Teil II

Schaltfunktionen

Teil II.1

Zahlendarstellung

b-adische Systeme

Sei $b \in \mathbb{N}$ mit $b > 1$ und $E_b = \{0, 1, \ldots, b-1\}$ (Alphabet).
Dann ist jede Fixpunktzahl z (mit n Vorkomma– und k Nachkommastellen) mit $0 \leq z \cdot b^k \leq b^{n+k} - 1$ (und $n \in \mathbb{N}$, $k \in \mathbb{N}_0$) eindeutig als Wort der Länge $n + k$ über E_b darstellbar durch

$$z = \sum_{i=-k}^{n-1} z_i b^i = (z_{n-1} z_{n-2} \ldots z_0 . z_{-1} z_{-2} \ldots z_{-k})_b, \quad z_i \in E_b.$$

- Dezimalsystem: $E_{10} = \{0, 1, \ldots, 9\}$
- Dual– oder Binärsystem: $E_2 = \{0, 1\}$
- Oktalsystem: $E_8 = \{0, 1, \ldots, 7\}$
- Hexadezimalsystem: $E_{16} = \{0, 1, \ldots, 9, A, B, C, D, E, F\}$

Umrechnung Dezimal $\rightarrow$ Basis b

Gegeben: $d_{10}\ldots$ eine Dezimalzahl (ohne Nachkommastellen)
Gesucht: $(z_{n-1}z_{n-2}\ldots z_0)_b = d_{10}$

Algorithmus:

```
i := 0                     // Stelle
while d != 0 do            // solange d ungleich 0
  z[i] := d mod b          // Rest von d div b
  d := d div b             // ganzzahlige Division
  i := i+1                 // erhöhe Stelle
endwhile
```

Nachkommastellen: Dezimal $\rightarrow$ Basis b

Gegeben: $d_{10} = (0.x_1 \ldots x_k)_{10}$: k Nachkommastellen
Gesucht: $(0.z_1 \ldots z_j)_b = d_{10}$

Algorithmus:

```
i := 1                       // Stelle
while d != 0 do              // solange d ungleich 0
  z[i] := truncate(d * b)    // Vorkommastelle (0 <= z[i] <= b-1)
  d := d * b - z[i]          // Nachkommastellen
  i := i+1                   // erhöhe Stelle
endwhile
```

Beispiel

25.3_{10}

$$\begin{array}{rcrcrl}
25 & div & 2 & = & 12 & Rest: 1 \\
12 & div & 2 & = & 6 & Rest: 0 \\
6 & div & 2 & = & 3 & Rest: 0 \\
3 & div & 2 & = & 1 & Rest: 1 \\
1 & div & 2 & = & 0 & Rest: 1
\end{array}$$

$$\begin{array}{rcrcrl}
0.3 & * & 2 & = & 0.6 & z_1 = 0 \\
0.6 & * & 2 & = & 1.2 & z_2 = 1 \\
0.2 & * & 2 & = & 0.4 & z_3 = 0 \\
0.4 & * & 2 & = & 0.8 & z_4 = 0 \\
0.8 & * & 2 & = & 1.6 & z_5 = 1 \\
0.6 & * & 2 & = & 1.2 & z_6 = 1 = z_2 \\
\ldots & & & & &
\end{array}$$

$$25.3_{10} = 11001.010011001\ldots_2 = 11001.0\overline{1001}_2$$

Umrechnung Basis $b \to$ Dezimal

Gegeben: $(z_{n-1}z_{n-2}\dots z_0)_b$
Gesucht: $d_{10} = (z_{n-1}z_{n-2}\dots z_0)_b$

Algorithmus:

```
i := n-1                // Stelle
d := 0                  // Initialisierung
while i >= 0 do         // solange i grösser oder gleich 0
  d := b*d + z[i]       // berechne nächste Stelle
  i := i-1              // vermindere Index
endwhile
```

Basiert auf der Äquivalenz (Hornerschema):
$\sum_{i=0}^{n-1} z_i b^i = z_0 + b(\sum_{i=1}^{n-1} z_i b^{i-1}) =$
$= z_0 + b(z_1 + b(\dots b(z_{n-2} + bz_{n-1})\dots))$

