# Magyar igealakok szintézise
## (Ragok előállítása Boole-vektorok szorzásának módszerével)

**1.** Immár több mint egy éve, hogy a Tudományos és Műszaki Tájékoztató c. folyóiratban lezajlott a vita a gépi fordításról, s ezen belül a magyarországi gépi fordítás lehetőségeiről (TMT 20(1973): 329—358, ill. 840—878). Anélkül, hogy az érvek és ellenérvek közül bármelyiket is kiragadnánk, megállapíthatjuk, hogy a vitapartnerek a szemantikai elemzés lehetetlenségével, ill. hiányosságaival magyarázzák a gépi fordítás eddigi állapotát. Maguk a gépi fordítás hívei és művelői is elsődlegesen az elméleti kérdések felé fordultak, s úgy tűnik, hogy a gyakorlati és konkrét problémák megoldását egy időre félretették. Természetesen az informatika egészére nézve születtek már konkrét eredmények is, mint pl. a tezauruszok, de úgy véljük, hogy a GF számára a jelenlegi helyzetben ezek még inkább elméleti, mint gyakorlati hasznúnak minősülnek.

Pedig a nyelvészet és a gépi fordítás területén sok olyan megoldható probléma van, amely nem függ a szemantikától. Ilyenek: a magyar nyelv független szintézisében a ragozott alakok létrehozása, a szótári keresés műveletének hatékonyságával és általában a nyelvészeti feladatok programozásának optimalizálásával kapcsolatos munkálatok.

Természetesen ilyenek már vannak. Az az úttörő jellegű kísérlet, amely során Hell György 1961-ben először fordított oroszról magyarra,[1] már szükségessé tette bizonyos szintetizálási feladatok megoldását. Ezt követte Papp Ferenc,[2] Igor A. Melcsuk,[3] Klauszer Judit[4] és Stein Mária[5] főnévragozással, Jánoska Sándor[6] igeragozással kapcsolatos algoritmusainak kidolgozása, ill. esetenként gépen való kipróbálása. Ezenkívül Papp Ferenc a TMT-ben[7] említést tesz saját készülő monográfiájáról, amelyben a főnévi ala-

[1] Hell György, Az orosz—magyar gépi fordítás elméleti és gyakorlati kérdései. ÁltNyT. Bp., 1964. 2:177—196.

[2] Papp Ferenc, A magyar főnévragozás három modellje. MNy. 64:194—206.

[3] Melcsuk, Igor A. (Мельчук, И. А.) Egy új főnévragozási modell. MNy. 64:176—192.

[4] Klauszer Judit, A magyar főnevek szintézisének kérdéséhez. ÁltNyT., Bp., 1965. 3:117—129.

[5] Stein Mária, Synthese des ungarischen Hauptwortes mit elektronischen Rechenmaschine. Computational Linguistics. 1966. 5:169—176. (Megegyezik a Papp F. 2. jegyzetében levő cikk függelékében közölt programmal.)

[6] Jánoska Sándor, A magyar ige automatikus toldalékolásának egy modellje. Nyelvtudományi Értekezések. A magyar nyelv története és rendszere. Bp., 1967. 58: 464—468.

[7] Papp Ferenc, Gépi fordítás és számítógépes nyelvészet. Tudományos és Műszaki Tájékoztató, 1973. 20:840—844.

kok szintézisének kérdéseivel továbbra is foglalkozik; valamint egyik munkatársának kéziratban levő cikkéről, amely az igealakok szintéziséről szól. Kéziratban volt módom megismerkedni LUGOSINÉ PAPP MÁRTA, az ÉGSZI munkatársa cikkével, amely szintén az igeragozás szintézisével foglalkozik. Ez utóbbi kipróbálásra is került a Tervhivatal ICL System 4—70 számítógépén, és egyben fordulópontot is jelent: ez az első olyan nyelvészeti program hazánkban, amelyet byte-szervezésű gépen futtattak.

A gépi szótárazás és elemzés (forrásnyelvi analízis) kérdésével KIEFER FERENC[8] és VARGA DÉNES[9] foglalkoztak, mindketten DÖMÖLKI BÁLINT módszerét[10] alkalmazva a gépi fordításban. VARGA DÉNES módszerét SZELEZSÁN IRÉN is felhasználta[11] a gyakorlatban.

A cikk megírásakor még nem jelent meg az 1973 óta hirdetett Papers in Computational Linguistics című könyv (Az 1971-ben Debrecenben megtartott nemzetközi számítógépes nyelvészeti kongresszus előadásai, PAPP FERENC és SZÉPE GYÖRGY szerkesztésében). Ezért az ebben lévő tanulmányok ismeretének hiányában további témába vágó munkáról nem sikerült tudomást szerezni.

**2.** Programom VARGA DÉNESnek a szukcesszív behatárolás elnevezésű módszerét használja fel, de nem elemzés, hanem szintézis céljaira. Az igeragozás nyelvészeti szempontjaihoz nem kívánok hozzászólni, de a változó tővégű rendhagyó igék tipizálásában talán sikerült némi egyszerűsítést elérni. A módszer ismertetésén kívül egy, a nyelvészeti szakirodalomban ritkán tárgyalt témával is foglalkozunk, mégpedig a nyelvi feladatok megoldásának néhány programozástechnikai kérdésével.

A számítástechnika és a gépek a magyarországi gépi fordítás kezdete óta (1961) sokat fejlődtek, kialakult bizonyos számítógépes kultúra is. (Ma már szinte hihetetlennek tűnik, hogy HELL GYÖRGY a MESZ-1-es jelfogós számítógépen a kinyomtatott szöveget még 10-es számrendszerbeli számokban kapta meg, amelyet át kellett alakítani betűkké.[12])

Hazánkban jelenleg kétféle típusú számítógépek találhatók: egyik fajtájuk az ún. szószervezésű típus, amelyben a legkisebb címezhető egység a szó, általában 24 bites, és ebben 4, egyenként 6-bites karakter (megjeleníthető betű, szám vagy jel) helyezhető el. Bár egyes típusok rendelkeznek ún. karaktermódú műveletekkel, ezek nem általánosak. A másik típus a byte-szervezésű gépek családja, amelyben a legkisebb címezhető egység a byte, amely 8 bit hosszúságú, és ebben egy karakter helyezhető el. Ez nyelvészeti szempontból is praktikusabb.

Már régóta használatosak a különféle magasszintű (feladatorientált) programnyelvek, mint a COBOL, FORTRAN, ALGOL és újabban a PL/1.

[8] KIEFER FERENC, Gépi fordítási kísérletek az M-3 számítógépen. IMDK = Időszerű műszaki dokumentációs kérdések 3—4. Gépi fordítás. Algoritmusok orosz nyelvű szövegek elemzésére. Bp., 1963. OMKDK. 69—142.

