50%-OS SZÁMÍTÓGÉP-IDŐ MEGTAKARÍTÁS A TIGIT SZOLGÁLTATÁSBAN

Válas György
Országos Műszaki Információs Központ és Könyvtár

A számítógépes szakirodalmi szelektív információterjesztési szolgáltatások – amilyen az OMIKK *TIGIT (Témára Irányuló Gépi Információterjesztés)* nevű szolgáltatása is – költségeinek az adatbázis előfizetése mellett legnagyobb tétele a számítógép-idő. Mivel hazánkban szelektív információterjesztésre több helyen is használják a *Horváth Iván* (KFKI) készítette BINAR programcsomagot, talán nem érdekltelen, hogy *milyen módon szorítottuk nagyjából felére* ezt a költségtételt az OMIKK-ban. Külön kiemelendő, hogy minimális szellemi munkatöbbleten kívül az alkalmazott eljárás semmilyen többlet-ráfordítást nem igényel.

Eljárásunk lényege: *maximális alkalmazkodás a keresőprogram működési logikájához* a profilszerkesztésben. A megoldás részleteit a *Függelék* tartalmazza.

A keresőprogram és a keresőprofilok

A szakirodalmi szelektív információterjesztés során olyan adatállományokat dolgozunk fel, amelyek minden logikai rekordja egy-egy új szakirodalmi dokumentum leírását tartalmazza. A felhasználó kérte témakört keresőprofillal fogalmazzuk meg, a keresőprogram által megkívánt formában. Minden keresőprofil egy-egy figyelt témakör jellemző szavait, kifejezéseit, deszkriptorait és esetleges egyéb index-kifejezéseit (osztályozási jelzet stb.) tartalmazza, meghatározott logikai összefüggésben. A releváns dokumentumok kiválasztása a keresőprofilok és az adatbázis friss anyagát tartalmazó adatállomány összehasonlításával történik.

A BINAR programcsomag keresőprogramja az adatbázis szempontjából szorosan (szekvenciálisan), a keresőprofilok szempontjából kötegelten (*batch* üzemmódban) végzi a feldolgozást. Az adatbázis soron következő rekordját sorban a köteg valamennyi keresőprofiljával összehasonlítja. Találat esetén a szóban forgó keresőprofil azonosítójával készít egy output-rekordot, majd folytatja az összehasonlítást a következő profillal. Így egyes adatbázis-rekordokból több output-rekord is készülhet, másokból esetleg egy se. Ha a rekord összehasonlítása a köteg valamennyi keresőprofiljával megtörtént, akkor kezdi a keresőprogram az összehasonlítást a következő rekorddal. A keresőprogram futása után a relevánsnak talált rekordokat a *rendezőprogram* gyűjti össze keresőprofilként.

Ez az összehasonlítási művelet nagyon gépidőigényes. A keresőprogram futása során a felhasznált CPU-idő* még multiprogramozott környezetben (R-35 számítógépen) is elérheti a start–stop idő 75–80%-át, annak ellenére, hogy a keresőprogram magvát képező, a tényleges összehasonlítást végző EXERT rutin a futás gyorsítása érdekében *assembly* nyelven készült.

A keresés során az egyes „keresőszavak" összehasonlítása „igaz" értéket ad akkor, ha a rekordban sikerült megtalálni a keresett karaktersorozatot, „hamis" értéket akkor, ha nem sikerült. Releváns a rekord egy keresőprofilhoz akkor, ha a keresőprofilt alkotó teljes logikai kifejezés „igaz" értékű.

A keresőprogram tulajdonságainak alapos vizsgálata során úgy találtuk, hogy ezekhez a tulajdonságokhoz megfelelően alkalmazkodva, közülük az előnyösöket jól kihasználva *jelentősen csökkenthetjük a keresés során szükséges összehasonlítások számát,* átlagosan kevesebb összehasonlítás nyomán kaphatjuk meg a keresőprofil logikai kifejezésének értékét.

