Fakulta informatiky a informačných technológií STU v Bratislave Ilkovičova 2, 842 16 Bratislava 4

Tibor Dulovec

Zadanie 3 – Binárne rozhodovacie diagramy

DSA LS 2020/21
Cvičiaci P. Lehoczký
Pondelok 11:00 – 12:50

Obsah

Opis programu

Program slúži na prácu s binárnymi rozhodovacími diagramami. Pre lepšiu pamäťovú efektívnosť program obsahuje aj zredukovanie a vráti výstup po vložení kombinácií vstupných premenných.

Objekt uzla

Uzly sú vytvárane podľa triedy „**Node**“. Jeden uzol obsahuje hodnotu a môže obsahovať ľavý a pravý uzol. Má veľkosť, ktorá symbolizuje počet uzlov pod sebou a hĺbku, ktorá symbolizuje ako hlboko je tento uzol v diagrame.

Trieda má iba dve funkcie. Jedna je pomocná na rozdelenie Vektora, ktorá sa používa v druhej rekurzívnej funkcií **insertToNode**, ktorá slúži na vkladanie nových uzlov do tohto objektu.

Časové a pamäťové zložitosti

BDD_create

Pamäťová zložitosť: **O(n)**
Časová zložitosť: **O(n)**

BDD_reduce

Pamäťová zložitosť: **O(n)**

Časová zložitosť: **O(n^3),** vzhľadom na trojitý for cyklus

BDD_use

Pamäťová zložitosť: **O(n)**
Časová zložitosť: **O(n)** v najhorších prípadoch. Ale väčšinou časová zložitosť býva **O(h)**, kde h je výška diagramu

Opis funkcií

BDD_create

Funkcia po vložený atribútu booleovskej funkcií vytvára rozhodovací diagram. Na opis tenjo funkcie je použitý vektor, ktorý je vo formáte String. Daný vektor sa potom v rekurzívnej funkcii rozdeľuje na polovice a jednu vloží do ľavého uzla, a druhú do pravého uzla. Toto sa deje dovtedy, kým tá časť vektora na rozdelenie má aspoň veľkosť 2. Funkcia pri vytváraní vypočítava z opisu funkcie počet použitých premenných a počet vytvorených uzlov, ktoré sa potom používajú na vypočítavanie efektívnosti po redukovaní diagramu.

```
BDD bdd = new BDD();
bdd.BDD_create( bf: "10101101");
bdd.print();
```

```
3:     10101101
2:    1010 1101
1:   10 10 11 01
0: 1 0 1 0 1 1 0 1
```

Obrázok 1, vizualizácia zostrojeného rozhodovacieho diagramu o veľkosti 3premenných s očíslovanými vrstvami podľa hĺbky

BDD_reduce

Po zostavení binárneho rozhodovacieho diagramu vždy vznikne veľké množstvo duplikujúcich uzlov, ktoré znižujú pamäťovú efektivitu. Preto je naprogramovaná funkcia BDD_reduce, ktorá odstraňuje duplikujúce uzly a nahrádza ich odkazmi na iné. Týmto spôsobom sú na každej vrstve iba unikátne uzly, na ktoré môže odkazovať aj niekoľko uzlov.

Funkčnosť funkcie spočíva v tom, že sa vyberie z diagramu množina unikátnych prvkov pre každú úroveň. Následne každý prvok z tejto množiny porovnáva svoje uzly s každým ďalším prvkom. Po nájdení duplikujúceho sa jeden z uzlov odstráni a nahradí odkazom na ten druhý uzol.

Po prejdení celej vrstvy sa prepočíta veľkosť diagramu a cez návratovú funkciu sa pošle počet odstránených uzlov.

V prípade, že sa do BDD_reduce funkcie pošle chybný uzol, funkcia vráti „-1“.

```
BDD bdd = new BDD();
bdd.BDD_create( bf: "10101101");
int countOfRemovedNodes = bdd.reduce();
System.out.println(countOfRemovedNodes + " nodes was removed");
bdd.print();
```

```
7 nodes was removed
3:     10101101
2:    1010 1101
1:   10 11 01
0: 1 0
=======Pocet uzlov 8=======
```

Obrázok 2, vizualizácia zostrojeného rozhodovacieho diagramu o veľkosti 3premenných s očíslovanými vrstvami po zredukovaní

BDD_use

Pomocou tejto funkcie vieme získať výsledok booleovskej funkcie pri zadaní rôznej kombinácií premenných.

