Hardwarebeschreibung

Digital-Design

Prof. Dr.-Ing. habil. Jürgen Kampe

Binary Decision Diagram (BDD)

1. Beschreibung kombinatorischer Systeme.

- Bestimmen Sie die ROBDDs für folgende Funktionen:
  - $x_2 + \overline{x_1} x_0$
  - $x_2 \overline{x_1} + x_2 x_1 + \overline{x_1} x_0 \overline{x_2}$

2. Überprüfen Sie beide Funktionen auf Gleichheit.

Motivation

Model checking

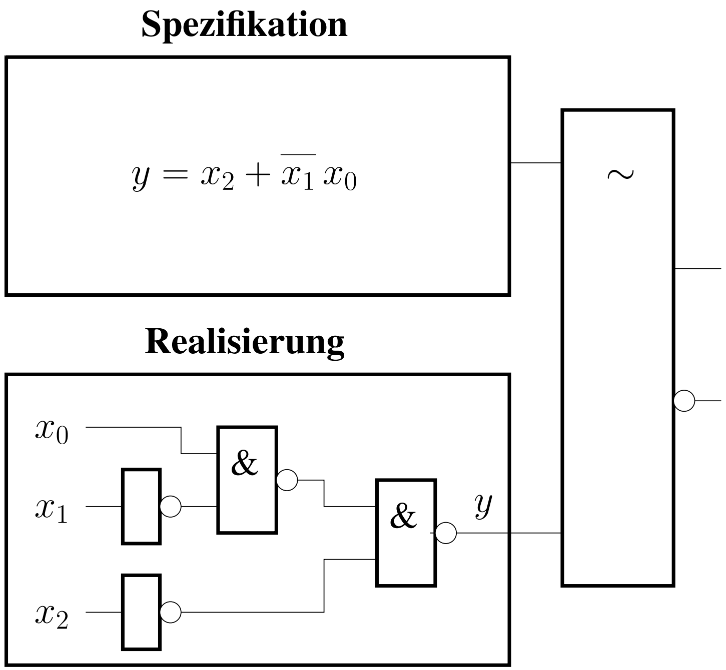
Überprüfung einer Spezifikation auf gewünschte Eigenschaften

Gibt es eine Eingangsbelegung
(hier: Kombination gedrückter Tasten),
bei der die Modell-Funktion $y = f(\underline{x}) = 1$ wird
(hier: ohne Geld Cola ausgeworfen wird)?

Motivation

Equivalence checking

Beweis der korrekten, spezifikationskonformen Schaltungsrealisierung



Wann sind zwei Funktionen identisch?

- Umformung der *Boole*'schen Gleichungen
- Wahrheitstafel, *Karnaugh*-Plan
- kanonische Normalformen
- ??

Bei komplexen Systemen (viele Eingangsvariable) sind viele Eingangsbelegungen zu betrachten:

z. B. $k = 30$ Variablen ergeben $e = 2^k = 1\,073\,741\,824$ Eingangsbelegungen!

- Suche nach einer kompakten, kanonischen Darstellung, die zur Beschreibung komplexer Systeme verwendet werden kann.

Kanonische Darstellungsformen (I)

1. Wertetabelle:

- $e = 2^k$ Zeilen erforderlich.

- Kombinatorische Funktionen mit vielen Eingangsvariablen ergeben sehr große Wertetabellen:
  Bei $k = 30$ Variablen sind $2^{30} = 1\,073\,741\,824$ (1 G) Tabelleneinträge erforderlich!

Wertetabelle:

$$y = f(\underline{x}) = x_2 + \overline{x_1} x_0, \quad k = 3$$

ϵ	x_2	x_1	x_0	y
0	0	0	0	?
1	0	0	1	?
2	0	1	0	?
3	0	1	1	?
4	1	0	0	?
5	1	0	1	?
6	1	1	0	?
7	1	1	1	?

Kanonische Darstellungsformen (II)

2. KDNF/KKNF:

$$\begin{aligned} y &= m_1 + m_4 + m_5 + m_6 + m_7 \\ &= M_0 \cdot M_2 \cdot M_3 \end{aligned}$$

• Die Normalform ist *kanonisch* (Regel-konform), wenn sie ausschließlich aus Mintermen oder aus Maxtermen besteht — wenn in jedem konjunktiven (DNF) oder disjunktiven (KNF) Term alle Eingangsvariable geordnet enthalten sind.

$\longrightarrow$ Erweiterung von $f(\underline{x}) = x_2 + \overline{x_1} x_0$ zur KDNF.

Das Ziel besteht in einer kompakten, standardisierten Darstellung, die für identische Funktionen übereinstimmt — es soll zu jeder Funktion genau eine kanonische Form existieren.

Kanonizität: Ein Darstellungstyp heißt *kanonisch*, wenn jede Funktion $f(\underline{x})$ genau eine Darstellung in dieser Form besitzt.

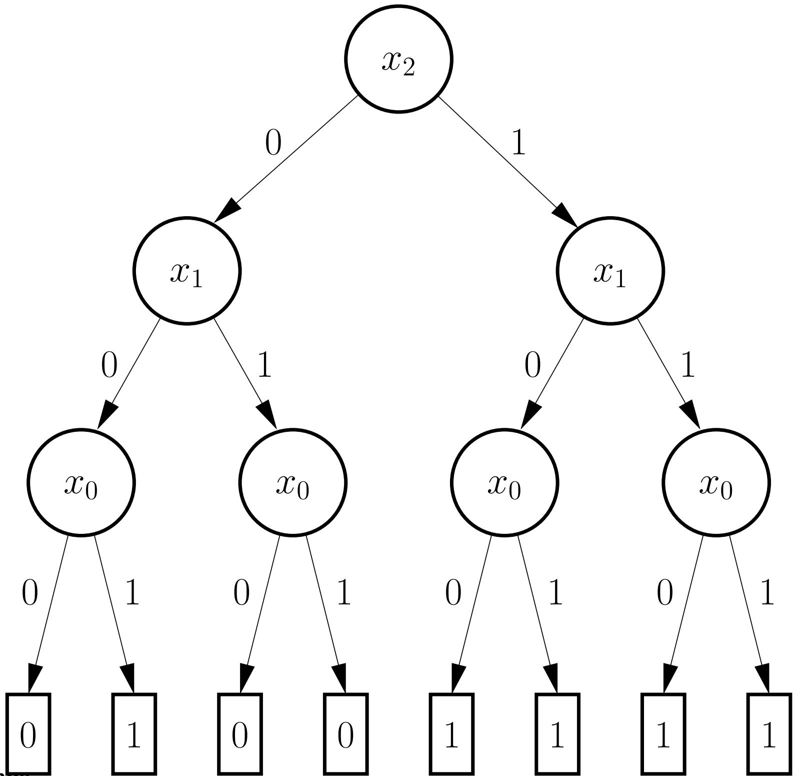
Kanonische Darstellungsformen (III)