A megoldás lényege az, hogy valahányszor a soron következő keresőprofilhoz a soron következő adatrekord irreleváns, *minél hamarabb ki kell léptetni a keresőprogramot a keresés folyamatából.* A nyilvánvaló irrelevanciát ne hosszú, sok gépidőt elpocsékoló összehasonlítás-sorozat után fedezze fel a program, hanem minél hamarabb, lehetőleg már az első néhány összehasonlítás nyomán. A keresőprogram erre azzal a tulajdonságával ad lehetőséget, hogy a logikai kifejezés végeredménye szempontjából közömbös összehasonlítási és logikai műveleteket nem hajtja fölöslegesen végre, valamint azzal,

---

* Central Processing Unit – a számítógép központi feldolgozó egysége – *A szerk.*

hogy a keresőprofil felépítése egyértelműen meghatározza a műveletek elvégzésének sorrendjét. Így a keresőprofil megszerkesztésekor nemcsak a felhasználó kérdése szempontjából tervezzük meg a keresési stratégiát, hanem – akár tudatosan tesszük ezt, akár nem – *részleteiben is megtervezzük az összehasonlítási műveletek sorrendjét.* Jobb tehát, ha ez utóbbit tudatosan tervezzük, úgy, ahogy az céljainknak megfelel.

A kidolgozott módszer kísérleti ellenőrzése

A keresőprogram megismerése alapján kidolgozott profilszerkesztési elvek gyakorlati kipróbálására 1980 decemberében kísérletet végeztünk. 12 régóta futó INSPEC keresőprofil-köteget szerkesztettünk át a kidolgozott elvek szerint.

Két keresőprofil-köteget állítottunk össze. Egyik köteg a 12 eredeti keresőprofilt tartalmazta, a másik ugyanazon keresőprofilok átszerkesztett változatát. Mindkét köteggel elvégeztük a keresést ugyanazon a számítógépen (az OMIKK R-20 gépén), teljesen azonos körülmények között, monoprogramozású környezetben, ugyanazon az adatállományon (az INSPEC adatbázis 1980/22-es mágnesszalagján). A két futás adatait az *1. táblázat* tartalmazza. A monoprogramozásra való tekintettel a start–stop idő jó mérőszáma a felhasznált gépidőnek; ezzel számolva az elvégzett kísérletben a gépidő-megtakarítás majdnem elérte az 50%-ot.

1. táblázat

Az ellenőrző kísérlet adatai

	A futás		
	kezdete	befejezése	időtartama
Eredeti profilokkal	14.46.03	16.54.43	2 óra 08 perc 40 mp.
Átírt profilokkal	17.32.37	18.45.04	1 óra 12 perc 27 mp.

Külön kiemelendő, hogy a keresőprofilok átszerkesztése (a „behatárolás" révén, ld. *Függelék) a keresés pontosságát is növelte.* Az eredeti keresőprofilok 671 találatot adtak. Az átszerkesztés nyomán 19-cel (0,28%) csökkent az irreleváns találatok száma, és 10-zel (0,15%) a csak részben releváns találatok száma. A valóban releváns találatok közül 3 (0,045%) veszett el, közülük 1 indexelési hiba miatt, 2 pedig (0,03%) profilszerkesztési hibából. Ez utóbbi veszteség nagyobb figyelemmel elkerülhető.

A tapasztalatok hasznosítása

Bár a 12 keresőprofil egyszeri futtatásával végzett kísérlet kevés, további párhuzamos futtatásra anyagi korlátok miatt nem volt módunk. Mindenesetre e korlátozott kísérletből is kitűnt, hogy a gépidő-megtakarítás jelentős, a találati veszteség pedig elhanyagolható, még ha ezek számszerű értékében maradt is némi bizonytalanság.

A kísérletet követő 8 hónap során az INSPEC adatbázishoz tartozó valamennyi keresőprofilunkat átszerkesztettük a gépidő-takarékosság szempontjai szerint. Ennek során a párhuzamos futtatás lehetőségének hiánya miatt különös gondossággal jártunk el. Ezzel egyidejűleg alkalmaztuk a kidolgozott elveket az INIS és a COMPENDEX adatbázis keresőprofiljaira is, figyelembe véve az adatbázisok egyedi tulajdonságait.

1981 eleje óta minden újonnan írt keresőprofilunkban tekintettel vagyunk a gépidő-takarékosság szempontjára. Azóta a leírt módszernek *semmilyen negatív hatását nem tapasztaltuk,* pozitív hatása azonban a gépidő-költségek csökkentésében számottevő.

