

Binary Decision Diagrams

Seminarski rad u okviru kursa
Automatsko rezonovanje
Matematički fakultet

Milana Kovačević
Ivan Ristović

jul 2018.

Binary Decision Diagrams (u daljem tekstu *BDD*) i njihova poboljšanja su strukture podataka za reprezentaciju bulovskih funkcija. Iako u osnovi slični binarnim drvetima, rešavaju problem velikog broja čvorova u drvetu uklanjajući redundantne grane (za bulovsku funkciju sa n argumenata, broj mogućih puteva u binarnom drvetu od korena do lista je 2^n , dok je broj čvorova znatno veći). U ovom radu ćemo detaljnije opisati intuiciju iza BDD struktura, načine za konstrukciju BDD, kao i *ROBDD* - redukovana i uredjena BDD. Ovaj rad će pratiti implementacija BDD u jeziku *C++*, uz prateće delove koda na nekim mestima.

[1]

Sadržaj

1	Bulovske funkcije	2
2	Binarna drveta odlučivanja	2
	Literatura	4

1 Bulovske funkcije

Bulovske funkcije su funkcije koje primaju bulovske argumente i vraćaju bulovsku vrednost. Bulovske vrednosti mogu biti *true* ili *false*. U nastavku ćemo sa 1 označavati *true*, a sa 0 *false*, što je uobičajena konvencija.

Za bulovsku funkciju sa n bulovskih argumenata, postoji 2^n mogućih ulaza. Pošto je povratna vrednost takodje bulovskog tipa, zaključujemo da postoji 2^{2^n} različitih bulovskih funkcija sa n argumenata, što se vidi iz sledeće jednakosti:

$$\underbrace{2 * 2 * 2 * \dots * 2}_{2^n} = 2^{2^n}$$

Funkcije koje primaju neoznačeni broj u opsegu $[0, 2^n - 1]$ se mogu zameniti sa n bulovskih funkcija sa n argumenata. Kao primer, neka je data funkcija F koja prima i vraća neoznačeni ceo broj. Zamenjujemo funkciju F bulovskim funkcijama f_i , gde $i = 0, 1, \dots, n-1$. Argumenti funkcije f_i su n binarnih cifara broja, dok je povratna vrednost f_i vrednost i -te binarne cifre rezultata funkcije F . Drugim rečima, svaka od funkcija f_i računa jednu cifru rezultata. Kao konkretan primer, s obzirom da su neoznačeni brojevi u računarima zauzimaju obično 32 bita, funkciju:

```
unsigned F(unsigned n);
```

možemo zameniti sa 32 bulovske funkcije:

```
bool f0(bool n0, bool n1, bool n2, ... , bool n31);
bool f1(bool n0, bool n1, bool n2, ... , bool n31);
...
bool f31(bool n0, bool n1, bool n2, ... , bool n31);
```

Naravno, moramo se uveriti da su cifre rezultata zaista jednake izlazima bulovskih funkcija. Jedan način za verifikaciju rezultata je primena obe tehnike nad svim mogućim ulazima i poredjenje dobijenih rezultata. Medjutim, čak i za ovako jednostavne funkcije reč je o oko 4 milijarde (2^{32}) mogućih ulaza. Drugi način je da se ove funkcije prikažu putem nekih struktura podataka, i da se funkcije porede tako što se porede njihove reprezentacije preko tih strukture. Drugim rečima, posmatramo funkcije kao podatke. U poglavljima koji slede će biti više reči o strukturama podataka koje se koriste za predstavljanje bulovskih funkcija.