[9] VARGA DÉNES, Morfológiai elemzés a szukcesszív behatárolás módszerével. IMDK 244—271.

[10] DÖMÖLKI BÁLINT, Jelsorozatok tulajdonságainak felismerésére szolgáló algoritmusok. Az MTA Számítástechnikai Központjának tájékoztatója, 8:63—68.

[11] SZELEZSÁN IRÉN, Orosz főnevek és melléknevek analízisének programozási vonatkozásai. A tudományos tájékoztatás elmélete és gyakorlata. 11. sz. Nyelvfeldolgozás és dokumentáció. Bp., 1967. OMKDK, 164—183.

[12] Vö. HELL GYÖRGY i. m. 181.

Egyetlen hátrányuk, hogy egyes betűkkel manipuláló műveleteik korlátozottak, és a karakternél kisebb adatokba beavatkozni nem lehet. Léteznek ún. szimbólumkezelő nyelvek is, ezeket nálunk azonban még nem használják. Így nyelvészeti feladatok megoldásához célszerű a géporientált assembly-szintű programozást felhasználni (az ebben a cikkben leírt program a SIEMENS System 4004/45 byte-szervezésű gépre, ASSEMBLER nyelven íródott).

Mivel a számítógépek kettes számrendszerű számaival dolgozni nehézkes, ezért 3 bitet összefogva nyolcas (oktális), 4 bitet összevéve tizenhatos (hexadecimális) számrendszerű számokat szokás írni. A cikk végén található táblázatok minden adat mindhárom alakját tartalmazzák.

A nyelvészeten belül bizonyos eljárásoknak különféle utasítások felelhetnek meg, a makroprogram (algoritmus) és a mikroprogram (gépi utasítások) nem feltétlenül egyeznek meg. A legáltalánosabb eljárás valamely adat felismerése, azonosítása.

Ez történhet az összes lehetséges eset „lekérdezésével". Pl. X betű magánhangzó ? Sorra összehasonlítjuk a kérdéses betűt (továbblépés „igen" esetén a zárójelben levő első, „nem" esetén a második számú utasításra):

(1) X = A ? (6,2)
(2) X = E ? (6,3)
(3) X = I ? (6,4)
(4) X = O ? (6,5)
(5) X = U ? (6,7)
(6) Az az utasítás, amelyet magánhangzó esetén kell végrehajtani;
(7) Az az utasítás, amelyet mássalhangzó esetén kell végrehajtani, vagy folytatás.

A másik lehetséges megoldás az ún. elágazási táblázat létrehozása (a FORTRAN-ban „kiszámított GO TO"-nak nevezik). Ebben az esetben az adat saját maga vezérli a programot. Az előbbi példa ekkor így módosul: A számítógépben az A betű valamilyen jelkombináció, egyben egy számnak is megfelel, pl. 21-nek. A kérdéses betű értékéből kivonunk 20-at, így az A betűből 1 lesz, a B betűből (22) 2, s.i.t.

(1) Vonj ki a betű értékéből 20-at!
(2) Ennek az utasításnak a címéhez [(2)] add hozzá a kivonás eredményét, és ugorj arra a címre!
(3) A-nak megfelelő cím. Ugorj (m) címre!
(4) B-nek megfelelő cím. Ugorj (n) címre!
(5) C-nek megfelelő cím. Ugorj (n) címre!
(6) D-nek megfelelő cím. Ugorj (n) címre!
(7) E-nek megfelelő cím. Ugorj (m) címre!
...
(m) Utasítás magánhangzó esetén;
(n) Utasítás mássalhangzó esetén, vagy folytatás.

A kétféle eljárás közül természetszerűleg az a gazdaságosabb, amelyik az adott feladat típusához jobban illik és rövidebb.

Egy szabály alkalmazását (pl. adott tőhöz milyen rag járulhat) úgy állapítják meg, hogy megvizsgálják, milyen fonémától (betűtől) függ. Amely-

nyiben ténylegesen a betűt vizsgáljuk, erre általában az előbbi két megoldás kínálkozik. Ellenben általános módszer, hogy a szótárban nyelvtani információt (típuskódot) helyezünk el, és ezeknek a lekérdezése mindig egyszerűbb, mint magának a betűnek a vizsgálata, nem is szólva arról, hogy a szó betűi közül a kérdéses meghatározó betűt ki is kell keresni.

Más kérdés az is, hogy a kivételek és szabályok aránya miként valósítható meg célszerűen. Nyilvánvaló, hogy azok a szabályok, amelyek érvényességi köre kicsi, tulajdonképpen kivételként működnek. Helyesírási szempontból nem léteznek olyan alakok, mint [*HALLLAK*], [*IZZZ*], bár egy generálási szabály ezeket hozná létre, és külön törlési szabályt kellene kidolgozni ezekre az esetekre. Kettősbetűt tartalmazó olyan szó, amely kedvéért külön programutasításokat kellene beiktatni, olyan kevés van,[13] hogy célszerűbb kivételnek tekinteni, hisz kivételkereső programrészlet minden programban van. Ugyanez érvényes a meghatározott témakörben mozgó fordítás szókincsére is. Ha egy — egyébként teljesen általános és ezen a szókincsen kívül eső szavakra gyakori érvényes — szabály a szótári anyagra nézve ritka, akkor ezt a szabályt törölni lehet, és az ilyen sajátosságú szavakat rendhagyóként lehet kezelni. Ilyen pl. a *-DZ* tőveghangú igék családja, amelynek *-DDZ* felszólító módú tővariánsait — az előreláthatólag tudományos célokat szolgáló gépi fordítási szótárakban egytől egyig kivételnek tekinthetők.[14]

Mielőtt bemutatnánk a logikai szorzás módszerét, érdemes megnézni azt is, hogy milyen más módszerekkel lehet pl. igeragokat létrehozni.

Erre is két fő lehetőség adódik. Példaként a kijelentő mód alanyi ragozás egyessz. 1. személy ragjait vegyük. A rag alakilag két tényezőtől függ, a hangrendtől és az ikességtől. Ha a lehetséges eseteket táblázatba (más neveken mátrixba, 2-dimenziós tömbbe) foglaljuk, akkor ilyen képet kapunk:

| | 1 | 2 | 3 |
|---|---|---|---|
| Hangrend: | mély | | magas |
| | | „e-soros" | „ö-soros" |
| 1 iktelen | *OK* | *EK* | *ÖK* |
| 2 ikes | *OM* | *EM* | *ÖM* |

Ha az ige a szótár nyelvtani információja alapján ikes és mély hangrendű, akkor a mátrix $a_{2,1}$ elemét kell a tőhöz illesztenünk. Hátránya, hogy rendkívül nagy tömböket kell lefoglalni, és minden rag ill. ragváltozat t e l j e s alakját tárolni kell.