*

Befejezésül köszönetemet fejezem ki *Lőcs Gyulá*-nak és *Bartucz József*-nek (KFKI), akik lehetővé tették számomra a BINAR programcsomag keresőprogramjának alapos megismerését, és *Nagy Károly*-nak (OMIKK), aki az ellenőrző kísérlet gyors és pontos elvégzésében volt segítségemre.

Függelék

A BINAR programcsomaghoz írható keresőprofil szerkezete a következő:

Az adatbázisban keresendő karaktersorozatokat „keresőszó"-nak nevezzük. A „keresőszó" lehet természetesen többszavas össszetett kifejezés vagy szócsonk is.

Minden „keresőszó"-hoz tartozik egy „keresési mód". Ez szabja meg, hogy a „keresőszó" karaktersorozat keresése az adatbázisban milyen illesztéssel történjen. A „keresőszó"-nak mind az eleje, mind a vége illeszthető mezőhatárra (jele: E), szóhatárra (jele: T) vagy tetszőleges karakterre (jele: *). Így a kezdet és a vég három-három lehetőségének kombinációi kilenc „keresési mód"-ot adnak: EE, ET, E*, TE, TT, T*, *E, *T, **.

Az azonos adatmezőben kereshető szavak „paraméter"-be csoportosíthatók, ha a keresésben logikai „VAGY" kapcsolatban állnak, tehát logikai összeg tagjai. A „paraméter"-hez tartozik egy „paraméter-azonosító" és egy „típus". A „paraméter-azonosító"-nak a logikai kifejezések felírásában van szerepe. A „típus" azt írja le, hogy a „paraméter"-be tartozó „keresőszavak" az adatbázis rekordjainak mely adatmezőjében vagy adatmezőiben keresendők.

A „paraméter"-ekből alkotott logikai kifejezés az, amellyel az adatbázis rekordjai közül végülis kiválasztjuk a relevánsakat. A logikai kifejezés három féle műveletet tartalmazhat: szorzást (AND), összeadást (OR), negált szorzást (ANDNOT). Ezenkívül tartalmazhat (közvetett úton kijelölt) zárójeleket.

A keresőprogramnak a gépidő-gazdálkodás szempontjából lényeges tulajdonságai a következők:

1. Minden „keresőszó"-ra csak akkor végzi el az összehasonlítást, amikor a logikai kifejezés feldolgozásában odáig ér. Amelyik „keresőszó"-ig nem jut el, arra nem is használ gépidőt.

2. A logikai műveletekre a szokásos precedencia-szabály érvényes: a szorzás és negált szorzás „erősebb kötés", mint az összeadás, tehát zárójel nélküli kifejezésben vagy adott zárójelen belül először a szorzatokat, negált szorzatokat számítja ki, majd a kész szorzatokat tekinti az összeg tagjainak.

3. Az azonos szintű műveleteket balról jobbra hajtja végre.

4. A logikai kifejezés feldolgozása során a zárójeles részkifejezéseket akkor számítja ki, mikor a kezdőzárójelig eljut. Ekkor a zárójelen belüli kifejezést teljes logikai kifejezésnek véve a 2.–4. pontok szerint számítja ki. A „paraméter" tekintendő a legbelső zárójelnek, ez mindig egy (esetleg csak egytagú) logikai összeget tartalmaz.

5. A logikai szorzat vizsgálatából kilép, amint valamelyik tényezőt „hamis" értékűnek találja. Ekkor már a szorzat értéke a további (nem vizsgált) tényezők értékétől függetlenül „hamis".

6. A logikai összeg vizsgálatából (beleértve a „paraméter"-en belül a „keresőszavak" vizsgálatát) kilép, amint valamelyik tagot „igaz" értékűnek találja. Ekkor már az összeg értéke a további (nem vizsgált) tagok értékétől függetlenül „igaz".

