Einführung in die Technische Informatik

Prof. Dr.-Ing. Stefan Kowalewski

WS 22/23

Kapitel 2: Darstellung Boolescher Funktionen

Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

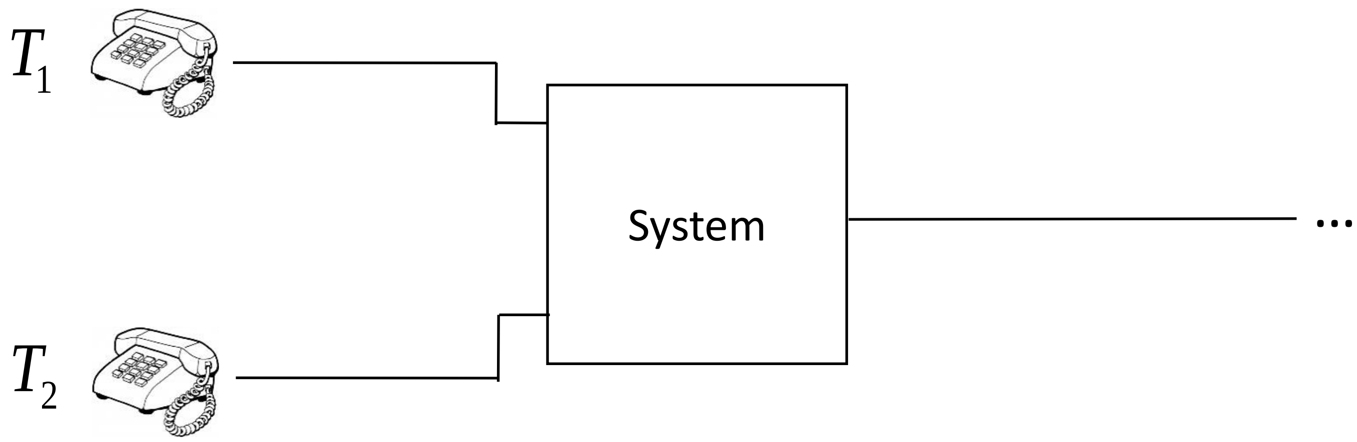
Abschnitt 2.1

Boolesche Algebra

- Gesetze einer Booleschen Algebra
- Anwendung einer Booleschen Algebra

Motivation

- 2 Zustände reichen aus, um Informationen zu speichern
- Beispiel:



- Aufgabe des Systems:
  - „Da nur eine Leitung genutzt wird, darf nur eines der beiden Telefone zu einem Zeitpunkt verwendet werden.“
- Formal ausgedrückt:

Boolesche Algebra

- **Boolesche Algebra** als formale Grundlage in der Schaltungstechnik und der Computerhardware
- In der Booleschen Algebra gibt es genau 2 Werte: 0 (false) und 1 (true)
- Unäre Verknüpfung: ¬
- Binäre Verknüpfungen: ∨, ∧

x	¬x
0	**1**
1	**0**

x	y	x∨y
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

x	y	x∧y
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

Gesetze in der Booleschen Algebra

(a) Kommutativgesetze:

$x \wedge y = y \wedge x$ $\qquad$ $x \vee y = y \vee x$

(b) Assoziativgesetze:

$(x \wedge y) \wedge z = x \wedge (y \wedge z)$ $\qquad$ $(x \vee y) \vee z = x \vee (y \vee z)$

(c) Distributivgesetze:

$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ $\qquad$ $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$

(d) Absorption:

$x \wedge (x \vee y) = x$ $\qquad$ $x \vee (x \wedge y) = x$

(e) Idempotenz:

$x \wedge x = x$ $\qquad$ $x \vee x = x$

(f) Nullelement: $\qquad$ **Einselement:**

$x \wedge 0 = 0$ $\qquad$ $x \vee 1 = 1$

Gesetze in der Booleschen Algebra

(g) Eindeutiges Komplement:

$(x \vee y = 1 \text{ und } x \wedge y = 0) \Leftrightarrow (x = \neg y)$

(h) Involution:

$\neg(\neg x) = x$

(h) Konstanten:

$\neg 0 = 1$ $\qquad$ $\neg 1 = 0$

(j) De Morgansche Regeln:

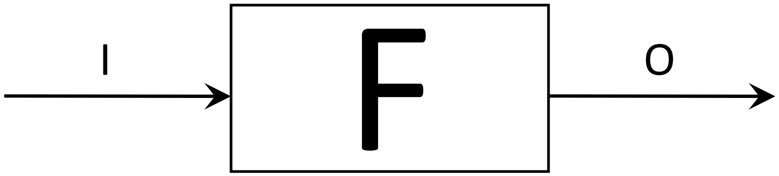
$\neg(x \vee y) = \neg x \wedge \neg y$ $\qquad$ $\neg(x \wedge y) = \neg x \vee \neg y$

Abschnitt 2.2

Boolesche Funktionen

- Schaltfunktionen
- 1-stellige Boolesche Funktionen
- 2-stellige Boolesche Funktionen
- Darstellung Boolescher Funktionen

Schaltfunktionen



Definition: Seien $n, m \in N,\ n, m \geq 1$. Dann heißt eine Funktion $F: B^n \to B^m$ **Schaltfunktion**

Beispiele:
- Addition von zwei 16-stelligen Dualzahlen
- Multiplikation von zwei 16-stelligen Dualzahlen
- Sortieren von 30 16-stelligen Dualzahlen
- Primzahltest einer 16-stelligen Dualzahl

Boolesche Funktionen

Eine Schaltfunktion $f: B^n \rightarrow B^1$ heißt (*n*-stellige) **Boolesche Funktion**

Zusammenhang zu Schaltfunktionen:

Sei $F: B^n \rightarrow B^m$ mit $F(x_{n-1}, \dots, x_1, x_0) = (y_{m-1}, \dots, y_1, y_0)$

Setzt man für jedes $i \in \{m-1, \dots, 0\}$

$$f_i: B^n \rightarrow B$$

definiert durch

$$f_i(x_{n-1}, \dots, x_1, x_0) = y_i$$

so ist *F* wie folgt darstellbar:

$$F(x_{n-1}, \dots, x_1, x_0) = (f_{m-1}(x_{n-1}, \dots, x_1, x_0), f_{m-2}(x_{n-1}, \dots, x_1, x_0), \dots, f_0(x_{n-1}, \dots, x_1, x_0))$$

für alle $x_{n-1}, \dots, x_0 \in B$

Beispiel für Zusammenhang Schaltfunktion – Boolesche Funktion

1-stellige Boolesche Funktion

$B \rightarrow B$

x	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	0	1	1
1	0	1	0	1

Es gilt: $f_0(x) = 0,\quad f_1(x) = x,\quad f_2(x) = \overline{x},\quad f_3(x) = 1$

2-stellige Boolesche Funktion

$B^2 \to B$

(1)		$x \cdot \overline{x}$	$x \cdot y$	$x \cdot \overline{y}$	x	$\overline{x} \cdot y$	y	$x \oplus y$	$x + y$
(2)		$\equiv 0$	Min	$>$	x	$<$	y	$\neq$	Max
(3)			$\wedge$	$\nrightarrow$	x	$\nleftarrow$	y	$\nleftrightarrow$	$\vee$
(4)			AND					XOR	OR
x	y	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

Sind das alle 2-stelligen Booleschen Funktionen?

2-stellige Boolesche Funktion

$B^2 \rightarrow B$

(1)		$\overline{x+y}$	$\overline{x \oplus y}$	$\overline{y}$	$x+\overline{y}$	$\overline{x}$	$\overline{x}+y$	$\overline{x \cdot y}$	$x+\overline{x}$
(2)		1-Max	=	1-y	$\geq$	1-x	$\leq$	1-Min	$\equiv 1$
(3)		$\downarrow$	$\leftrightarrow$	¬y	$\leftarrow$	¬x	$\rightarrow$	$\uparrow$	
(4)		NOR	XNOR					NAND	
x	y	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

Beispiel für eine 3-stellige Funktion

Einschlägige Indizes

i	x_2	x_1	x_0	$f(x_2, x_1, x_0)$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Wie viele 3-stellige Boolesche Funktionen gibt es?

Müssen wir die alle in Funktionstabellen definieren, bevor wir sie benutzen können?

Darstellung Boolescher Funktionen

Folgende Darstellungen werden vorgestellt:

- Disjunktive und Konjunktive Normalform (DNF, KNF)
- Directed Acyclic Graph (DAG)
- Ordered Binary Decision Diagram (OBDD)

Abschnitt 2.3

Disjunktive und Konjunktive Normalform

- Minterme
- Darstellungssatz für Boolesche Funktionen
- Folgerung aus dem Darstellungssatz
- Maxterme
- Grundbausteine zur Realisierung Boolescher Funktionen

Minterme

Gegeben: B^3 in dieser Darstellung.

Minterm - eine Anzahl von Literalen (booleschen Variablen wie z.B. x_2), die alle durch ein **UND (∧)** verknüpft sind

i	x_2	x_1	x_0
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

Beispiele für **Minterme:**

$$m_3(x_2, x_1, x_0) = \overline{x_2} \cdot x_1 \cdot x_0$$
$$m_4(x_2, x_1, x_0) = x_2 \cdot \overline{x_1} \cdot \overline{x_0}$$

Darstellungssatz für Boolesche Funktionen

Jede Boolesche Funktion $f: B^n \rightarrow B$ ist eindeutig darstellbar als **Summe der Minterme ihrer einschlägigen Indizes**.

D.h: Ist $I \subseteq \{0, \ldots, 2^n - 1\}$ die Menge der einschlägigen Indizes von f, so gilt

$$f = \sum_{i \in I} m_i$$

und keine andere Minterm-Summe stellt f dar.

Die Summe der Minterme der einschlägigen Indizes wird als **Disjunktive Normalform (DNF)** bezeichnet.

DNF Beispiel

Sei $f: B^3 \rightarrow B$ gegeben durch:

i	x_2	x_1	x_0	$f(x_2, x_1, x_0)$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

DNF:

$$
\begin{aligned}
f(x_2, x_1, x_0) &= m_3 + m_5 + m_7 \\
&= \overline{x_2} x_1 x_0 + x_2 \overline{x_1} x_0 + x_2 x_1 x_0
\end{aligned}
$$

Folgerung aus dem Darstellungssatz

- Alle *n*-stelligen Booleschen Funktionen lassen sich mit den 2-stelligen Funktionen UND und ODER und der 1-stelligen Funktion NICHT darstellen.