3. binärer Entscheidungsbaum (*binary decision tree*, BDT):

Der binäre Entscheidungsbaum ist ein gerichteter, schleifenloser Graph

$$\boldsymbol{G} = \{\boldsymbol{V}, \boldsymbol{E}\}$$

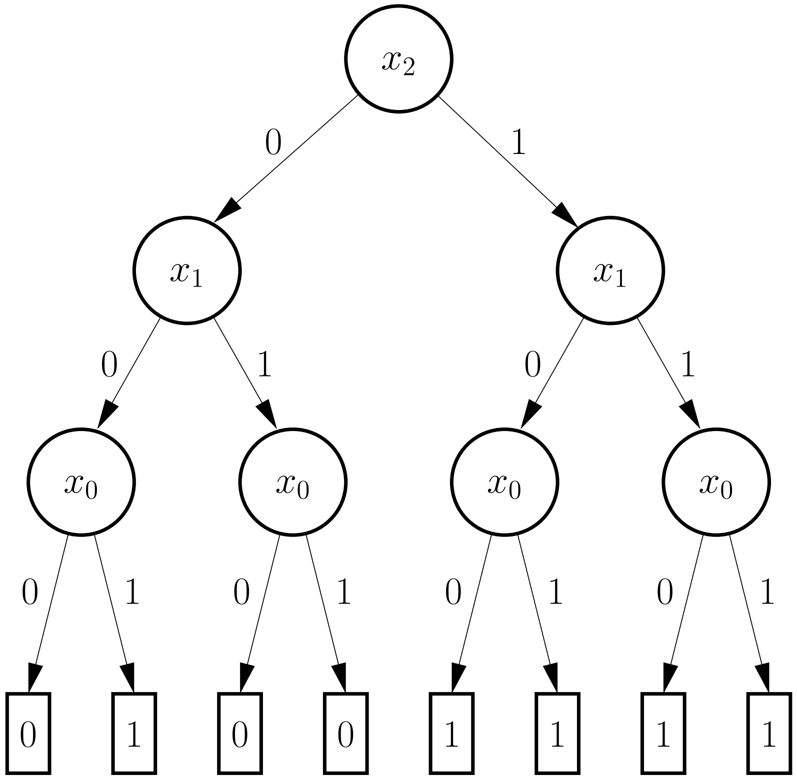
mit der Knotenmenge (*vertices*) $\boldsymbol{V}$ und der Kantenmenge (*edges*) $\boldsymbol{E}$.



- Ausgehend von einem Wurzelknoten (*root*) wird in jeder Entscheidungsebene eine Variable bewertet,
- jede Eingangsbelegung $\underline{x}_\epsilon$ wird durch einen Pfad von der Wurzel zu einem Blattknoten repräsentiert,
- der Blattknoten enthält den Funktionswert y_ϵ für diese Eingangsbelegung.

Binary Decision Tree (BDT)

Definition: Ein binärer Entscheidungsbaum in k Variablen ist ein Binärbaum:



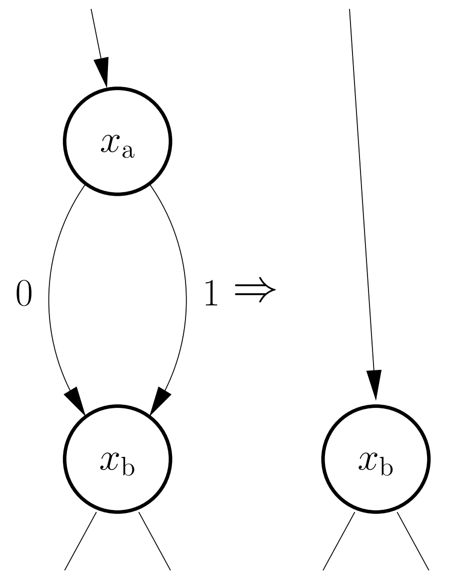
- dessen $2^k - 1$ Wurzel- und innere Knoten mit den Variablen x_κ markiert sind,
- jeder dieser Knoten 2 abgehende Kanten mit den Gewichten „0“ und „1“ besitzt, die den Belegungen der Variablen mit den Werten $x_\kappa \in \{0, 1\}$ entsprechen,
- 2^k Blattknoten mit den Funktionswerten „0“ und „1“ enthält.

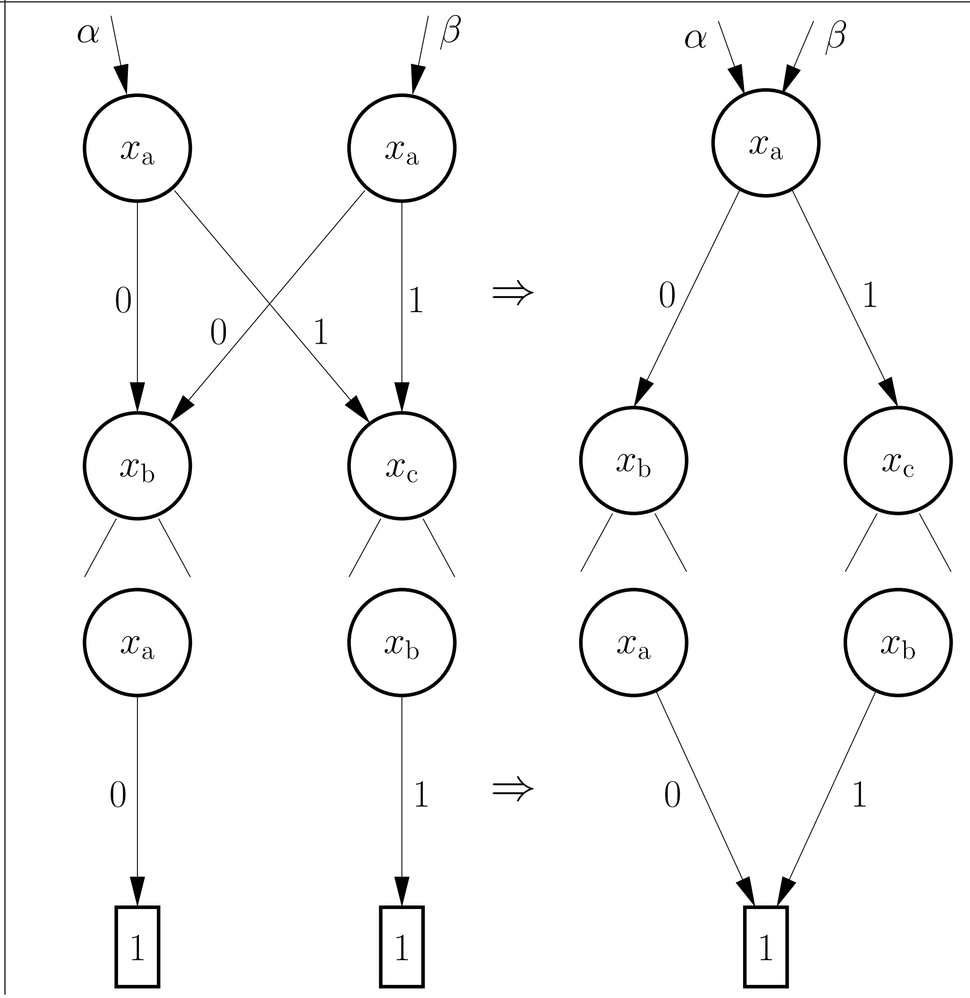
- Der Entscheidungsbaum enthält identische (isomorphe) und redundante Knoten
  $\hookrightarrow$ *Reduced Binary Decision Diagram* (RBDD)