A másik eljárásban a rag kiinduló alakja *VC*, ahol *V* *o-e-ö* „értéket vehet fel", *C* pedig *k-m* értékeket. A programban külön-külön le kell kérdezni a *V* megállapításához a hangrend, a *C*-hez az ikesség típusát.

A programok összeállítása szempontjából általában is kétféle típust lehet megkülönböztetni. Van olyan program, amelyben az adatok szoros szabály szerint vannak tömbökbe rendezve, ebben az esetben az adat kezelése rend-

[13] A Vég Sz.-ban 100 *-LL* végű ige van (224—225), ezek közül a *-lak/-lek* ragos alak szempontjából a *hall* igén kívül legfeljebb a *szégyell*, *restell* és *vall* jöhet számításba. *Izzik* és *átizzik* igén kívül a VégSz.-ban más nincs is (197).

[14] *Edz*, *pedz*, *megedz* (VégSz. 523), valamint 67 *-DZIK* végű ige (190). Ezek közül egy gépi fordítási szótárba, ha a fele is bekerül, legfeljebb 15—20 igének van szükség a felszólító módú alakjára.

kívül egyszerű programot kíván, de nagy, rossz kihasználtságú adattömbökre van szükség. A másik fajta esetén az adatok rendezettségi foka alacsonyabb, így kisebb helyet is foglalnak el, de a program bonyolultabb. Egy szemléletes példa: Ha a szótárban a szavak abc-rendben szorosan egymás után következnek, akkor egy 1000-szavas szótár kis helyet foglal el, de a szó megtalálása viszonylag körülményes. De ha a szavakat az első két betűjük szerint külön lapra írjuk, *AA—AB—AC—...—ZY—ZZ* sorrendben, akkor a keresés egyszerűvé válik, ugyanakkor a terjedelem megnövekszik, és egyes lapokon nagyon kevés, másokon esetleg egy szó sem lesz. A vektorszorzás elve nagyjából a két véglet között félúton foglal helyet.

Varga Dénes módszere, amely a nyelvi jelsorozatok felismerésének újszerű módja, a Boole-vektorok konjunkcióján, azaz halmazok metszetének megállapításán alapszik. Vegyünk egy tetszőleges nagyságú halmazt, amelynek minden részhalmaza egy-egy (nyelvi) tulajdonságot reprezentál. Minden nyelvi elemhez rendeljünk hozzá egy Boole-vektort, azaz 0-kból és 1-esekből álló számsorozatot, amelyek rendre azt jelzik, hogy a nyelvi elem melyik tulajdonságú halmazokba tartozik (ott a vektorban 1-es áll). Amennyiben több elem egyszerre jelentkezik, ezek együttes előfordulása csökkenti a lehetőségek körét, és csak azok a tulajdonságok maradnak érvényben, amelyek mindegyik vektorban közösek. A számítógépben a közös rész (konjunkció, metszet) megállapítása a logikai *ÉS* műveletével történik; az *ÉS*-művelet eredménye akkor és csak akkor 1, ha minden tagja is 1.

Két példa: 1. Ha *A*, *B*, *C* és *D* tulajdonságok, *x*, *y*, *z* pedig elemek. Ha *x* *A* és *C* tulajdonságú, akkor Boole-vektora 1010, s tegyük fel, hogy *y*-é 1101, *z*-é 0111. Ha *x* és *y* együttesen fordul elő, akkor a két vektor konjunkciója

| | |
|---|---|
| *x* | 1010 |
| *y* | 1101 |
| *x ÉS y*: | 1000, |

vagyis feltételezhetjük, hogy *x* és *y* együttes előfordulásakor csak az *A* tulajdonság van érvényben.

2. Konkrétan, igeelemzésre felhasználva, *A—B—C* részhalmazok a kijelentő mód alanyi ragozás egyes szám három személyének feleljenek meg. A ragokban (*-ok*, *-ek*, *-ök*, *-om*, *-em*, *-öm*; *-sz*, *-asz*, *-esz*, *-ol*, *-el*, *-öl*; Ø,*-ik*) előforduló betűk abc-sorrendben: *A*, *E*, *I*, *K*, *L*, *M*, *O*, *Ö*, *SZ*, valamint a Ø. Vektoraik:

*A*: 010, *E*: 110, *I*: 001, *K*: 101, *L*: 010,
*M*: 100, *O*: 110, *Ö*: 110, *SZ*: 010, Ø: 001.

Tegyük fel, hogy az elemzendő szó tövét megtaláltuk, a teljes szóalakból pedig leválasztottuk a végződés betűit. Ezek vektorait összeszorozva,

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *—ASZ*: | *A*: | 010 | *—OL*: | *O*: | 110 | *—IK*: | *I*: | 001 |
| | *SZ*: | 010 | | *L*: | 010 | | *K*: | 101 |
| *A ÉS SZ*: | | 010, | *O ÉS L*: | | 010, | *I ÉS K*: | | 001, |

megkapjuk, hogy a ragok e. sz. 2., 2., ill. 3. személyű alakot jelentenek.

Az analízis módszerét megfordítva állíthatjuk: Ha tudjuk, milyen jeleknek kell előfordulni egy adott tulajdonság teljesülése esetén, akkor a feltételek teljesülése azokat és csak azokat a jeleket választja ki, amelyekre szükségünk van.

Példánknál maradva, az e. sz. 1. személy ragjai az *O—E—Ö—K—M* betűkből kerülnek ki. Ötelemű vektorokkal kifejezhetjük, hogy a megfelelő hangrendi és ikességi típusban melyik betűk fordulhatnak elő:

| | |
|---|---|
| Mély hangrendű igék: | 10011 |
| Magas ajakkerekítés nélküli („*e*-soros"): | 01011 |
| Magas ajakkerekítéses („*ö*-soros"): | 00111 |
| Iktelen igék: | 11110 |
| Ikesek: | 11101 |
| Betűsor: | *OEÖKM* |

A szótári információ szerint az ige mély hangrendű ikes:

| | |
|---|---|
| hangrendi vektor: | 10011 |
| ikesség vektor: | 11101 |
| konjunkció: | 10001, |

amely alapján megállapíthatjuk, hogy az *O* és az *M* helyén áll 1-es, vagyis a rag: *-om*.

| | |
|---|---|
| Vagy: magas „*ö*-soros" hangrend: | 00111 |
| iktelen: | 11110 |
| eredményvektor: | 00110, |

vagyis a rag *Ö—K*, *-ök*.