Beispiel

$$
\begin{aligned}
& 0 * 2 + 1 \\
= \; & 1 * 2 + 1 \\
= \; & 3 * 2 + 0 \\
= \; & 6 * 2 + 0 \\
= \; & 12 * 2 + 1 \\
= \; & 25
\end{aligned}
$$

Darstellung negativer Zahlen

$z = (z_{n-1} \dots z_0 . z_{-1} \dots z_{-k})_2$ eine $n{+}k$ Dualzahl in Fixpunktdarstellung

- Einerkomplement:
  $K_1(z) = (\overline{z}_{n-1}\overline{z}_{n-2} \dots \overline{z}_0 . \overline{z}_{-1} \dots \overline{z}_{-k})_2$
  mit $z_i = 0 \leftrightarrow \overline{z}_i = 1$, und $\overline{\overline{z}}_i = z_i$.

- Zweierkomplement:
  $K_2(z) = (K_1(z) + 1\ ulp) modulo 2^n = 2^n - z$

- b-Komplement für $n + k$ Fixpunktzahlen:
  $K_b(z) = (\tilde{z}_{z-1}\tilde{z}_{n-2} \dots \tilde{z}_0 . \tilde{z}_{-1} \dots \tilde{z}_{-k})_b + 1\ ulp$

Beispiel

Bei einer Wortlänge von $n = 8$ Bits lauten die Darstellungen von $+92$ und -92 im Einer– bzw. Zweierkomplement:

Komplement	$+92$	-92
Einer–	dual 01011100 hexadezimal 5C	dual 10100011 hexadezimal A3
Zweier–	dual 01011100 hexadezimal 5C	dual 10100100 hexadezimal A4

Subtraktion mit Zweier-Komplement

$x - y = x + K_2(y)$

Für $n = 8$, berechne $45 - 92$

$$
\begin{array}{ll}
45 = 2^5 + 2^3 + 2^2 + 2^0\text{:} & (00101101)_2 \\
+\ K_2(92)\text{:} & (10100100)_2 \\
\hline
 & (11010001)_2 = K_2(z)
\end{array}
$$

Aus $K_2(z) = (11010001)_2$ folgt:

$$
\begin{aligned}
z &= K_1(K_2(z) - 1\ ulp) \\
&= K_1((11010001)_2 - 1) \\
&= (00101111)_2 \\
&= 2^5 + 2^3 + 2^2 + 2^1 + 2^0 = 47
\end{aligned}
$$

Gleitkommazahlen nach IEEE 754/854

Vorzeichen–Bit

$\downarrow$ Exponent: l Bit Mantisse: m Bit

S	$e_{l-1}e_{l-2}\cdots\cdots\cdots e_1e_0$	$f_{m-1}f_{m-2}\cdots\cdots\cdots\cdots f_1f_0$

$$\text{Exponentencharakteristik}: \quad E = (e_{l-1}e_{l-2}\cdots\cdots e_1e_0)_2$$
$$\text{Mantisse}: \quad M = (f_{m-1}f_{m-2}\cdots\cdots f_1f_0)$$

DargestellteZahl :

$(-1)^S \cdot (1.M)_2 \cdot 2^{(E-E_{Bias})}$	$1 \leq E \leq 2 \cdot E_{Bias}$	normalisiert
$(-1)^S \cdot (0.M)_2 \cdot 2^{1-E_{Bias}}$	$E = 0,\ M \neq 0$	denormalisiert
$(-1)^S \cdot 0$	$E = 0,\ M = 0$	Null mit Vorz.
$(-1)^S \cdot \infty$	$E = 2 \cdot E_{Bias} + 1,\ M = 0$	∞ mit Vorz.
NaN	$E = 2 \cdot E_{Bias} + 1,\ M \neq 0$	Not a Number

Bereich und Genauigkeit

- Mantisse $m \ldots$ Genauigkeit
- Exponent $l - E_{Bias} \ldots$ Größe des Zahlenbereichs.

