Rechnerstrukturen

Michael Engel und Peter Marwedel

TU Dortmund, Fakultät für Informatik

SS 2013

Hinweis: *Folien a. d. Basis von Materialien von Gernot Fink und Thomas Jansen*

29. April 2013

1. Rechnerarithmetik
   - Gleitkommazahlen-Arithmetik (Wiederholung)
   - Gleitkommaarithmetik: Fehlerquellen

2. Optimierung von Schaltnetzen
   - Einleitung
   - Algebraische Vereinfachungen

3. KV-Diagramme
   - Beschreibung und Beispiel
   - Minimalpolynome

Gleitkommazahlen-Arithmetik

Darstellung gemäß IEEE 754-1985

$$
\begin{aligned}
x &= (-1)^{s_x} \cdot m_x \cdot 2^{e_x} \\
y &= (-1)^{s_y} \cdot m_y \cdot 2^{e_y}
\end{aligned}
$$

s Vorzeichenbit

m Mantisse (Binärdarstellung, inklusive impliziter 1)

e Exponent (Exzessdarstellung, $b = 2^{l-1} - 1$)

Ergebnis $z = (-1)^{s_z} \cdot m_z \cdot 2^{e_z}$

Vereinfachung Wir ignorieren das Runden.

Aber: Wichtiger Teil des IEEE 754 Standards!

Weitere Details z.B. in:

David Goldberg (1991): What every computer scientist should know about floating-point arithmetic. ACM Computing Serveys 23(1):5–48.

Multiplikation von Gleitkommazahlen

$$
\begin{aligned}
x &= (-1)^{s_x} \cdot m_x \cdot 2^{e_x} \\
y &= (-1)^{s_y} \cdot m_y \cdot 2^{e_y} \\
z = x \cdot y &= (-1)^{s_z} \cdot m_z \cdot 2^{e_z}
\end{aligned}
$$

Beobachtung $z = (-1)^{s_x \oplus s_y} \cdot (m_x \cdot m_y) \cdot 2^{e_x + e_y}$

also

1. $s_z := s_x \oplus s_y$ (trivial)
2. $m_z := m_x \cdot m_y$ (Multiplikation von Betragszahlen wie gesehen, implizite Einsen nicht vergessen!)
3. $e_z := e_x + e_y$ (Addition, wegen Exzessdarstellung $e_x + e_y - b$ berechnen)

Addition von Gleitkommazahlen

$$\begin{aligned} x &= (-1)^{s_x} \cdot m_x \cdot 2^{e_x} \\ y &= (-1)^{s_y} \cdot m_y \cdot 2^{e_y} \\ z = x + y &= (-1)^{s_z} \cdot m_z \cdot 2^{e_z} \end{aligned}$$

Beobachtung einfach, wenn $e_x = e_y$
dann $m_1 \cdot 2^e + m_2 \cdot 2^e = (m_1 + m_2) \cdot 2^e$

Plan

1. Ergebnis wird „so ähnlich" wie Zahl mit größerem Exponenten, darum Mantisse der Zahl mit kleinerem Exponenten anpassen
2. Mantissen auf jeden Fall addieren, bei unterschiedlichen Vorzeichen dazu eine Mantisse negieren (Zweierkomplement)
3. anschließend normalisieren

Algorithmus zur Addition

1. Falls $e_x < e_y$, dann x und y komplett vertauschen.
2. Falls Vorzeichen ungleich, dann Vorzeichen von s_y invertieren und Übergang von y zu $-y$ im Zweierkomplement („$\overline{y} + 1$"). $s_z := s_x$
3. Mantisse m_y um $e_x - e_y$ Stellen nach rechts verschieben (Exponenten „virtuell" jetzt angeglichen)
   Achtung   Kann zum "Verlust" signifikanter Stellen führen!
4. $m_z := m_x + m_y$
   Falls $e_x = e_y$, Vorzeichenwechsel möglich. Dann s_z invertieren.
5. $e_z := e_x$. Ergebnis normalisieren

Achtung   Bei Mantissen an implizite Einsen denken!

klar   Keine separate Subtraktion erforderlich.
Vorzeichenwechsel trivial.
Addition negativer Zahlen enthalten durch Zweierkomplement.

