Technische Informatik
Minimierung mit Quine-McCluskey

Thorsten Thormählen

22. November 2022

Teil 5, Kapitel 3

Dies ist die Druck-Ansicht.

Aktiviere Präsentationsansicht

Steuerungstasten

→ nächste Folie (auch Enter oder Spacebar).
← vorherige Folie
d schaltet das Zeichnen auf Folien ein/aus
p wechselt zwischen Druck- und Präsentationsansicht
CTRL + vergrößert die Folien
CTRL - verkleinert die Folien
CTRL 0 setzt die Größenänderung zurück

Notation

Typ	Schriftart	Beispiele
Variablen (Skalare)	kursiv	a, b, x, y
Funktionen	aufrecht	$\mathrm{f}, \mathrm{g}(x), \max(x)$
Vektoren	fett, Elemente zeilenweise	$\mathbf{a}, \mathbf{b} = \begin{pmatrix} x \\ y \end{pmatrix} = (x, y)^\top$, $\mathbf{B} = (x, y, z)^\top$
Matrizen	Schreibmaschine	$\mathtt{A}, \mathtt{B} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$
Mengen	kalligrafisch	$\mathcal{A}, B = \{a, b\}, b \in \mathcal{B}$
Zahlenbereiche, Koordinatenräume	doppelt gestrichen	$\mathbb{N}, \mathbb{Z}, \mathbb{R}^2, \mathbb{R}^3$

Inhalt

- Minimierung eines booleschen Ausdrucks mit dem Quine-McCluskey-Verfahren
  - Finden der Primimplikanten (Phase 1)
  - Aufstellen der Primimplikantentafel (Phase 2)
  - Lösen des Überdeckungsproblems (Phase 3)
  - Lösen zyklischer Überdeckungsprobleme mit dem Verfahren von Petrick

Quine-McCluskey-Verfahren

- Das Quine-McCluskey-Verfahren erlaubt, boolesche Funktionen zu minimieren
- Während bei KV-Diagrammen Funktionen mit bis zu 4 Variablen leicht minimiert werden können, ist das Quine-McCluskey-Verfahren auch für Funktionen mit mehr Variablen geeignet
- Großer Vorteil des Quine-McCluskey-Verfahrens ist, dass es sich relativ einfach auf einem Computer implementieren lässt

Quine-McCluskey-Verfahren

Eingabe: Eine Funktion $y = \mathrm{f}(x_{n-1}, \ldots, x_1, x_0)$ der Stelligkeit n

Angabe z.B. durch DNF
Beispiel:

$$
\begin{aligned}
y = \mathrm{f}(x_{n-1}, \ldots, x_1, x_0) = \\
&= m_0 \vee m_4 \vee m_6 \vee m_{11} \vee m_{12} \vee m_{13} \vee m_{14} \\
&= \bar{x}_3\bar{x}_2\bar{x}_1\bar{x}_0 \vee \bar{x}_3 x_2 \bar{x}_1 \bar{x}_0 \vee \\
&\quad \bar{x}_3 x_2 x_1 \bar{x}_0 \vee x_3 \bar{x}_2 x_1 x_0 \vee \\
&\quad x_3 x_2 \bar{x}_1 \bar{x}_0 \vee x_3 x_2 \bar{x}_1 x_0 \vee \\
&\quad x_3 x_2 x_1 \bar{x}_0
\end{aligned}
$$

oder Wahrheitstafel (siehe rechts)

Ausgabe: Ein minimaler boolescher Ausdruck als ODER-Verknüpfung von Monomen (= UND-Verknüpfungen von Literalen)

Möglichst wenig ODER-Verknüpfungen

Möglichst wenig UND-Verknüpfungen

	x_3	x_2	x_1	x_0	y
0:	0	0	0	0	1
1:	0	0	0	1	0
2:	0	0	1	0	0
3:	0	0	1	1	0
4:	0	1	0	0	1
5:	0	1	0	1	0
6:	0	1	1	0	1
7:	0	1	1	1	0
8:	1	0	0	0	0
9:	1	0	0	1	0
10:	1	0	1	0	0
11:	1	0	1	1	1
12:	1	1	0	0	1
13:	1	1	0	1	1
14:	1	1	1	0	1
15:	1	1	1	1	0