Binary Decision Diagram (BDD)

Reduzierungsregeln

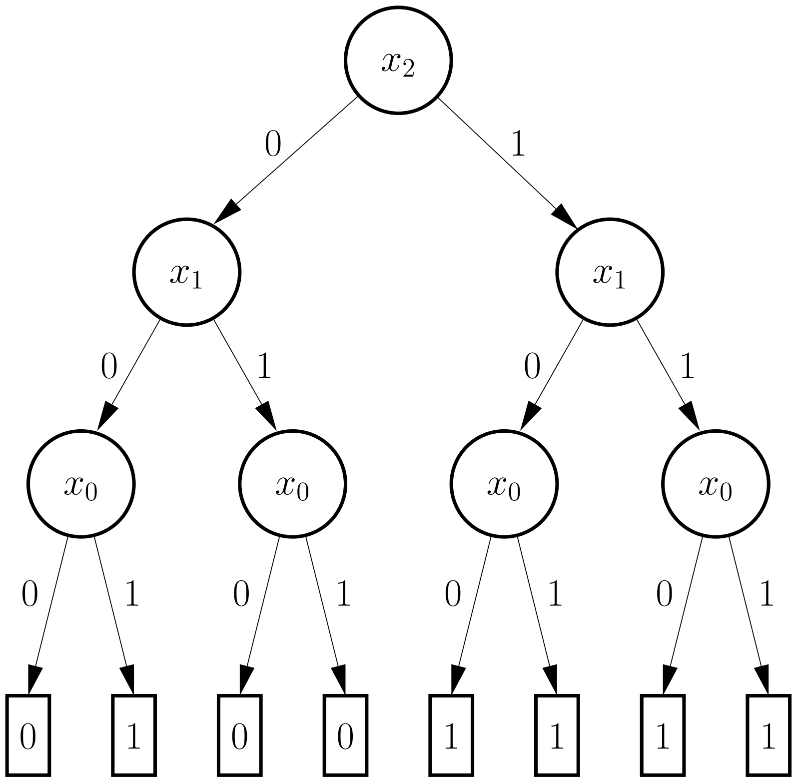
Eliminierung redundanter Knoten	Verschmelzung isomorpher (identischer) Knoten





Anwendung der Reduzierungsregeln:

$$y = f(\underline{x}) = x_2 + \overline{x_1} x_0$$



$\Rightarrow$ **?**

If-Then-Else Normalform

If-then-else Operator: Jede Entscheidungsebene für eine Eingangsvariable x_κ kann durch den *if-then-else Operator* $x_\kappa \rightarrow y_1, y_0$:

$$\textit{if } x_\kappa \textit{ then } y_1 \textit{ else } y_0$$

dargestellt werden:

$$x_\kappa \rightarrow y_1, y_0 \quad \equiv \quad x_\kappa \cdot y_1 + \overline{x_\kappa} \cdot y_0$$

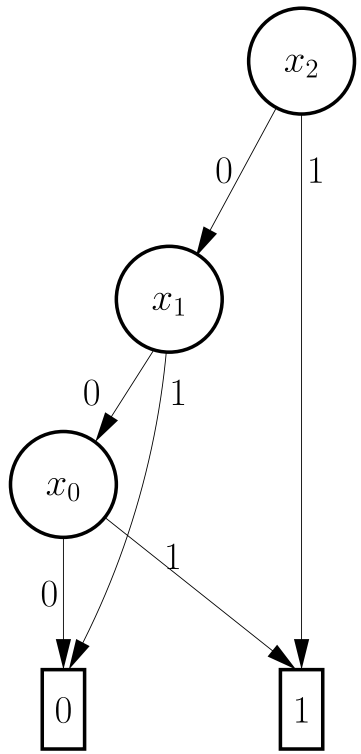
Geoff Draper: "A Fork in the Path"

Die ***If-Then-Else Normalform*** **(INF)** wird ausschließlich durch if-then-else Operatoren, deren Testausdrücke aus den Variablen x_κ bestehen, und durch die Konstanten 0 und 1 gebildet.

If-Then-Else Normalform:

Im Beispiel $f(\underline{x}) = x_2 + \overline{x_1} x_0$ ergibt die INF:

$$f(\underline{x}) = x_2 \rightarrow 1, (x_1 \rightarrow 0, (x_0 \rightarrow 1, 0))$$



Schaltungsinterpretation des BDD

Aus dem ROBDD kann die Funktion wieder ausgelesen werden, allerdings ist das Ergebnis nicht minimal!

DNF: Jeder Pfad vom Wurzelknoten zu einem „1“-Blattknoten ergibt einen Term der DNF, wobei eine Variable x_κ

- nicht negiert einzusetzen ist, wenn das Gewicht der abgehenden Kante 1 $\qquad x_\kappa \xrightarrow{1} \quad x_\kappa$
- negiert einzusetzen ist, wenn das Gewicht der abgehenden Kante 0 $\qquad x_\kappa \xrightarrow{0} \quad \overline{x_\kappa}$

beträgt.