2 Binarna drveta odlučivanja

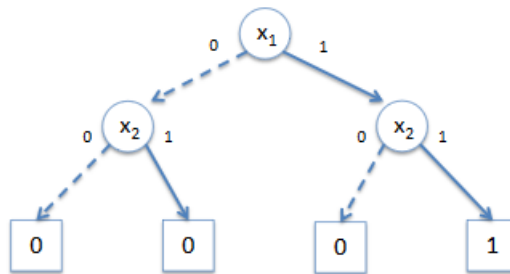
Binarna drveta odlučivanja su u osnovi jako slična binarnim drvetima. Na ovom nivou radimo sa bulovskim funkcijama. Neka je dato n promenljivih x_1, x_2, \dots, x_n koje predstavljaju ulaze u bulovsku funkciju f . U korenom čvoru testiramo jednu promenljivu (bez umanjenja opštosti, krenućemo redom, te neka je to x_1). U zavisnosti od vrednosti te promenljive formiraju se dva pod-drveta - jedno u kome je $x_1 = 0$ (*nisko* pod-drvo), a drugo u kome je $x_1 = 1$ (*visoko* pod-drvo). U svakom pod-drvetu se rekurzivno testiraju ostale promenljive na

isti način. Do listova se dolazi kada više nema preostalih ulaznih promenljivih. Posmatrajući jedan put od korena do lista, dobijamo valuaciju za skup $\{x_1, \dots, x_n\}$.

Uzmimo za primer funkciju:

```
bool f_and(bool x1, bool x2) { return x1 & x2; }
```

Matematički zapis ove funkcije bi bio $x_1 \wedge x_2$. Binarno drvo odlučivanja za ovu funkciju je dat na slici 2.1. Polazeći od promenljive x_1 , formiramo dve grane na osnovu toga da li je $x_1 = 0$ ili $x_1 = 1$. Od sada pa u buduće ćemo grane u kojima je vrednost 0 crtati isprekidanom linijom, a grane u kojima je vrednost 1 punom linijom, zarad preglednosti.

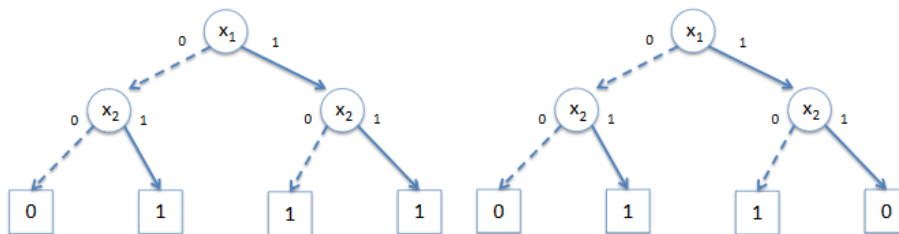


Slika 2.1: Binarno drvo odlučivanja za funkciju \wedge

Slično se mogu definisati i ostale korisne funkcije, na primer f_{or} (\vee) i f_{xor} (\oplus), sa dijagramima na slici 2.2:

```
bool f_or(bool x1, bool x2) { return x1 || x2; }

bool f_xor(bool x1, bool x2) {
    return (x1 && !x2) || (!x1 && x2); 1
}
```



Slika 2.2: Binarna drveta odlučivanja za funkcije \vee i \oplus , redom

¹Ne možemo koristiti operator \wedge jer on operiše nad promenljivima tipa *int*, a ne *bool*.

Binarna drveta odlučivanja imaju neke veoma loše osobine. Najveći problem je njihova veličina. Binarno drvo za n ulaznih promenljivih će imati 2^{n-1} unutrašnjih čvorova i 2^n listova.

Uprkos tome, postoje i neke dobre osobine, pre svega *kanoničnost*. Ukoliko testiramo promenljive uvek u istom redosledu (tada se binarno drvo odlučivanja naziva *uredjeno*), onda je drvo jedinstveno za svaku bulovsku funkciju. Stoga se test ekvivalencije dve bulovske funkcije svodi na testiranje ekvivalentnosti njihovih binarnih drveta odlučivanja. Nažalost, zbog velikog broja čvorova u drvetima, problem je eksponencijalne složenosti u odnosu na broj ulaznih parametara.

Kako bi se rešio problem eksplozije broja čvorova u drvetu, formiraju se unapredjenja binarnih drveta odlučivanja - binarni dijagrami odlučivanja (*BDD*). O njima će biti više reči u poglavlju koje sledi.

Literatura

- [1] BDD. on-line at:
<https://www.cs.cmu.edu/~fp/courses/15122-f10/lectures/19-bdds.pdf>.