**3.** Ezek után következhet a program részletes ismertetése. A szintetizáló programrészlet lényegében csak egy nagyobb program egyik szubrutinja, ezért részére a kiinduló adatokat („aktuális paramétereket") egy másik programrészlet szolgáltatja. Ezeket a kiinduló adatokat a program önálló futása esetén lyukkártyából kellett beolvasni. A lyukkártyán egy ige szótári sorszáma és négy nyelvtani adata szerepel (SORSZÁM, VP, LI, J, K néven) a következő jelentésekben (a továbbiakban a feltüntetett rövidítéseket használva):

1—2. számjegy (VP és LI):

0 1 — főnévi igenév, FNIN
0 2 — folyamatos,
0 3 — befejezett és
0 4 — előidejű melléknévi igenév, FMIN—BMIN—EMIN
0 5 — folyamatos és
0 6 — befejezett határozói igenév, FHIN—BHIN
1 1 — kijelentő mód jelen idő, KJE
1 2 — kijelentő mód múlt idő, KM
1 3 — feltételes mód jelen idő, FTJ
1 4 — felszólító mód, FSZ
1 5 — kijelentő mód jövő idő, KJÖ
1 6 — feltételes mód múlt idő, FTM

A 3—4. számjegy (J és K) értéke 0 0 minden igenév esetén, a 3. számjegy (J) személyragos alakoknál 1 — alanyi, 2 — tárgyas ragozást jelent (AR-TR), 3 — a főnévi igenév ragozott alakjait, FNINR. A 4. számjegy (K) 1 és 6 között e. sz. 1., 2., 3. t. sz. 1., 2. és 3. személyt jelent, 7 — e. sz. 1. sz., 2. személyre mutató raggal (*-lak*, *-lek*, IMPL).

A kisméretű tőszótár szóegységeinek felépítése általában: nyelvtani információ, szabályos tőalak, rendhagyó tőváltozatok száma, a rendhagyó tő nyelvtani információi és maga a rendhagyó tőalak. A szabályos tőalak nyelvtani információi 3 byte-ban, 24 biten helyezkednek el: 1—8. HOSSZKÓD, 9—10. HR, 11. IK, 12—13. J2, 14. J3, 15—16. THG, 17—19. FSZT, 20—21. SZL, 22. KH és 23—24. TVLT. Ezek értékei az alábbiak lehetnek:

HR (hangrend): 00 — mély hangrendű,
01 — magas „*e*-soros",
11 — magas „*ö*-soros".

IK (ikesség): 0 — iktelen,
1 — ikes.

J2 (múlt idő): 00 — minden alak *-t*—ragos,
11 — minden alak *-t*—ragos, kivéve az AR e. sz. 3. személyt, amely *-Vtt* alakú,
01 — minden alak *-Vtt*—ragos.

J3 (FNIN—FTJ): 0 — a főnévi igenév és a feltételes mód képzése *-n-*, kötőhang nélkül,
1 — ua. kötőhanggal.

SZL: 00 — a KJE AR e. sz. 2. sz. ragja *-SZ*,
01 — a rag *-ol*, *-el*, *-öl*,
11 — a rag *-asz*, *-esz*.

KH (kötőhang): 0 — KJE AR t. sz. 2. és 3. sz. kötőhang nélküli,
1 — ua. kötőhangos.

TVLT (tőváltozat):

00 — csak szabályos tő van,
01 — egy rendhagyó tő is van,
11 — több rendhagyó tő van.

Mielőtt a többi típusjel ismertetésére rátérnénk, bemutatjuk a rendhagyó tő szótári felépítését. Ha több rendhagyó tő van, akkor az adatokat egy szám vezeti be, amely jelzi, hány tőváltozat van. Egy rendhagyó tő esetén ilyen bevezető szám nincs. Egy rendhagyó tő alakja: hosszkód, esetvektorok száma, esetvektorok, rendhagyó tőalak betűi. Az esetvektor mutatja, hogy a rendhagyó tő milyen alak képzéséhez használandó fel. 8 bitjéből az 1. jelzi, hogy önálló (0), vagy továbbragozandó alak (1) a rendhagyó tő. A 2—7. bit pedig a bemenő adat rövidített alakja. A 2—3—4. bit az LI értékét veszi fel, az 5—8. bit értéke 0000 az igeneveknél, az 5. bit 0, ha AR, 1, ha TR alakra vonatkozik, a 6—7—8. bit a K értékét veszi fel. Mivel a főnévi igenév ragozott alakjait sohasem képezzük rendhagyó tőből, ezért ennek a jelölésére nem volt szükség. (Példa: KJE AR e. sz. 1. sz. — 1 1 1 1 bemenő adat — esetvektora önálló alaknál 0—001—0—001, FSZ TR t. sz. 2. sz. — 1 4 2 5 bemenő adat — esetvektora továbbragozandó alaknál 1—100—1—101.)

Az FSZT típuskódjai jelzik, hogy a felszólító mód és a KJE TR e. sz. 3. sz., t. sz. 1—3. sz. szempontjából milyen variánsokat különböztetünk meg:

| | | | | |
|---|---|---|---|---|
| 0 | (000) | *-j-* | (*verjük*) | *-j-* | (*verjük*) |
| 1 | (001) | *-s-* | (*mossuk*) | *-s-* | (*mossuk*) |
| 2 | (010) | *-sz-* | (*játsszuk*) | *-sz-* | (*játsszuk*) |
| 3 | (011) | *-z-* | (*hozzuk*) | *-z-* | (*hozzuk*) |
| 4 | (100) | *-sztj-* | (*ébresztjük*) | *-ssz-* | (*ébresszük*) |
| 5 | (101) | *-tj-* | (*vetjük*) | *-ss-* | (*vessük*) |
| 6 | (110) | *-tj-* | (*gyüjtjük*) | *-ts-* | (*gyűjtsük*) |
| 7 | (111) | *-tssz-* | (*metsszük*) | *-(t)ss(z)* | (*metsszük* vagy *messük*) |

A TGH típuskód lehetséges értékeinek ismertetéséhez először közölnünk kell, hogy az ige tőbeli változó hangjait milyen típusokba soroltuk:

Tőbeli változó magánhangzók:

W és X lehetséges értékei: Ø, *O*, *E*, *Ö*;
# és | lehetséges értékei: Ø, *U*, *E*, *Ü*.

W és # minden esetben kiesik (Ø), ha a végződés magánhangzóval kezdődik. Ellenkező esetben a három hangrendi típusnak megfelelő értéket veheti fel. X és | ugyanez korlátozott változatban, hangkiesés csak jelen időben történhet, valamint ha a folyamatos melléknévi igenevet *V*-vel képezzük.

Tőbeli változó mássalhangzók:

*D* és *Z* az alábbi értékeket veheti fel:

*SSZ*: KJE TR t. sz. 1. sz., ill. mély hangrendűeknél e. sz. 3. sz. és t. sz. 2—3. személyben is;
*SZ*: a kijelentő mód jelen idő egyéb alakjaiban;
*V*: ha a FMIN-et *V*-vel képezzük.