Fehlerquellen bei der Gleitkommaarithmetik

- Rundung, wenn Berechnungsergebnis zur korrekten Darstellung *mehr* signifikante Bits (d.h. i.d. Mantisse) erfordert als verfügbar (kann bei Multiplikation *und* Addition auftreten)
- "Verlust" niederwertiger Bits durch Angleich der Exponenten während der Addition

  Worst Case: $x \gg y$ und $y \neq 0$ aber $x + y = x$ ⚡

Und nicht zu vergessen: Darstellung nur einer *extrem kleinen* Auswahl der rationalen Zahlen möglich, variiert mit der Größenordnung der repräsentierten Zahlen!

Probleme bei der Addition: Ein Szenario

Gegeben: Folge von n Gleitkommazahlen $[x_i]$ mit $0 \leq i \leq n$
(z.B. gespeichert in einem Feld/Array `x[i]`)

Aufgabe: Berechne Summe S ... möglichst exakt
(d.h. mit den Möglichkeiten der Gleitkommaarithmetik)

Naive Lösung: Direkte Summation, d.h. berechne:

$$S\uparrow = \sum_{i=0}^{n-1} x_i \quad \text{oder lieber} \quad S\downarrow = \sum_{i=n-1}^{0} x_i$$

~~Theorie/Intuition: Beide Summationen liefern dasselbe Ergebnis!~~

Praxis: $S\uparrow$ und $S\downarrow$ sind i.a. nicht gleich! $\longrightarrow$ Beispielprogramm (in C)

Mögliche Abhilfe: Erhöhung der Genauigkeit (i.d.R. schwierig)
... oder "schlauere" Berechnung ;-)

Fehlerreduktion: Kahan-Summation

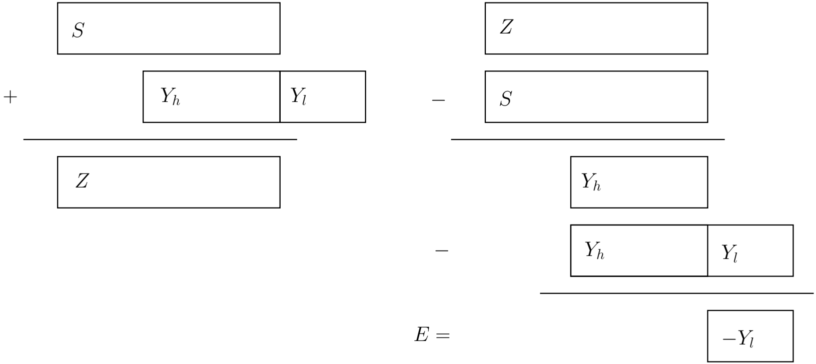
Algorithmus zur numerisch stabileren Berechnung von $S = \sum_{i=0}^{n-1} x_i$:

```
S = 0;                    /* Summe */
E = 0;                    /* geschaetzter Fehler */
for i = 0 to n-1 {
        Y = x[i] - E;     /* bish. Fehler beruecksichtigen *
        Z = S + Y;        /* neues Summationsergebnis */
        E = (Z - S) - Y;  /* neue Fehlerschaetzung */
        S = Z;
        }
```

$\longrightarrow$ Beispielprogramm (in C)

Fehlerreduktion: Kahan-Summation (2)

Veranschaulichung des fehlerkompensierenden Berechnungsablaufs:



Optimierung von Schaltnetzen

Was bedeutet Optimierung?

klar bestmögliche Lösung finden

also

1. Lösungen finden
2. beweisen, dass es keine bessere gibt

hier meist nur
Verbesserung, nicht Optimierung

Strukturierter Schaltnetz-Entwurf

Schaltnetz-Entwurf bisher

- ad hoc
- Normalformen

Wunsch Systematisierung, Strukturierung

Hoffnungen

- einfacher zu guten Entwürfen
- Schaltnetze verständlicher
- Schaltnetze besser verifizierbar

Systematisierung Schaltnetz-Entwurf

banale(?) Grundidee Wiederverwendung guter Schaltnetze als Komponenten

klar schon gemacht
(z. B. bei allen Addierern HA verwendet)

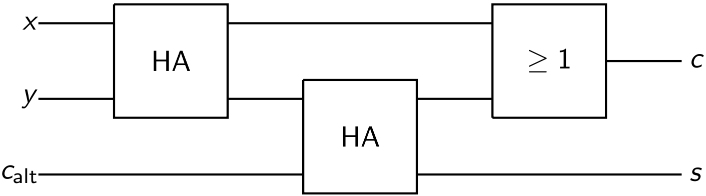
zum Einstieg noch elementareres Beispiel

Kann man VA *sinnvoll* aus HA bauen?