7. Egy „keresőszó" keresési ideje erősen függ a „keresési mód"-tól, azon belül is elsősorban a karaktersorozat kezdetének illesztésétől. Ha minden szókezdetre illesztve el kell végezni az összehasonlítást, ez sokkal több összehasonlítás, mint ha csak a mezőkezdetre illesztve hasonlítunk. Még sokkal több összehasonlítást jelent, ha minden egyes karakterre illesztve hasonlítunk. Például az INSPEC adatbázis deszkriptoraiban végzett keresés során az E*, T* és ** „keresési mód"-ok időigényének aránya durva becslés szerint 1 : 2,5 : 25 körül lehet.

8. Egy „keresőszó" keresési ideje erősen függ az őt tartalmazó „paraméter" „típus"-ától. Minél többszörösen ismétlődő mezőben, minél több, minél hosszabb szóból álló mezőben kell keresni, annál lassúbb a keresés. Leggyorsabb a keresés általában az osztályozási jelzet, szekciókód típusú mezőkben, ennél valamivel időigényesebb a deszkriptorokban végzett keresés. Nagyon drága az *abstract* teljes szövegében vagy három-négyféle adatmezőben egyszerre végzett keresés. Minden adatbázisra külön kell meghatározni, hogy az egyes „típus"-ok időigénye hogyan aránylik egymáshoz.

9. Minden zárójelezés rekurzív szubrutinhívást eredményez, ezért minden fölöslegesen beiktatott zárójel időpazarlás.

10. Ha már feldolgozott logikai részkifejezés megismétlődik, nem végzi el ismételten az összehasonlítást, hanem a korábban már meghatározott és tárolt logikai értéket helyettesíti be.

11. A „keresőszó"-val való tényleges összehasonlítás előtt a szóhosszak összehasonlítását végzi a keresőprogram. Ha az adatrekord összehasonlításra kerülő szava rövidebb, mint a „keresőszó" első szava, meg sem történik a tényleges összehasonlítás. Ezért, ha a „keresőszó" első szava nagyon hosszú, ez nemhogy nem növeli, de még csökkenti is a keresési időt, hiszen az esetek többségében csak a hosszak összehasonlításáig jut el a keresőprogram.

A keresőprogram felsorolt tulajdonságai olyan profilszerkesztési elveket sugallnak, amelyek alkalmazása a gépidő-igény szempontjából előnyös.

a) Ha van olyan egy vagy néhány (rendszerint csonkolt) osztályozási jelzet, szekciókód, vagy ezek hiányában legalább deszkriptor, amelyek valamelyike minden releváns rekordban elő kell forduljon, akkor ezekkel „behatárolhatjuk" a keresőprofilt, vagyis olyan szorzatként építhetjük fel, amelynek első tényezője a kérdéses (csonkolt) osztályozási jelzeteket, szekciókódokat, deszkriptorokat keresi. Mivel az irreleváns rekordok többségére ez a tényező „hamis" értéket ad, a legtöbb rekordra ezzel be is fejeződik a keresés, csak viszonylag kisszámú rekord esetén lép be a keresőprogram a keresés időigényesebb részébe (1–5. tulajdonságok).

b) Ha egyes „keresőszavak"-kal nagyon időigényes „típus"-ban vagy „keresési mód"-dal kell keresnünk (pl. *abstract* szövegében, ** „keresési mód"-dal), akkor azt úgy kell a keresőprofilban elhelyeznünk, hogy minél kevesebb rekordban jusson el a keresés eddig a pontig. Olyan logikai szorzatban kell tehát elhelyeznünk, amelynek előző tényezői már a legtöbb irreleváns rekordon „hamis" értéket eredményeznek (1–8. tulajdonságok).

c) A logikai szorzat tényezőit olyan sorrendben célszerű felírni, hogy a nagy valószínűséggel „hamis" értéket adó tényezők vizsgálata történjen meg előbb (1–5. tulajdonságok).

d) A logikai összeg tagjait olyan sorrendben célszerű felírni, hogy a nagy valószínűséggel „igaz" értéket adó tagok vizsgálata történjen meg előbb (1–4, 6. tulajdonságok). Ez érvényes a „paraméter"-en belüli „keresőszavak" sorrendjére is.

e) Kerüljük az időigényes „keresési mód"-okat. Egy T* „keresési mód"-ú „keresőszó" helyett két E* „keresési mód"-ú, egy ** „keresési mód"-ú helyett tíz T* vagy húsz E* „keresési mód"-ú rendszerint időnyereség (7. tulajdonság).

