MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 20/1978.

BONYOLULT LOGIKAI KIFEJEZÉSEK KIÉRTÉKELÉSÉNEK SZÁMITÁSTECHNIKAI ÉS OPTIMALIZÁLÁSI PROBLÉMÁI

Ratkó István

Az ÁSzSz HwB gépén történő kórházi morbiditási adatfeldolgozás (ld. [1]) vetette fel a következő feladatot.

Adott egy fix hosszúságú rekordokból álló adatfile. A szakembereket igen gyakran csak speciális feltételeknek elegettevő adatok érdeklik. Ezek a feltételek logikai kifejezések formájában irhatók fel. A feltételekkel kapcsolatban a következőket tudjuk: a) sok van belölük, b) nem rögzithető előre minden lehetséges feltétel, amire szükség lesz. A programozási nyelv szokásos eszközei nem biztositanak olyan lehetőséget, amellyel ez a két követelmény kielégithető. Cikkünkben mutatunk egy megoldási módszert, arra a nem lényegtelen szempontra is ügyelve, hogy a feltételvizsgálat ideje minél kisebb legyen.

1. A modell

A rekord álljon N adatelemből, az i-edik adatelem értékét jelölje τ_i, τ_i lehetséges értékeinek halmazát pedig H_i. (H_i lehet pl. egy intervallum, de lehet bonyolultabb halmaz is). Valamely konkrét adatfeldolgozásnál az adatfile bizonyos feltételeknek elegettevő rekordjaira van csak szükség. Hogyan irjunk fel egy az ezen rekordokat kiválasztó logikai kifejezést? A kifejezés akkor és csak akkor legyen igaz, ha a rekordra szükségünk lesz. Tegyük fel, hogy a logikai kifejezésben az $i_1, i_2, \ldots, i_M$ sorszámú adatelemekre vonatkozó feltételek szerepelnek. A k-adik adatelemmel kapcsolatos feltétek igy néznek ki:

$$(\tau_k \in A) \quad \text{ahol} \quad A \subset H_k.$$

A $(\tau_k \in A)$ itélet tulajdonképpen $|A|$ számú diszjunkcióból áll; itt $|A|$ az A halmaz elemszáma.

Jelölje $A_{i_k, j}$ H_{i_k} azon részhalmazait, amelyek $(\tau_{i_k} \in A_{i_k,j})$ formában szerepelnek a rekordokat kiválasztó logikai kifejezésekben ($j = 1,2, \ldots, r_k$, $k = 1,2, \ldots, M$). Nyilvánvaló, hogy logikai kifejezésünknek legalább

$$\sum_{k=1}^{M} \sum_{j=1}^{r_k} |A_{i_k,j}| \quad \text{tagja (konjunkció, diszjunkció vegyesen) van.}$$

Ezt hagyományos módon beépiteni a programba reménytelen, sőt legtöbbször lehetetlen. A fenti tagszám ugyanis többezeres nagyságrendü is lehet a bevezetőben emlitett feladatnál.

2. A módszer

Mit tehetünk? Redukáljuk a kifejezésben szereplő változók számát. Ezt a következő egyszerü ötlettel érhetjük el.

Definiáljuk a $Z_{i_{k,j}}(s)$ függvényt a következőképpen:

$$Z_{i_{k,j}}(s) = \begin{cases} 0, \text{ ha } s \epsilon A_{i_{k,j}} \\ \\ 1, \text{ ha } s \notin A_{i_{k,j}} \end{cases} \qquad (s \epsilon H_k)$$

Ezzel elértük, hogy a $(\tau_k \epsilon A_{i_{k,j}})$ $|A_{i_{k,j}}|$ számú diszjunkció helyett az egyetlen $Z_{i_{k,j}}(\tau_k) = 0$ tag áll, ugyanis nyilvánvaló, hogy $(\tau_k \epsilon A_{i_{k,j}})$ akkor és csak akkor igaz, ha $Z_{i_{k,j}}(\tau_k) = 0$ igaz. Megjegyezzük, hogy kifejezésünk most legalább $\sum_{k=1}^{M} r_k$ tagból áll.