KNF: Jeder Pfad vom Wurzelknoten zu einem „0“-Blattknoten ergibt einen Term der KNF, wobei eine Variable x_κ

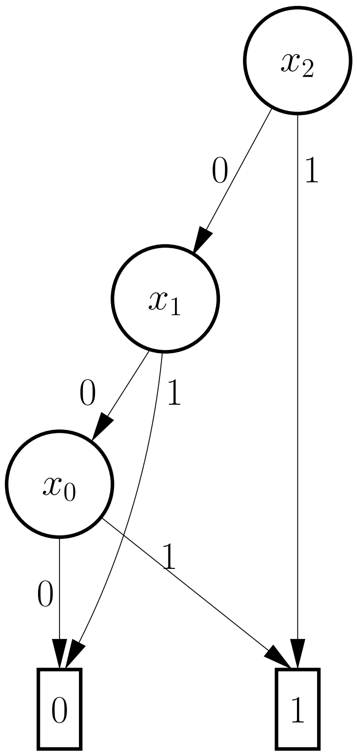
- nicht negiert einzusetzen ist, wenn das Gewicht der abgehenden Kante 0 $\qquad x_\kappa \xrightarrow{0} \quad x_\kappa$
- negiert einzusetzen ist, wenn das Gewicht der abgehenden Kante 1 $\qquad x_\kappa \xrightarrow{1} \quad \overline{x_\kappa}$

beträgt.

Entsprechend kann auf diese Weise aus dem BDT die KDNF bzw. die KKNF ausgelesen werden.

Schaltungsinterpretation:

$$y = f(\underline{x}) = x_2 + \overline{x_1} x_0$$

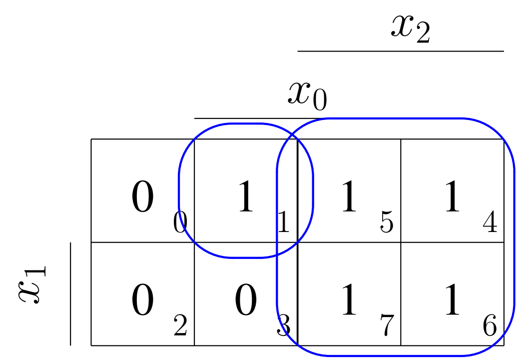


DNF:

Pfad	Term
$x_2 \xrightarrow{1} \boxed{1}$	$x_2 +$
$x_2 \xrightarrow{0} x_1 \xrightarrow{0} x_0 \xrightarrow{1} \boxed{1}$	$\overline{x_2}\,\overline{x_1}\,x_0$

Ergebnis: $y = x_2 + \overline{x_2}\,\overline{x_1}\,x_0$

minimal: $y = x_2 + \overline{x_1}\,x_0$

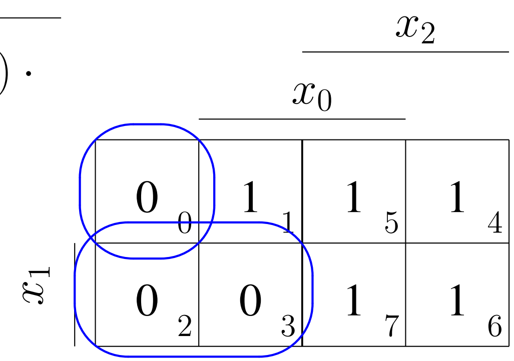


KNF:

Pfad	Term
$x_2 \xrightarrow{0} x_1 \xrightarrow{0} x_0 \xrightarrow{0} \boxed{0}$	$(x_2 + x_1 + x_0)\,\cdot$
$x_2 \xrightarrow{0} x_1 \xrightarrow{1} \boxed{0}$	$(x_2 + \overline{x_1})$

Ergebnis: $y = (x_2 + x_1 + x_0) \cdot (x_2 + \overline{x_1})$

minimal: $y = (x_2 + x_0) \cdot (x_2 + \overline{x_1})$



Schaltungsinterpretation des BDD
BDD-Teilung

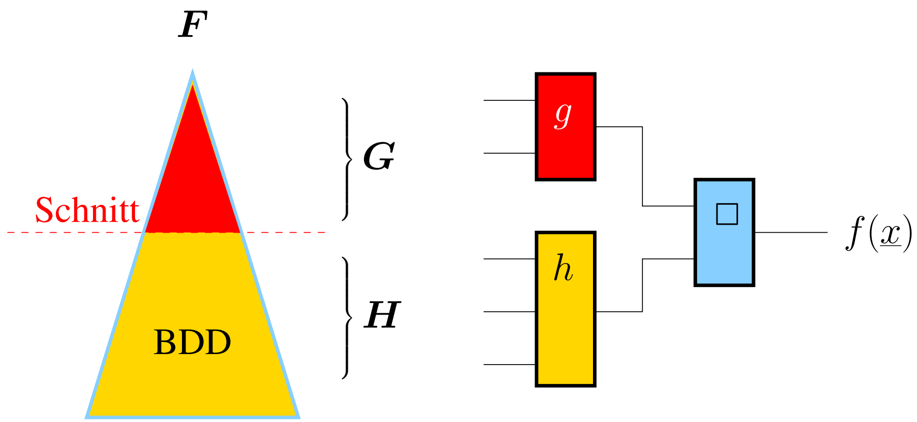
Das Problem der Wurzelknoten-Präsenz in jedem Term kann mit dem BDD-Teilungsverfahren gelöst werden:

Allgemein: Für eine Reihe von *Boole*'schen Operationen $\square$:

$$f() = g() \square h()$$

kann eine Teilung vorgenommen werden:

- Anwendung von Schnitten, wobei der Wurzel- und die Blattknoten auf unterschiedlichen Seiten des Schnittes liegen.

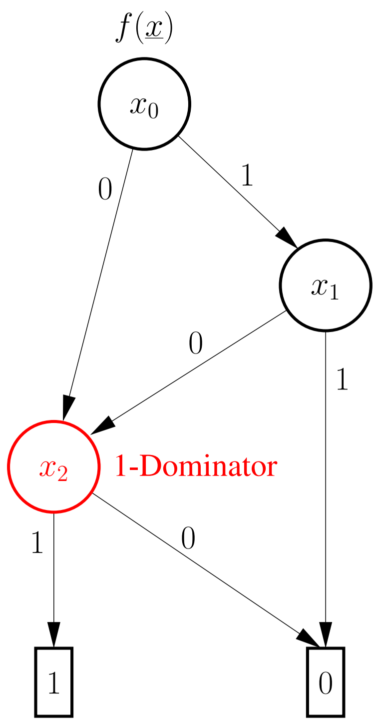


Wo können die Schnitte sinnvoll angewendet werden, so dass bei der Teilung kein „Rest“ entsteht?

BDD-Teilung
Schnittebene „1-Dominanz“

Auswahl eines 1-Dominators als Schnittebene.