Quine-McCluskey-Verfahren

Die Minimierung nach dem Quine-McCluskey-Verfahren läuft in 3 Phasen ab:

- Phase 1: Finden der Primimplikanten
- Phase 2: Aufstellen der Primimplikantentafel
- Phase 3: Lösen des Überdeckungsproblems

Phase 1: Finden der Primimplikanten

Um die Primimplikanten zu finden, wird (wie bei den KV-Diagrammen) das Vereinigungstheorem verwendet:

$$(a \wedge b) \vee (a \wedge \neg b) = a$$

Bei KV-Diagrammen wird das Vereinigungstheorem rekursiv angewendet, indem immer größere Blöcke aus benachbarten Blöcken gebildet werden

Beim Quine-McCluskey-Verfahren wird analog vorgegangen, nur dass statt mit graphischen Blöcken mit Tabelleneinträgen gearbeitet wird

Phase 1: Finden der Primimplikanten

- Die erste Tabelle enthält die Implikanten der Ordnung 0. Sie wird aus der 1-Menge der Wahrheitstafel extrahiert
- Die zweite Tabelle enthält die Implikanten der Ordnung 1. Sie wird aus der vorgegangenen Tabelle konstruiert, indem diejenigen Zeilen zusammengefasst werden, die sich in genau einer Variablen unterscheiden
- Implikanten, die zusammengefasst wurden, werden geeignet markiert (hier wird das Zeichen → verwendet)
- Selbst wenn Implikanten schon zusammengefasst wurden, stehen sie noch zur Verfügung, um mit anderen Implikanten zusammengefasst zu werden
- Implikanten, die nicht zusammengefasst wurden, sind die gesuchten Primimplikanten. Sie werden ebenfalls markiert (hier wird das Zeichen ✓ verwendet)
- Dieses Verfahren wird rekursiv für Implikanten höherer Ordnung fortgesetzt bis keine Zusammenfassung mehr möglich ist

Phase 1: Finden der Primimplikanten

Wahrheitstafel:

	x_3	x_2	x_1	x_0	y
0:	0	0	0	0	1
1:	0	0	0	1	0
2:	0	0	1	0	0
3:	0	0	1	1	0
4:	0	1	0	0	1
5:	0	1	0	1	0
6:	0	1	1	0	1
7:	0	1	1	1	0
8:	1	0	0	0	0
9:	1	0	0	1	0
10:	1	0	1	0	0
11:	1	0	1	1	1
12:	1	1	0	0	1
13:	1	1	0	1	1
14:	1	1	1	0	1
15:	1	1	1	1	0

Implikanten (Ordnung 0):

	x_3	x_2	x_1	x_0	
0:	0	0	0	0	$\rightarrow$
4:	0	1	0	0	$\rightarrow$
6:	0	1	1	0	$\rightarrow$
11:	1	0	1	1	✓
12:	1	1	0	0	$\rightarrow$
13:	1	1	0	1	$\rightarrow$
14:	1	1	1	0	$\rightarrow$

Phase 1: Finden der Primimplikanten

Implikanten (Ordnung 0):

	x_3	x_2	x_1	x_0	
0:	0	0	0	0	→
4:	0	1	0	0	→
6:	0	1	1	0	→
11:	1	0	1	1	✓
12:	1	1	0	0	→
13:	1	1	0	1	→
14:	1	1	1	0	→

Implikanten (Ordnung 1):

	x_3	x_2	x_1	x_0	
0, 4:	0	-	0	0	✓
4, 6:	0	1	-	0	→
4, 12:	-	1	0	0	→
6, 14:	-	1	1	0	→
12, 13:	1	1	0	-	✓
12, 14:	1	1	-	0	→

Implikanten (Ordnung 2):

	x_3	x_2	x_1	x_0	
4, 6, 12, 14:	-	1	-	0	✓

Phase 2: Aufstellen der Primimplikantentafel

Die Primimplikatentafel besteht aus allen Primimplikanten
(diese wurden in Phase 1 mit ✓ markiert)