Genauigkeit	Exponent	Mantisse	E_{Bias}	Bereich	Stellen
single: 32 Bit	$l = 8$	$m = 23$	127	$1.5 \cdot 10^{-45} \ldots 3.4 \cdot 10^{38}$	$7 - 8$
double: 64 Bit	$l = 11$	$m = 52$	1023	$5.0 \cdot 10^{-324} \ldots 1.7 \cdot 10^{308}$	$15 - 16$

Beispiel: 1.0

`single`

$$
\begin{aligned}
1.0 * 2^0 = 1.0 * 2^{127-127} \quad &\widehat{=} \quad 0\ \overbrace{01111111}^{\text{Exponent}}\ \overbrace{00000000000000000000000}^{\text{Mantisse}} \\
&= \quad \$3F800000
\end{aligned}
$$

`double`

$$
\begin{aligned}
1.0 * 2^0 = 1.0 * 2^{1023-1023} \quad &\widehat{=} \quad 0\ \overbrace{01111111111}^{\text{Exponent}}\ \underbrace{\overbrace{00000..............00000}^{\text{Mantisse}}}_{\text{52 Nullen}} \\
&= \quad \$3FF0000000000000
\end{aligned}
$$

Umwandlung Dezimal $\longrightarrow$ Binär

- Gegeben: z
- Schritt 1: Mantisse $M = z * 2^y$ sodass $1 <= M < 2$
- Schritt 2: Exponent $E = y + E_{Bias}$

Beispiel:

$$\begin{aligned}
6.5 &= 2^2 \cdot 1.625 \\
&= 2^2 \cdot \left(1 + \frac{1}{2} + \frac{0}{2^2} + \frac{1}{2^3}\right) \\
&= 2^{129-127} \cdot \left(1 + \frac{1}{2} + \frac{0}{2^2} + \frac{1}{2^3}\right) \\
&= 2^{129-127} \cdot (1.101)_2 \\
&\widehat{=}\ 0\ 10000001\ 10100000000000000000000 \\
&= \$40D00000.
\end{aligned}$$

Teil II.2

Boolesche Algebra

Boolesche Algebra

Für $B = \{0, 1\}$, $x, y \in B$ seien folgende Verknüpfungen definiert:

$$\begin{aligned} x \cup y &:= \text{Max}(x, y), \\ x \cap y &:= \text{Min}(x, y), \\ \overline{x} &:= 1 - x, \end{aligned}$$

$(B, \cup, \cap, \overline{\ })$ ist eine Boolesche Algebra

Gesetze der Booleschen Algebra

a) Kommutativgesetze: $x \cup y = y \cup x, \quad x \cap y = y \cap x$

b) Assoziativgesetze: $(x \cup y) \cup z = x \cup (y \cup z), \quad (x \cap y) \cap z = x \cap (y \cap z)$

c) Verschmelzungsgesetz: $(x \cup y) \cap x = x, \quad (x \cap y) \cup x = x$

d) Distributivgesetze: $x \cap (y \cup z) = (x \cap y) \cup (x \cap z), \quad x \cup (y \cap z) = (x \cup y) \cap (x \cup z)$

e) Komplementgesetz: $x \cup (y \cap \overline{y}) = x, \quad x \cap (y \cup \overline{y}) = x$

f) $x \cup 0 = x, \quad x \cap 0 = 0, \quad x \cap 1 = x, \quad x \cup 1 = 1$

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

h) $x = x \cup x = x \cap x = \overline{\overline{x}}$

Potenzmenge

Für $B = \mathcal{P}(A)$ die Menge aller Teilmengen von A ist $(B, \cup, \cap, \bar{\ })$ ebenfalls eine Boolesche Algebra

Teil II.3

Schaltfunktionen

Schaltfunktionen

- Schaltfunktion: $F : B^n \to B^m$

- Totale Schaltfunktion: für alle 2^n Inputs gibt es einen eindeutigen Output, der alle m Bits belegt.