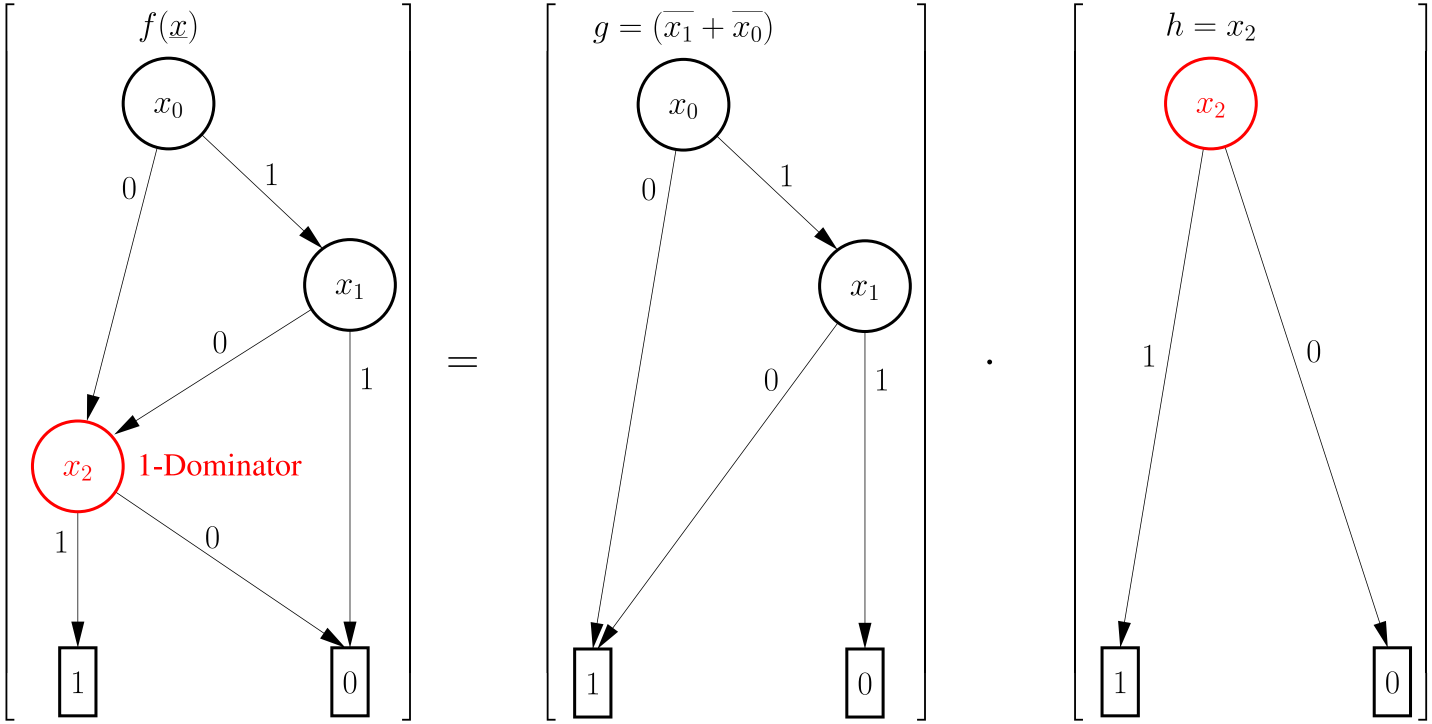
Definition: Ein **1-Dominator** ist ein Knoten, der auf jedem möglichen Pfad vom Wurzelknoten zum 1-Blattknoten liegt. Für eine geeignete Beispielfunktion:



KNF: $f(\underline{x}) = \mathbf{?}$

BDD-Teilung

Schnittebene „1-Dominanz“



$$f(\underline{x}) = g \cdot h = (\overline{x_1} + \overline{x_0}) \cdot x_2$$

$g()$:

		x_0	x_0, x_2	x_2
	1	1	1	1
x_1	1	0	0	1

$\cdot$ $h()$:

		x_0	x_0, x_2	x_2
	0	0	1	1
x_1	0	0	1	1

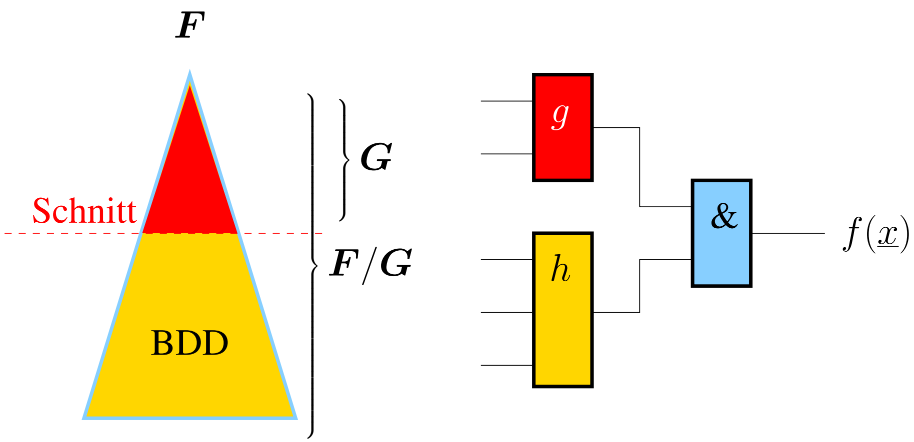
$=$ $f(\underline{x})$:

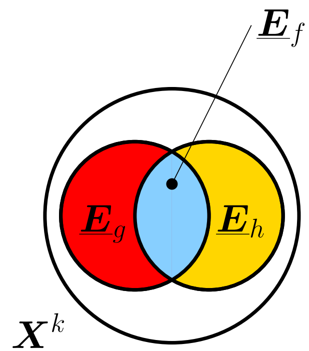
		x_0	x_0, x_2	x_2
	0	0	1	1
x_1	0	0	0	1

- Der 1-Dominator repräsentiert eine **konjunktive** Zerlegung $f(\underline{x}) = g \cdot h$.