f) Kerüljük az időigényes „típus"-okat. Csak akkor használjunk ilyet, ha másképpen nem érhető el a célunk (8. tulajdonság).

g) Ha a precedencia-szabály (2. tulajdonság) segítségével elkerülhető a zárójelezés, kerüljük is el (9. tulajdonság). Kivétel ez alól, ha a zárójelezés ismételhetővé tesz egy logikai részkifejezést. Ez utóbbi esetben a zárójelezés előnyös (10. tulajdonság).

h) Ha a „keresőszó" egyetlen nagyon hosszú szóból áll, azt nem érdemes csonkolni. Gyorsabb a keresés, ha teljesen kiírjuk a hosszú szót (11. tulajdonság).

A felsorolt elvek közül elsősorban az a) és b) elvek érvényesítése jelent döntő gépidő-nyereséget.

VÁLAS György: 50%-os számítógép-idő megtakarítás az OMIKK TIGIT szolgáltatásában

A számítógépes szakirodalmi szelektív információterjesztés (SDI) költségeinek az adatbázis előfizetése mellett legnagyobb tétele a számítógép-idő. Az OMIKK-ban (Országos Műszaki Információs Központ és Könyvtár) ezt a költség-tételt nagyjából felére sikerült leszorítani. Az eljárás lényege: a maximális alkalmazkodás a keresőprogram működési logikájához a profilszerkesztésben. Így a gépidő-nyereséget, némi szellemi munkától eltekintve, befektetés nélkül lehetett elérni. A cikk ezt a munkát foglalja össze. A BINAR keresőprogram felhasználói részére a Függelék ismerteti az alkalmazott módszer részleteit, és az ezek alapjául szolgáló program-tulajdonságokat.

* * *

ВАЛАШ, Дь.: 50%-ая экономия машинного времени в службе машинного избирательного распространения информации (ИРИ) в ОМИКК

Наибольшая составляющая расходов на машинное ИРИ — после расходов на приобретение магнитоленточной базы данных — это затраты машинного времени. В ОМИКК эти затраты удалось сократить примерно в два раза. Суть метода заключается в максимальном приспособлении при составлении поискового профиля к поисковой логике программы поиска информации. Благодаря этому экономия машинного времени достигается без всяких дополнительных затрат, не считая незначительного труда. Для пользователей поисковой программы пакета БИНАР в приложении рассматриваются подробности этого метода и особенности программы, которые были при этом использованы.

* * *

VÁLAS, Gy.: A technique for 50 per cent computer time saving in the SDI service OMIKK/TIGIT

One of the highest cost components of the Selective Dissemination of Information (SDI) is, apart from the database subscription fee, the computer time. In the SDI service TIGIT of the National Technical Information Centre and Library (OMIKK) the computer cost component has been reduced to about half of its original costs. The technique is based on the maximum compliance with the logic of the search program operation to be kept in mind during profile construction. It allowed to achieve machine time gain without investment, utilizing only intellectual effort. The author outlines the basic concepts of his method. In addition, for those using the BINAR search program, a description of the program features and the details of the time saving technique based on them is given in the Appendix.

* * *

VÁLAS, GY.: 50%-ige Maschinenzeiteinsparung bei der rechnergestützten selektiven Informationsverbreitung „TIGIT" im OMIKK

Den grössten Kostenanteil der rechnergestützten selektiven Informationsverbreitung betragen – ausser den Kosten für die Abonnierung der Datenbasis – die Maschinenzeiten. In der OMIKK (Ungarisches Technisches Informationszentrum und Bibliothek) ist es gelungen, diesen Kostenposten auf fast die Hälfte herunterzudrücken. Das Wesen des Verfahrens besteht in der maximalen Anpassung bei der Profilaufbereitung an die Funktionslogik des Suchprogramms. Der Gewinn an Rechnerzeit konnte – abgesehen von der geistigen Arbeit – ohne Investitionen erzielt werden. Der Artikel beschreibt diese Arbeit. Für die Verwender des Suchprogramms BINAR werden im Anhang die Details der angewandten Methode und die diesen zugrunde liegenden Programmcharakteristika beschrieben.

* * *