Splošno razčlenitveno pravilo za OBDD, OFDD in 0-sup-BDD

Robert Meolic, Zmago Brezočnik
Fakulteta za elektrotehniko, računalništvo in informatiko
Univerza v Mariboru
Smetanova 17, 2000 Maribor, Slovenija
{*meolic,brezocnik*}*@uni-mb.si*

Abstract

Binary decision diagrams (BDD) are very successful data structure for representation and manipulation of Boolean functions. Various types of BDDs have been proposed. In this paper relations between OBDD, OFDD, and 0-sup-BDD are shown in a new way. A generic decomposition rule for them is introduced. Using this rule a set of 288 types of BDDs is defined together with their reduction rules.

1 Uvod

Binarni odločitveni graf (BDD) je učinkovita podatkovna struktura za predstavitev logičnih funkcij. BDD je usmerjeni, binarni, aciklični graf $G = (V, P)$, ki ima natanko en koren in je podan z množico vozlišč V ter množico povezav P. Koren grafa imenujemo *vrhnje vozlišče*, povezavi do naslednikov pa imenujemo *'else'* in *'then' povezava*. Podgrafa imenujemo *'else'* in *'then' naslednik* vozlišča. *Razčlenitveno pravilo* pove, kako iz danega odločitvenega grafa dobimo logično funkcijo, ki jo ta graf predstavlja. Odločitveni graf lahko preoblikujemo (dodamo ali odvzamemo vozlišča) tako, da še vedno predstavlja isto funkcijo. Pri tem uporabljamo *pravilo minimizacije*. Odločitveni graf je minimalen glede na dano pravilo minimizacije, če ga po tem pravilu ne moremo preoblikovati.

Obstaja mnogo različnih vrst BDD. Največ se uporabljajo OBDD [3], OFDD [4] in 0-sup-BDD [6]. V 2. razdelku podamo splošno razčlenitveno pravilo za te tri vrste BDD. Naš pristop je bolj splošen kot v [1], ker upošteva tudi 0-sup-BDD. Prikazano razčlenitveno pravilo definira 288 različnih vrst BDD, med njimi tudi OBDD, OFDD in 0-sup-BDD. V 3. razdelku opišemo pripadajoča pravila minimizacije za vseh 288 vrst BDD. Uporabo splošnega razčlenitvenega pravila nazorno razložimo na treh primerih v 4. razdelku. V zaključku ovrednotimo svoje delo.

2 Splošno razčlenitveno pravilo

Imejmo logično funkcijo $f = f(x_1, x_2, ..., x_n)$ in urejeni binarni odločitveni graf G ($i < j \Rightarrow x_i < x_j$), ki predstavlja funkcijo f. Naj bo x_i ($G.top$) spremenljivka v vrhnjem vozlišču grafa, g in h pa funkciji, ki ju predstavljata 'else' ($G.f$) in 'then' ($G.t$) naslednik vrhnjega vozlišča. Če je graf G OBDD, OFDD ali 0-sup-BDD, veljajo naslednja razčlenitvena pravila:

$$\text{za OBDD:} \qquad f = \overline{x_i} \cdot g + x_i \cdot h \tag{1}$$

$$\text{za OFDD:} \qquad f = g \oplus x_i \cdot h \tag{2}$$

$$\begin{aligned} \text{za 0-sup-BDD:} \quad & i) & g &= \overline{x_i} \cdot f|_{x_i=0} \\ & ii) & h &= \overline{x_i} \cdot f|_{x_i=1} \\ & iii) & f &= g + x_i \cdot h|_{x_i=0} \end{aligned} \tag{3}$$

Definirajmo *razčlenitveno funkcijo* $R(x_i, g, h)$, za katero velja [1] [1]:

$$f = R(x_i, g, h) \ \wedge \ x_i \notin sup(g) \cup sup(h) \tag{4}$$

ter *topološko funkcijo* $T(x_i, k)$, ki naj bo odvisna od k ($k \in sup(T)$). Splošno razčlenitveno pravilo lahko zapišemo v obliki rekurzivnega algoritma 1. Funkcijo f dobimo iz grafa s klicem $val(G, n, 1)$.

```
val(G,n,j) {
  if (j > n) return G; /* G is 0 or 1 */
  if (G.top == x_j)
    return R(x_j, val(G.f,n,j+1), val(G.t,n,j+1));
  else return T(x_j, val(G,n,j+1));
}
```