- Boolesche Schaltfunktion: $f : B^n \to B$

- Komponenten Darstellung:

$$F(x_1, \ldots, x_n) = (f_1(x_1, \ldots, x_n), f_2(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)).$$

- Anzahl verschiedener Boolescher Funktion: 2^{2^n}

Beispiele

- Addition zweier 2 Bit Zahlen: $A : B^4 \rightarrow B^3$
  $A(x_1, x_2, x_3, x_4) =$
  $(a_1(x_1, x_2, x_3, x_4), a_2(x_1, x_2, x_3, x_4), a_3(x_1, x_2, x_3, x_4))$
  mit

$$
\begin{array}{rcl}
a_1(x_1, x_2, x_3, x_4) & = & x_2 \oplus x_4 \\
a_2(x_1, x_2, x_3, x_4) & = & x_1 \oplus x_3 \oplus (x_2 \wedge x_4) \\
a_3(x_1, x_2, x_3, x_4) & = & (x_1 \wedge x_3) \\
& \vee & (x_1 \wedge x_2 \wedge x_4) \\
& \vee & (x_3 \wedge x_2 \wedge x_4)
\end{array}
$$

- Sortieren von 30 16 Bit Zahlen: $S : B^{30*16} \rightarrow B^{30*16}$

Boolesche Funktionen für $n = 2$

	(1)	$x \cdot \overline{x}$	$x \cdot y$	$x \cdot \overline{y}$	x	$\overline{x} \cdot y$	y	$\oplus$	$x + y$
	(2)	$\equiv 0$	Min	$>$	x	$<$	y	$\neq$	Max
	(3)		$\wedge$	$\not\rightarrow$	x	$\not\leftarrow$	y	$\not\leftrightarrow$	$\vee$
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

	(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)	$\downarrow$	$\leftrightarrow$	$\neg y$	$\leftarrow$	$\neg x$	$\rightarrow$	$\uparrow$	
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

Disjunktive Normalform

- $i = (i_1 i_2 \ldots i_n)_2$ ist ein einschlägiger Index zu $f : B^n \to B$ genau dann wenn $f(i_1, i_2, \ldots, i_n)_2 = 1$

- für einen einschlägigen Index i einer Booleschen Funktion f, heißt die Funktion $m_i(x_1, x_2, \ldots, x_n) := x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$ der i-te Minterm von f, wobei

$$x_j^{i_j} := \begin{cases} x_j & \text{falls } i_j = 1 \\ \overline{x}_j & \text{falls } i_j = 0 \end{cases}$$

- Darstellungssatz: Für jede Boolesche Funktion f gilt (DNF):

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

Beispiel

Gegeben sei die folgende Boolesche Funktion $f : B^3 \to B$ durch die Wertetabelle

i	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
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

Einschlägigen Indizes sind 3, 5, 7.
Die zugehörigen Minterme sind:
$m_3(x_1, x_2, x_3) = \overline{x}_1 x_2 x_3$
$m_5(x_1, x_2, x_3) = x_1 \overline{x}_2 x_3$
$m_7(x_1, x_2, x_3) = x_1 x_2 x_3.$
$f(x_1, x_2, x_3) = \overline{x}_1 x_2 x_3 + x_1 \overline{x}_2 x_3 + x_1 x_2 x_3$

Funktionale Vollständigkeit

Ein System $\mathcal{B} = \{f_1, \ldots, f_n\}$ Boolescher Funktionen heißt (funktional) vollständig, wenn jede Boolesche Funktion allein durch Einsetzungen bzw. Kompositionen von Funktionen aus $\mathcal{B}$ dargestellt werden kann.

a) $\{+, \cdot, ^{-}\}$ ist funktional vollständig.

b) $\{+, ^{-}\}$ ist funktional vollständig.

c) $\{\cdot, ^{-}\}$ ist funktional vollständig.

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

e) $\{\text{NOR}\}$ ist funktional vollständig.

Konjunktive Normalform

- i-ter Maxterm: $M_i(x_1, \ldots, x_n) := \overline{m_i(x_1, ..., x_n)} = x_1^{\overline{i_1}} + \ldots + x_n^{\overline{i_n}}$

- Konjunktive Normalform: Für jede Boolesche Funktion f gilt (KNF):

$$f = \prod_{i \notin I} M_i$$

$$KNF(f) = \overline{DNF(\overline{f})}$$

- Beispiel:

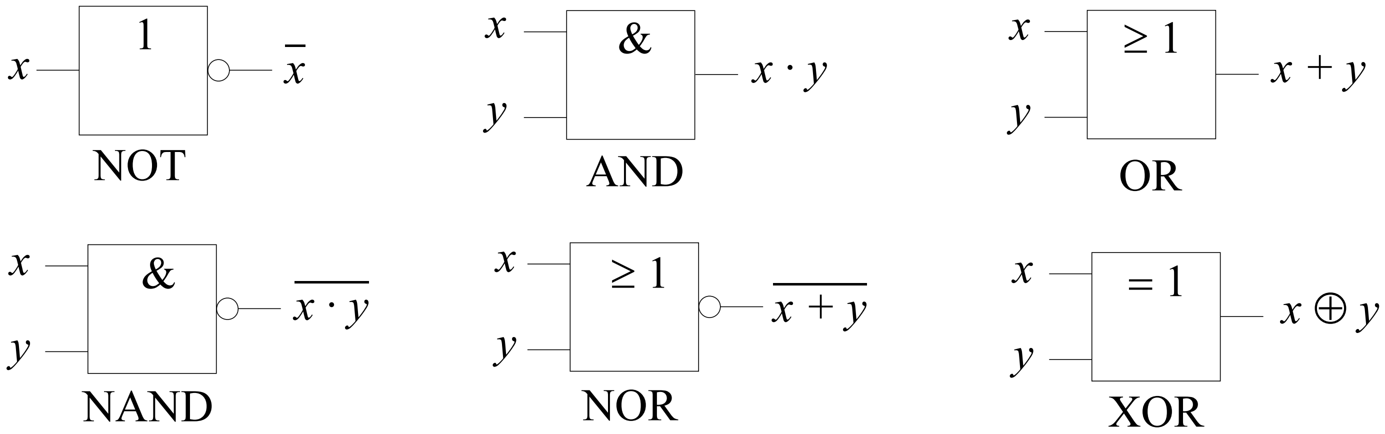
$$\begin{aligned} f(x_1, x_2, x_3) &= \overline{x}_1 x_2 x_3 + x_1 \overline{x}_2 x_3 + x_1 x_2 x_3 \\ &= M_0 \cdot M_1 \cdot M_2 \cdot M_4 \cdot M_6 \\ &= (x_1 + x_2 + x_3) \cdot (x_1 + x_2 + \overline{x}_3) \cdot (x_1 + \overline{x}_2 + x_3) \\ &\quad \cdot (\overline{x}_1 + x_2 + x_3) \cdot (\overline{x}_1 + \overline{x}_2 + x_3). \end{aligned}$$