BDD-Teilung

Schnittebene „1-Dominanz“





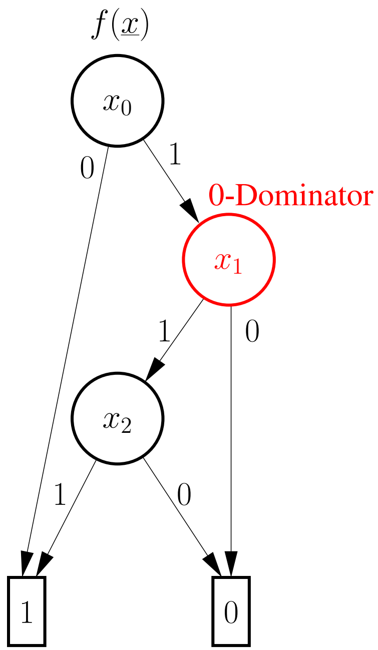
- Der obere Teil bildet den Divisor g;
- der untere Teil bildet den Quotient h.

$$\begin{aligned}\boldsymbol{E}_g &= \{0, 1, 2, 4, 5, 6\}\\ \boldsymbol{E}_h &= \{4, 5, 6, 7\}\\ \boldsymbol{E}_f &= \boldsymbol{E}_g \cap \boldsymbol{E}_h = \{4, 5, 6\}\end{aligned}$$

BDD-Teilung
Schnittebene „0-Dominanz“

Auswahl eines 0-Dominators als Schnittebene.

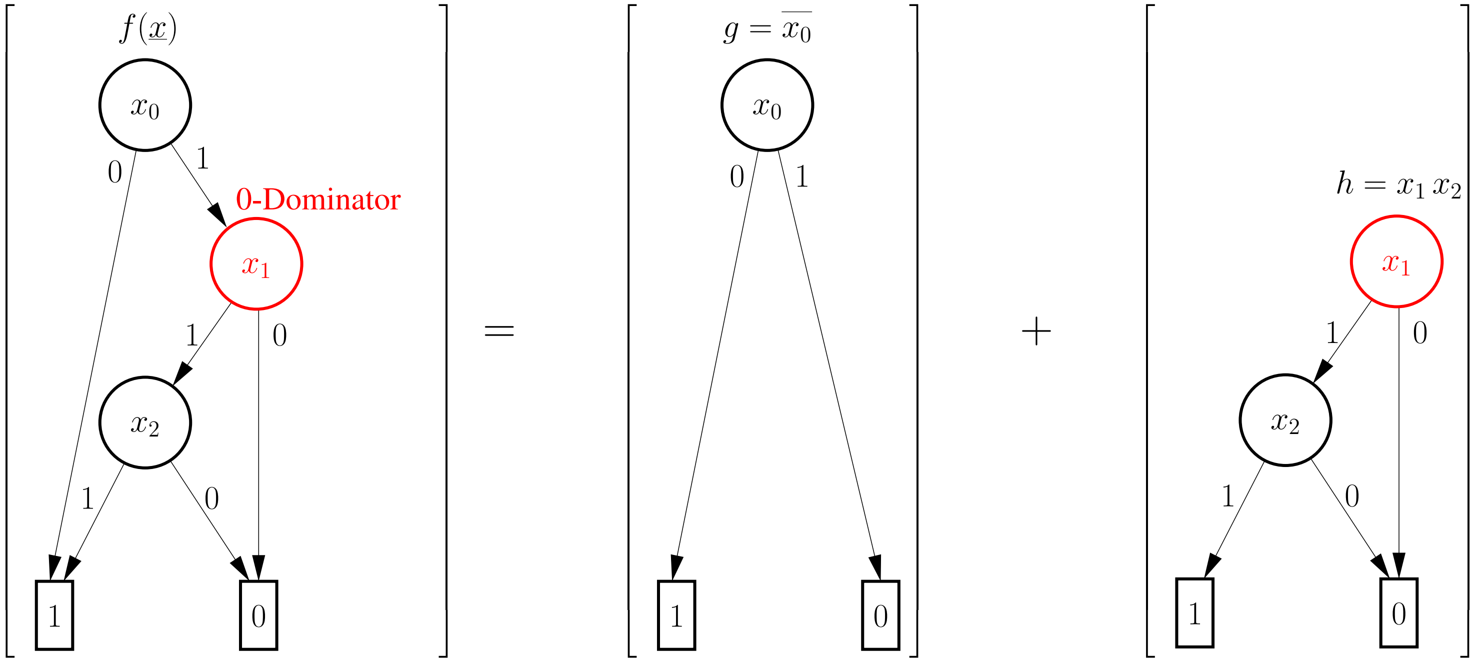
Definition: Ein **0-Dominator** ist ein Knoten, der auf jedem möglichen Pfad vom Wurzelknoten zum 0-Blattknoten liegt. Für eine geeignete Beispielfunktion:



DNF: $f(\underline{x}) = \mathbf{?}$

BDD-Teilung

Schnittebene „0-Dominanz“



$= \overline{x_0} + x_1\,x_2$

$g()$:

		x_0	x_0	
			x_2	x_2
	1	0	0	1
x_1	1	0	0	1

+

$h()$:

		x_0	x_0	
			x_2	x_2
	0	0	0	0
x_1	0	0	1	1

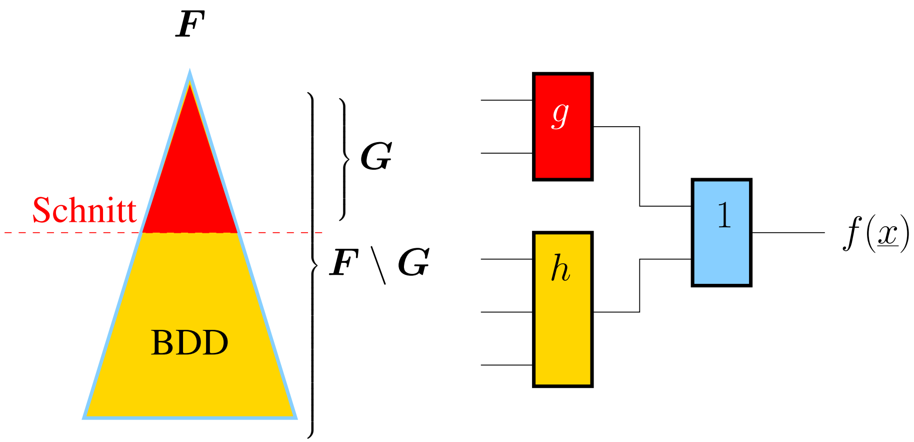
=

$f(\underline{x})$:

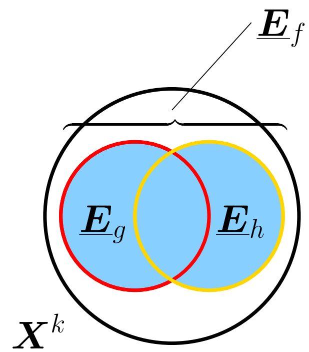
		x_0	x_0	
			x_2	x_2
	1	0	0	1
x_1	1	0	1	1

- Der 0-Dominator repräsentiert eine **disjunktive** Zerlegung $f(\underline{x}) = g + h$.