- Man sagt:
  Das System $\{\wedge, \vee, \bar{\ }\}$ ist **funktional vollständig**.

Maxterme

Gegeben: B^3 in dieser Darstellung.

i	x_2	x_1	x_0
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

Maxterm - eine Anzahl von Literalen (booleschen Variablen wie z.B. x_2), die alle durch ein **ODER (V)** verknüpft sind

Beispiele für **Maxterme:**

$$M_3(x_2, x_1, x_0) = x_2 + \overline{x_1} + \overline{x_0}$$

$$M_4(x_2, x_1, x_0) = \overline{x_2} + x_1 + x_0$$

Konjunktive Normalform (KNF)

- Analog zu DNF: jede Boolesche Funktion ist eindeutig darstellbar als das Produkt der Maxterme ihrer NICHT-einschlägigen Indizes.
- Es gilt:
  - Sei m_i i-ter Minterm von f
  - Dann heißt $M_i = \overline{m_i}$ i-ter Maxterm von f

Einführung in die Technische Informatik | WS 22/23
Prof. Dr.-Ing. Stefan Kowalewski | Robin Mroß, M.Sc. | Alexander Kruschewsky, M.Sc.
Informatik 11 Embedded Software
RWTH AACHEN UNIVERSITY

KNF Beispiel

Sei $f: B^3 \to B$ gegeben durch:

i	x_2	x_1	x_0	$f(x_2, x_1, x_0)$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

KNF:

$$\begin{aligned} f(x_2, x_1, x_0) &= M_0 \cdot M_1 \cdot M_2 \cdot M_4 \cdot M_6 \\ &= (x_2 + x_1 + x_0) \cdot (x_2 + x_1 + \overline{x_0}) \cdot (x_2 + \overline{x_1} + x_0) \cdot (\overline{x_2} + x_1 + x_0) \cdot (\overline{x_2} + \overline{x_1} + x_0) \end{aligned}$$

Abschnitt 2.4

Funktionale Vollständigkeit

- Funktionale Vollständigkeit von NAND
- Funktionale Nicht-Vollständigkeit von {→,1}

Funktionale Vollständigkeit von NAND

Bekannt: $\{\wedge, \vee, \neg\}$ ist funktional vollständig

Frage: Ist $\{\uparrow\}$ funktional vollständig?

Vorgehen: Stelle ein bekanntes fkt. vollst. System mit $\{\uparrow\}$ dar.

(1)		$\overline{x \cdot y}$
(2)		1-Min
(3)		$\uparrow$
(4)		NAND
x	y	f_{14}
0	0	1
0	1	1
1	0	1
1	1	0

1. Nicht $\{\neg\}$: $\bar{x} = \bar{x} \vee \bar{x} = \overline{\overline{\bar{x} \vee \bar{x}}} = \overline{x \wedge x} = x \uparrow x$

Funktionale Vollständigkeit von NAND

2. Oder $\{\vee\}$: $x \vee y = \overline{\overline{x \vee y}} = \overline{\bar{x} \wedge \bar{y}}$

$$= \bar{x} \uparrow \bar{y} = (x \uparrow x) \uparrow (y \uparrow y)$$

3. Und $\{\wedge\}$: $x \wedge y = \overline{\overline{x \wedge y}} = \overline{x \uparrow y}$

$$= (x \uparrow y) \uparrow (x \uparrow y)$$

→ Funktional Vollständig

(1)		$\overline{x \cdot y}$
(2)		1-Min
(3)		$\uparrow$
(4)		NAND
x	y	f_{14}
0	0	1
0	1	1
1	0	1
1	1	0

Funktionale Nicht-Vollständigkeit von {→,1}

Bekannt: $\{\wedge, \vee, \neg\}$ ist funktional vollständig

Frage: Ist $\{\rightarrow, 1\}$ nicht funktional vollständig?

Vorgehen: Versuche ein bekanntes fkt. vollst. System mit $\{\rightarrow, 1\}$ darzustellen.

Zeige, dass sich eine der Funktionen mit keiner Kombination darstellen lässt.

(1)		$\overline{x} + y$
(2)		$\leq$
(3)		$\rightarrow$
(4)		
x	y	f_{13}
0	0	1
0	1	1
1	0	0
1	1	1

Einführung in die Technische Informatik | WS 22/23
Prof. Dr.-Ing. Stefan Kowalewski | Robin Mroß, M.Sc. | Alexander Kruschewsky, M.Sc.
Informatik 11
Embedded Software
RWTH AACHEN UNIVERSITY

Funktionale Nicht-Vollständigkeit von {→,1}

1. Nicht $\{\neg\}$: $\bar{x} = ?$

Mögliche Kombinationen von x mit $\{\rightarrow, 1\}$:

$$1 \rightarrow 1 = 1$$
$$1 \rightarrow x = x$$
$$x \rightarrow 1 = 1$$
$$x \rightarrow x = 1$$

(1)		$\bar{x} + y$
(2)		$\leq$
(3)		$\rightarrow$
(4)		
x	y	f_{13}
0	0	1
0	1	1
1	0	0
1	1	1

Es lassen sich keine neuen Funktionen und insbesondere kein Nicht $\{\neg\}$ darstellen.

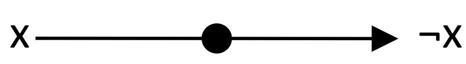
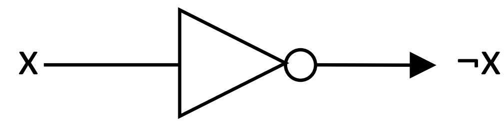
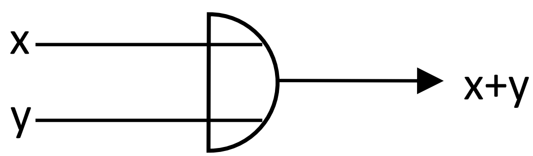
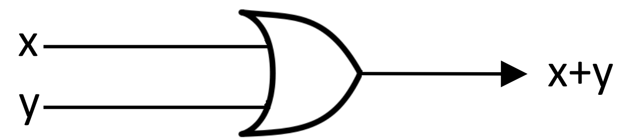
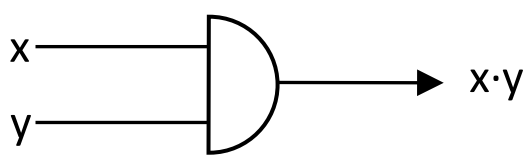
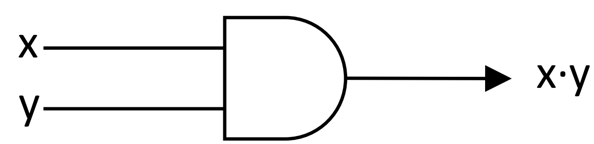
→ Nicht funktional Vollständig

Abschnitt 2.5

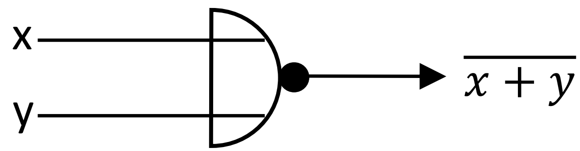
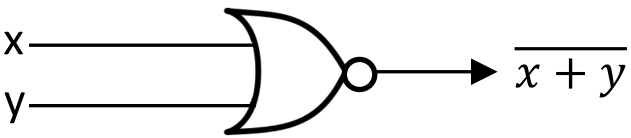
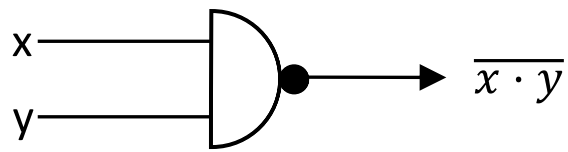
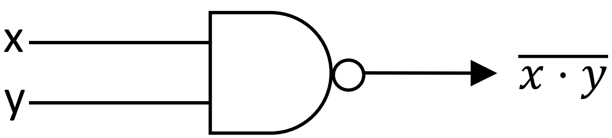
Schaltnetze

- DAG-Darstellung
- Anwendung: Schaltungsabhängige Fehlerdiagnose

Grundbausteine zur Realisierung Boolescher Funktionen

Funktion	Unser Symbol	IEEE-Symbol
Negation (Komplement-Gatter)		
Addition (Oder-Gatter)		
Multiplikation (Und-Gatter)		

Grundbausteine zur Realisierung Boolescher Funktionen

Funktion	Unser Symbol	IEEE-Symbol
NOR-Gatter		
NAND-Gatter		

Beispiel

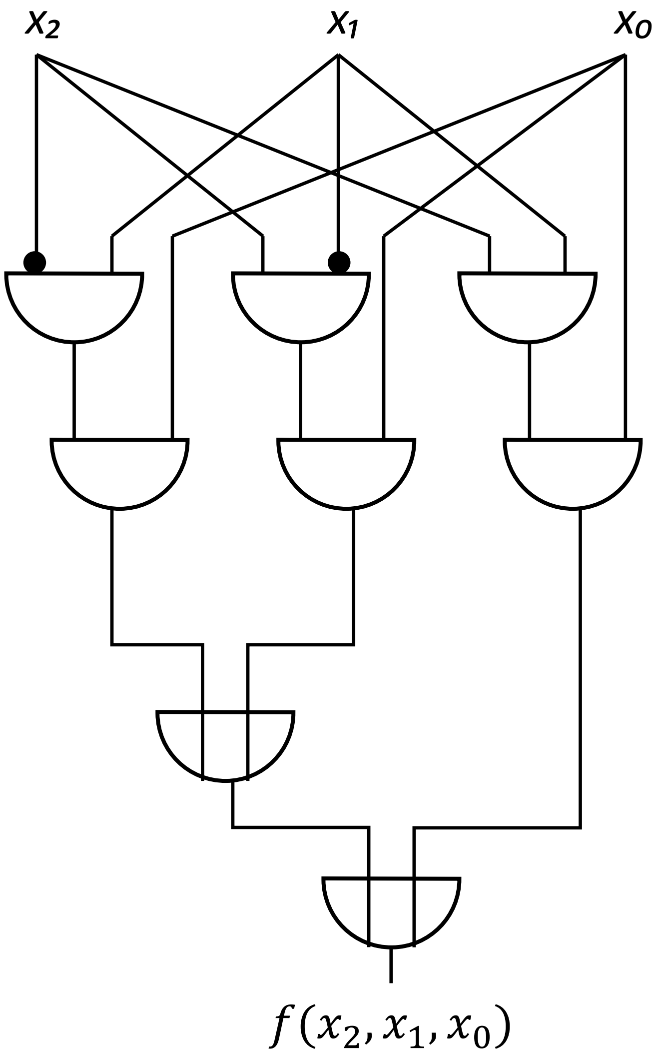
Sei $f: B^3 \to B$ gegeben durch:

i	x_2	x_1	x_0	$f(x_2, x_1, x_0)$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