Q változása *V*, *SZ*, *SSZ* betűkké azonos az előbbiekkel, egyébként
**Ø**: EMIN képzésekor (*VEQ—ENDOE — VEENDOE*):
***N***: FNIN és FTJ képzésekor (*VEN-*):
*T*: BMIN és KM képzésekor (*VET-*);
*GY*: FSZ képzésekor (*VEGY-*);
az őt megelőző mgh: határozói igenevekben (*VE-E-VE*).

A * a *ró*, *sző*, *vi* típusú igék véghangzója. Átalakítási szabály: * és *T*: *TT* * és egyéb mássalhangzó: * → Ø, * és magánhangzó: * → *V* (*RO*—T*: *ROTT*, *RO*—NA*: *RONA*, *RO*—OM*: *ROVOM*).

Ennek alapján THG („tőhang") típuskódjai: 01 — ha a *D* és *Z* végű igék a FMIN-et *V* nélkül képzik; 11 — ha *V*-vel. Minden más esetben a kód 00.

Azok az alakok, amelyek ezek a szabályok szerint nem képezhetők, vagy elfogadható, de kevésbé használatos alakok, rendhagyó tőként szerepelnek a szótárban (pl. *FEK # D* nem *FEKUEDVE*, hanem *FEKVE*, ugyanakkor a szabály szerint *FEKUEDVEEN*).

A program próbafuttatásához használt szótár összes igéit az 1. táblázat sorolja fel.

A program végrehajtása: Miután a bemenő adatokat a gép elhelyezte a SORSZÁM, VP, LI, J, K változókban, először a megfelelő ige kikeresése következik. A SZÓTÁR nevű tömb kezdőcíméhez a sorszámot hozzáadva, megkapjuk a kérdéses ige adatainak kezdetét. A nyelvtani információt tartalmazó két byte-ot a GRINF nevű munkarekeszbe visszük át, a hosszkód alapján pedig a VERB nevű rekeszbe annyi betűt, ahány betűs a szabályos tő. Amennyiben nem a főnévi igenév valamelyik ragozott alakját kell előállítani (1 1 3 K), a TVLT típuskód alapján lekérdezzük, hogy van-e rendhagyó tő. Ha van, akkor a bemenő adatokból elő kell állítani az 1 byte-os „esetvektor" alakot, hogy a rendhagyó tő esetvektoraival össze tudjuk hasonlítani. Ha valamelyik rendhagyó tő esetvektora és a bemenő adat rövidített alakja egybeesik, akkor a szabályos tő helyett a VERB rekeszbe a rendhagyó tő kerül. Ha az esetvektor 1. bitje 0, akkor a kérdéses alak előállítása befejeződött, ha 1-es, akkor folytatódik a program.

Ha a bemenő adat (1 5 J K) vagy (1 6 J K) alakú, tehát KJÖ vagy FTM alakot kell képezni, az eljárás a következő:

KJÖ: a tőhöz *-NI-*, *-ANI* vagy *-ENI* végződés kerül (J3 típuskód lekérdezésével), valamint a *FOG* szótő. A nyelvtani információ adatait kicseréljük a *FOG* adataival (csupa 0), a bemenő adat LI adatát 1-esre állítjuk (KJE-nek felel meg). FTM: a bemenő adat LI adatát 2-esre állítjuk (KM), és egy VAUX nevű jelzőbe 1-es értéket viszünk be.

Amennyiben a THG típusjelző *D*, *Z* vagy *Q* véghangzót jelez, a megfelelő esetekre való rákérdezéssel végre kell hajtani a módosítást. Mivel nem tudjuk, hogy a végződés milyen betűvel fog kezdődni, ezért a tőbeli magánhangzó és a * változtatására csak a végén kerülhet sor. Amennyiben módosítás történt, egy JELZŐ nevű változóba beviszünk 1-et, mert később szükség lesz rá.

Tőbeli módosítást kell végrehajtani a KJE és FSZ megfelelő eseteiben is:

Ha az FSZT típuskód 2 (010) vagy 7 (111) (*-sz* végű igék), akkor TR t. sz. 1. sz., ill. mély hangrendű igék esetén esz. 3., t. sz. 2—3. személyben a tővégi *Z*-t el kell hagyni (KJE). Ha FSZT 0 (000), 1 (001), 3 (011) és 6 (110) típusú, akkor FSZ képzésénél módosítás nincs. Egyébként:

2 (010) tővégi *Z* levágása,
4 (100) tővégi *ZT* levágása,
5 (101) tővégi *T* helyett *S* beírása, és
7 (111) tővégi *TSZ* helyett (*T*)*S* beírása szükséges.

Ezután következhet a megfelelő alak alapvektorának kiválasztása. Anélkül, hogy ezt részletesen tárgyalnánk, a 2. táblázat „választóvektor" elnevezésű részét kell megmagyaráznunk. Mivel az egyes alakok különböző típusjelektől függnek, ezért azt is fel kellett tüntetni, hogy melyiktől. Az igenevek, a KJE és FSZ képzésében ez többféle is lehet, ezt a program egy 8-menetes ciklusban kérdezi le. A KM és FTJ képzésekor mindig ugyanazokat a vektorokat használjuk, ezért ott a választóvektorokat nem kellett feltüntetni.

Az alapvektor „betűválasztó vektor" elnevezésű részét egymás után összeszorozzuk a kiválasztott típusvektorokkal. Az eredmény szintén egy Boole-vektor, amely a megfelelő betűsor lekérdezését vezérli. Ahányadik helyen az eredményvektorban 1-es áll, ugyanannyiadik betűt kell sorban

egymás után betenni a SUFFIX nevű rekeszbe. Mivel a betűsorok nem egyforma hosszúak, ezért a betűsor lekérdezéséhez különböző menetszámú ciklusok is elegendők (14-29-22-20, ill. 29 menet). Abban az esetben, ha *D, Z, Q* típusú tőnél tőmódosítás történt (erről szerzünk tudomást a JELZŐ nevű változó tartalma alapján), akkor a lekérdezést KJE és FSZ esetén nem az 1., hanem csak a 9. bittől kell elkezdeni.

A könyvtári tájékoztatásban használt fénylyukkártyák (más néven: vizuális vagy átnézeti lyukkártyák) szintén a konjunkciós módszeren alapszanak. Ezért ennek az eljárásnak a modellezése is elvégezhető számítógép nélkül. A vektorokat kis papírcsíkok helyettesítik, amelyek az 1-esek helyén ki vannak vágva. A csíkokat a betűsorra helyezve azok és csak azok a betűk maradnak láthatók, amelyek a kívánt ragot adják. A gépi eljárást és a „barkácsmódszert" egyaránt az 1. ábra szemlélteti.