Algoritem 1: Splošno razčlenitveno pravilo

Obstaja 24 različnih razčlenitvenih funkcij, ki zadostijo pogoju (4) [1]. Dve po dve izmed njih sta simetrični (dobljena grafa se razlikujeta le v tem, da sta v vsakem vozlišču 'else' in 'then' naslednik medsebojno zamenjana). V tabeli 1 je prikazanih 12 razčlenitvenih funkcij, ostalih 12 dobimo tako, da v formulah zamenjamo mesti g in h. [2]

[1] $x_i \in sup(f)$ če in samo če $f(x_1, ...x_{i-1}, 0, x_{i+1}, ...x_n) \neq f(x_1, ...x_{i-1}, 1, x_{i+1}, ...x_n)$.

[2] Shannonova razčlenitev je zapisana kot $\overline{x_i} \cdot g \oplus x_i \cdot h$ [5].

Tabela 1: Razčlenitvene funkcije

1.	$\overline{x_i} \cdot g \oplus x_i \cdot h$	2.	$\overline{x_i} \cdot \overline{g} \oplus x_i \cdot h$
3.	$\overline{x_i} \cdot g \oplus x_i \cdot \overline{h}$	4.	$\overline{x_i} \cdot \overline{g} \oplus x_i \cdot \overline{h}$
5.	$g \oplus \overline{x_i} \cdot h$	6.	$g \oplus \overline{x_i} \cdot \overline{h}$
7.	$\overline{g} \oplus \overline{x_i} \cdot h$	8.	$\overline{g} \oplus \overline{x_i} \cdot \overline{h}$
9.	$x_i \cdot g \oplus h$	10.	$x_i \cdot \overline{g} \oplus h$
11.	$x_i \cdot g \oplus \overline{h}$	12.	$x_i \cdot \overline{g} \oplus \overline{h}$

Vseh topoloških funkcij je 12, zapišemo jih lahko na naslednji način: k, $\overline{k}$, $x_i + k$, $\overline{x_i} + k$, $x_i \cdot k$, $\overline{x_i} \cdot k$, $x_i \oplus k$, $\overline{x_i} \oplus k$, $x_i \mid k$, $\overline{x_i} \mid k$, $x_i \downarrow k$ in $\overline{x_i} \downarrow k$.

Splošno razčlenitveno pravilo (algoritem 1) definira $24 \cdot 12 = 288$ različnih vrst urejenih binarnih odločitvenih grafov, med njimi tudi OBDD, OFDD in 0-sup-BDD. Vrsto BDD označimo z indeksoma $R(x_i, g, h)$ in $T(x_i, k)$, ki povesta uporabljeno razčlenitveno in topološko funkcijo. Tako lahko zapišemo: OBDD = BDD $_{\overline{x_i} \cdot g \oplus x_i \cdot h\ ,\ k}$, OFDD = BDD $_{g \oplus x_i \cdot h\ ,\ k}$ in 0-sup-BDD = BDD $_{\overline{x_i} \cdot g \oplus x_i \cdot h\ ,\ \overline{x_i} \cdot k}$.

3 Minimizacija

V nadaljevanju imejmo graf $G = (x_i, g, h)$, kjer je x_i spremenljivka v vrhnjem vozlišču, g in h pa sta funkciji, ki ju predstavljata 'else' in 'then' naslednik. Vsaki vrsti BDD pripada eno od naslednjih 12 pravil minimizacije:

1.	$(x_i, k, k) = k$	2.	$(x_i, \overline{k}, \overline{k}) = k$
3.	$(x_i, \overline{k}, k) = k$	4.	$(x_i, k, \overline{k}) = k$
5.	$(x_i, k, 0) = k$	6.	$(x_i, \overline{k}, 0) = k$
7.	$(x_i, k, 1) = k$	8.	$(x_i, \overline{k}, 1) = k$
9.	$(x_i, 0, k) = k$	10.	$(x_i, 0, \overline{k}) = k$
11.	$(x_i, 1, k) = k$	12.	$(x_i, 1, \overline{k}) = k$

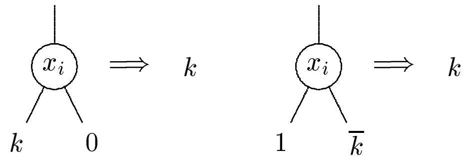
Izrek: Naj graf G, ki je vrste $BDD_{R(x_i,g,h),T(x_i,k)}$, predstavlja logično funkcijo $f = f(x_1, x_2, ..., x_n)$. Pripadajoče pravilo minimizacije naj bo eno izmed zgornjih 12, označimo ga z $(x_i, g, h) = k$. Potem velja:

$$R(x_i, g, h) = T(x_i, k) \qquad (5)$$

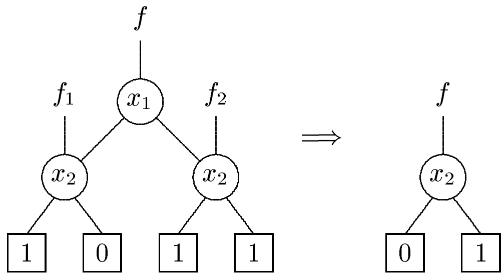
Dokaz: Po algoritmu 1 dobimo $f = val(G, n, i) = R(x_i, g, h)$. Po uporabi pravila minimizacije nad vrhnjim vozliščem dobimo graf G_{min}, v katerem ni vozlišča x_i. Po algoritmu 1 dobimo zanj $f = val(G_{min}, n, i) = T(x_i, k)$. Ker G in G_{min} predstavljata isto funkcijo, formula 5 velja. ■

S pomočjo formule 5 dobimo tabelo 2. V njej so zbrana pravila minimizacije za vseh 288 različnih vrst BDD. Pravilo minimizacije $(x_i, g, h) = k$ je v tabeli zapisano kot par g, h.

Na sliki 1 sta prikazani pravilo minimizacije št. 5 (pripada OFDD in 0-sup-BDD) in pravilo št. 12. Slednje pripada BDD$_{g \oplus \overline{x_i} \cdot h\ ,\ x_i + k}$. Na sliki 2 je primer minimizacije za to vrsto BDD. Funkcija f, ki jo graf predstavlja, je $x_1 + \overline{x_2}$, 'else' in 'then' naslednik vozlišča x_1 pa sta funkciji $f_1 = 1$ in $f_2 = x_2$.



Slika 1: Dve od 12 pravil minimizacije (št. 5 in 12)

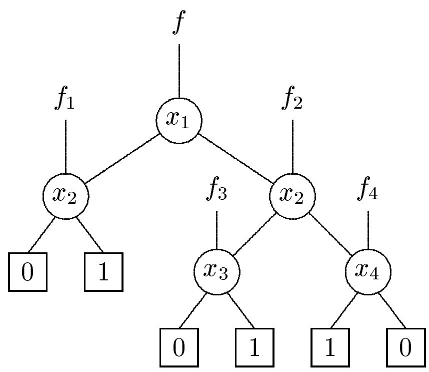


Slika 2: Minimizacija BDD$_{g \oplus \overline{x_i} \cdot h\ ,\ x_i + k}$

4 Primeri

4.1 OBDD

Imejmo OBDD s slike 3, ki predstavlja logično funkcijo $f : \mathbb{B}^4 \to \mathbb{B}$, kjer so spremenljivke označene z x_1, x_2, x_3 in x_4.