DNF:

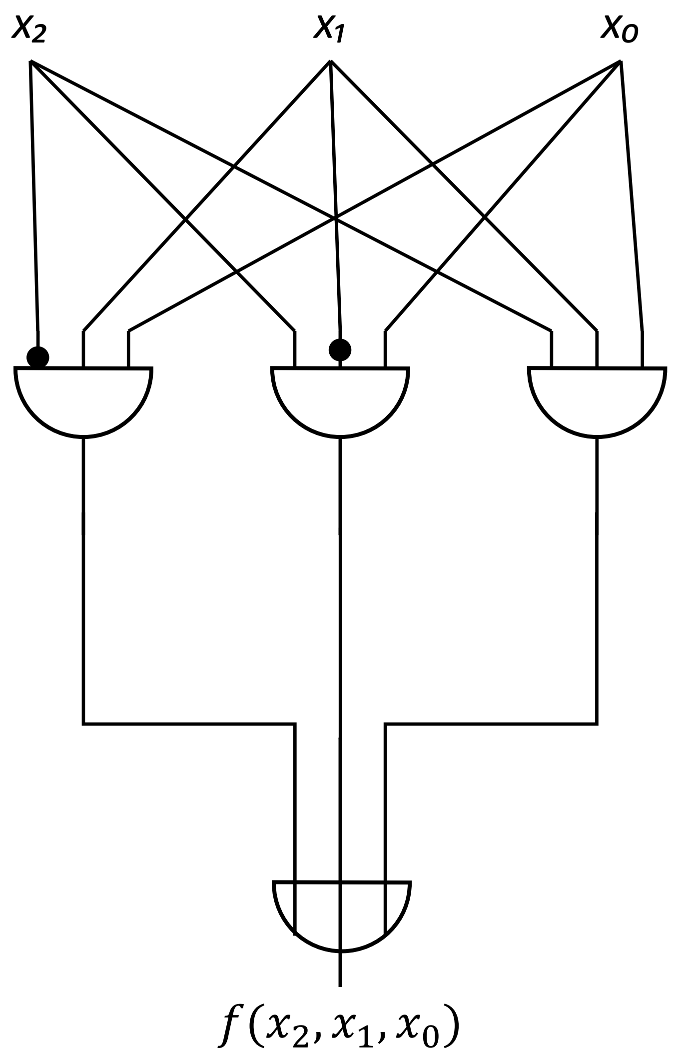
$$\begin{aligned} f(x_2, x_1, x_0) &= m_3 + m_5 + m_7 \\ &= \overline{x_2} x_1 x_0 + x_2 \overline{x_1} x_0 + x_2 x_1 x_0 \end{aligned}$$

Beispiel



Beispiel

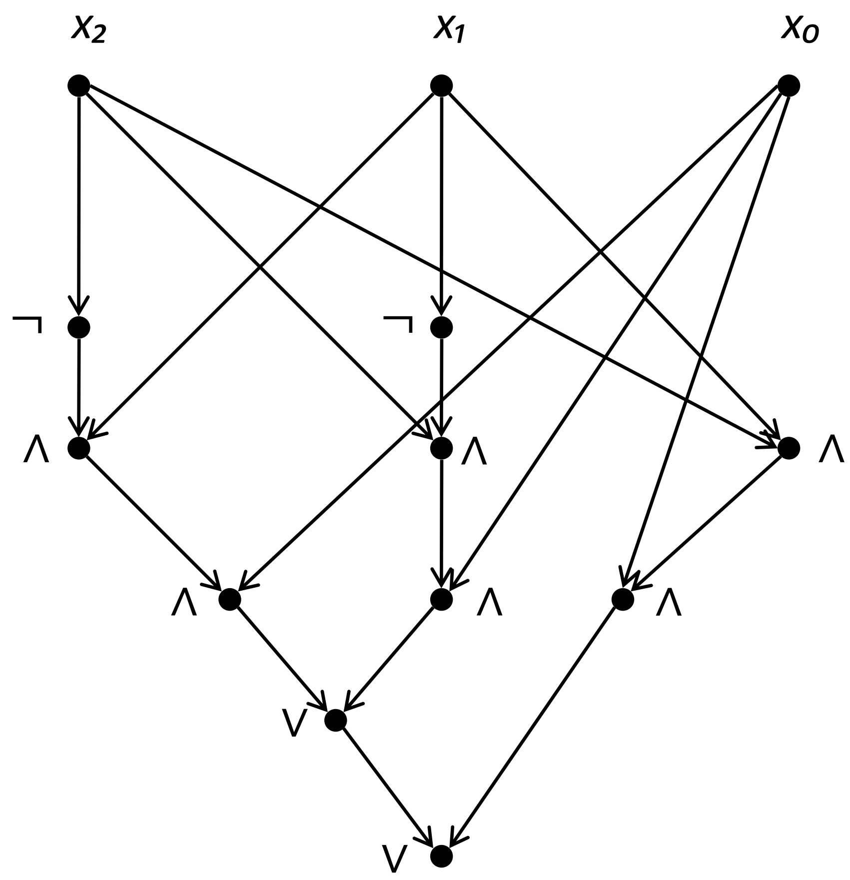
Alternative Schaltung:



DAG-Darstellung

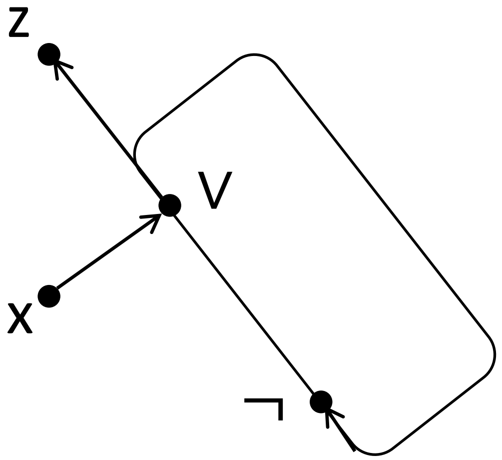
DAG=
Directed
Acyclic
Graph

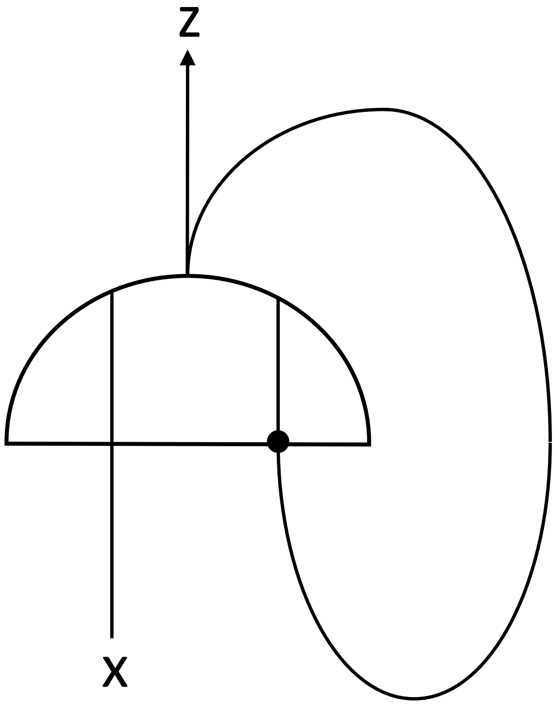
gerichteter
azyklischer
Graph



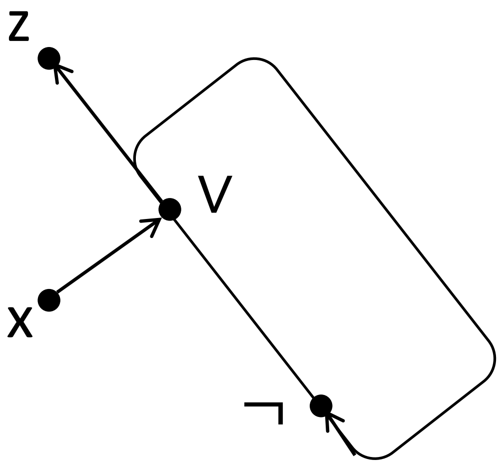
$f(x_2, x_1, x_0)$

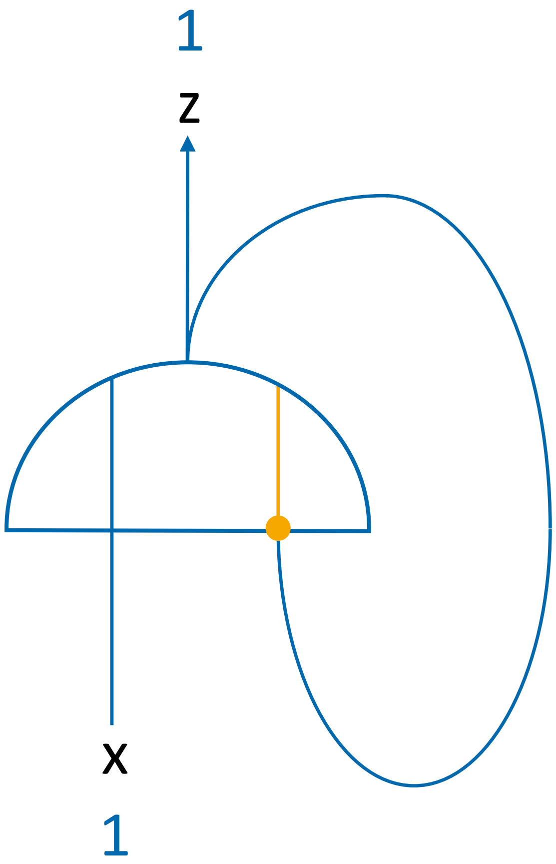
Beispiel für zyklischen Graph: Flimmerschaltung



