Prof. Dr. Christoph Scholl
Tobias Seufert

Freiburg, 14. Juni 2023

Technische Informatik
Musterlösung zu Übungsblatt 8

Hinweis: Auf diesem Blatt befindet sich eine „Bonusaufgabe“. Diese Aufgabe zählt nicht in die Gesamtheit der Aufgaben, bei sinnvoller Bearbeitung wird sie jedoch zur Menge der sinnvoll bearbeiteten Aufgaben gerechnet.

Aufgabe 1 (3 Punkte)

Die Funktion $f : \mathbb{B}^4 \to \mathbb{B}$ sei durch ihre *ON*-Menge gegeben:

$$ON(f) = \{0000,\ 0001,\ 0100,\ 0101,\ 0110,\ 0111,\ 1000,\ 1010,\ 1100,\ 1110,\ 1111\}$$

Sie haben auch bereits durch Quine-McCluskey die Menge der Primimplikanten bestimmt:

$$Prim(f) = \{--00,\ -1-0,\ -11-,\ 1--0,\ 0-0-,\ 01--\}$$

Erstellen Sie eine Primimplikantentafel zu f und reduzieren Sie die Tafel. Geben Sie in jedem Schritt an welche Spalten/Zeilen Sie für die Reduktion betrachten und welche Reduktionsregel Sie anwenden um Spalten/Zeilen zu löschen. Es ist nicht erforderlich nach jedem Reduktionsschritt die reduzierte PIT anzugeben. Geben Sie nur die reduzierte PIT an, wenn es notwendig oder hilfreich ist um Ihre Vorgehensweise nachzuvollziehen. Wie lautet das berechnete Minimalpolynom?

Hinweis: Das Resultat ist eine leere Tabelle. Die Methode von Petrick muss nicht angewendet werden.

Lösung:

Hinweise:
Für falsche Einträge -0.25 P, aber max. -1.5 Punkte für falsche Einträge. Folgefehler sind ok.
Einmalig -1 P wenn Regeln allgemein falsch angewendet werden.
-0.25 P wenn zum Schluss nicht ersichtlich wird was denn das Minimalpolynom ist.

PIT:

	0000	0001	0100	0101	0110	0111	1000	1010	1100	1110	1111
--00	1		1				1		1		
-1-0			1		1				1	1	
-11-					1	1				1	1
1--0							1	1	1	1	
0-0-	1	1	1	1							
01--			1	1	1	1					

- 2. Regel: 0000, 0100, 0101 dominieren 0001 ⇒ Lösche 0000, 0100 und 0101

	0001	0110	0111	1000	1010	1100	1110	1111
`--00`				1		1		
`-1-0`		1				1	1	
`-11-`		1	1				1	1
`1--0`				1	1	1	1	
`0-0-`	1							
`01--`		1	1					

⇒ diese Tabelle ist nicht zwingend erforderlich.

- 2. Regel: 0110, 0111 und 1110 dominieren 1111 ⇒ Lösche 0110, 0111 und 1110

	0001	1000	1010	1100	1111
`--00`		1		1	
`-1-0`				1	
`-11-`					1
`1--0`		1	1	1	
`0-0-`	1				
`01--`					

⇒ diese Tabelle ist nicht zwingend erforderlich.

- 2. Regel: 1000 und 1100 dominieren 1010 ⇒ Lösche 1000 und 1100

	0001	1010	1111
`--00`			
`-1-0`			
`-11-`			1
`1--0`		1	
`0-0-`	1		
`01--`			

- 1. Regel: `0-0-`, `1--0` und `-11-` sind offensichtlich wesentlich.

Alternativ: Nur Anwendung von Regel 1.

Leere Tabelle, kein Petrick notwendig.

Minimalpolynom: $x1'x3' + x1x4' + x2x3$ bzw. 0-0- + 1--0 + -11-.

Aufgabe 2 (4 Punkte)

Sei für eine boolesche Funktion die folgende Primimplikantentafel für die Minterme $\{a, b, c, d, e, f\}$ und die Primimplikanten $\{A, B, C, D, E, F\}$ bereits ermittelt worden:

	a	b	c	d	e	f
A		1		1		
B	1		1	1		
C			1	1		1
D		1			1	
E			1	1	1	
F	1	1				1

Reduzieren Sie die Primimplikantentafel so weit wie möglich, geben Sie dabei die einzelnen verwendeten Reduktionsregeln an. Wenden Sie dann die Methode von Petrick an, um das aus der gegebenen Primimplikantentafel resultierende Minimalpolynom zu bestimmen. Gehen Sie dabei davon aus, dass alle Primimplikanten die gleichen Kosten haben.