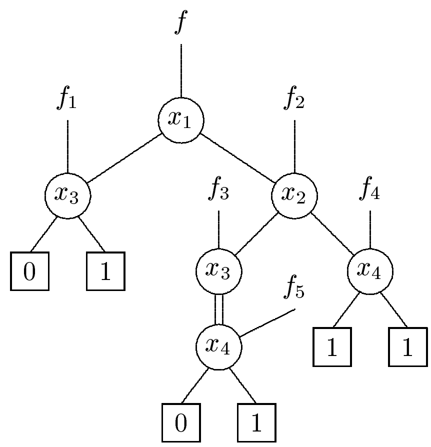
Slika 3: Primer OBDD

Tabela 2: Pravila minimizacije za različne vrste BDD-jev

$R \quad \backslash \quad T$	k	$\overline{k}$	$x_i + k$	$\overline{x_i} + k$	$x_i \cdot k$	$\overline{x_i} \cdot k$	$x_i \oplus k$	$\overline{x_i} \oplus k$	$x_i \mid k$	$\overline{x_i} \mid k$	$x_i \downarrow k$	$\overline{x_i} \downarrow k$
$\overline{x_i} \cdot g \oplus x_i \cdot h$	k, k	$\overline{k}, \overline{k}$	$k, 1$	$1, k$	$0, k$	$k, 0$	$k, \overline{k}$	$\overline{k}, k$	$1, \overline{k}$	$\overline{k}, 1$	$\overline{k}, 0$	$0, \overline{k}$
$\overline{x_i} \cdot \overline{g} \oplus x_i \cdot h$	$\overline{k}, k$	$k, \overline{k}$	$\overline{k}, 1$	$0, k$	$1, k$	$\overline{k}, 0$	$\overline{k}, \overline{k}$	k, k	$0, \overline{k}$	$k, 1$	$k, 0$	$1, \overline{k}$
$\overline{x_i} \cdot g \oplus x_i \cdot \overline{h}$	$k, \overline{k}$	$\overline{k}, k$	$k, 0$	$1, \overline{k}$	$0, \overline{k}$	$k, 1$	k, k	$\overline{k}, \overline{k}$	$1, k$	$\overline{k}, 0$	$\overline{k}, 1$	$0, k$
$\overline{x_i} \cdot \overline{g} \oplus x_i \cdot \overline{h}$	$\overline{k}, \overline{k}$	k, k	$\overline{k}, 0$	$0, \overline{k}$	$1, \overline{k}$	$\overline{k}, 1$	$\overline{k}, k$	$k, \overline{k}$	$0, k$	$k, 0$	$k, 1$	$1, k$
$g \oplus \overline{x_i} \cdot h$	$k, 0$	$\overline{k}, 0$	$1, \overline{k}$	$k, \overline{k}$	k, k	$0, k$	$\overline{k}, 1$	$k, 1$	$\overline{k}, k$	$1, k$	$0, \overline{k}$	$\overline{k}, \overline{k}$
$g \oplus \overline{x_i} \cdot \overline{h}$	$k, 1$	$\overline{k}, 1$	$1, k$	k, k	$k, \overline{k}$	$0, \overline{k}$	$\overline{k}, 0$	$k, 0$	$\overline{k}, \overline{k}$	$1, \overline{k}$	$0, k$	$\overline{k}, k$
$\overline{g} \oplus \overline{x_i} \cdot h$	$0, k$	$0, \overline{k}$	$\overline{k}, k$	$\overline{k}, 1$	$k, 0$	k, k	$1, k$	$1, \overline{k}$	$k, 1$	$k, \overline{k}$	$\overline{k}, \overline{k}$	$\overline{k}, 0$
$\overline{g} \oplus \overline{x_i} \cdot \overline{h}$	$1, k$	$1, \overline{k}$	k, k	$k, 1$	$\overline{k}, 0$	$\overline{k}, k$	$0, k$	$0, \overline{k}$	$\overline{k}, 1$	$\overline{k}, \overline{k}$	$k, \overline{k}$	$k, 0$
$x_i \cdot g \oplus h$	$\overline{k}, 0$	$k, 0$	$0, \overline{k}$	$\overline{k}, \overline{k}$	$\overline{k}, k$	$1, k$	$k, 1$	$\overline{k}, 1$	k, k	$0, k$	$1, \overline{k}$	$k, \overline{k}$
$x_i \cdot \overline{g} \oplus h$	$\overline{k}, 1$	$k, 1$	$0, k$	$\overline{k}, k$	$\overline{k}, \overline{k}$	$1, \overline{k}$	$k, 0$	$\overline{k}, 0$	$k, \overline{k}$	$0, \overline{k}$	$1, k$	k, k
$x_i \cdot g \oplus \overline{h}$	$0, \overline{k}$	$0, k$	$\overline{k}, \overline{k}$	$\overline{k}, 0$	$k, 1$	$k, \overline{k}$	$1, \overline{k}$	$1, k$	$k, 0$	k, k	$\overline{k}, k$	$\overline{k}, 1$
$x_i \cdot \overline{g} \oplus \overline{h}$	$1, \overline{k}$	$1, k$	$k, \overline{k}$	$k, 0$	$\overline{k}, 1$	$\overline{k}, \overline{k}$	$0, \overline{k}$	$0, k$	$\overline{k}, 0$	$\overline{k}, k$	k, k	$k, 1$

