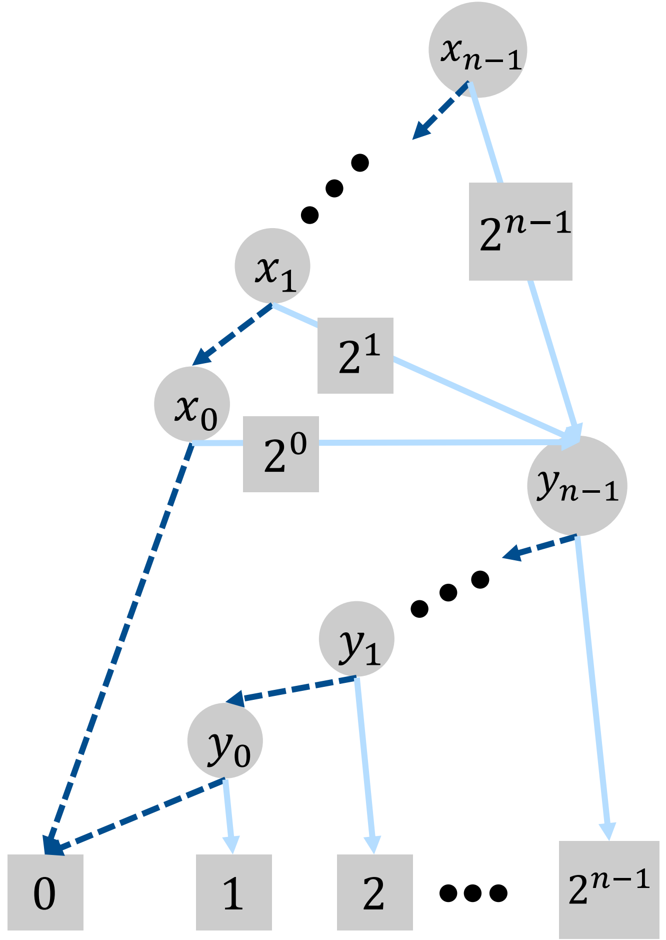
Beispiel: n-Bit Multiplikation

Beschreibung eines effizienten Multiplizieres

Compiler

*BMD der Realisierung

$\stackrel{?}{=}$



Goldene Spezifikation

Fazit

- **DDs erlauben „häufig" speicherplatz-effiziente Darstellung**
  - Boolesche Funktionen mit vielen Ein- und Ausgängen darstellbar
  - Logiksynthese und Verifikation großer Funktionen potentiell möglich

- **einige in der Praxis vorkommende Funktionen nur mit exponentiellem Aufwand darstellbar**
  - Zum Beispiel: BDDs für die Multiplikation

→ kein Allheilmittel, aber besser als bisher bekannte kanonische Darstellungen

Überblick

Teil 1: Der Rechneraufbau (Kapitel 2-5)	Teil 2: Der Funktionalitätsaufbau (Kapitel 6-12)

• Rechner im Überblick
• Pipelining
• Speicher
• Parallelverarbeitung
	• Kodierung
	• Grundbegriffe, Boolesche Funktionen
	• **Darstellungsmöglichkeiten**
	  • Zweistufige Logiksynthese
	  • **Binäre Entscheidungsdiagramme**
	• Schaltkreise, Synthese, spezielle Schaltkreise