Noch einmal zum Volladdierer

			VA „$c_{\text{alt}} + x + y$“		HA „$x + y$“		HA „$c_{\text{alt}} + A_s$“		
c_{alt}	x	y	c	s	A_c	A_s	B_c	B_s	$A_c \vee B_c$
0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0	1	0
0	1	0	0	1	0	1	0	1	0
0	1	1	1	0	1	0	0	0	1
1	0	0	0	1	0	0	0	1	0
1	0	1	1	0	0	1	1	0	1
1	1	0	1	0	0	1	1	0	1
1	1	1	1	1	1	0	0	1	1

Strukturierter Volladdierer



Größe 5 Tiefe 3

Erinnerung Halbaddierer
Größe 2, Tiefe 1

Erinnerung „alter“ Volladdierer
Größe 5, Tiefe 3

also immerhin nichts verloren
im Vergleich zum sorgfältigen „ad hoc-Entwurf“

Multiplexer

Erinnerung vereinfachte Wertetabelle

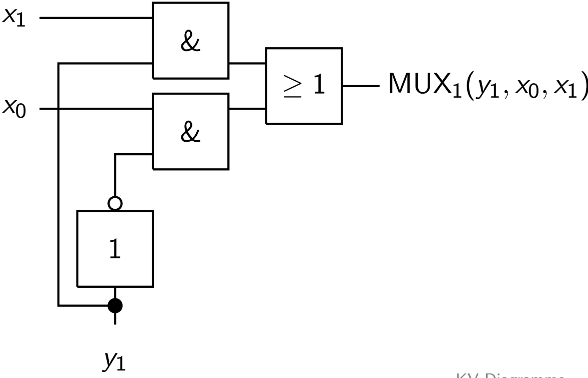
y_1	$\mathrm{MUX}_1(y_1, x_0, x_1)$
0	x_0
1	x_1

normale (ausführliche) Wertetabelle

y_1	x_1	x_0	$\mathrm{MUX}_1(y_1, x_0, x_1)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$\mathrm{MUX}_1\colon \{0,1\}^3 \to \{0,1\}$

y_1	x_1	x_0	$\mathrm{MUX}_1(y_1, x_0, x_1)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



Strukturiert zum Schaltnetz für MUX_2

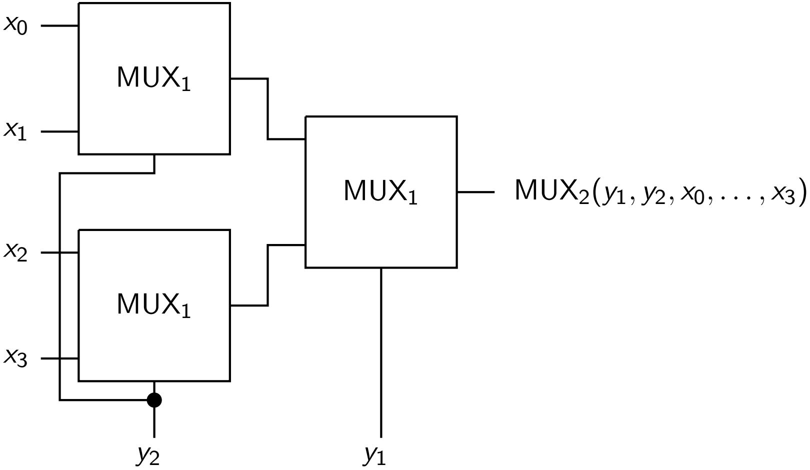
y_1	y_2	$\text{MUX}_2(y_1, y_2, x_0, x_1, x_2, x_3)$
0	0	x_0
0	1	x_1
1	0	x_2
1	1	x_3