Teil II.4

Schaltnetze

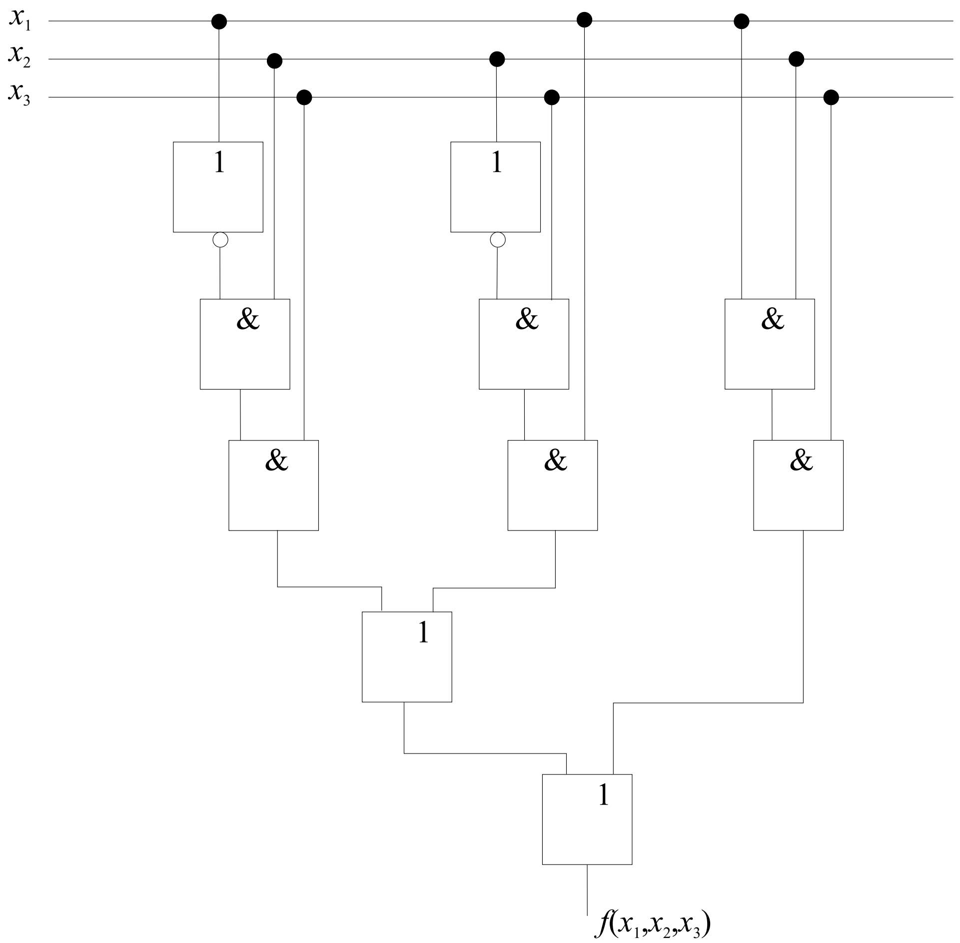
Bausteine

Neue DIN-NORM



Beispiel

$$f(x_1, x_2, x_3) = \overline{x}_1 x_2 x_3 + x_1 \overline{x}_2 x_3 + x_1 x_2 x_3$$



Kriterien für Schaltungen

- Geschwindigkeit: Minimiere Stufen

- Größe: Minimiere Anzahl der Gatter

- Fan-In, Fan-Out: Minimiere Anzahl der Inputs und Outputs

Graphen

- Punktmenge $P \subseteq \mathbb{N}$.

- Kantenmenge $K \subseteq P \times P$.
  nicht-reflexiv

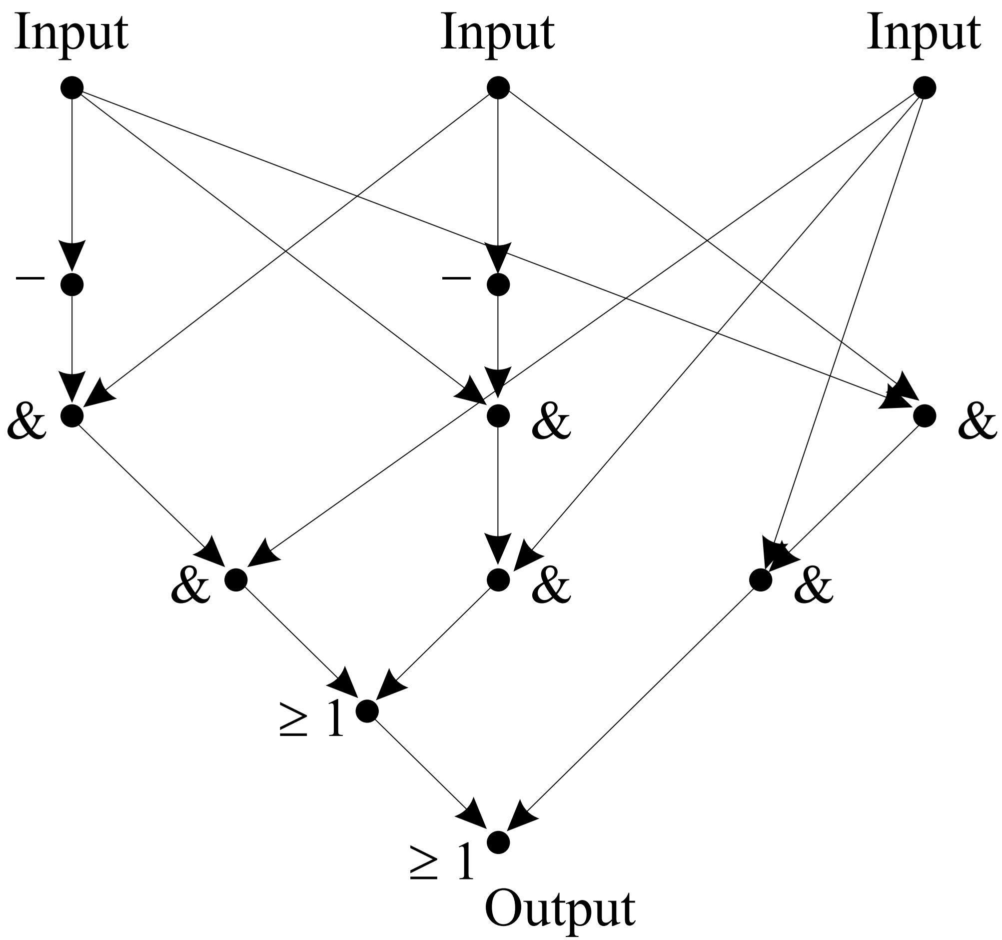
- Graph $G = (P, K)$
  ungerichtet: K symmetrisch. gerichtet: K nicht symmetrisch