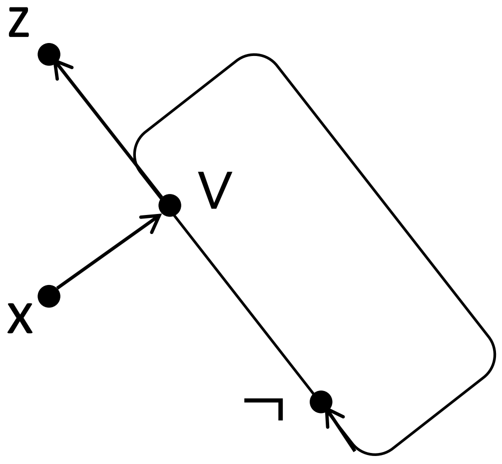


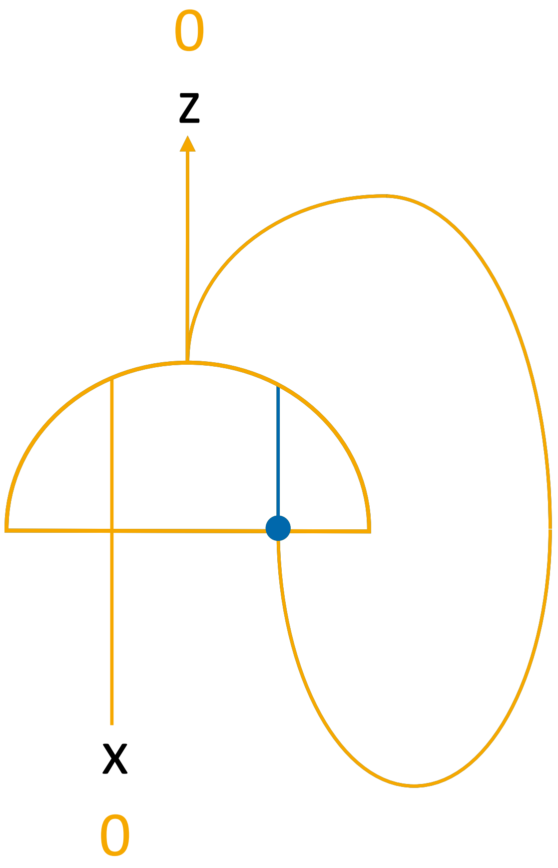
Beispiel für zyklischen Graph: Flimmerschaltung





Beispiel für zyklischen Graph: Flimmerschaltung





Anwendung: Schaltungsabhängige Fehlerdiagnose

Beispiel:

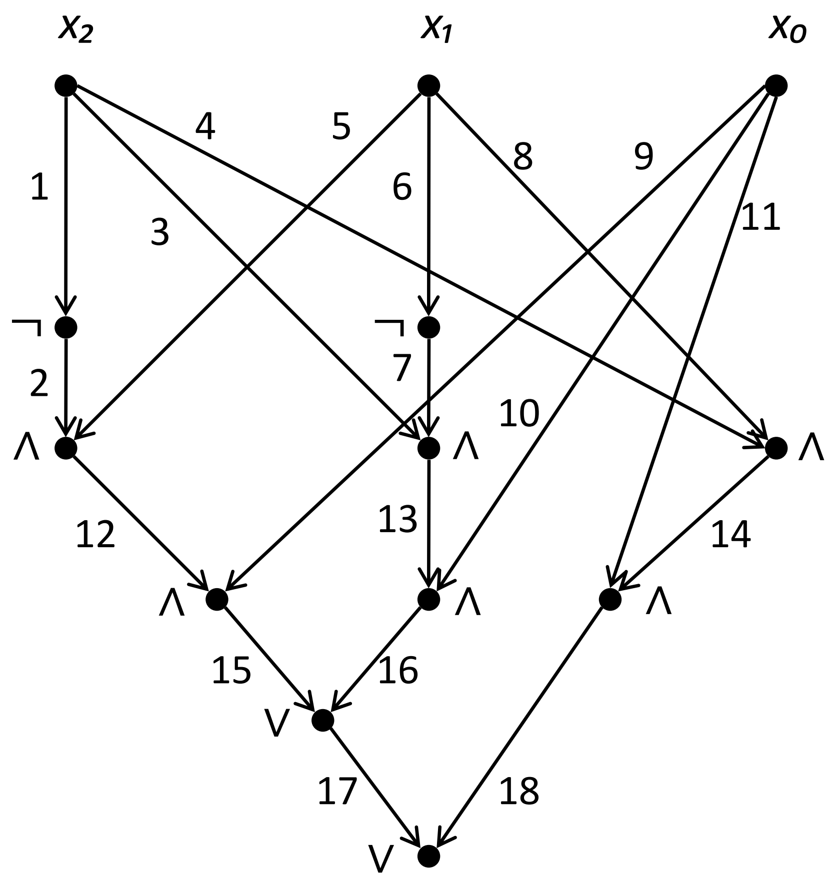
$$f(x_2, x_1, x_0) = \overline{x_2}x_1x_0 + x_2\overline{x_1}x_0 + x_2x_1x_0$$

Annahmen:

- Es tritt im gegebenen Schaltnetz höchstens ein Fehler auf
- Der Defekt, welcher den Fehler verursacht, ist ein gerissener Verbindungsdraht

Hier: **0-Verklemmung** bzw. **Stuck-at-Zero-Fault**

DAG mit Drahtnummern



$f(x_2, x_1, x_0)$

Darstellung von f

$$f_1 = \overline{0} \cdot x_1x_0 + x_2\overline{x_1}x_0 + x_2x_1x_0 = x_1x_0 + x_2x_0$$
$$f_2 = 0 \cdot x_1x_0 + x_2\overline{x_1}x_0 + x_2x_1x_0 = x_2\overline{x_1}x_0 + x_2x_1x_0 = x_2x_0$$
$$f_3 = \overline{x_2}x_1x_0 + x_2x_1x_0 = x_1x_0$$
$$f_4 = \overline{x_2}x_1x_0 + x_2\overline{x_1}x_0$$
$$f_5 = x_2\overline{x_1}x_0 + x_2x_1x_0 = x_2x_0$$
$$f_6 = \overline{x_2}x_1x_0 + x_2x_0$$
$$f_7 = x_1x_0$$
$$f_8 = \overline{x_2}x_1x_0 + x_2\overline{x_1}x_0$$
$$f_9 = x_2x_0$$
$$f_{10} = x_1x_0$$
$$f_{11} = \overline{x_2}x_1x_0 + x_2\overline{x_1}x_0$$
$$f_{12} = 0 \cdot x_0 + x_2\overline{x_1}x_0 + x_2x_1x_0 = x_2x_0$$

$$f_{13} = x_1x_0$$
$$f_{14} = \overline{x_2}x_1x_0 + x_2\overline{x_1}x_0$$
$$f_{15} = x_2x_0$$
$$f_{16} = x_1x_0$$
$$f_{17} = x_2x_1x_0$$
$$f_{18} = \overline{x_2}x_1x_0 + x_2\overline{x_1}x_0$$

Hinweis: $xyz + x\overline{y}z = xz$

Fehlermöglichkeiten (Ausfalltafel/-matrix)

x_2	x_1	x_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	1	1	0	1	1	1	0
1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1	1	0	1	1
1	1	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	1	1	1	0	1

x_2	x_1	x_0	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	1	1	0	1	0	1
1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	1	0	1	1	0	0	1
1	1	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	1	1	0	1	1	1	0

Reduzierte Ausfallmatrix

Es gilt:

$$f_1 = f_6$$
$$f_2 = f_5 = f_9 = f_{12} = f_{15}$$
$$f_3 = f_7 = f_{10} = f_{13} = f_{16}$$
$$f_4 = f_8 = f_{11} = f_{14} = f_{18}$$

x_2	x_1	x_0	f	f_1	f_2	f_3	f_4	f_{17}
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	1	1	1	1	0	1	1	0
1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
1	1	0	0	0	0	0	0	0
1	1	1	1	1	1	1	0	1

Fehlermatrix

Zeilen-Nr.	x_2	x_1	x_0	f	$f \oplus f_1$	$f \oplus f_2$	$f \oplus f_3$	$f \oplus f_4$	$f \oplus f_{17}$
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	1	1	1	0	1	0	0	1
4	1	0	0	0	0	0	0	0	0
5	1	0	1	1	0	0	1	0	1
6	1	1	0	0	0	0	0	0	0
7	1	1	1	1	0	0	0	1	0

Abschnitt 2.6

Ordered binary decision diagrams

- Kofaktoren
- Grundoperationen zur Vereinfachung von OBDDs
- Variablenordnung

Ordered Binary Decision Diagrams

- Geordnete Binäre Entscheidungs-Diagramme (OBDD)

- Vereinfachung von OBDDs

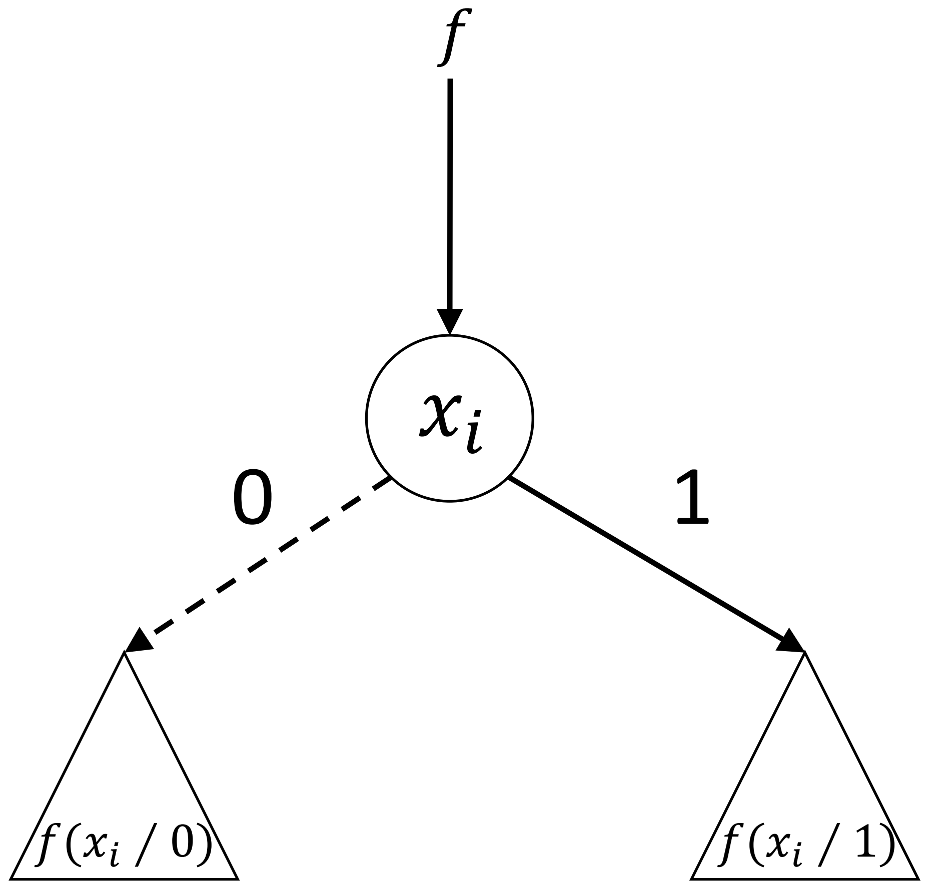
Kofaktoren

$$f(x_i/a) = f(x_{n-1}, \dots, x_{i+1}, a, x_{i-1}, \dots, x_0)$$

Dabei sei *a* ein fester Wert:

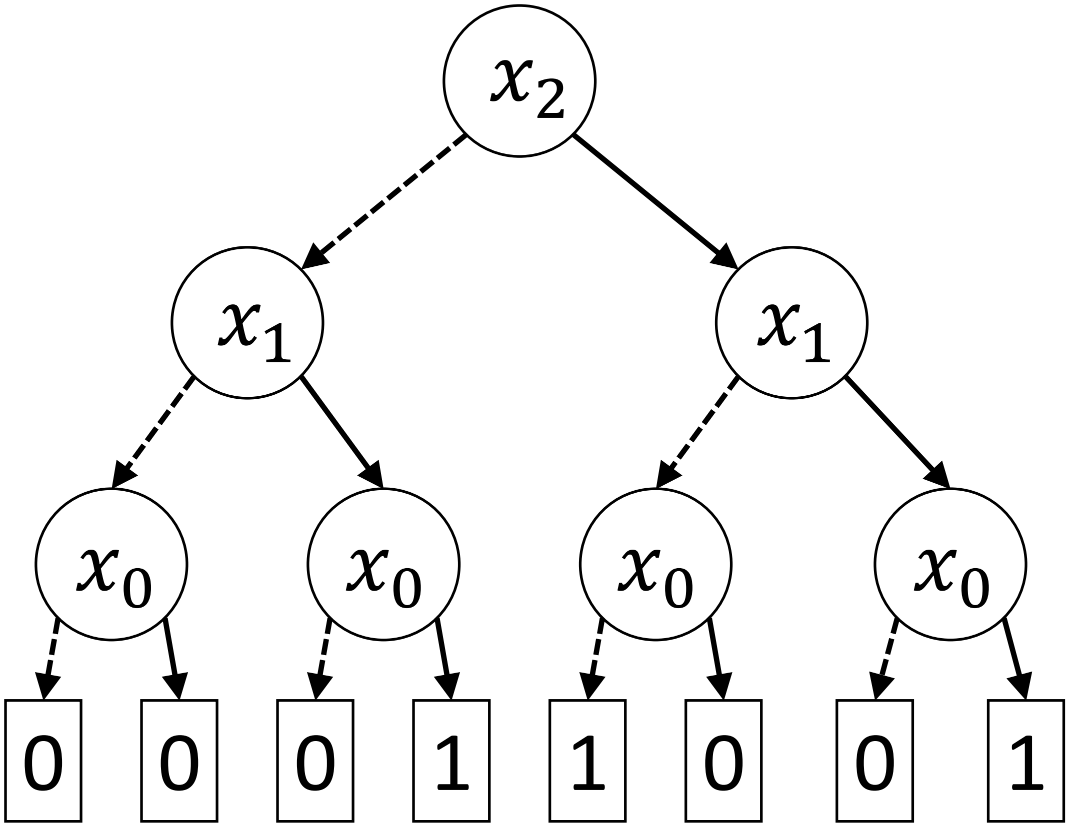
- **Positiver Kofaktor**: $f(x_i/a) = f(x_{n-1}, \dots, x_{i+1}, 1, x_{i-1}, \dots, x_0)$
- **Negativer Kofaktor**: $f(x_i/a) = f(x_{n-1}, \dots, x_{i+1}, 0, x_{i-1}, \dots, x_0)$

Baumdarstellung einer Booleschen Funktion anhand der Kofaktoren

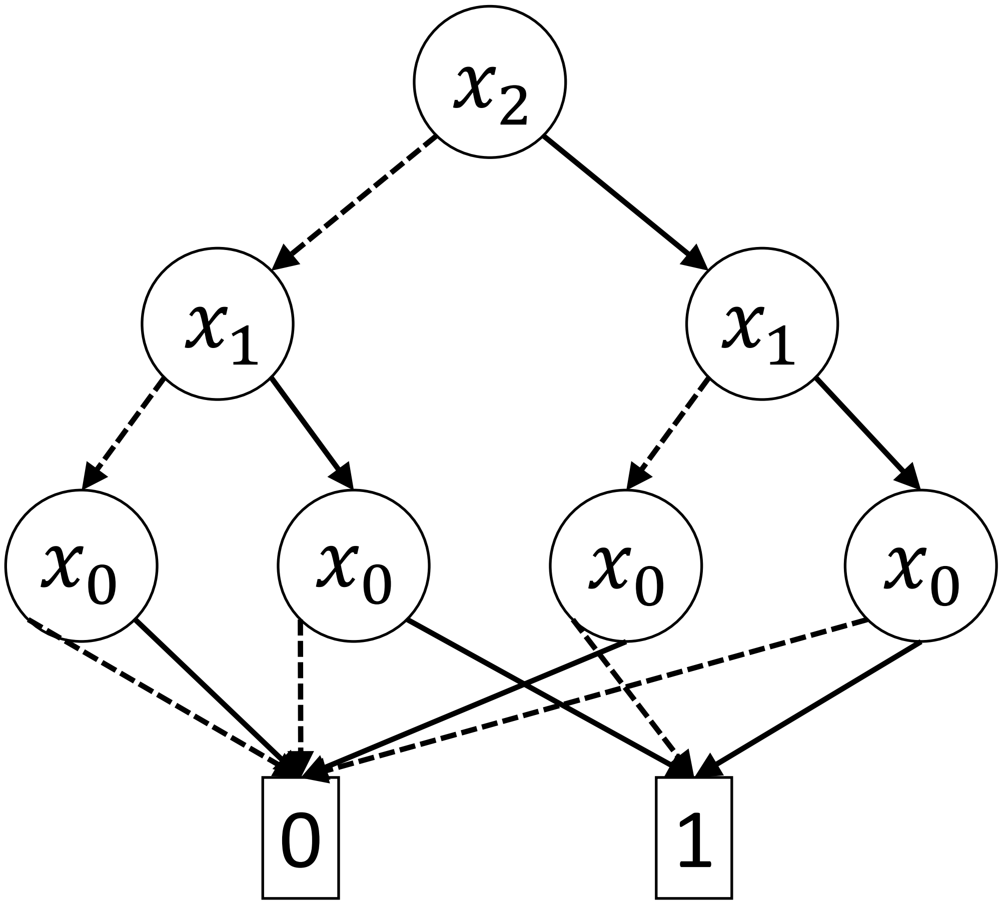


Beispiel: Funktion als Entscheidungsbaum

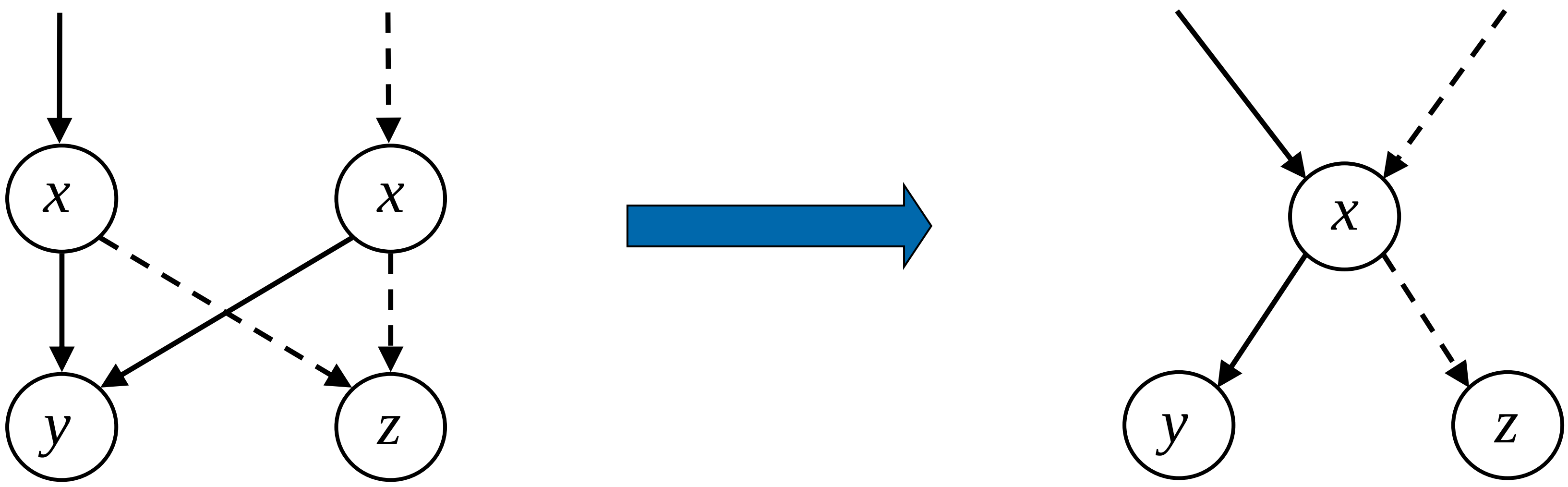
x_2	x_1	x_0	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



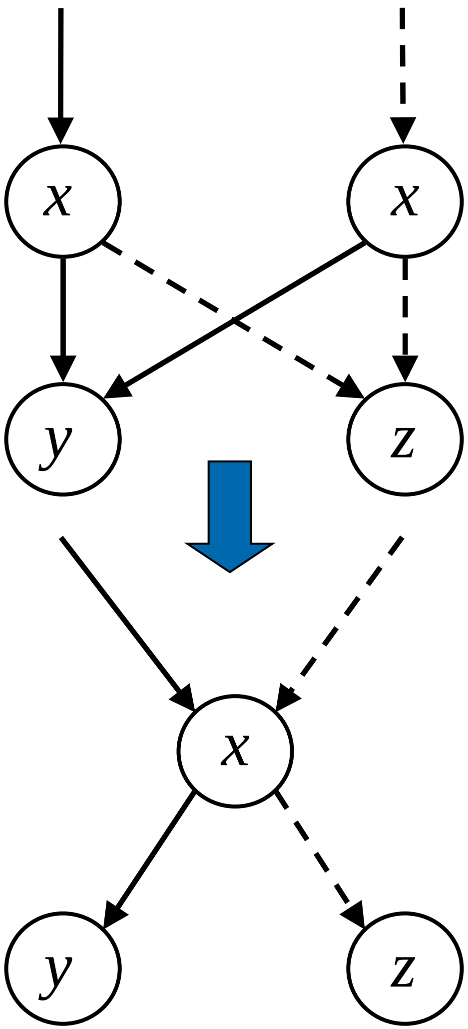
Beispielbaum nach Zusammenlegen der Blätter