Za OBDD velja $R(x_i, g, h) = \overline{x_i} \cdot g \oplus x_i \cdot h$ in $T(x_i, k) = k$. Po razčlenitvenem pravilu dobimo:

$$
\begin{aligned}
&f = val(f,4,1) = R(x_1, val(f_1,4,2), val(f_2,4,2)) = \\
&= R(x_1, R(x_2, val(0,4,3), val(1,4,3)), \\
&\quad R(x_2, val(f_3,4,3), val(f_4,4,3))) = \\
&= R(x_1, R(x_2, T(x_3, val(0,4,4)), T(x_3, val(1,4,4))), \\
&\quad R(x_2, R(x_3, val(0,4,4), val(1,4,4)), \\
&\quad T(x_3, val(f_4,4,4)))) = \\
&= R(x_1, R(x_2, T(x_3, T(x_4, val(0,4,5))), T(x_3, \\
&\quad T(x_4, val(1,4,5)))), R(x_2, R(x_3, T(x_4, \\
&\quad val(0,4,5)), T(x_4, val(1,4,5))), T(x_3, \\
&\quad R(x_4, val(1,4,5), val(0,4,5))))) = \\
&= R(x_1, R(x_2, T(x_3, 0), T(x_3, 1)), \\
&\quad R(x_2, R(x_3, 0, 1), T(x_3, R(x_4, 1, 0)))) = \\
&= R(x_1, R(x_2, 0, 1), R(x_2, x_3, T(x_3, \overline{x_4}))) = \\
&= R(x_1, x_2, R(x_2, x_3, \overline{x_4})) = \\
&= R(x_1, x_2, \overline{x_2} \cdot x_3 \oplus x_2 \cdot \overline{x_4}) = \\
&= \overline{x_1} \cdot x_2 \oplus x_1 \cdot (\overline{x_2} \cdot x_3 \oplus x_2 \cdot \overline{x_4}) = \\
&= \overline{x_1} \cdot x_2 \oplus x_1 \cdot \overline{x_2} \cdot x_3 \oplus x_1 \cdot x_2 \cdot \overline{x_4}
\end{aligned}
$$



Slika 4: Primer 0-sup-BDD

4.2 0-sup-BDD

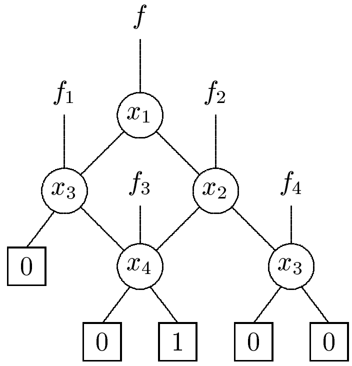
Imejmo 0-sup-BDD s slike 4, ki predstavlja logično funkcijo $f : \mathbb{B}^4 \rightarrow \mathbb{B}$, kjer so spremenljivke označene z x_1, x_2, x_3 in x_4.