- Pfad: $(p_1, p_2, \ldots, p_n)$

- Zyklus: $(p_1, p_2, \ldots, p_n, p_1)$

- DAG: Gerichteter, azyklischer Graph
  (Directed Acyclic Graph)

Beispiel



Teil II.5

Ringsummennormalform

Ringsummennormalform

RNF:

Sei $f : B^n \to B$ und $I = \{\alpha_1, \ldots, \alpha_k\}$ die Menge der einschlägigen Indizes zu f. Dann gilt:

$$\begin{aligned} f &= m_{\alpha_1} \oplus m_{\alpha_2} \oplus \ldots \oplus m_{\alpha_k} \\ &= \sum_{i=1}^{k} m_{\alpha_i} \end{aligned}$$

Äquivalenzen

Für alle $x, y, z \in B$ gilt:

a) $x \oplus 1 = \overline{x}, \quad x \oplus 0 = x$

b) $x \oplus x = 0, \quad x \oplus \overline{x} = 1$

c) $x \oplus y = y \oplus x$ (Kommutativität)

d) $x \oplus (y \oplus z) = (x \oplus y) \oplus z$ (Assoziativität)

e) $x \cdot (y \oplus z) = x \cdot y \oplus x \cdot z$ (Distributivität bzgl. $\cdot$)

f) $0 \oplus 0 \oplus \ldots \oplus 0 = 0$

g) $\underbrace{1 \oplus 1 \oplus \ldots \oplus 1}_{n--\text{mal}} = \begin{cases} 1 & \text{falls } n \text{ ungerade} \\ 0 & \text{falls } n \text{ gerade} \end{cases}$

Komplementfreie RNF

Reed–Muller Form:

Jede Boolesche Funktion $f : B^n \to B$ ist eindeutig darstellbar als Polynom (Multinom) in den Variablen $x_1, x_2, \ldots, x_n$ mit den Koeffizienten $a_0, a_1, \ldots, a_{1\ldots n} \in B$. Die Darstellung ist wie folgt:

$$
\begin{aligned}
f \;=\; & a_0 \\
& \oplus a_1 x_1 \oplus a_2 x_2 \oplus \ldots \oplus a_n x_n \\
& \oplus a_{12} x_1 x_2 \oplus \ldots \oplus a_{n-1,n} x_{n-1} x_n \\
& \qquad \vdots \\
& \oplus a_{1\ldots n} x_1 x_2 \cdot \ldots \cdot x_n.
\end{aligned}
$$

Herleitung der komplementfreien RNF

1. Bestimme DNF über einschlägige Indizes
2. Ersetze $+$ durch $\oplus$
3. Ersetze $\overline{x}$ durch $x \oplus 1$
4. Multipliziere aus unter Verwendung der Äquivalenzen für $\oplus$.

Beispiel:

$$\begin{aligned}
f(x_1, x_2, x_3) &= \overline{x}_1 x_2 x_3 + x_1 \overline{x}_2 x_3 + x_1 x_2 x_3 \\
&= \overline{x}_1 x_2 x_3 \oplus x_1 \overline{x}_2 x_3 \oplus x_1 x_2 x_3 \\
&= (x_1 \oplus 1) x_2 x_3 \oplus x_1 (x_2 \oplus 1) x_3 \oplus x_1 x_2 x_3 \\
&= x_1 x_2 x_3 \oplus x_2 x_3 \oplus x_1 x_2 x_3 \oplus x_1 x_3 \oplus x_1 x_2 x_3 \\
&= x_2 x_3 \oplus x_1 x_3 \oplus x_1 x_2 x_3.
\end{aligned}$$