Prof. Dr. Rolf Drechsler, drechsler@informatik.uni-bremen.de, MZH 3510
Dipl.-Inform. Daniel Große, grosse@informatik.uni-bremen.de, MZH 3490

9. Übungsblatt zur Vorlesung

Technische Informatik 1

Aufgabe 1 (5 Punkte)

Die Boolesche Funktion $f \in B_{4,1}$ sei gegeben durch die Menge ihrer Primimplikanten $Prim(f)$ mit:

$$\text{Prim}(f) = \{\overline{x}_1 x_2 x_4, x_2 \overline{x}_3 x_4, x_1 \overline{x}_3 x_4, x_1 \overline{x}_2 x_4, \overline{x}_2 x_3 x_4, \overline{x}_1 x_3 x_4\}.$$

a) Stelle die Primimplikantentafel für f auf.
b) Bestimme *alle* Minimalpolynome von f mit Hilfe der Methode von Petrick.

Aufgabe 2 (3 Punkte)

Konstruiere zu der Funktion $f \in B_{5,1}$ mit

$$f(x_1, x_2, x_3, x_4, x_5) = x_1 x_2 + x_3(x_4 + x_5)$$

das reduzierte OBDD mit Variablenordnung $x_1 < x_2 < x_3 < x_4 < x_5$.
Erstelle dazu zuerst den vollständigen Entscheidungsbaum durch iterierte Anwendung der Shannon-Entwicklung und führe dann alle Reduktionsschritte elementar durch.

Aufgabe 3 (4 Punkte)

Berechne für die Funktion $f \in B_{4,1}$ mit

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

ein reduziertes KFDD mit folgender Variablenordnung π und Zerlegunstyp-Liste d:

a) $\pi : x_1 < x_2 < x_3 < x_4$ und $d = (S_{@x_1}, pD_{@x_2}, S_{@x_3}, pD_{@x_4})$.
b) $\pi : x_2 < x_1 < x_4 < x_3$ und $d = (pD_{@x_1}, nD_{@x_2}, pD_{@x_3}, nD_{@x_4})$.

Dabei bedeutet $D_{@y}$, dass für die Variable y die Zerlegung D benutzt werden soll.

Aufgabe 4 (4 Punkte)

In Abbildung 1 ist ein KFDD *vor* und eines *nach* diversen Reduktionsschritten dargestellt.

a) Gib alle durchgeführten Reduktionsschritte mit Knoten und Typ an. Die Knoten des linken KFDDs wurden zu diesem Zweck mit den Bezeichnungen a–f versehen.
b) Wieviele Möglichkeiten für Dekompositionstypenlisten gibt es theoretisch bei KFDDs mit 3 Variablen? Welche kommen für das KFDD in der Abbildung in Frage?

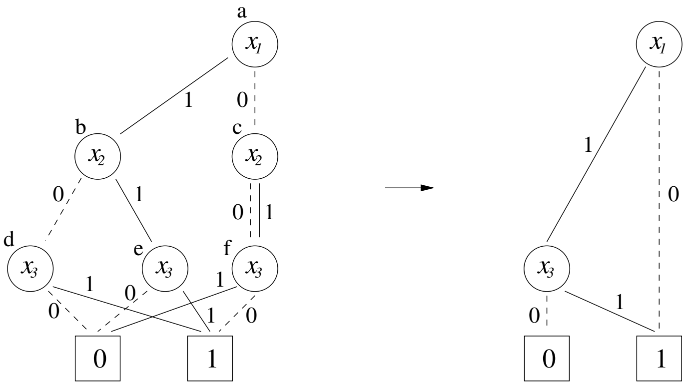

Abbildung 1: Nicht-reduziertes und reduziertes KFDD.

Aufgabe 5 (4 Punkte)

Gegeben sei die Funktion $f \in B_{n,1}$ als vollständiger OBDD G (G ist also nicht reduziert). Im Folgenden wird ein Algorithmus beschrieben, der die Kardinalität der ON-Menge von f ($\#ON(f)$) auf G bestimmt.

Jedem Knoten v des Graphen sei eine Zahl $a_v \in \mathbf{Z}$ zugeordnet, die mit -1 initialisiert ist. Der initiale Aufruf des Algorithmus' ist $\#ON(f) := on\_set(Wurzel(G))$.

```
int on_set(Knoten v)
{
        Falls a_v ≠ −1
                return a_v
        Falls v ein Terminal 1 ist
                return 1
        Falls v ein Terminal 0 ist
                return 0
        Ansonsten
        a_v = on_set(low(v)) + on_set(high(v))
        return a_v
}
```

a) Verändere den Algorithmus so, dass er auch auf einem reduzierten OBDD das richtige Ergebnis liefert. Welche Komplexität hat dein Algorithmus?
b) Welche Komplexität hat der Algorithmus, wenn Du die Zahlen a_v nicht verwenden kannst?
c) Welche Schwierigkeiten treten auf, wenn Du das Verfahren auf reduzierte OKFDDs mit einer DTL $d = (pD, \ldots, pD)$ verallgemeinern willst?

Abgabetermin: zu Beginn der Vorlesung am 27.06.2005