Za 0-sup-BDD velja $R(x_i, g, h) = \overline{x_i} \cdot g \oplus x_i \cdot h$ in $T(x_i, k) = \overline{x_i} \cdot k$. Po algoritmu 1 dobimo:

$$
\begin{aligned}
&f = val(f,4,1) = R(x_1, val(f_1,4,2), val(f_2,4,2)) = \\
&= R(x_1, T(x_2, val(f_1,4,3)), R(x_2, val(f_3,4,3), \\
&\quad val(f_4,4,3))) = \\
&= R(x_1, T(x_2, R(x_3, val(0,4,4), val(1,4,4))), \\
&\quad R(x_2, R(x_3, val(f_5,4,4), val(f_5,4,4)), \\
&\quad T(x_3, val(f_4,4,4)))) = \\
&= R(x_1, T(x_2, R(x_3, T(x_4, val(0,4,5)), T(x_4, \\
&\quad val(1,4,5)))), R(x_2, R(x_3, R(x_4, val(0,4,5), \\
&\quad val(1,4,5)), R(x_4, val(0,4,5), val(1,4,5))), \\
&\quad T(x_3, R(x_4, val(1,4,5), val(1,4,5))))) = \\
&= R(x_1, T(x_2, R(x_3, T(x_4, 0), T(x_4, 1))), \\
&\quad R(x_2, R(x_3, R(x_4, 0, 1), R(x_4, 0, 1)), \\
&\quad T(x_3, R(x_4, 1, 1)))) = \\
&= R(x_1, T(x_2, R(x_3, 0, \overline{x_4})), R(x_2, R(x_3, x_4, x_4), \\
&\quad T(x_3, 1))) = \\
&= R(x_1, T(x_2, x_3 \cdot \overline{x_4}), R(x_2, x_4, \overline{x_3})) = \\
&= R(x_1, \overline{x_2} \cdot x_3 \cdot \overline{x_4}, \overline{x_2} \cdot x_4 \oplus x_2 \cdot \overline{x_3}) = \\
&= \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} \oplus x_1 \cdot (\overline{x_2} \cdot x_4 \oplus x_2 \cdot \overline{x_3}) = \\
&= \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} \oplus x_1 \cdot \overline{x_2} \cdot x_4 \oplus x_1 \cdot x_2 \cdot \overline{x_3}
\end{aligned}
$$

4.3 $\mathrm{BDD}_{g\oplus\overline{x_i}\cdot h\ ,\ x_i+k}$

Imejmo $\mathrm{BDD}_{g\oplus\overline{x_i}\cdot h\ ,\ x_i+k}$ s slike 5, ki predstavlja logično funkcijo $f : \mathbb{B}^4 \to \mathbb{B}$, kjer so spremenljivke označene z x_1, x_2, x_3 in x_4.



Slika 5: Primer $\mathrm{BDD}_{g\oplus\overline{x_i}\cdot h\ ,\ x+k}$

Velja $R(x_i, g, h) = g \oplus \overline{x_i} \cdot h$ in $T(x_i, k) = x_i + k$. Razčlenitveno pravilo uporabimo tako:

$$
\begin{aligned}
f &= val(f,4,1) = R(x_1, val(f_1,4,2), val(f_2,4,2)) = \\
&= R(x_1, T(x_2, val(f_1,4,3)), R(x_2, val(f_3,4,3), \\
&\quad val(f_4,4,3))) = \\
&= R(x_1, T(x_2, R(x_3, val(0,4,4), val(f3,4,4))), \\
&\quad R(x_2, T(x_3, val(f_3,4,4)), R(x_3, val(0,4,4), \\
&\quad val(0,4,4)))) = \\
&= R(x_1, T(x_2, R(x_3, T(x_4, val(0,4,5)), R(x_4, \\
&\quad val(0,4,5), val(1,4,5)))), R(x_2, T(x_3, R(x_4, \\
&\quad val(0,4,5), val(1,4,5))), R(x_3, T(x_4, \\
&\quad val(0,4,5)), T(x_4, val(0,4,5))))) = \\
&= R(x_1, T(x_2, R(x_3, T(x_4,0), R(x_4,0,1))), \\
&\quad R(x_2, T(x_3, R(x_4,0,1)), R(x_3, T(x_4,0), \\
&\quad T(x_4,0)))) = \\
&= R(x_1, T(x_2, R(x_3, x_4, \overline{x_4})), R(x_2, T(x_3, \overline{x_4}), \\
&\quad R(x_3, x_4, x_4))) = \\
&= R(x_1, T(x_2, x_4 \oplus \overline{x_3} \cdot \overline{x_4}), R(x_2, x_3 + \overline{x_4}, x_4 \oplus \\
&\quad \overline{x_3} \cdot x_4)) = \\
&= R(x_1, x_2 + (x_4 + \overline{x_3}), (x_3 + \overline{x_4}) \oplus \overline{x_2} \cdot x_3 \cdot x_4) = \\
&= (x_2 + x_4 + \overline{x_3}) \oplus \overline{x_1} \cdot ((x_3 + \overline{x_4}) \oplus \overline{x_2} \cdot x_3 \cdot x_4) = \\
&= x_1 \oplus \overline{x_2} \cdot x_3 \cdot \overline{x_4} \oplus \overline{x_1} \cdot \overline{x_3} \cdot x_4 \oplus \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot x_4
\end{aligned}
$$

5 Zaključek

Za vsako od številnih vrst odločitvenih grafov obstaja razred logičnih funkcij, pri katerih velikost grafa raste eksponentno s številom spremenljivk. Če gledamo le najslabši primer, potem nobena vrsta urejenih odločitvenih grafov ni bistveno učinkovitejša od druge. V praksi pa se nekatere vrste BDD obnesejo boljše kot druge.

S primerjavo med OBDD in OFDD [2] je bilo ugotovljeno, da za nekatere razrede logičnih funkcij velikost OBDD raste eksponentno s številom spremenljivk, velikost OFDD pa ne. Velja tudi obratno. Velikosti OBDD in 0-sup-BDD za dano logično funkcijo n spremenljivk pa se lahko razlikujeta največ za faktor $n + 1$ [6].

Vseh 288 BDD iz tabele 2 ima podobne lastnosti kot OBDD, OFDD in 0-sup-BDD, zato njihova uporaba ne prinaša bistvenih prednosti. Lahko pa prikazano splošno razčlenitveno pravilo uporabimo kot nadgradnjo OKFDD, ki so sedaj kombinacija le OBDD in OFDD, ne pa tudi 0-sup-BDD.

Splošno razčlenitveno pravilo za OBDD, OFDD in 0-sup-BDD tudi na nov način pojasnjuje njihove lastnosti in lahko pomeni izhodišče za nadaljnje primerjave njihovih zahtevnosti in osnovo za splošen algoritem, s katerim bi lahko tvorili vse tri vrste BDD.

Literatura

[1] Bernd Becker, Rolf Drechsler. How many Decomposition Types do we need ? In *IEEE European Design and Test Conference*, pages 438–443, 1995.

[2] Bernd Becker, Rolf Drechsler, Ralph Werchner. On the Relation Between BDDs anf FDDs. In *LATIN95, LCNS*, 1995.

[3] K. S. Brace, R. L. Rudell, R. E. Bryant. Efficient Implementation of a BDD Package. *27. ACM/IEEE DAC*, pages 40–45, 1990.

[4] R. Drechsler in A. Sarabi in M. Theobald in B. Becker in M. A. Perkowski. Efficient Representation and Manipulaton of Switching Functions Based on Ordered Kronecker Functional Decision Diagrams. Technical report, 1994.

[5] Tsutomu Sasao. AND-EXOR Expressions and their optimization. In Tsutomu Sasao, editor, *Logic Synthesis and Optimization*, chapter 13, pages 287–312. Kluwer Academic Publishers, 1993.

[6] Olaf Schröer, Ingo Wegener. The Theory of Zero-Suppressed BDDs and the Number of Knight's Tours. Technical report, FB Informatik, LS II, Univ. Dortmund, 1995.