Beobachtung

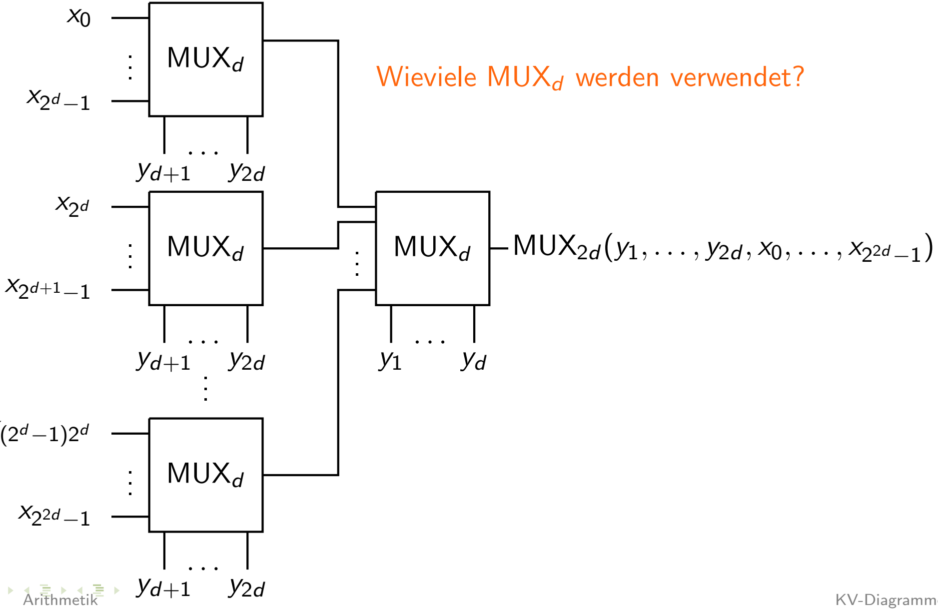
- $y_2 = 0 \Rightarrow \text{MUX}_2(y_1, y_2, x_0, x_1, x_2, x_3) \in \{x_0, x_2\}$
- $y_2 = 1 \Rightarrow \text{MUX}_2(y_1, y_2, x_0, x_1, x_2, x_3) \in \{x_1, x_3\}$

also ein MUX_1 wählt mittels y_2 aus $\{x_0, x_1\}$
ein MUX_1 wählt mittels y_2 aus $\{x_2, x_3\}$
ein MUX_1 wählt mittels y_1 aus den Ergebnissen

Strukturierter MUX_2



Strukturierter MUX_{2d}



Wieviele MUX_d werden verwendet?

Entwurf kleiner Schaltnetze

Erinnerung Normalformen (DNF, RNF, KNF) direkt in Schaltnetz umsetzbar

klar auch dabei jede Variable höchstens einmal negieren

ab jetzt Wir zählen Negationsgatter nicht mehr.
klar für große Schaltnetze nicht wesentlich
klar höchstens „Fehler“ Summand n

dann Schaltnetzgröße $\widehat{=}$ Anzahl Minterme/Maxterme $+ 1$

Verbesserung äquivalenter kleinerer boolescher Ausdruck führt zu kleinerem Schaltnetz

Vorsicht Kleinerer Ausdruck ist keine Normalform mehr!

Algebraische Vereinfachungen

Erinnerung Rechengesetze für boolesche Algebra (Satz 3)

Beispiel $f\colon \{0,1\}^4 \to \{0,1\}$ mit Wertevektor
$(1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0)$

DNF dazu
einschlägige Indizes $0,\ 2,\ 4,\ 6,\ 8,\ 9,\ 10,\ 11$

Minterm zu i ist Funktion $m_i\colon \{0,1\}^4 \to \{0,1\}$ mit
$m_i(x_1, x_2, x_3, x_4) = 1 \Leftrightarrow (x_1\, x_2\, x_3\, x_4)_2 = i$

am Beispiel $i = 2$ $m_2(x_1, x_2, x_3, x_4) = \overline{x_1}\,\overline{x_2}\,x_3\,\overline{x_4}$