Zeilen:

- Pro Primimplikant jeweils eine Zeile

Spalten:

- Jede Spalte steht für einen Minterm m_i der DNF
- In jeder Spalte wird markiert, ob der Primimplikant den Minterm überdeckt
  (hier wird das Zeichen ○ verwendet)
- Ist der Primimplikant der einzige Primimplikant, der diesen Minterm überdeckt, so wird er "essentieller Primimplikant" genannt und besonders markiert
  (hier wird das Zeichen ● verwendet)

Essentielle Primimplikanten müssen in jedem Fall Teil des minimalen booleschen Ausdrucks sein und werden sofort übertragen

Phase 2: Aufstellen der Primimplikantentafel

Implikanten (Ordnung 0):

	x_3	x_2	x_1	x_0	
0:	0	0	0	0	→
4:	0	1	0	0	→
6:	0	1	1	0	→
11:	1	0	1	1	✓
12:	1	1	0	0	→
13:	1	1	0	1	→
14:	1	1	1	0	→

Implikanten (Ordnung 1):

	x_3	x_2	x_1	x_0	
0, 4:	0	-	0	0	✓
4, 6:	0	1	-	0	→
4, 12:	-	1	0	0	→
6, 14:	-	1	1	0	→
12, 13:	1	1	0	-	✓
12, 14:	1	1	-	0	→

Implikanten (Ordnung 2):

	x_3	x_2	x_1	x_0	
4, 6, 12, 14:	-	1	-	0	✓

Primimplikantentafel:

	x_3	x_2	x_1	x_0	0	4	6	11	12	13	14	
4, 6, 12, 14:	-	1	-	0		○	●		○		●	$(x_2\bar{x}_0)$
0, 4:	0	-	0	0	●	○						$(\bar{x}_3\bar{x}_1\bar{x}_0)$
12, 13:	1	1	0	-					○	●		$(x_3x_2\bar{x}_1)$
11:	1	0	1	1				●				$(x_3\bar{x}_2x_1x_0)$

Extrahierte essentielle Primimplikanten: $(\bar{x}_3\bar{x}_1\bar{x}_0)$, $(x_2\bar{x}_0)$, $(x_3\bar{x}_2x_1x_0)$, $(x_3x_2\bar{x}_1)$

Phase 2: Aufstellen der Primimplikantentafel

Ist es zufällig so, dass die essentiellen Primimplikanten gemeinsam alle Minterme überdecken, kann der Algorithmus bereits hier beendet werden

Der minimale boolesche Ausdruck für das gezeigte Beispiel lautet daher:
$y = (\bar{x}_3\bar{x}_1\bar{x}_0) \vee (x_2\bar{x}_0) \vee (x_3\bar{x}_2x_1x_0) \vee (x_3x_2\bar{x}_1)$

Ist dies nicht der Fall, muss in Phase 3 eine geeignete Überdeckung gefunden werden

Phase 3: Lösen des Überdeckungsproblems

- Die essentiellen Primimplikanten müssen in jedem Fall Teil des minimalen booleschen Ausdrucks sein
- Daher können nach der Übertragung in den booleschen Ausdruck diese Zeilen in der Primimplikantentafel gestrichen werden
- Ebenfalls können diejenigen Spalten gestrichen werden, die von den essentiellen Primimplikanten überdeckt werden
- Es entsteht eine "reduzierte Primimplikantentafel"

Phase 3: Lösen des Überdeckungsproblems

Die reduzierte Primimplikantentafel kann folgendermaßen vereinfacht werden:

Zeilenregel:

- Existiert für eine Zeile z_i eine andere Zeile z_j, die die gleichen und noch weitere Minterme überdeckt, so wird z_i gestrichen
- Überdecken zwei Zeilen die gleichen Minterme, so wird die Zeile des Primimplikants behalten, der weniger Literale enthält