Bizonyos mértékü egyszerüsitést már elértünk, ha azonban a logikai kifejezés kiértékelési idejét, s igy a futási időt csökkenti akarjuk, további meggondolásra van szükségünk.

A rekordokat kiválasztó L logikai kifejezésünket hozzuk diszjunktiv normálalakra:

$L = L_1 VL_2 VL_3 V \ldots VL_{d-1} VL_d$, ahol tehát L_i konjunkciókból áll.

Ez mindig megtehető (ld. [2]).

Bár a következőkben a mondottakat FORTRAN nyelven szemléltetjük, hasonlót állithatunk COBOL nyel alkalmazása esetén.

Nyilvánvaló, hogy az

IF(L.EQ.FALSE) GO TO 2 (a rekord kihagyandó)

utasitás végrehajtása több időt vesz igénybe, mint az

IF(L_1.EQ.TRUE) GO TO 1
IF(L_2.EQ.TRUE) GO TO 1
.
.
.
IF(L_d.EO.TRUE) GO TO 1
GO TO 2
1 szükség van a rekordra
utasitáscsoporté.

Mig ugyanis az első esetben L értékének eldöntéséhez az összes konjunkciót ki kell értékelni, a második esetben a kiértékelés hamarabb befejeződik.

Az IF(L_i.EQ.TRUE) GO TO 1 tovább bontható.

Legyen evégből: $L_i = L_{i1} \wedge L_{i2} \wedge L_{i3} \ldots \wedge L_{iq_i}$

Az IF(L_{i1}.EQ.FALSE) GO TO C_{i+1}
IF(L_{i2}.EQ.FALSE) GO TO C_{i+1}

(1) IF(L_{iq_i}.EQ.FALSE) GO TO C_{i+1}
GO TO 1

C_{i+1} } Mint fent, csak az L_{i+1}-et bontva

utasitáscsoport végrehajtása hamarabb fejeződik be, mint az IF(L_i.EQ.TRUE) GO TO 1 utasitásé.

A T futási idő ujabb csökkentését érhetjük el a $Z_{i_{k,j}}(\tau_k)$.EQ.0 események relativ gyakoriságának ismeretében.

Ehhez a következő minimum feladatot kell megoldani. Az 1,2,3, . . . , d számok egy permutációja legyen $m_1, m_2, \ldots, m_d$, az $1,2, \ldots, q_i$ számoké pedig $n_1, n_2, \ldots, n_{q_i}$ ($i = 1,2, \ldots, d$). Ha (1)-ben az L_i diszjunkciók sorrendje $L_{m_1}, L_{m_2}, \ldots, L_{m_d}$, továbbá az L_i szétbontásában $L_{in_1}, L_{in_2}, \ldots, L_{in_{q_i}}$ a kiértékelések várható száma meghatározható. A kérdés: milyen sorrend esetén lesz ez a várható érték minimális?

3. A kiértékelési szám minimalizálása

Tegyük fel, hogy az $L = L_1 V L_2 V \ldots V L_d$ felirásnál L_k-ban ($k = 1,2, \ldots, d$) az l_k-adik konjunkció hamis, az előtte lévő konjunkciók igazak, továbbá, hogy L_{k+1} igaz. Ekkor a $\rho(L) = l_1 + l_2 + \ldots + l_k + a_{k+1}$ számot L kiértékelési számának nevezzük, ahol a_{k+1} L_{k+1} konjunkcióinak száma. Ha L_1 igaz, akkor $\rho(L) = a_1$

$\rho(L)$ minimalizálására törekszünk.