BDD-Teilung
Schnittebene „0-Dominanz“

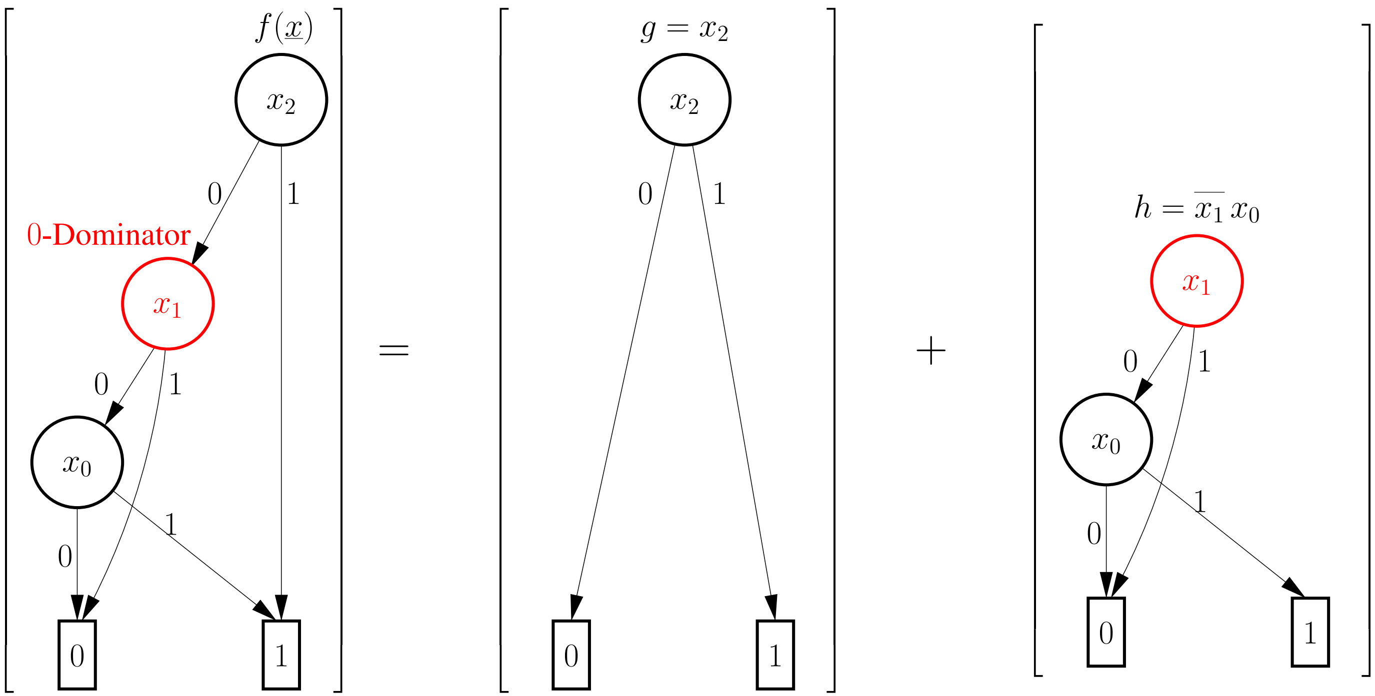


- Der obere Teil bildet den Subtrahend g;
- der untere Teil bildet die Differenz h.



$$\begin{aligned} \boldsymbol{E}_g &= \{0, 2, 4, 6\} \\ \boldsymbol{E}_h &= \{6, 7\} \\ \boldsymbol{E}_f &= \boldsymbol{E}_g \cup \boldsymbol{E}_h = \{0, 2, 4, 6, 7\} \end{aligned}$$

$y = x_2 + \overline{x_1} x_0, \quad k = 3$:



$= x_2 + \overline{x_1}\, x_0$

$g()$:

		x_0	x_0, x_2	x_2
			1	1
x_1			1	1

$+$

$h()$:

		x_0	x_0, x_2	x_2
		1	1	
x_1				

$=$

$f(\underline{x})$:

		x_0	x_0, x_2	x_2
		1	1	1
x_1			1	1

Konstruktion des *Reduced Binary Decision Diagram* (RBDD)

Konstruktion:

- BDT aus Wertetabelle und Anwendung der Reduktionsregeln
- BDT aus KDNF/KKNF und Anwendung der Reduktionsregeln
- BDD oder RBDD aus *Shannon*-Theorem, ggf. Anwendung der Reduktionsregeln

Die Konstruktion eines BDD basierend auf dem *Shannon*'schen Expansionstheorem:

$$f(x_0, \ldots, x_i, \ldots, x_\mathrm{n}) = x_i \; f(x_0, \ldots, 1, \ldots, x_\mathrm{n}) + \overline{x_i} \; f(x_0, \ldots, 0, \ldots, x_\mathrm{n})$$

- minimierte Restausdrücke, Überspringen nicht vorhandener Variable in den Restausdrücken und Erkennung mehrfach identisch auftretender Restausdrücke $\Rightarrow$ reduzierter BDD (RBDD)
- vollständige Zerlegung $\Rightarrow$ vollständiger BDD/BDT

Konstruktion des ROBDD mit dem *Shannon*'schen Expansionstheorem:

$$\begin{aligned} f(x_0, \ldots, x_i, \ldots, x_\mathrm{n}) &= \overline{x_i}\ f(x_0, \ldots, 0, \ldots, x_\mathrm{n}) + x_i\ f(x_0, \ldots, 1, \ldots, x_\mathrm{n}) \\ &= \overline{x_i}\ g() \qquad\qquad\qquad\quad + x_i\ h() \end{aligned}$$

$y = f(\underline{x}) = x_2 + \overline{x_1} x_0$ in der Reihenfolge x_0 — x_1 — x_2:

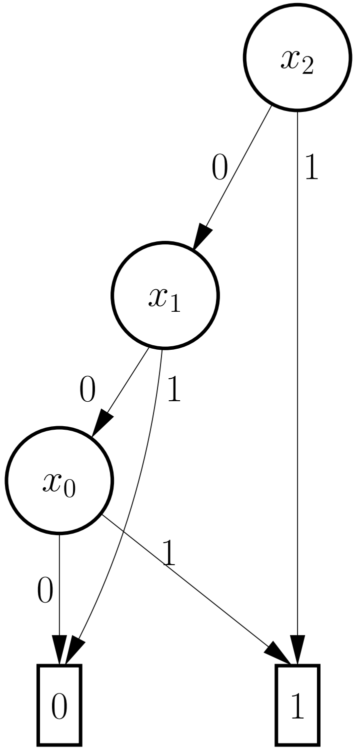
$$y = \textbf{?}$$

Reduced Ordered Binary Decision Diagram (ROBDD) (I)