denn $\overline{x_1}\,\overline{x_2}\,x_3\,\overline{x_4} = 1$

$\Leftrightarrow (x_1 = 0) \wedge (x_2 = 0) \wedge (x_3 = 1) \wedge (x_4 = 0)$

$\Leftrightarrow (x_1\, x_2\, x_3\, x_4)_2 = 2$

Algebraische Vereinfachung

$f\colon \{0,1\}^4 \to \{0,1\}$ mit $(1,0,1,0,1,0,1,0,1,1,1,1,0,0,0,0)$

DNF dazu

$$f(x_1,x_2,x_3,x_4) = \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4} \vee \overline{x_1}\,\overline{x_2}\,x_3\,\overline{x_4} \vee \overline{x_1}\,x_2\,\overline{x_3}\,\overline{x_4} \vee \overline{x_1}\,x_2\,x_3\,\overline{x_4}$$
$$\vee\, x_1\,\overline{x_2}\,\overline{x_3}\,\overline{x_4} \vee x_1\,\overline{x_2}\,\overline{x_3}\,x_4 \vee x_1\,\overline{x_2}\,x_3\,\overline{x_4} \vee x_1\,\overline{x_2}\,x_3\,x_4$$

Vereinfacht $f(x_1,x_2,x_3,x_4) = \overline{x_1}\,\overline{x_2}\,\overline{x_4} \vee \overline{x_1}\,x_2\,\overline{x_4} \vee x_1\,\overline{x_2}\,\overline{x_3} \vee x_1\,\overline{x_2}\,x_3$

Größe (in neuer Zählweise) $8+1=9$, vereinfacht: $4+1=5$

Wie können wir das verkleinern?

Resolution $x_i\,x_j \vee \overline{x_i}\,x_j = x_j$

Algebraische Vereinfachung

$f\colon \{0,1\}^4 \to \{0,1\}$ mit $(1,0,1,0,1,0,1,0,1,1,1,1,0,0,0,0)$

dazu
$f(x_1,x_2,x_3,x_4) = \overline{x_1}\,\overline{x_2}\,\overline{x_4} \vee \overline{x_1}\,x_2\,\overline{x_4} \vee x_1\,\overline{x_2}\,\overline{x_3} \vee x_1\,\overline{x_2}\,x_3$

Vereinfacht $f(x_1,x_2,x_3,x_4) = \overline{x_1}\,\overline{x_4} \vee x_1\,\overline{x_2}$

Größe (in neuer Zählweise) $4+1=5$, vereinfacht: $2+1=3$
Wie können wir das verkleinern?

Resolution $x_i\,x_j \vee \overline{x_i}\,x_j = x_j$

klar Größe $9 \rightsquigarrow 3$ beeindruckend
aber recht schwierig

KV-Diagramme

Maurice Karnaugh (1953)
Edward W. Veitch (1952)

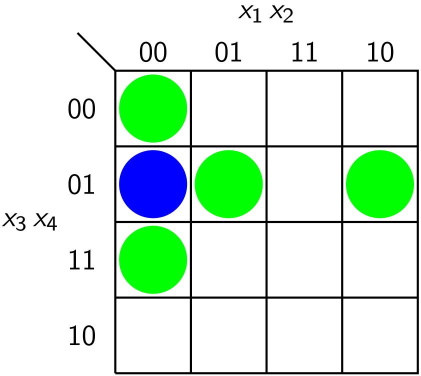
KV-Diagramme — systematischer, anschaulicher und viel übersichtlicherer Weg, Funktionen $f\colon \{0,1\}^3 \to \{0,1\}$ und $f\colon \{0,1\}^4 \to \{0,1\}$ zu vereinfachen

aber — schon für $f\colon \{0,1\}^5 \to \{0,1\}$ unübersichtlich

$\Rightarrow$ darum vor allem im HaPra wichtig.

KV-Diagramm für $f\colon \{0,1\}^4 \to \{0,1\}$

Nachbarschaften, d.h. Variablenbelegungen unterscheiden sich nur in einer Stelle



Hinweis: Nicht alle möglichen Nachbarschaften gezeigt!

Beispiel KV-Diagramm für $f\colon \{0,1\}^4 \to \{0,1\}$

$f\colon \{0,1\}^4 \to \{0,1\}$ mit $(1,0,1,0,1,0,1,0,1,1,1,1,0,0,0,0)$
(Beispielfunktion wie bereits gesehen)

x_1	x_2	x_3	x_4	$f(x_1,\dots,x_4)$
0	0	0	0	1
	$\vdots$			$\vdots$
1	0	1	1	1
	$\vdots$			$\vdots$
1	1	1	1	0

$x_3\,x_4$ \ $x_1\,x_2$	00	01	11	10
00	1	1	0	1
01	0	0	0	1
11	0	0	0	1
10	1	1	0	1

Nullen weglassen
für mehr Übersichtlichkeit

Was fällt bei der Variablenbelegung auf? Warum ist das so?

