Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Inkrementális tranzitív lezárt számítás gráfokban

Szabó Tamás (KCTZMJ), IV. évf, MSc mérnök inf. szakos hallgató
Konzulens: Bergmann Gábor Phd hallgató, MIT
Szolgáltatásbiztos rendszertervezés szakirány
Önálló laboratórium I. absztrakt
2010/11. II. félév

Egy irányított (irányítatlan) gráf tranzitív lezártja olyan (***u***;***v***) csúcspárokból áll, amelyekre teljesül az, hogy a gráfban létezik irányított (irányítatlan) út az ***u*** csúcsból a ***v*** csúcsba. A tranzitív lezárt ismeretének nagy jelentősége van például azokban az esetekben, amikor egy telekommunikációs, szállítmányozási vagy közösségi hálózatot gráfokkal modellezünk. A VIATRA2 modelltranszformációs keretrendszerben például típus leszármazási vagy tartalmazási viszonyok vizsgálatakor lenne nagy segítség egy olyan könyvtár, amely több tranzitív lezáró algoritmust is tartalmazna. Fontos azonban megjegyezni, hogy az alapgráf megváltozása esetén a tranzitív elérhetőségek halmaza is módosul. A telekommunikációs hálózat esetében például egy összeköttetés hibája miatt elszigeteltté válhatnak egyes csomópontok. A nagyméretű gráfok esetén azonban a szükséges számítások elvégzése már kritikus a teljesítményre nézve.

A félév során több statikus és inkrementális tranzitív lezártat számító algoritmust is implementáltam. A statikus algoritmusok - mint amilyen a Floyd-Warshall, a gráfbejárásokon alapuló algoritmusok – a gráf módosítása esetén a tranzitív elérhetőségek teljes halmazát újra és újra kiszámítják. Ez nagy gráfok esetén egyrészt sok ideig tart, másrészt fölösleges is, mivel a tranzitív elérhetőségekben történt változásokra lehet következtetni a gráfban történt változásokból. Az inkrementális algoritmusok ezt az ötletet használják ki, a tranzitív elérhetőségek inicializálása után a gráfban történt módosításoknak megfelelően karban tartják a korábban kiszámított elérhetőségek halmazát. Inkrementális algoritmusra példa a Counting [1], DRED [1] és a King [2] algoritmusok, utóbbinak a félév során egy optimalizált változatát is elkészítettem.

Az implementációhoz a Java programozási nyelvet választottam, a tesztelés pedig JUnit felhasználásával történt. Az inkrementális algoritmusok esetén a futási idő két részből tevődik össze: egyrészt magának az inicializációnak az idejéből, másrészt pedig a gráfban történt változások után a frissítésnek az idejéből. A futási idő mellett a memória felhasználását is megmértem az egyes algoritmusoknak. A mérési eredmények alapján elmondhat, hogy a King algoritmus alacsony frissítési idővel rendelkezik általános gráfokon, a DAG jellegű gráfokra a Counting algoritmus azonban még ennél is gyorsabb. A DRED algoritmus azonban csak él beszúrás esetén rendelkezik alacsony frissítési idővel. Mindazonáltal, a King algoritmus esetén alkalmazott komplex adatszerkezetek miatt az inicializáció ideje gyorsan növekedik a gráf méretének növekedésével.

Hivatkozások

[1] Ashish Gupta, Inderpal Singh Mumick, and V. S. Subrahmanian. 1993. Maintaining views incrementally. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data* (SIGMOD '93), Peter Buneman and Sushil Jajodia (Eds.). ACM, New York, NY, USA, 157-166.

[2] Valerie King. 1999. Fully Dynamic Algorithms for Maintaining All-Pairs Shortest Paths and Transitive Closure in Digraphs. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science* (FOCS '99). IEEE Computer Society, Washington, DC, USA, 81-.