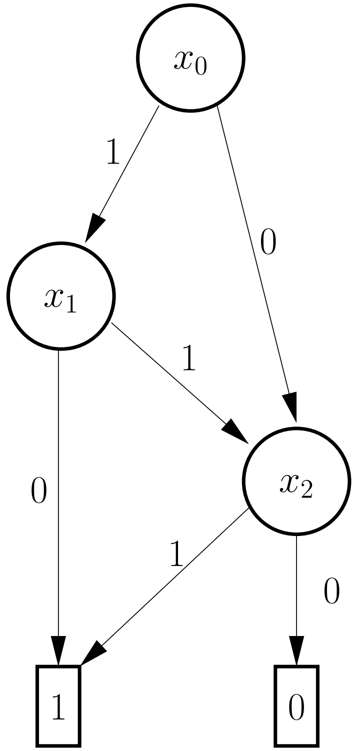
- Es ergeben sich unterschiedliche Graphen für die gleiche Funktion, wenn die Variablenreihenfolge verändert wird.

$y = f(\underline{x}) = x_2 + \overline{x_1} x_0$:

Reihenfolge $x_2 — x_1 — x_0$



Reihenfolge $x_0 — x_1 — x_2$



Reduced Ordered Binary Decision Diagram (ROBDD) (II)

- Als *kanonische* Darstellung muss eine Variablenreihenfolge festgelegt werden (*ordering*)
  $\hookrightarrow$ *Reduced Ordered Binary Decision Diagram* (ROBDD)
  - ROBDDs sind eine *kanonische* Darstellung für Schaltfunktionen,
  - werden zur Spezifikation und zur Verifikation komplexer kombinatorischer Funktionen verwendet.

Zwei Schaltfunktionen sind identisch, wenn ihre ROBDD bei identischer Variablenordnung zueinander isomorph sind.

Überprüfung der Gleichheit:

Ist die Funktion $y = f(\underline{x}) = x_2 + \overline{x_1}x_0$ identisch zu
$y_2 = f(\underline{x}) = x_2\overline{x_1} + x_2x_1 + \overline{x_1}x_0\overline{x_2}$?

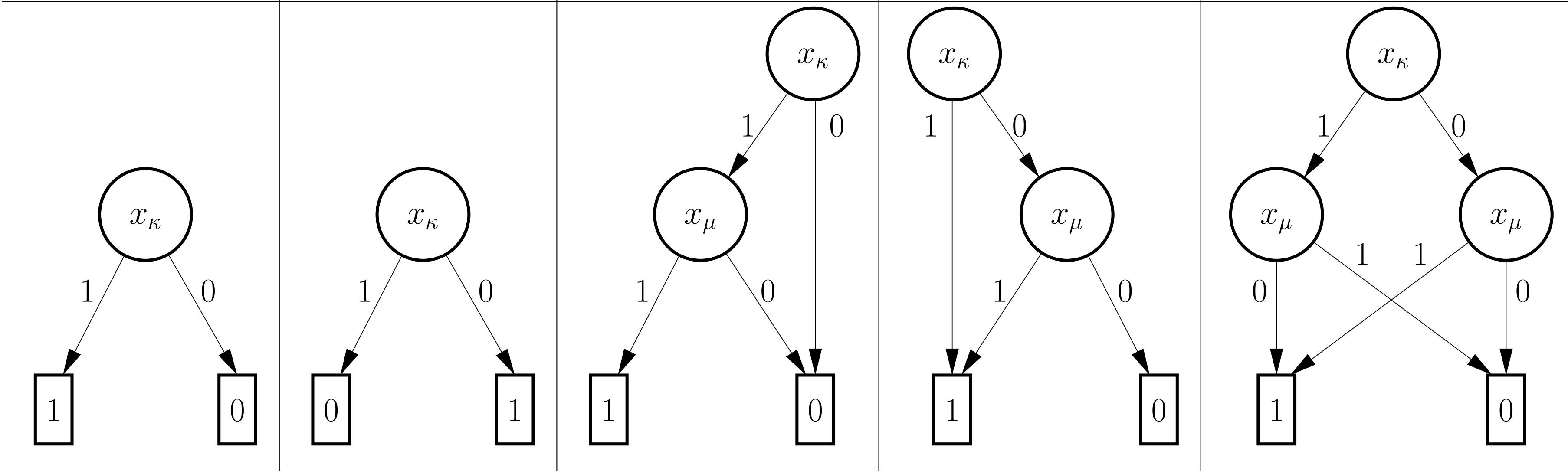
Entwicklung von $y_2 = f(\underline{x}) = x_2\overline{x_1} + x_2x_1 + \overline{x_1}x_0\overline{x_2}$ in der gleichen Reihenfolge x_0 — x_1 — x_2:

$y_2 =$ **?**

$\hookrightarrow$ identisch?

Beispiele elementarer BDDs

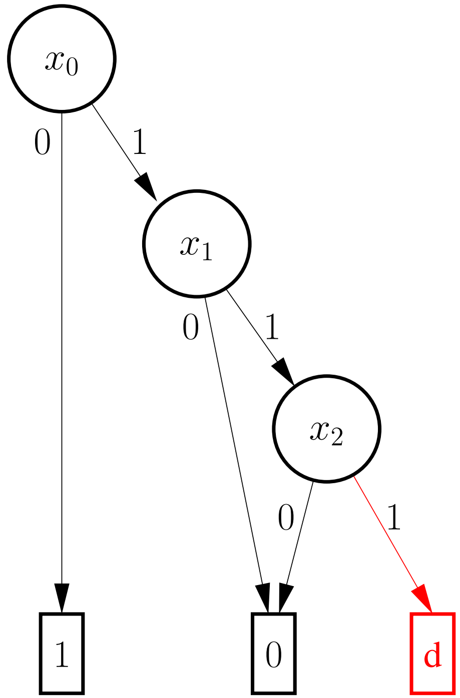
x_κ	$\overline{x_\kappa}$	$x_\kappa\, x_\mu$	$x_\kappa + x_\mu$	$x_\kappa \not\sim x_\mu$



BDDs mit *don't care*

Entsprechend der Überlegung, dass jeder Pfad im BDD vom Wurzelknoten zu einem Blattknoten einer möglichen Eingangsbelegung entspricht, können *don't care* als Pfade zu einem dritten Blattknoten eingetragen werden (dreiwertige BDDs).

ϵ	x_2	x_1	x_0	y
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	d



Verzeichnis der Präsentationen