KV-Diagramm für Beispielfunktion $f\colon \{0,1\}^4 \to \{0,1\}$

$x_3\,x_4$ \ $x_1\,x_2$	00	01	11	10
00	1	1		1
01				1
11				1
10	1	1		1

$x_1 = 1, x_2 = 0 \Rightarrow x_1\,\overline{x_2}$

$x_1 = 0, x_4 = 0 \Rightarrow \overline{x_1}\,\overline{x_4}$

$x_2 = 0, x_4 = 0 \Rightarrow \overline{x_2}\,\overline{x_4}$

$f(x_1, x_2, x_3, x_4)$
$= x_1\,\overline{x_2} \vee \overline{x_1}\,\overline{x_4}$

Suche alle größten Rechtecke mit Zweierpotenzlänge.
Bilde für jedes Rechteck passendes Monom.
Decke alle Einsen durch sparsame Rechteckauswahl ab.
Bilde f als Disjunktion der korrespondierenden Monome.

Einordnung KV-Diagramme

Was leisten KV-Diagramme?

Beobachtung Wir lösen mit KV-Diagrammen ein Problem, das wir noch gar nicht definiert haben.

klar Wir holen das jetzt nach.

vorab Begriffsfestlegungen

- Variable
  Beispiele $x_1, x_2, x_3, \ldots$
- Literale Variable und Negationen
  Beispiele $x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots$
- Monom Konjunktion einiger Literale
  Beispiele $x_1 \overline{x_3} x_4$, x_2
- Polynom Disjunktion einiger Monome
  Beispiel $\overline{x_1}\, x_3 \vee x_2 \vee \overline{x_4}\, x_5$

Weitere Begriffsdefinitionen

Wir haben schon Variable, Literal, Monom, Polynom.

- Implikant von f Monom m mit folgender Eigenschaft: $\forall x \in \{0,1\}^n\colon m(x) = 1 \Rightarrow f(x) = 1$
  Beispiel Monom eines Polynoms für f
- Verkürzung eines Monoms m Monom m', für das m Implikant ist
  Beispiel $x_1 \overline{x_3}$ ist Verkürzung von $x_1 \overline{x_3}$
- echte Verkürzung eines Monoms m Monom m', das Verkürzung von m ist und echt weniger Literale enthält
  Beispiel $\overline{x_3}$ ist echte Verkürzung von $x_1 \overline{x_3}$
- Primimplikant von f Implikant von f, für den es keine echte Verkürzung gibt, die auch Implikant von f ist
  Beispiel $\overline{x_2}\, x_3$ ist Primimplikant von $x_1 \vee \overline{x_2}\, x_3$

Verbindung zu Schaltnetzen

Erinnerung   Wir zählen keine Negationen mehr.

Monom mit i Literalen   Und-Gatter mit Fan-In i oder
$i - 1$ Und-Gatter mit Fan-In 2
angemessen   Kosten i

Polynom mit j Monomen   zusätzlich Oder-Gatter mit Fan-In j oder
$j - 1$ Oder-Gatter mit Fan-In 2
angemessen   zusätzlich Kosten j

also   direkte Verbindung zwischen
Polynom-Kosten und Schaltnetz-Kosten

noch ein letzter Begriff

Minimalpolynom zu f Polynom für f mit minimalen Kosten
unter allen Polynomen für f

Minimalpolynome

also Minimalpolynome liefern „günstigste“ Schaltnetze...

Vorsicht Stimmt nicht so ganz!
nur richtig, wenn man sich auf
direkte Polynomrealisierung einschränkt

trotzdem Wir suchen Minimalpolynome.

Wie finden wir systematisch Minimalpolynome?

Minimalpolynome und Primimplikanten

Theorem Minimalpolynome enthalten nur Primimplikanten.