Spaltenregel

- Wird ein Minterm immer überdeckt, wenn auch ein anderer überdeckt wird (alle Primterme behandeln diese beiden gleich), so kann eine der beiden zugehörigen Spalten gestrichen werden
- Die Spaltenregel kann die reduzierte Primimplikantentafel übersichtlicher machen, ist aber letztendlich nicht wichtig für das Ergebnis des Verfahrens

Phase 3: Lösen des Überdeckungsproblems

Wahrheitstafel:

	x_3	x_2	x_1	x_0	y
0:	0	0	0	0	1
1:	0	0	0	1	0
2:	0	0	1	0	1
3:	0	0	1	1	1
4:	0	1	0	0	0
5:	0	1	0	1	0
6:	0	1	1	0	0
7:	0	1	1	1	1
8:	1	0	0	0	1
9:	1	0	0	1	1
10:	1	0	1	0	0
11:	1	0	1	1	0
12:	1	1	0	0	1
13:	1	1	0	1	0
14:	1	1	1	0	1
15:	1	1	1	1	1

Implikanten (Ordnung 0):

	x_3	x_2	x_1	x_0	
0:	0	0	0	0	→
2:	0	0	1	0	→
3:	0	0	1	1	→
7:	0	1	1	1	→
8:	1	0	0	0	→
9:	1	0	0	1	→
12:	1	1	0	0	→
14:	1	1	1	0	→
15:	1	1	1	1	→

Implikanten (Ordnung 1):

	x_3	x_2	x_1	x_0	
0, 2:	0	0	-	0	✓
0, 8:	-	0	0	0	✓
2, 3:	0	0	1	-	✓
3, 7:	0	-	1	1	✓
7, 15:	-	1	1	1	✓
8, 9:	1	0	0	-	✓
8, 12:	1	-	0	0	✓
12, 14:	1	1	-	0	✓
14, 15:	1	1	1	-	✓

Phase 3: Lösen des Überdeckungsproblems

Primimplikantentafel:

	x_3	x_2	x_1	x_0	0	2	3	7	8	9	12	14	15	
0, 2:	0	0	-	0	○	○								$(\bar{x}_3\bar{x}_2\bar{x}_0)$
0, 8:	-	0	0	0	○				○					$(\bar{x}_2\bar{x}_1\bar{x}_0)$
2, 3:	0	0	1	-		○	○							$(\bar{x}_3\bar{x}_2x_1)$
3, 7:	0	-	1	1			○	○						$(\bar{x}_3x_1x_0)$
7, 15:	-	1	1	1				○					○	$(x_2x_1x_0)$
8, 9:	1	0	0	-					○	●				$(x_3\bar{x}_2\bar{x}_1)$
8, 12:	1	-	0	0					○		○			$(x_3\bar{x}_1\bar{x}_0)$
12, 14:	1	1	-	0							○	○		$(x_3x_2\bar{x}_0)$
14, 15:	1	1	1	-								○	○	$(x_3x_2x_1)$

Extrahierte essentielle Primimplikanten: $(x_3\bar{x}_2\bar{x}_1)$

Phase 3: Lösen des Überdeckungsproblems

Reduzierte Primimplikantentafel (Iteration 0):

	x_3	x_2	x_1	x_0	0	2	3	7	12	14	15	
0, 2:	0	0	-	0	●	○						$(\bar{x}_3\bar{x}_2\bar{x}_0)$
2, 3:	0	0	1	-		○	○					$(\bar{x}_3\bar{x}_2x_1)$
3, 7:	0	-	1	1			○	○				$(\bar{x}_3x_1x_0)$
7, 15:	-	1	1	1				○			○	$(x_2x_1x_0)$
12, 14:	1	1	-	0					●	○		$(x_3x_2\bar{x}_0)$
14, 15:	1	1	1	-						○	○	$(x_3x_2x_1)$

Extrahierte essentielle Primimplikanten: $(\bar{x}_3\bar{x}_2\bar{x}_0)$, $(x_3x_2\bar{x}_0)$

Reduzierte Primimplikantentafel (Iteration 1):

	x_3	x_2	x_1	x_0	3	7	15	
3, 7:	0	-	1	1	●	○		$(\bar{x}_3x_1x_0)$
7, 15:	-	1	1	1		○	●	$(x_2x_1x_0)$