Funkcia prechádza od root uzla až po list v danom diagrame, podľa zadanej kombinácie. Na konci funkcie, funkcia vráti správny výsledok 0 alebo 1.

Návratová hodnota je vo formáte znaku. Ak sa vráti znak „-“, tak počas vykonávaní funkcii vznikla chyba.

Testovanie

print

Funkcia pre objekt BDD, slúžiaca na vizualizáciu zostrojeného stromu. Funkcia po redukovaní stromu zobrazuje iba unikátne uzly a nie odkazy na uzly.

generateBf

Funkcia na vygenerovanie náhodnej booleovskej funkcie. Vstupný atribút je počet premenných. Podľa tohto atribútu funkcia vygeneruje a vráti adekvátne dlhú booleovskú funkciu.

checkBf

Funkcia slúži na overenie zostrojeného diagramu. Vstupný parameter je diagram, ktorý chceme overiť. Následne sa automaticky skúšajú postupne všetky možné kombinácie premenných pomocou funkcie BDD_use() a ich výsledky spája. Po prejdení všetkých kombinácií sa porovná výsledný spojený String s pôvodnou booleovsky funkciou a ak sú hodnoty identické, tak rozhodovací diagram bol zostrojený správne.

testManyBDDs

Hlavná testovacia funkcia. Testuje celý proces vytvárania, overovania a redukovania. Vstupnými parametrami je počet testov, ktoré sa majú vykonať a počet premenných, podľa ktorých sa náhodne vygenerujú booleovské funkcie.

Priebeh jedného testu prebieha nasledovne.
Vygeneruje sa náhodná booleovská funkcia funkciou **generateBf**. Vytvorí sa nový binárny rozhodovací diagram pomocou funkcie **BDD_create**. Overení sa pomocou funkcie **checkBf**. Ak zostrojený strom je vytvorený nesprávne, celé testovanie sa pozastaví.
Ďalšia funkcia je **BDD_reduce**, ktorá diagram zredukuje a znova sa overí pomocou **checkBf**, či diagram stále funguje správne a nebol nijako poškodený.

Po vykonaní všetkých testov sa vypíše kompletné vyhodnotenie testov.

Vyhodnotenie testov

Po skončení hlavnej testovacej funkcie **testManyBDDs**, sa vypíše vyhodnotenie testov.

Toto vyhodnotenie obsahuje **Reduction rate** ako pomer uzlov pred a po redukovaní.
V priemere to je **92%**, čo je výrazne zefektívnenie práce s rozhodovacími diagramami.

Počas vykonávania funkcie **testManyBDDs,** sa časovo merajú všetky funkcie a vo vyhodnotení sú spočítané všetky tieto časy z konkrétnych časti kódu.
Najviac času zaberá redukovanie stromu. Časovo redukovanie zaberá v priemere **75%** celého testu. Oproti 2% z celkového času, ktoré zaberá vytváranie stromu to je obrovský rozdiel, ale je to výhodný kompromis za pamäťovú efektívnosť, ktorú týmto získame.

Posledný časový blok údajov sú priemerné časy za celý test a priemerné časy pre danú časť kódu. V prípade 14 premenných je priemerný čas okolo **30 sekúnd,** čo osobne považujem za veľmi dobrý čas. V prípade 13 premenných to je 9 sekúnd.

```
====SUMMARY FOR ALL TESTS====
> Count of tests: 2000
> Count of variables: 14
-----------------------------
> Nodes created: 65534000
> Nodes removed: 60900087
> Reduction rate: 92,93%
-----------------------------
> Total time: 68250 ms
> Total time for create: 1294 ms
> Total time for use: 4436 ms
> Total time for reduce: 57260 ms
-----------------------------
> Average time for test: 34 ms
> Average time for create: 0 ms
> Average time for use: 2 ms
> Average time for reduce: 28 ms
```

Obrázok 3, vyhodnotenie testu funkciou testManyBDDs pre 2000 testov a 14 premenných