Beweis durch Widerspruch

Annahme $f = m_1 \vee m_2 \vee \cdots \vee m_i$ Minimalpolynom,
m_1 kein Primimplikant zu f

gemäß Definition $\exists m'$: m_1 ist Implikant von m',
m' ist echte Verkürzung von m_1 und
m' ist Implikant von f

klar $m_1 = 1 \Rightarrow m' = 1$, da m_1 Implikant von m'
Beobachtung $m' = 1 \Rightarrow f = 1$, da m' Implikant von f ist
also $m' \vee m_2 \vee \cdots \vee m_i$ ist günstigeres Polynom für f
Widerspruch zur Voraussetzung, dass $m_1 \vee m_2 \vee \cdots \vee m_i$
Minimalpolynom für f $\square$

Minimalpolynomberechnung

Idee für Minimalpolynomberechnung

1. Berechne alle Primimplikanten von f.
2. Berechne günstigste „Überdeckung“ von f mit diesen Monomen.

Beobachtung Dieser Ansatz ist sicher schlecht, wenn
das Minimalpolynom zu f klein ist,
f aber viele Primimplikanten hat.

Wir verfolgen diesen Ansatz dennoch.

Minimalpolynomberechnung

Behauptung Wir haben mit KV-Diagrammen Minimalpolynome berechnet.

klar Wir haben günstigste Überdeckung von f gesucht.

offen Entsprechen maximale Rechtecke mit Zweierpotenzseitenlängen genau Primimplikanten?

klar Für $f\colon \{0,1\}^4 \to \{0,1\}$ gibt es Monome der Längen 0, 1, 2, 3 und 4.

Wir schauen uns die Situation für jede mögliche Monomlänge an.

Primimplikanten der Länge 0

$x_3 x_4$ \ $x_1 x_2$	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

Monom 1

Primimplikanten der Länge 1

$x_3\,x_4$ \ $x_1\,x_2$	00	01	11	10
00	1	1	1	1
01			1	1
11			1	1
10	1	1	1	1

Monom x_1

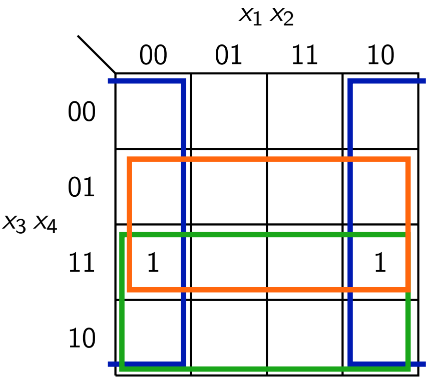
Monom $\overline{x_4}$

Primimplikanten der Länge 2

$x_3\,x_4$ \ $x_1\,x_2$	00	01	11	10
00				
01				
11	1	1		
10	1	1		

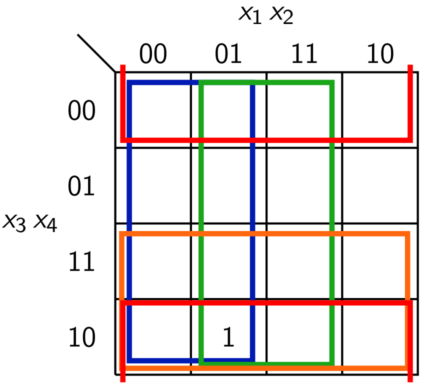
Monom $\overline{x_1}\, x_3$

Primimplikanten der Länge 3



Monom $\overline{x_2}\, x_3\, x_4$

Primimplikanten der Länge 4



Monom $\overline{x_1}\,x_2\,x_3\,\overline{x_4}$

Beobachtung: Primimplikanten entsprechen genau KV-Rechtecken

Minimalpolynombestimmung mit KV-Diagramm

Aufgabe Bestimme für $f\colon \{0,1\}^4 \to \{0,1\}$ ein Minimalpolynom.