Extrahierte essentielle Primimplikanten: $(\bar{x}_3x_1x_0)$, $(x_2x_1x_0)$

Phase 3: Lösen des Überdeckungsproblems

Nach rekursiver Anwendung des Streichens der Zeilen der essentiellen Primimplikanten und Vereinfachung der reduzierten Primimplikantentafel terminiert das Verfahren häufig

Der minimale boolesche Ausdruck kann dann durch die essentiellen Primimplikanten aus allen Rekursionsschritten gebildet werden

Der minimale boolesche Ausdruck für das gezeigte Beispiel lautet daher:
$y = (x_3\bar{x}_2\bar{x}_1) \lor (\bar{x}_3\bar{x}_2\bar{x}_0) \lor (x_3x_2\bar{x}_0) \lor (\bar{x}_3x_1x_0) \lor (x_2x_1x_0)$

Es gibt aber auch zyklische Überdeckungsprobleme, bei denen das Verfahren nicht terminiert

Zyklische Überdeckungsprobleme

Ob ein zyklisches Überdeckungsprobleme vorliegt, kann daran erkannt werden, dass in der Primimplikantentafel (oder der reduzierten Primimplikantentafel) keine essentiellen Primimpliktanten gefunden werden können

Damit ist es nicht eindeutig, welcher Primimplikant in die Lösung aufgenommen werden soll und welcher nicht

Eine einfache Lösung ist es, alle möglichen Kombinationen von Primimplikanten auszuprobieren, und die Kombination zu verwenden, die eine Überdeckung mit minimalen Kosten liefert

Die Anzahl der möglichen Kombinationen steigt jedoch schnell an. Es gibt 2^k Kombinationen, wenn k die Anzahl der beteiligten Primimplikanten ist

Also z.B. $2^{32} = 4294967296$ bei 32 Primimplikanten

Viele dieser Kombinationen liefern gar keine Überdeckung. Andere überdecken Minterme unnötigerweise mehrfach und erzeugen nicht minimale Lösungen.

Wie kann die Anzahl der betrachteten Kombinationen reduziert werden?

Verfahren von Petrick

Zunächst wird jeder Zeile k (bzw. dem korrespondierenden Primterm) eine boolesche Variable p_k zugeordnet

$(p_k = 1)$, wenn der Primterm in der Lösung verwendet wird, sonst $(p_k = 0)$

Mit den Variablen p_k wird ein boolescher Ausdruck aufgestellt, der nur wahr ist, wenn alle Spalten überdeckt werden

- Der boolesche Ausdruck besteht dabei aus so vielen UND-Verknüpfungen wie es Spalten gibt:
  (Spalte 1 überdeckt) UND (Spalte 2 überdeckt) UND ....
- Die Aufgabe, eine Spalte zu überdecken, kann von einem beliebigen Primterm übernommen werden. Daher wird eine ODER-Verknüpfung aus alle Primtermen gebildet, die die jeweilige Spalte überdecken, z.B.:
  (p_1 ODER p_3) UND (p_3 ODER p_5) UND ...

Anschließend werden die Klammern aufgelöst und der Ausdruck vereinfacht

Vereinfacht wird dabei mit den beiden Theoremen:
Idempotenz: $p_k \wedge p_k = p_k$ und Absorption: $p_k \vee (p_k \wedge p_j) = p_k$

Nach Auflösung besteht der Ausdruck aus ODER-Verknüpfungen von Monomen der Variablen p_k. Jedes Monom beschreibt eine mögliche Lösung

Verfahren von Petrick

Wahrheitstafel:

	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	1
4:	1	0	0	1
5:	1	0	1	1
6:	1	1	0	0
7:	1	1	1	1

Implikanten (Ordnung 0):

	x_2	x_1	x_0	
0:	0	0	0	→
2:	0	1	0	→
3:	0	1	1	→
4:	1	0	0	→
5:	1	0	1	→
7:	1	1	1	→

Implikanten (Ordnung 1):

	x_2	x_1	x_0	
0, 2:	0	-	0	✓
0, 4:	-	0	0	✓
2, 3:	0	1	-	✓
3, 7:	-	1	1	✓
4, 5:	1	0	-	✓
5, 7:	1	-	1	✓