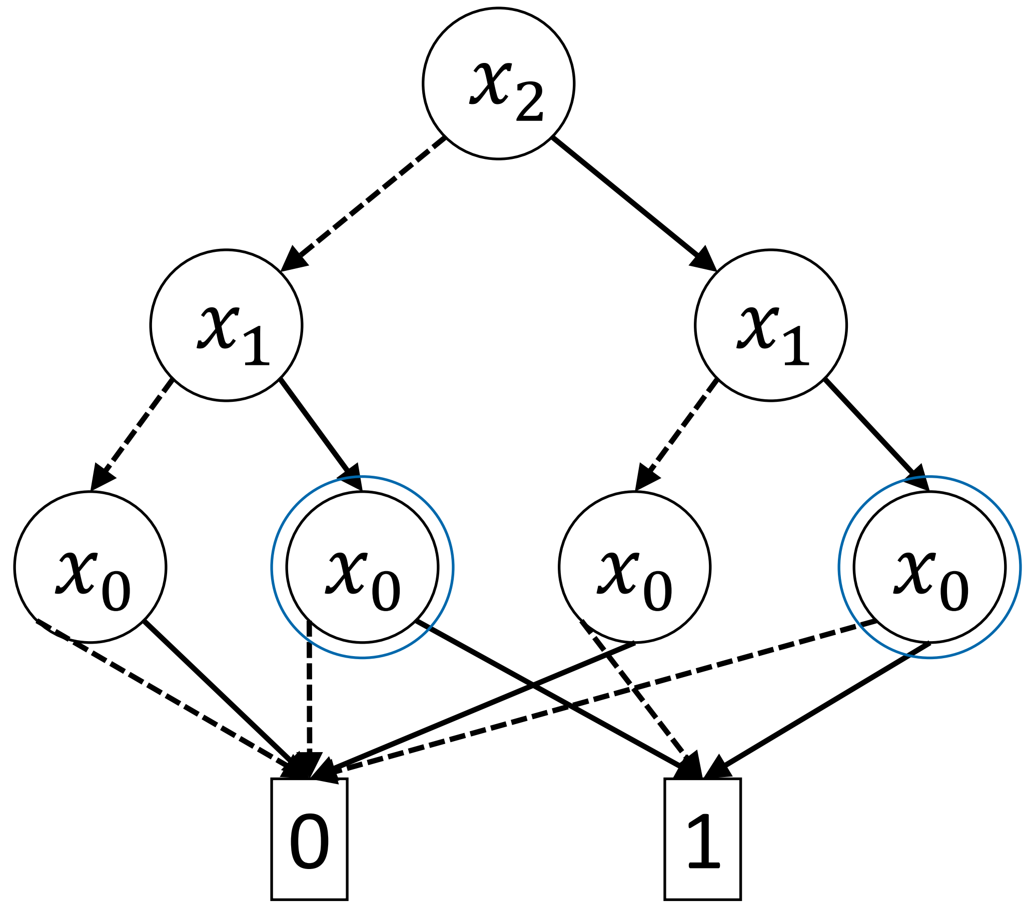


1. Grundoperation zur Vereinfachung von OBDDs: Verjüngung (4-3 Regel)

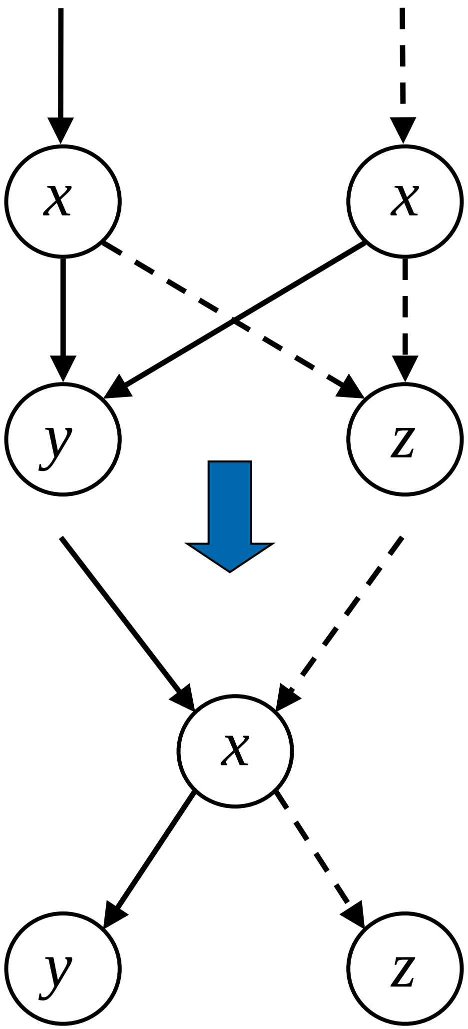


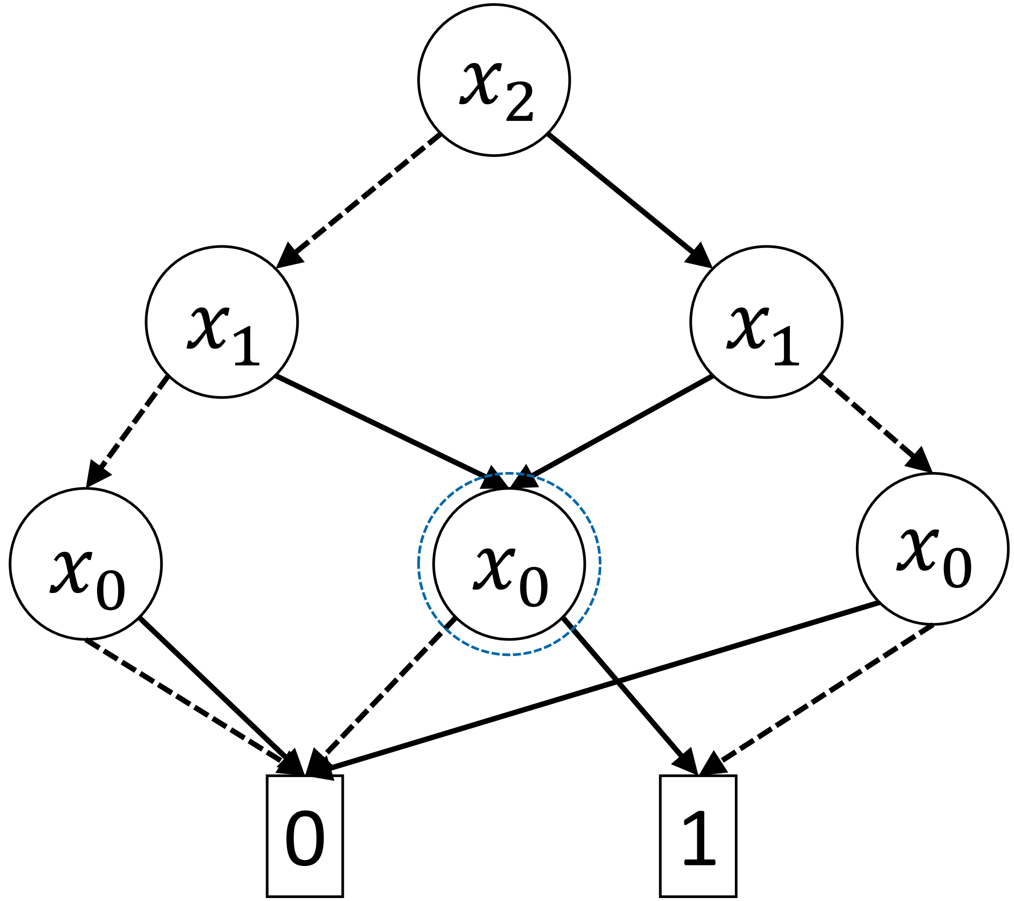
1. Grundoperation zur Vereinfachung von OBDDs: Verjüngung (4-3 Regel)





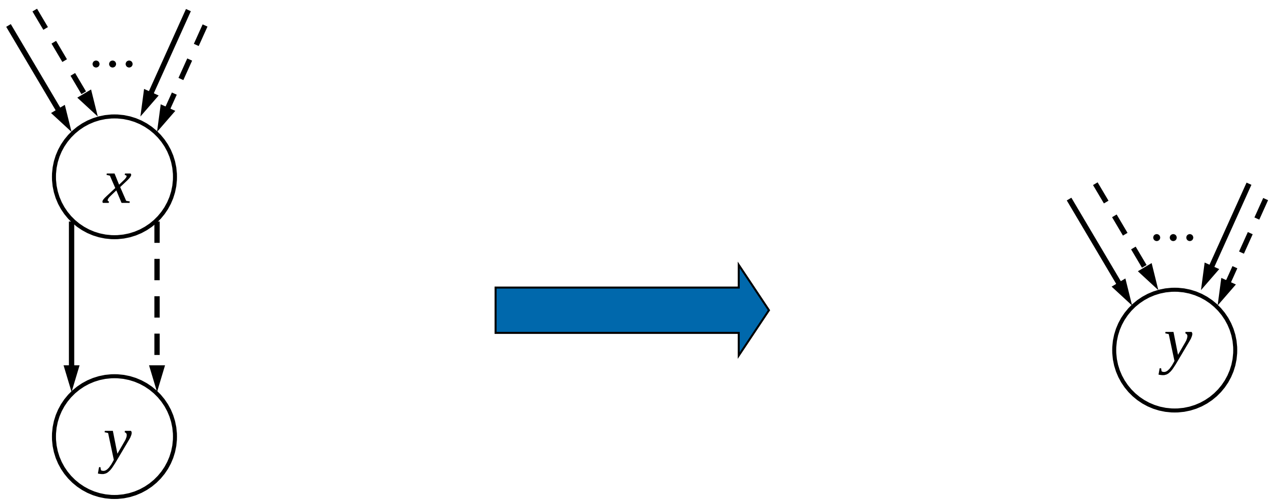
Beispielbaum nach Zusammenlegen identischer Teilbäume