Először vizsgáljuk azt a kérdést, hogy az

$$L = \ldots V(\ldots L_1 \wedge L_2 \wedge \ldots) \ldots \quad \text{és} \quad \bar{L} = \ldots V(\ldots \wedge L_2 \wedge L_1) \ldots) V \ldots$$

kifejezések kiértékelési számairól mit tudunk mondani.
Vezessük be a következő jelöléseket:

$$L' = \ldots \wedge L_1 \wedge L_2 \wedge \ldots \qquad L'' = \ldots \wedge L_2 \wedge L_1 \wedge \ldots$$

$p = P(L_1 = i)$ (azaz hogy L_1 igaz) $\quad q = P(L_2 = i)$

$A' = \{$ L'-ben minden L_1 előtti konjukció igaz és minden L' előtti diszjunkció hamis , továbbá $L_1 = i$, $L_2 = h$ $\}$

$A'' = \{$ L''-ben minden L_2 előtti konjukció igaz és minden L'' előtti diszjunkció hamis , továbbá $L_2 = i$, $L_1 = h\}$

$p_1 = P(A')$, $p_2 = P(A'')$

1. Tétel.

$\rho(L) \leqslant \rho(\overline{L})$ akkor és csak akkor, ha $p(1-q)p_1 \leqslant (1-p)qp_2$.

Bizonyitás.

Legyen ξ_1 és η_1 (ξ_2 és η_2) $L(\overline{L})$ kiértékelési száma L' előtt és után (L'' előtt és után) azon feltevés mellett, hogy $L_1 = i$, $L_2 = h$ és L'-ben minden L_1 előtti konjunkció igaz ($L_1 = h$, $L_2 = i$ és L''-ben minden L_2 előtti konjunkció igaz). Jelölje k az L_1 előtti konjunkciók számát (L'-ben, L''-ben), tovább p' annak valószinüségét, hogy L'-ben minden L_1 előtti konjunkció igaz. Nyilván p' annak valószinüségét is megadja, hogy L''-ben minden L_2 előtti konjunkció igaz.

Elemi meggondolásokból következik, hogy $\rho(L) \leqslant \rho(\overline{L})$ akkor és csak akkor, ha

$$E(\xi_1 + (k+2)\chi_A + \eta_1\chi_A)p'p(1-q) +$$
$$+ E(\xi_2 + (k+1)\chi_{A''} + \eta_2\chi_{A''})p'(1-p)q \leqslant$$
$$\leqslant E(\xi_2 + (k+2)\chi_{A''} + \eta_2\chi_{A''})p'(1-p)q.$$

Ez pedig ekvivalens (1)-gyel, ami állitásunkat bizonyitja. Megjegyezzük, hogy ha sem L_1, sem L_2 nem szerepel az L' előtti diszjunkciókban, akkor (1) a $p \leqslant q$ egyenlőtlenséggel ekvivalens.

Azt vizsgáljuk ezek után, hogy az $L = \ldots VL'VL''V \ldots$ és az $\overline{L} = \ldots VL''VL'V \ldots$ kifejezésekre mit mondhatunk $\rho(L)$-ről és $\rho(\overline{L})$-ról.

Jelölje a illetve b L'-ben illetve L''-ben a konjunkciók számát. A következő jelöléseket használjuk:

p_1: annak valószinüsége, hogy L' minden tagja igaz, L'' és az L' előtti tagok mindegyike hamis

η: L''-ben az első hamis tag sorszáma, feltéve, hogy $L' = i$

p_2: annak valószinűsége, hogy L',L'' minden tagja igaz, L' előtti tagok mindegyike hamis

p_3: annak valószinűsége, hogy L'' minden tagja igaz, L' és az L'' előtti tagok mindegyike hamis

ξ: L'-ben az első hamis tag sorszáma, feltéve, hogy $L'' = i$

2.Tétel.

$\rho(L) \leqslant \rho(\overline{L})$ akkor és csak akkor, ha

(2) $$ap_2 + p_3 E\xi \leqslant p_1 E\eta + bp_2$$

Bizonyitás.

Legyen

ξ_1: L esetén L'-ig ($\overline{L}$ esetén L''-ig) a kiértékelt tagok száma azon feltétel mellett, hogy $L' = i$

ξ_2: $L(\overline{L})$ esetén L'-ig (L''-ig) a kiértékelt tagok száma azon feltétel mellett, hogy $L' = L'' = i$

ξ_3: $L(\overline{L})$ esetén L'-ig (L''-ig) a kiértékelt tagok száma azon feltétel mellett, hogy $L'' = i$

Nyilván $\rho(L) \leqslant \rho(\overline{L})$ akkor és csak akkor igaz, ha

$$E(\xi_1 + a)p_1 + E(\xi_2 + a)p_2 + E(\xi_3 + \xi + b)p_3 \leqslant$$
$$\leqslant E(\xi_1 + \eta + a)p_1 + E(\xi_2 + b)p_2 + E(\xi_3 + b)p_3.$$

Ennek átrendezésével adódik (2).