Néhány megjegyzés a betűsorokhoz:

KJE (29 betű): *AEJSZ JSZNI IULAE NAOOE TOOEK KMLD* 3-4-5. és 6-7-8. pozíció egyaránt J-S-Z, a TR ragjai (*J, S, SZ, Z*), de az egyik csoport „hangrendtől független", a másik „hangrendtől függő". 10-11. pozíció I, az egyik az -IK rag betűje, a másik a FNIN ragozott alakjaiban a *-NIA, -NIE, -NIUK, -NIUEK* ragé. 18—19., és 22—23. pozíció O betűi közül az első valódi O (mély hangrendű), a második az *Ö* (*OE*) *O* betűje( azaz „magas hangrendű"). 25—26. pozíció *K*, az egyik az ikességtől független *K*, a másik az ikes ill. a tsz. alakok *K* betűje.

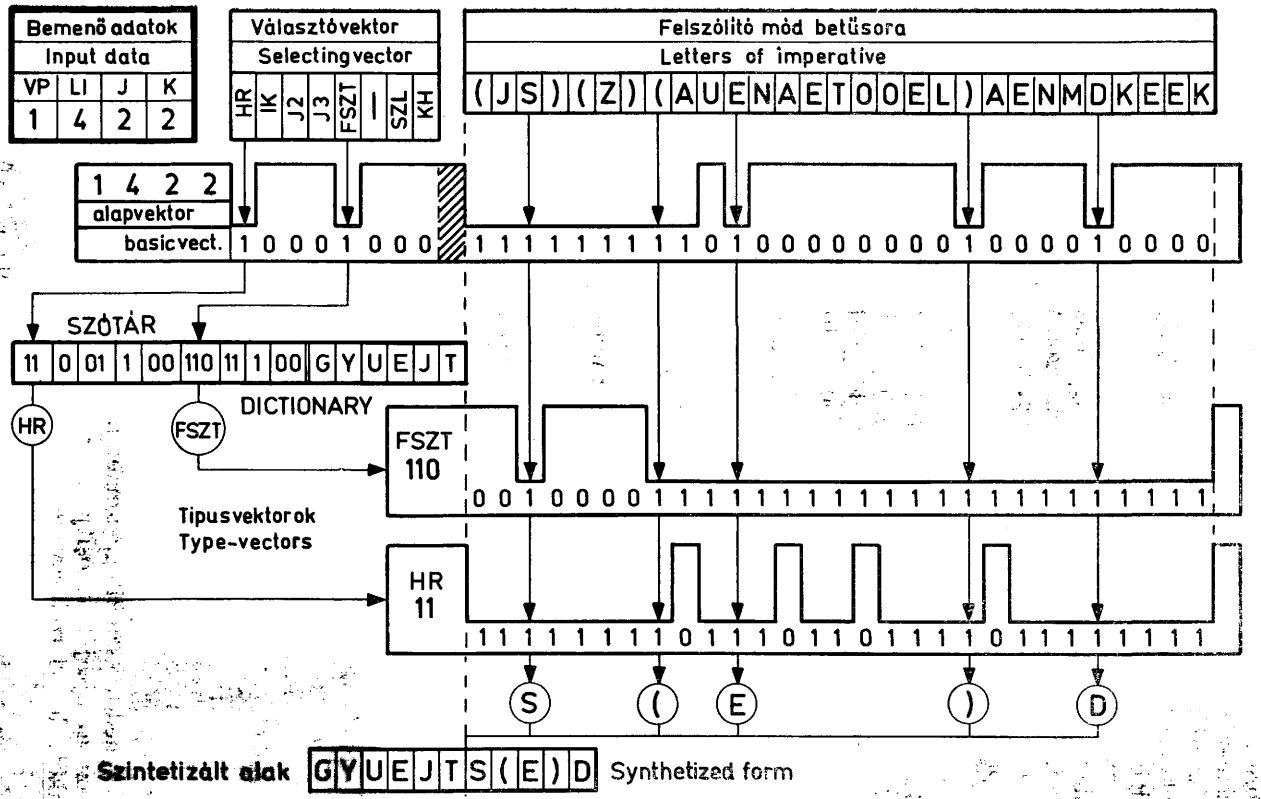
FSZ (29 betű): (*JS*) ( *Z*) (*AU ENAET OOEL*) *AENMD KEEK*. A zárójelek a 2. személyű alakok rövid és hosszabb formáját különböztetik meg, a következőképpen: (Egyes alakoknál felesleges többletzárójellel, de ennek kiküszöbölése bonyolította volna a betűsort és a vektorokat.)

| | |
|---|---|
| 0 (000) *VERJ(EEL)* | *VER(J(E)D* |
| 0 (001) *MOSS(AAL).* | *MOS(S(A)D* |
| 2 (010) *JAATSSZ(AAL)* | *JAATS(S)Z(A)D* |
| 3 (011) *HOZZ(AAL)* | *HOZ(Z(A)D* |
| 4 (100) *VAALASSZ(AAL)* | *VAALAS(S)Z(A)D* |
| 5 (101) *VESS(EEL)* | *VES(S(E)D* |
| 6 (110) *GYUEJTS(EEL)* | *GYUEJTS(E)D* |
| 7 (111) *ME(T)SS(Z)(EEL)* | *ME(T)S(S)(Z)(E)D* |

A 27—28—29. pozícióban *EEK* az ikes igék AR esz. 3. személyének ragja.

Feltételes mód (20 betű): *AENAA EEEET NAOEL AEKMD* 6—7. 8—9. pozícióban *E* betűk, a 6—7. hangrendtől függő *e-é* betű, a 8—9. az AR esz. 1. személy hangrendtől független *-ék* ragjának eleme. A FTJ ragjainál az ikes igék ragozását nem vettük figyelembe.

Ezután hátra van, hogy a SUFFIX első betűjét megnézzük: magánhangzóval vagy *T*-vel kezdődik-e. Amennyiben valamelyik feltétel teljesül, a MGHFLAG, ill. TFLAG nevű változók egyikébe 1-et kell bevinni. A tőben (VERB rekeszben) meg kell vizsgálni, hogy van-e benne W, X, |, #, vagy *. Ha igen, a szükséges módosítást végre kell hajtani: a végződés elejére a jelző-(FLAG-)változóra, a hangrendre a HR kódra történő rákérdezéssel kapunk választ. A kijelentő mód jövő idejének képzésekor a korábban bemutatott elv szerint járunk el, a KM és FTM alakokat pedig úgy különböztetjük meg, hogy rákérdezünk a VAUX flag értékére. Ha az 1, akkor a FTM-t a *VOLNA* szó hozzátevésével kapjuk meg.