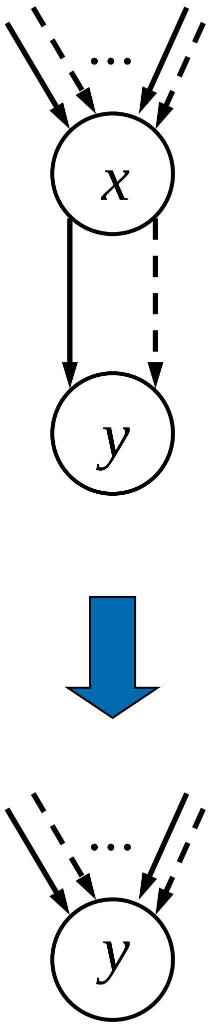


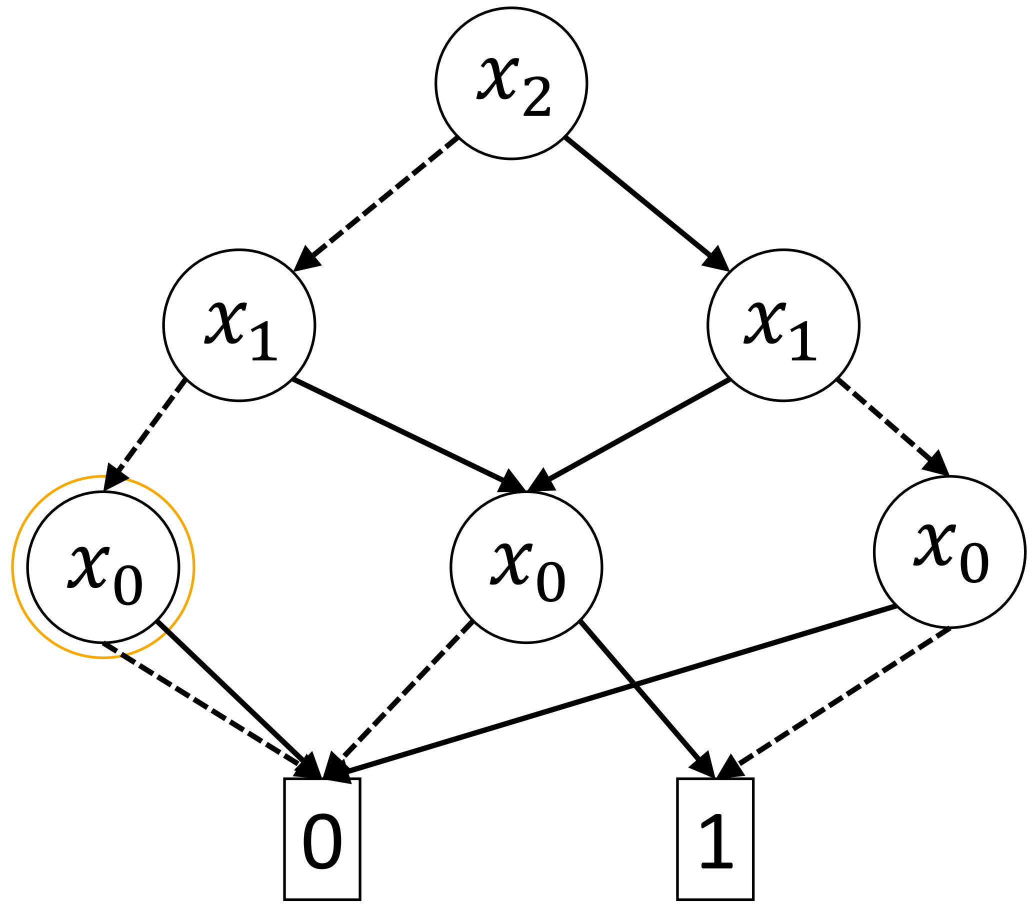
Informatik 11
Embedded Software
RWTH AACHEN UNIVERSITY

2. Grundoperation zur Vereinfachung von OBDDs: Elimination (2-1 Regel)

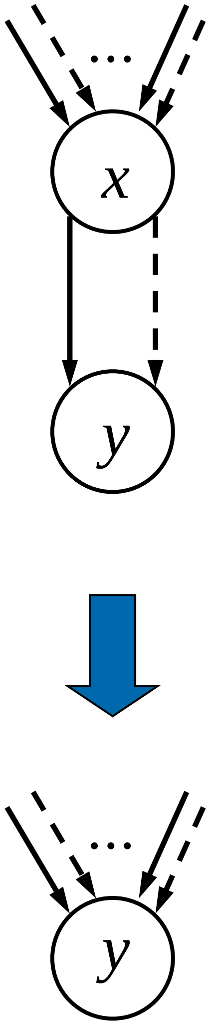


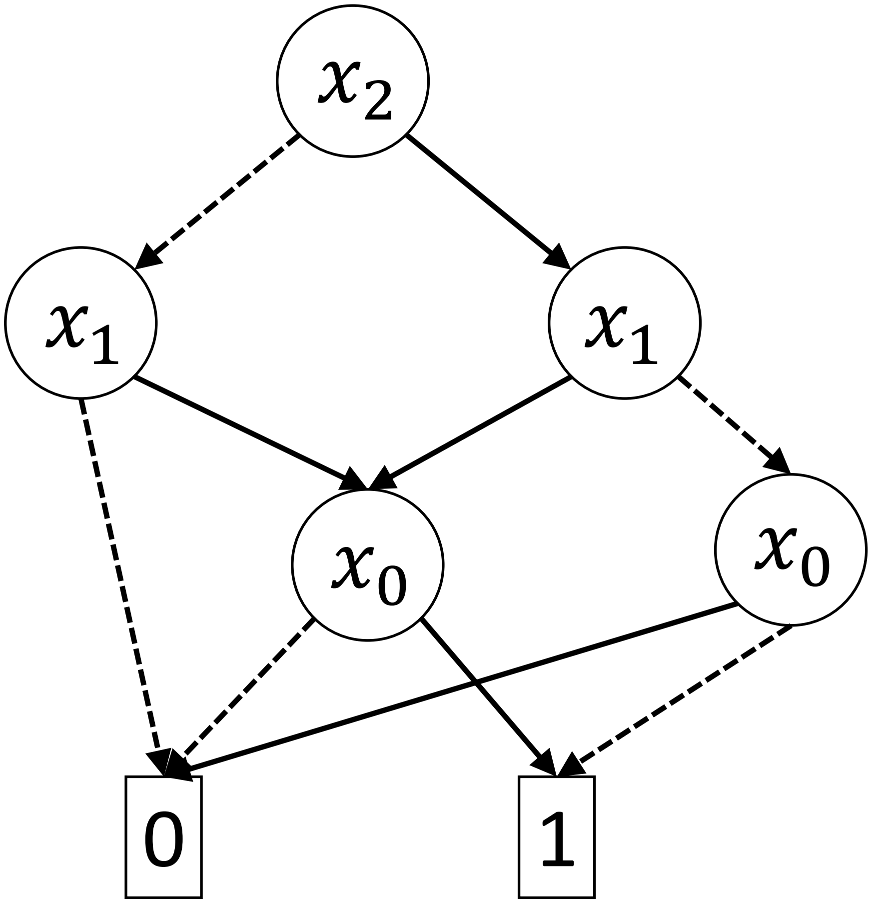
2. Grundoperation zur Vereinfachung von OBDDs: Elimination (2-1 Regel)





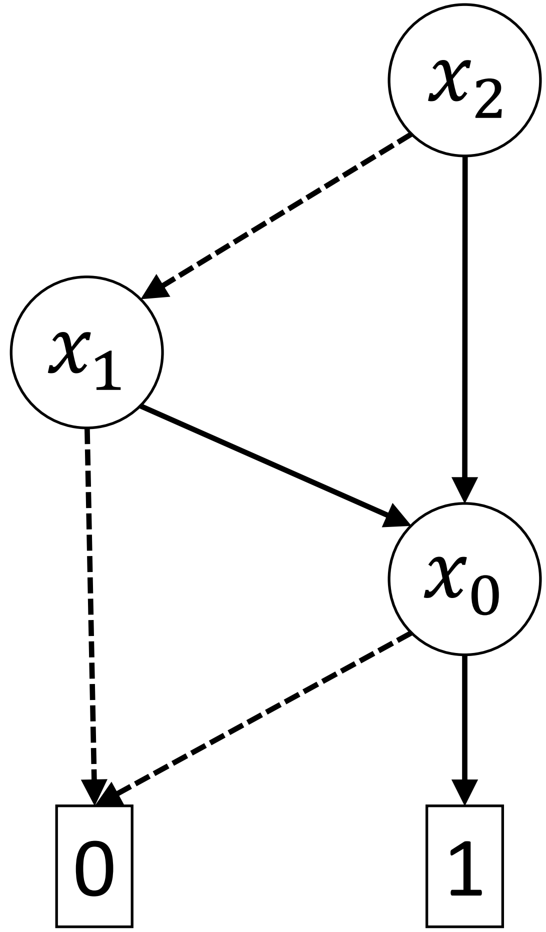
Beispielbaum nach Elimination des linken x_0-Teilbaums