Verfahren von Petrick

Primimplikantentafel:

	x_2	x_1	x_0	0	2	3	4	5	7	
0, 2:	0	-	0	○	○					$(\bar{x}_2\bar{x}_0) \equiv p_0$
0, 4:	-	0	0	○			○			$(\bar{x}_1\bar{x}_0) \equiv p_1$
2, 3:	0	1	-		○	○				$(\bar{x}_2 x_1) \equiv p_2$
3, 7:	-	1	1			○			○	$(x_1 x_0) \equiv p_3$
4, 5:	1	0	-				○	○		$(x_2\bar{x}_1) \equiv p_4$
5, 7:	1	-	1					○	○	$(x_2 x_0) \equiv p_5$

$(p_0 \vee p_1)(p_0 \vee p_2)(p_2 \vee p_3)(p_1 \vee p_4)(p_4 \vee p_5)(p_3 \vee p_5)$

$\Leftrightarrow (p_0 \vee p_0p_2 \vee p_0p_1 \vee p_1p_2)(p_1p_2 \vee p_2p_4 \vee p_1p_3 \vee p_3p_4)(p_3p_4 \vee p_4p_5 \vee p_3p_5 \vee p_5)$

$\Leftrightarrow (p_0 \vee p_1p_2)(p_1p_2 \vee p_2p_4 \vee p_1p_3 \vee p_3p_4)(p_3p_4 \vee p_5)$

$\Leftrightarrow (p_0p_1p_2 \vee p_0p_2p_4 \vee p_0p_1p_3 \vee p_0p_3p_4 \vee p_1p_2 \vee p_1p_2p_4 \vee p_1p_2p_3 \vee p_1p_2p_3p_4)(p_3p_4 \vee p_5)$

$\Leftrightarrow (p_0p_2p_4 \vee p_0p_1p_3 \vee p_0p_3p_4 \vee p_1p_2)(p_3p_4 \vee p_5)$

$\Leftrightarrow (p_0p_2p_3p_4 \vee p_0p_2p_4p_5 \vee p_0p_1p_3p_4 \vee p_0p_1p_3p_5 \vee p_0p_3p_4 \vee p_0p_3p_4p_5 \vee p_1p_2p_3p_4 \vee p_1p_2p_5)$

$\Leftrightarrow (p_0p_2p_4p_5 \vee p_0p_1p_3p_5 \vee p_0p_3p_4 \vee p_1p_2p_3p_4 \vee p_1p_2p_5)$

Verfahren von Petrick

Jede Lösung kann bezüglich ihrer Kosten bewertet werden

Die geringste Anzahl an Primtermen haben die beiden Lösungen $p_0 p_3 p_4$ und $p_1 p_2 p_5$

Jedes Literal p_k steht dabei für einen Primterm, der in die Lösung eingeht

Gibt es gleich viele Literale, wie im gezeigten Beispiel, entscheidet die Länge der einzelnen Primterme (im gezeigten Beispiel auch gleich)

Im gezeigten Beispiel sind daher folgende zwei Lösungen gleichwertig:

Lösung 1: $y = (\bar{x}_2 \bar{x}_0) \lor (x_1 x_0) \lor (x_2 \bar{x}_1)$

Lösung 2: $y = (\bar{x}_1 \bar{x}_0) \lor (\bar{x}_2 x_1) \lor (x_2 x_0)$

Interaktiver Quine-McCluskey Rechner

Durch Klicken auf die grauen Elemente kann die boolesche Funktion verändert werden

Anzahl der Variablen: 3 Don't-Cares erlauben: nein **Generiere zufälliges Beispiel**

Wahrheitstafel:

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

Minimaler boolescher Ausdruck:

$y = 0$

Gibt es Fragen?

Anregungen oder Verbesserungsvorschläge können auch gerne per E-mail an mich gesendet werden: [Kontakt]

[Weitere Vorlesungsfolien]

[Impressum, Datenschutz,]