Bemenő adatok
Input data
VP LI J K
1 4 2 2
Választóvektor
Selecting vector
HR IK J2 J3 FSZT — SZL KH
Felszólító mód betűsora
Letters of imperative
(J S) (Z) (A U E N A E T O O E L) A E N M D K E E K
1 4 2 2
alapvektor
basic vect. 1 0 0 0 1 0 0 0
1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0
SZÓTÁR
11 0 01 1 00 110 11 1 00 G Y U E J T
DICTIONARY
HR
FSZT
FSZT
110
0 0 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Típusvektorok
Type-vectors
HR
11
1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1
S ( E ) D
Szintetizált alak
G Y U E J T S ( E ) D
Synthetized form

*1. ábra*

Végül a tő és a végződés egymás mellé tett betűit az eredménymező SZINTETIZÁLT IGEALAK nevű oszlopába helyezve, kiiratjuk a kész sort (2. ábra).

Az ismertetett program bizonyos feladatokat nem tud ellátni, ezt más programrészletekre bízza. Ilyenek: szórendi kérdések (különösen tagadó alak és igekötős ige ragozásánál), az *IS* szó elhelyezése, stb., sőt azt sem, hogy az adott ige hiányos ragozású-e. Ezt véleményünk szerint az idegen nyelvű szótár magyar szóra vonatkozó adataihoz kell elhelyezni. Ugyanott lehetne feltüntetni azt is, hogy az ige igekötős-e, vagy jellegzetes tővégződésű, és amennyiben igen, akkor a program ezeket az igéket „ál-összetettként" kezelné. A jellegzetes végződéseket (főleg a képzőket) önálló igetövekként lehetne tekinteni, pl.:

| | HR | IK | J2 | J3 | THG | FSZT | SZL | KH | TVLT |
|---|---|---|---|---|---|---|---|---|---|
| 1. | — | 0 | 01 | 0 | 00 | 101 | 00 | 0 | 00 |
| 2. | — | 0 | 00 | 0 | 00 | 000 | 00 | 0 | 00 |
| 3. | — | — | 11 | 0 | 00 | 011 | 01 | 0 | 00 |
| 4. | — | 0 | 11 | 0 | 00 | 000 | 00 | 0 | 00 |

Az egyes típusokba a következő képzők tartoznak: 1. *-gat, -get, -ogat, -eget, -hat, -het, -at, -et, -tat, -tet;*, 2. *-ál, -él, dogál, degél, -dögél;*, 3. *-oz, -ez, -öz, -ozik, -ezik, -özik;*, 4. *-kod, -ked, -köd;* amelyek típuson belül csak hangrendben és az ikességben különböznek egymástól.

Lehet, hogy a típuskódok közül a KH nevű típus helytelenül veszi egybe a KJE AR t. sz. 2. és 3. személy kötőhangzósságát. De mivel a gépi fordítású szövegekben tsz. 2. személyű alak feltehetőleg ritkán fog előfordulni, ezért az esetleges szokatlan képzések megbocsáthatók. Így az *IGYEKSZIK* igét kétféle típuskóddal ellátva is lehetne szerepeltetni a szótárban. Párhuzamosan a kétféle változat: (KH kódja 0 vagy 1)

*IGYEK | Z* (THG kódja 11), *IGYEKEZ* (THG kódja 00)

| | |
|---|---|
| *igyekszem* | *igyekezem* |
| *igyekszel* | *igyekezel* |
| *igyekszik* | *igyekezik* |
| *igyekszünk* | *igyekezünk* |
| *igyekszetek* | *igyekeztek* |
| *igyekszenek* | *igyekeznek* |
| *igyekvő* | *igyekező* |

A praktikus cél mindenek előtt nem az volt, hogy minden képzett alak stilisztikailag is tökéletes legyen, hanem az, hogy a kész, produkált alak ellen nyelvtanilag ne lehessen kifogást emelni. Így ez a modell szándékában a modellezésnek azt a fajtáját valósítja meg, amikor nem lehet képezni az összes helyes alakot, de a létrehozott alakok között nincs hibás.

**4.** Ezután tekintsük át azt a két munkát, amelyről a bevezetésben már történt említés.

JÁNOSKA SÁNDOR igeragozási programja a Magyar Nyelv Szóvégmutató Szótára alapján készült, ebből elsősorban a tőtípusok felosztását veszi át.

(azonos kódszámozással 1—52 között. Vö.: VégSz. 16—17. pp.) Minden egyes kódszámmal ellátott típus a ragozás szempontjából egy vagy több tőpéldányt (legfeljebb 4 tőváltozatot) tartalmazhat. A ragokat egy másik táblázat sorolja fel, a következő kódolási rendszerben 3-jegyű számokkal:

1.. KJE, 2.. kijelentő mód elbeszélő múlt,
3.. KM, 4.. FTJ, 5.. FSZ, 7.. FNIN és FNINR
.1. AR, .2. IMPL, .3. TR.
..1—..6 között: e. sz. és t. sz. 1—3. sz.

Külön alakja a 632-es, amely a felszólító mód tárgyas ragozás e. sz. 2. személy rövid alakja.

Minden rag képzése egy-egy ún. „kulcsalakra" van visszavezetve, amelyek a következők:

1. 114: KJE AR t. sz. 1. sz.
2. 112: KJE AR e. sz. 2. sz.
3. 115: KJE AR t. sz. 2. sz.
4. 116: KJE AR t. sz. 3. sz.
5. 134: KJE TR t. sz. 1. sz.
6. 311: KM AR e. sz. 1. sz.
7. 313: KM AR e. sz. 3. sz.
8. 413: FTJ AR e. sz. 3. sz.
9. 513: FSZ AR e. sz. 3. sz.
10. 632: FSZ TR e. sz. 2. sz. rövid alakja.

A ragok 3 fő hangrendi típusba vannak sorolva, a hangrendi típusok 7—7 alcsoportba. A tanulmány szerint a hangrendi típust szintén a VégSz. kódja szolgáltatja (i. m. 467), de ha ezt összevetjük a VégSz. kódolási rendszerével (VégSz. 19, D / 1 kód jelentései), akkor nem világos, hogy a magas ajakkerekítéses („ö-soros") típusnak mi a kódja. A táblázat nem minden részében tartalmaz kiírt toldalékot, hanem ezt valamilyen utalás oldja meg.

Az 1. táblázat, amelyben a tőtípusok és tőpéldányok felsorolása található, tartalmazza a képzési utasításokat is: A tővariáns száma nevű oszlop jelzi 1—4 közötti számmal, hogy típuson belül hányadik tőpéldány adatai következnek, a 4. oszlop a hangrend típusa, (nincs kitöltve, csak jegyzet utal arra, hogy a VégSz. kódja kerül erre a helyre), és az 5—14. számú tíz oszlopban a fenti tíz kulcsalak képzésére vonatkozó típuskód található.Ezek jelzik, hogy a ragozási táblázat hangrendi típusán belül az 1—7. altípus közül melyik ragot kell a tőhöz, ill. a megfelelő (rendhagyó) tővariánshoz illeszteni.