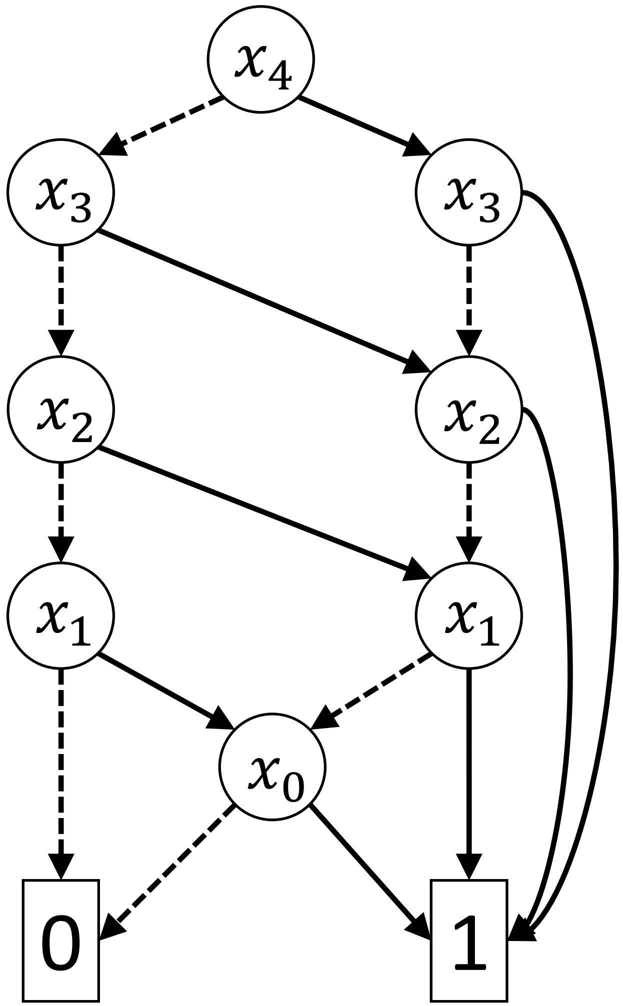


Beispiel 2

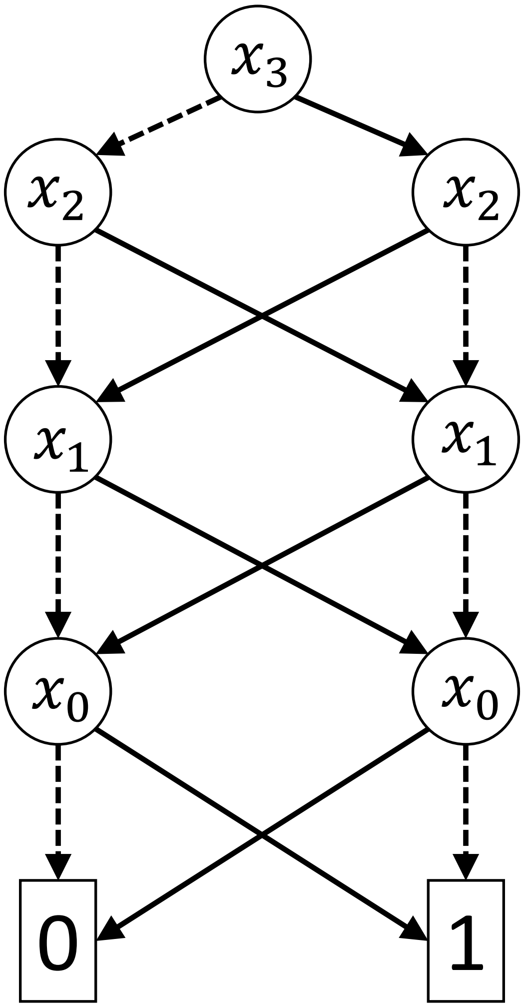
$f = ?$



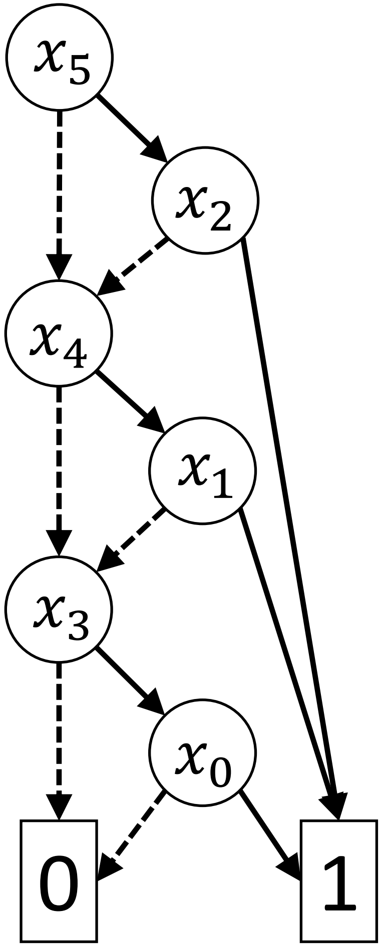
Beispiel: OBDD für die „Schwellenwert-Funktion“ $f(x_4, x_3, x_2, x_1, x_0) = T_2^5$



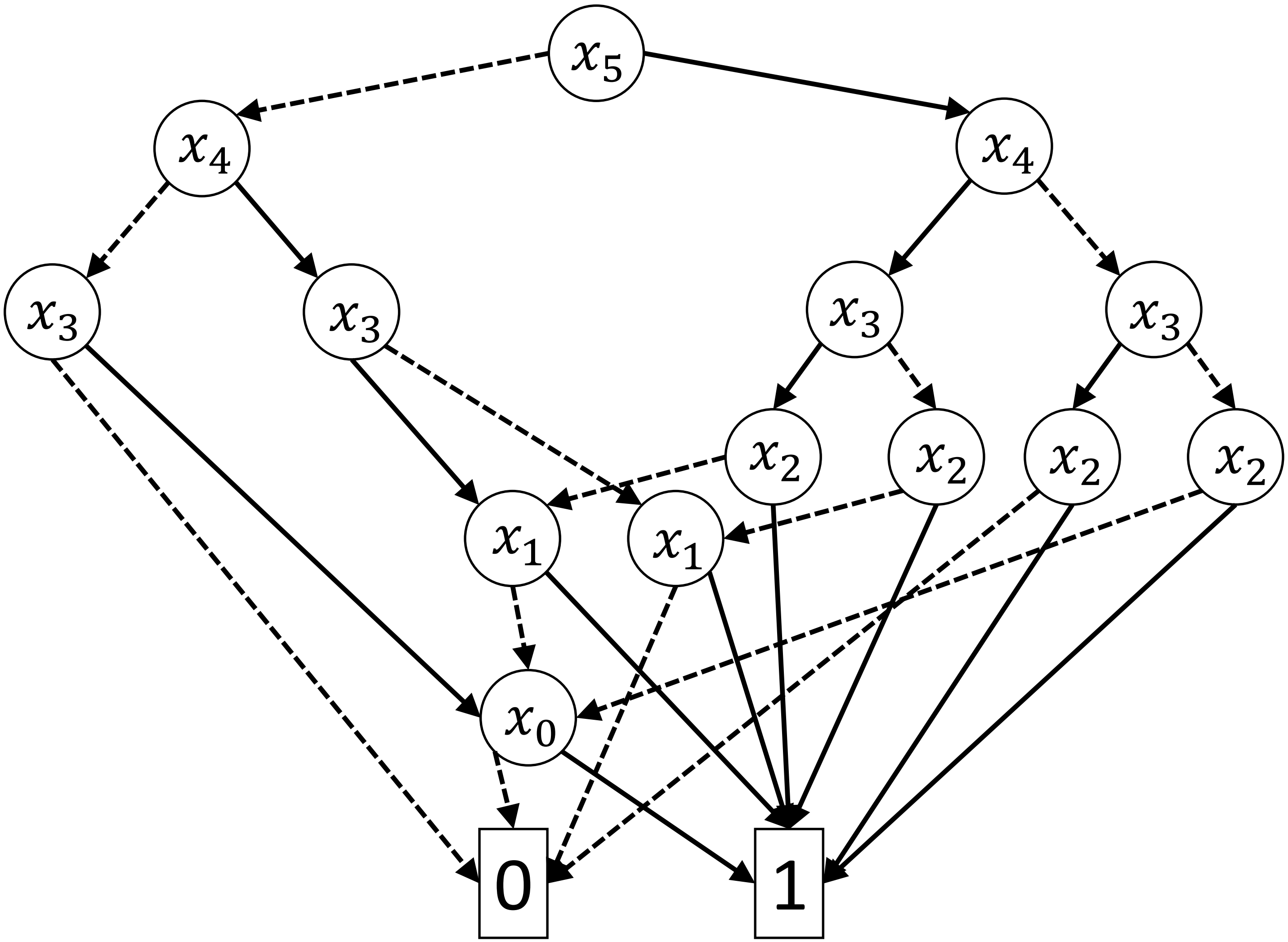
Beispiel: OBDD für die „Ungerade-Paritäts-Funktion“



OBDD zur Variablenordnung $V_1 = x_5 <= x_2 <= x_4 <= x_1 <= x_3 <= x_0$



OBDD zur Variablenordnung $V_2 = x_5 <= x_4 <= x_3 <= x_2 <= x_1 <= x_0$



Abschnitt 2.7

Gray-Code

Gray-Code

- Generierungsverfahren zur **robusten** Übertragung

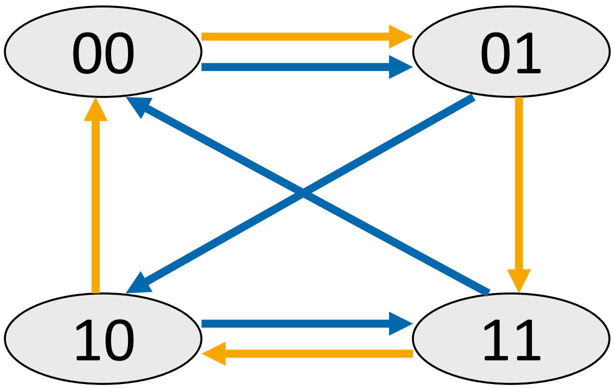
- Eigenschaft:

  Die Darstellung zweier benachbarter Zahlen
  unterscheidet sich nur durch **1 Bit**

- Generierung (eine Möglichkeit):
  1. Zahl im Binärcode darstellen $x_1 = (11)_2$
  2. Links-Shift um 1 Bit $x_2 = x_1 \ll 1 = (110)_2$
  3. XOR-Verknüpfung $x_3 = x_1 \oplus x_2 = (101)_2$
  4. Rechts-Shift um 1 Bit $x_4 = x_3 \gg 1 = (10)_2$

Beispiel: 2-Bit Gray-Code

Dualzahl	Gray-Code
00	00
01	01
10	11
11	10



Normale Reihenfolge

Gray-Code

Beispiel: 3-Bit Gray-Code

c_2	c_1	c_0	C_2	C_1	C_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Beispiel: 3-Bit Gray-Code

$$C_2 = f(c_2, c_1, c_0) = c_2 \cdot \overline{c_1} \cdot \overline{c_0} + c_2 \cdot \overline{c_1} \cdot c_0 + c_2 \cdot c_1 \cdot \overline{c_0} + c_2 \cdot c_1 \cdot c_0$$

$$C_1 = g(c_2, c_1, c_0) = \overline{c_2} \cdot c_1 \cdot \overline{c_0} + \overline{c_2} \cdot c_1 \cdot c_0 + c_2 \cdot \overline{c_1} \cdot \overline{c_0} + c_2 \cdot \overline{c_1} \cdot c_0$$

$$C_0 = h(c_2, c_1, c_0) = \overline{c_2} \cdot \overline{c_1} \cdot c_0 + \overline{c_2} \cdot c_1 \cdot \overline{c_0} + c_2 \cdot \overline{c_1} \cdot c_0 + c_2 \cdot c_1 \cdot \overline{c_0}$$

Zusammenfassung

- **Boolesche Algebra** als formale Grundlage in der Schaltungstechnik und der Computerhardware
- Mehrere Möglichkeiten Boolesche Funktionen darzustellen: KNF, DNF, DAGs, OBDDs
- Gray-Code: Generierungsverfahren zur **robusten** Übertragung