Lösung:

Regel 2: d dominiert $c \Rightarrow$ Lösche d.

	a	b	c	e	f
A		1			
B	1		1		
C			1		1
D		1		1	
E			1	1	
F	1	1			1

Regel 3: D (oder F) dominiert $A \Rightarrow$ Lösche A.

	a	b	c	e	f
B	1		1		
C			1		1
D		1		1	
E			1	1	
F	1	1			1

Keine weitere Reduktionsregel mehr anwendbar $\Rightarrow$ Petrick:

$$(B+F)(D+F)(B+C+E)(D+E)(C+F) =$$
$$\underline{EF} + DEF + CEF + BDF + BCD + CDF + BEF + CDEF + BDEF + BCEF + BCDF + BCDE + BCDEF$$

Minimalpolynom: $E + F$.

Punkte: [1.5] pro Reduktionsregel; [1] für Petrick; einmalig [-0.5] wenn Reduktionsregel korrekt angewandt aber nicht angegeben. [-0.5] wenn Petrick korrekt aber Minimalpolynom nicht angegeben.

Aufgabe 3 (*Bonusaufgabe:* 2 + 1 + 2 + 2 + 1 Punkte)

Gegeben sei die Boolesche Funktion $xor_n : \mathbb{B}^n \to \mathbb{B}$ mit

$$xor_n(x_1, \ldots, x_n) = \begin{cases} 1 & \text{wenn } \sum_{i=1}^{n} x_i \text{ ungerade} \\ 0 & \text{wenn } \sum_{i=1}^{n} x_i \text{ gerade} \end{cases}$$

a) Geben Sie die ON-Menge zu $xor_3(x_1, x_2, x_3)$ an. Verwenden Sie anschließend das Verfahren von Quine-McCluskey um alle Primimplikanten zu berechnen. Bestimmen Sie zum Schluss mithilfe der Primimplikantentafel das Minimalpolynom.

b) Leiten Sie nun das Minimalpolynon zu xor_n für $n \in \mathbb{N} \setminus \{0\}$ her. Erläutern Sie Ihre Vorgehensweise und geben Sie die Anzahl der Primimplikanten im Minimalpolynom an.

c) Sie wollen das Minimalpolynom realisieren und erkennen, dass sowohl die Kosten für eine PLA als auch die Kosten für eine direkte Schaltkreisübersetzung exponentiell mit n anwächst. Sie wollen jedoch eine kostengünstigere Lösung. Sei $g(xor_i(x_1, \ldots, x_i), xor_{n-i}(x_{i+1}, \ldots, x_n))$ für

$1 \leq i \leq n-1$ eine alternative Definition für xor_n. Leiten Sie die Funktion für g mithilfe einer Fallunterscheidung her.

d) Skizzieren Sie auf Basis von Aufgabenteil c) einen Schaltkreis für $xor_8(x_1, ..., x_8) = g(xor_i(x_1, ..., x_i), xor_{8-i}(x_{i+1}, ..., x_8))$ für $1 \leq i \leq 7$ mit möglichst geringer Tiefe.

Hinweis: Überlegen Sie sich, wie Sie i wählen müssen um eine möglichst geringe Tiefe zu erreichen.

e) Benutzen Sie Ihren Schaltkreis aus Aufgabenteil d) um sich einen Schaltkreis für xor_n zu überlegen, der ebenfalls eine möglichst geringe Tiefe hat. Dabei sei $n = 2^k$, mit $k \in \mathbb{N}$. Geben Sie in Abhängigkeit von n die Kosten und Tiefe ihres Schaltkreises an. In diesem Aufgabenteil ist *keine* Skizzierung des Schaltkreises erforderlich.

Lösung:

a) 0.5P für Minterme
0.5P Quine-McCluskey
0.5P Primimplikantentafel
0.5P Minimalpolynom

Minterme: $x_1'x_2'x_3,\ x_1'x_2x_3',\ x_1x_2'x_3',\ x_1x_2x_3$

Quine-McCluskey:

$L_0^{\{x_1,x_2,x_3\}}$
001
010
100
111

$\Rightarrow Prim(xor_3) = \emptyset$

Es gibt keine Kombinationsmöglichkeiten der Minterme aus L_0. Daher sind alle L_1-Mengen leer und die Primmenge im nächsten Schritt enthält alle vier Minterme. Der Grund ist, dass sich alle Minterme in (mindestens) zwei Literalen unterscheiden, d.h. alle Minterme sind Primimplikanten.