Megjegyzések

a.) A modell konkrét megvalosulásának leirása [1]-ben, [5]-ben és [3]-ban megtalálható. [4]-ben a problémát csak épp, hogy megemlitjük, jelen cikk az ott leirtak továbbfejlesztését adja. [6]-ban példán keresztül világitjuk meg a probléma lényegét.

b.) A minimalizálás programmal történő végrehajtása nem volt célunk, csupán a probléma feltevése és matematikai megoldása. A szükséges program megirása – a felvetődő igényeknek megfelelően – az [1]-ben leirt rendszer hatékonyságát fogja növelni.

c.) A forditóprogramok Boole-kifejezések optimalizálására való törekvése, illetve esetleges megvalósitása a logikai kifejezés 2. pontban leirt felbontását feleslegessé teheti, de a 3. pontban emlitett tételek ekkor is lényegesek.

d.) Eredményeink döntési eljárások (pl. döntési táblázatok) használatánál is alkalmazhatók.

I r o d a l o m

[1] Korházi morbiditási adatok feldolgozását szolgáló statisztikai programrendszer (MTA SzTAKI, témadokumentáció)

[2] Ruzsa Imre – Urbán János: Matematikai logika (Tankönykiadó, 1969)

[3] Ruda Mihály: Egy széles körben alkalmazható programoptimalizálási módszer (MTA SzTAKI, Közlemények 20.)

[4] Krámli András – Ratkó István – Ruda Mihály – Soltész János: A sttisztikai adatfeldolgozás matematikai és számitástechnikai problémái, MTA SzTAKI, Tanulmányok (70/1977)

[5] Ratkó István: Egy számitástechnikai eszköz bonyolult logikai kifejezések leirására orvosstatisztikai alkalmazásokban (Számitástechnikai és kibernetikai módszerek alkalmazása az orvostudományban és a biólogiában. 3. Kollokvium, Szeged, 1977.)

[6] Halassy Béla – Zentai Tamás: Döntési táblázatok (NSZÁMOK, 1973.)

РЕЗЮМЕ

Проблемы оптимализации логических выражений в языке программирования

Иштван Ратко

Пусть L_k состоит из ℓ_k конъюкций в логическом выражении

$$L = L_1 v\ L_2 v \ldots v\ L_d$$

Число

$$\alpha(L) = \ell_1+\ell_2+\ldots+\ell_k+a_{j+1}$$

называется стоимостью L, если ℓ_j-ая конъюкция в L_j ложна, 1-ая,...,(ℓ_j-1)-ая конъюнкция в L_j верна; если L_1 верна, пусть $\rho(L) = a_1$.

Если порядок дисюнкций и конъюнкций в дисюнкции изменяется, тогда $\rho(L)$ тоже изменяется.

В данной работе автор занимается минимализацией $\rho(L)$ и практической задачей выдвинула проблему.

Summary

On optimization problems of logical expressions in programming languages

István Ratkó

In the logical expression

$$L = L_1 \vee L_2 \vee \ldots \vee L_d$$

let L_k consist of the conjuctions I_k $(k = 1,2, \ldots, d)$.
We call the number

$$\rho(L) = I_1 + I_2 + \ldots + I_j + a_{j+1}$$

evaluating number of L iff the I_j-th conjuction of is false, 1-st, . . . , $(I_j$-1)-st conjuctions of L_j are true and the number of the conjuction signs in L_{j+1} is a_{j+1}; if L_1 is true let $\rho(L)$ be a_1.

If we make a change in the order of the disjunctions and that of the conjunctions of its members, $\rho(L)$ changes, too.

In the paper we deal with minimizing $\rho(L)$ and with a partical problem which led to ours.