Vorgehen

1. Eintragen der Funktion ins KV-Diagramm
2. Finden aller maximaler Zweierpotenz-Rechtecke
3. Finden eines Primimplikanten für jedes Rechteck
4. Finden einer Überdeckung aller Einsen durch eine minimale Monomauswahl

jetzt noch ein Beispiel
$f\colon \{0,1\}^4 \to \{0,1\}$ mit $(1,0,1,1,0,1,0,1,0,1,0,1,0,0,1,1)$

Eintragen der Funktion ins KV-Diagramm

$f : \{0,1\}^4 \to \{0,1\}$ mit $(\overset{0}{1}, \overset{1}{0}, \overset{2}{1}, \overset{3}{1}, \overset{4}{0}, \overset{5}{1}, \overset{6}{0}, \overset{7}{1}, \overset{8}{0}, \overset{9}{1}, \overset{10}{0}, \overset{11}{1}, \overset{12}{0}, \overset{13}{0}, \overset{14}{1}, \overset{15}{1})$

$x_3\,x_4$ \ $x_1\,x_2$	00	01	11	10
00	1			
01		1		1
11	1	1	1	1
10	1		1	

$0 = (00\,00)_2$
$1 = (00\,01)_2$
$2 = (00\,10)_2$
$3 = (00\,11)_2$
$4 = (01\,00)_2$
$5 = (01\,01)_2$
$6 = (01\,10)_2$
$7 = (01\,11)_2$
$8 = (10\,00)_2$
$9 = (10\,01)_2$
$10 = (10\,10)_2$
$11 = (10\,11)_2$
$12 = (11\,00)_2$
$13 = (11\,01)_2$
$14 = (11\,10)_2$
$15 = (11\,11)_2$

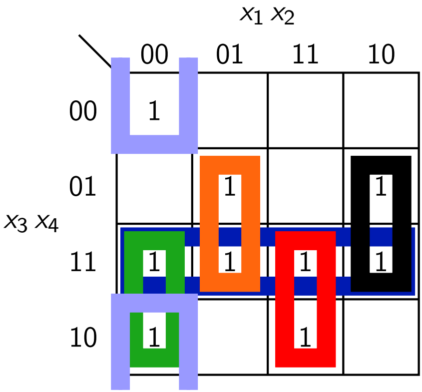
Finden aller maximaler Zweierpotenz-Rechtecke

$f : \{0,1\}^4 \to \{0,1\}$ mit $(1,0,1,1,0,1,0,1,0,1,0,1,0,0,1,1)$

$x_3\,x_4$ \ $x_1\,x_2$	00	01	11	10
00	1			
01		1		1
11	1	1	1	1
10	1		1	

Finden eines Primimplikanten für jedes Rechteck

$f : \{0,1\}^4 \to \{0,1\}$ mit $(1,0,1,1,0,1,0,1,0,1,0,1,0,0,1,1)$



$x_3\,x_4$

$\overline{x_1}\,\overline{x_2}\,x_3$

$\overline{x_1}\,x_2\,x_4$

$x_1\,x_2\,x_3$

$x_1\,\overline{x_2}\,x_4$

$\overline{x_1}\,\overline{x_2}\,\overline{x_4}$

Überdeckung aller Einsen durch minimale Monomauswahl

$f\colon \{0,1\}^4 \to \{0,1\}$ mit $(1,0,1,1,0,1,0,1,0,1,0,1,0,0,1,1)$

$x_3\,x_4$ \ $x_1\,x_2$	00	01	11	10
00	1			
01		1		1
11	1	1	1	1
10	1		1	

$x_3\,x_4$	beste Wahl
$\overline{x_1}\,\overline{x_2}\,x_3$	
$\overline{x_1}\,x_2\,x_4$	**erforderlich**
$x_1\,x_2\,x_3$	**erforderlich**
$x_1\,\overline{x_2}\,x_4$	**erforderlich**
$\overline{x_1}\,\overline{x_2}\,\overline{x_4}$	**erforderlich**

also $x_3\,x_4 \lor \overline{x_1}\,x_2\,x_4 \lor x_1\,x_2\,x_3 \lor x_1\,\overline{x_2}\,x_4 \lor \overline{x_1}\,\overline{x_2}\,\overline{x_4}$ Minimalpolynom

Fazit KV-Diagramme

Mit KV-Diagrammen effizient beide Schritte zur Minimalpolynomberechnung durchführbar

1. Bestimmung aller Primimplikanten
2. Bestimmung einer minimalen Überdeckung

für Funktionen $f\colon \{0,1\}^n \to \{0,1\}$ für $n \in \{3,4\}$

klar Das reicht nicht aus.

Wie bestimmen wir Minimalpolynome für $f\colon \{0,1\}^n \to \{0,1\}$ für größeres n?

klar Wir suchen einen Algorithmus.