PIT(xor_3):

	1	2	4	7
$x_1'x_2'x_3$	1			
$x_1'x_2x_3'$		1		
$x_1x_2'x_3'$			1	
$x_1x_2x_3$				1

Es ist direkt ersichtlich, dass alle Primimplikanten wesentlich sind (Regel 1). Die Disjunktion aller Minterme ist also das einzige Minimalpolynom für xor_3.

b) 0.25P für Ausdruck
0.75P für Erklärung

$$\bigvee_{\substack{(b_1,...,b_n) \in \mathbb{B}^n \text{ mit} \\ \sum_{i=1}^n b_i \text{ ungerade}}} \left(x_1^{b_1} \wedge \cdots \wedge x_n^{b_n}\right), \qquad \text{wobei } x_i^0 := x_i' \text{ und } x_i^1 := x_i$$

Der angegebene Ausdruck ist die Disjunktion aller Minterme für xor_n. In Aufgabenteil a) haben wir schon gesehen, dass alle Minterme wesentlich sind um xor_n darzustellen, da weder durch Quine-McCluskey noch

Primimplikantentafel Reduktionen erreicht werden können. Deswegen beschreibt auch hier die Disjunktion der Minterme das Minimalpolynom. Es sind 2^{n-1} Primimplikanten, je mit Länge n.

c) 0.5P für $g = xor_2$
1.5P für Herleitung

Aus der Aufgabenstellung ist gegeben: $xor_n(x_1, ..., x_n) = g(xor_i(x_1, ..., x_i), xor_{n-i}(x_{i+1}, ..., x_n))$ für $1 \leq i \leq n-1$.
Sei $b_1 := xor_i(x_1, ..., x_i),\ b_2 := xor_{n-i}(x_{i+1}, ..., x_n)$.

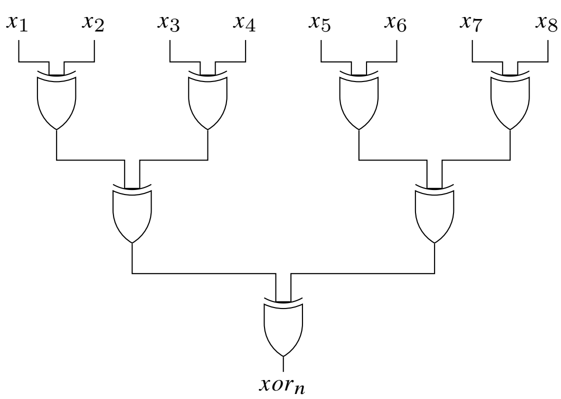
1. Fall $b_1 = 0,\ b_2 = 0$
   Dann ist $\sum_{j=1}^{i} x_j$ gerade, $\sum_{j=i+1}^{n} x_j$ gerade
   $\Rightarrow \sum_{j=1}^{n} x_j$ gerade $\Rightarrow g(0,0) = 0 = xor_2(0,0)$
2. Fall $b_1 = 1,\ b_2 = 0$
   Dann ist $\sum_{j=1}^{i} x_j$ ungerade, $\sum_{j=i+1}^{n} x_j$ gerade
   $\Rightarrow \sum_{j=1}^{n} x_j$ ungerade $\Rightarrow g(1,0) = 1 = xor_2(1,0)$
3. Fall $b_1 = 0,\ b_2 = 1$
   Dann ist $\sum_{j=1}^{i} x_j$ gerade, $\sum_{j=i+1}^{n} x_j$ ungerade
   $\Rightarrow \sum_{j=1}^{n} x_j$ ungerade $\Rightarrow g(0,1) = 1 = xor_2(0,1)$
4. Fall $b_1 = 1,\ b_2 = 1$
   Dann ist $\sum_{j=1}^{i} x_j$ ungerade, $\sum_{j=i+1}^{n} x_j$ ungerade
   $\Rightarrow \sum_{j=1}^{n} x_j$ gerade $\Rightarrow g(1,1) = 0 = xor_2(1,1)$

Zusammengefasst:

b_1	b_2	g
0	0	0
0	1	1
1	0	1
1	1	0

$\Rightarrow\ g = xor_2$
$xor_n(x_1, ..., x_n) = xor_2(xor_i(x_1, ..., x_i), xor_{n-i}(x_{i+1}, ..., x_n))$ für $1 \leq i \leq n-1$.

d) -0.25P wenn es ein Baum ist der zwar nicht geringste Tief hat, aber auch nicht gerade den längsten möglichen Pfad (depth $= n-1$) hat.
-0.5P wenn es gerade der Schaltkreis mit dem längsten Pfad ist.
Ein balancierter Baum hat die geringste Tiefe. D.h. wähle $i = n/2$.



e) Je 0.5P für Kosten und Tiefe.

Für einen balancierten Baum ist Kosten = #XOR-Gatter = $n-1$ und längster Pfad = $\lceil log(n) \rceil$. Bzw. da vorgegeben, dass $n = 2^k$ gilt, ist auch $log(n)$ korrekt.

Abgabe: 21. Juni 2023, $13^{\underline{00}}$ über das Übungsportal