Ha több tőpéldány azonos oszlopában találunk számokat, akkor ez azt jelenti, hogy ugyanazon alaknak többféle formájú képzése lehetséges (pl. *SEPER-T, SEPR-ETT*).

Mivel a munka — valószínűleg a kongresszus főleg nyelvészeti beállítottsága miatt — nem közöl részleteket a program számítástechnikai vonatkozásairól, ezért csak feltevésekre támaszkodhatunk. Nem derül ki, hogy azok az alakok, amelyek nincsenek kiírva, hanem utalással találhatók meg, milyen társzükségletűek. Feltehető, hogy ezek mindössze 1-bites jelzők, de az is, hogy ugyanakkora hosszúságúak, mint azok a tömbelemek, amelyek a ragot kiírt formájában tartalmazzák (ezek többnyire kétszavasak, mert vannak köztük 4-karakteresnél hosszabbak is). Ezért ennek a programnak az előnyeit-hátrányait a következőkben foglalhatjuk össze: előnye, hogy párhuzamos alakváltozatok képezhetők, az összes ragozott alak helyesen állít-

ható elő, és viszonylag kicsi adatkezelő program szükséges hozzá. Hátránya, hogy még így is csaknem 2000 karakter szükséges az összes kiírt raghoz, összességében is nagyon nagy tömbök foglalása szükséges.

Lugosiné Papp Márta „A magyar igeszintézis egy modellje" c., közeljövőben megjelenő cikke Elekfi Lászlónak a „Szókészletünk nyelvtani alakrendszere" c. munkájában levő rendszeren alapszik,[15] némi változtatással. Ez azt jelenti, hogy minden egyes alak külön kódszámot kapott, és a nagyon hiányos ragozású igealakok egyszerű leírásához egy ún. ÜRES típus is csatlakozik. Mivel az Elekfi-rendszer „emberi használatra" készült, ezért abban minden egyes típus tételesen kiírva tartalmazza azokat az eltérő igeragokat, amelyek abban a típusban a kivételeket jelentik; így ugyanazt a ragot többször is felsorolják. Ez a gépi rendszerben egy visszautaló szisztéma („rekurzió") segítségével egyszerűsödik, s a gép max. 6-szoros rekurzióval megtalálja a tényleges ragot.

A programmal az elbeszélő múlt és az összetett igealakok kivételével 63 alakot lehet képezni: az összes személyragos alakot, az igeneveket, a főnévi igenév ragozott alakjait, ezenkívül a *-hat, -ható, -ás, -és,* (műveltető) *-at, -tat* és (szenvedő) *-atik, -tatik* alakokat is. A program a ható, szenvedő és műveltető alakokat továbbragozni is tudja. Mivel minden ige az Elekfi-rendszerben jellemző végződése, igekötő nélküli alakja alapján valamely típusba tartozik, így kivételek, rendhagyóságok egyáltalán nem fordulhatnak elő ebben a rendszerben (önálló típust képeznek azok az ún. egyedi igék is, amelyek típusába összesen egy ige tartozik), és ezért az ÉrtSz. mintegy 15 000 igéjének ragozásához az 515 típusnak megfelelő adatokon, az igekötők és jellemző végződések táblázatán kívül csak egy, kb. 3000—3500 egyedű alapige szótárra van szükség.

Minden ige ragozása visszavezethető a két fő (mély és magas "e-soros") alaptípus ragozására, vagy valamelyik másik típusra. A DISELT nevű mágneslemez-adatkészlet (disk file) tartalmazza mind az 515 típus 63 képezhető alakjára vonatkozó alapadatokat, egyenként egy-egy 63+1 bites vektorban. Ha a 64. bit értéke 0, akkor az utalás valamely alaptípusra közvetlenül történik, ha 1, akkor valamely másikra. Az utalt típus kódját egy 5-karakteres mező tünteti fel. A 63 bit közül azoknak az alakoknak a bitje 0, amelyek nem különböznek az utalt típus ragjaitól, a kivételeket pedig az 1-esek jelzik. A program egyszeres vagy többszörös utalás során eljut egy olyan típus bitjéhez, ahol feltétlenül 1-es található, vagyis az ahhoz tartozó rag különbözik minden más ragtól. Ennek a ragnak egy száma van, s a ragok kiírt alakjait számszerint növekvő sorrendben a DISRAG mágneslemez-file tartalmazza. Minden eltérést a program úgy kezel, hogy amikor valamelyik DISELT vektorban 1-eshez ér, megvizsgálja, hogy abban a vektorban ez hányadik egyes. Pl. Az ige 7. alakját keressük, és az egyik vektorban ezt találjuk: 0010001.... Ez abban a vektorban a második 1-es. Ugyanannak a típusnak a kódszáma szerint a DISMIN disk file-ban a második elem egy ragszámot tartalmaz. (515 DISELT vektornak ugyancsak 515 DISMIN adatrészlet felel meg, ezek különböző hosszúságúak, aszerint, hogy az adott típusban hány többitől eltérő rag van.) A ragszám szerinti ragot az ige tövéhez másolva, az alak létrehozása befejeződik.

[15] Teljes változata nem jelent meg, rövidített összefoglalását l. Elekfi László A magyar szövégek és toldalékok rendszere. MNy. 68:303—309, 412—429.

A rag tőváltozás esetén 1. karakterével vezérli a tőváltozás módját, háromféleképpen:

a) *VA:LASZT* − 2SZA = *VA:LASSZA* (szám jelzi, hogy a tőből hány betűt kell levágni);
b) *CSICSEREG* − *XEK* = *CSICSERGEK* (X jelzi, hogy a tőbeli utolsó magánhangzó kiesik), és
c) *UGRIK* − *YTOK* = *UGORTOK* (Y jelzi a hangbetoldást).

Ezenkívül a rag tartalmazza az alakváltozatokat, beleértve a felszólító mód e. sz. 2. személy rövid alakját is.

Ez a logikailag rendkívül következetes rendszer nemcsak igék ragozására, hanem (a szerző véleménye szerint) más magyar szavak szintézisére, sőt idegen szavak elemzésére is képes, csak a megfelelő adatokat kell hozzá összeállítani, a program változtatása nélkül. Így felfogható tágabb értelemben általános adatkezelő programként is.

A program által szolgáltatott eredmények, bármely ige teljes ragozása, különböző statisztikai vizsgálatok végrehajthatósága, információszolgáltató képessége (hiányos ragozás, stílusminősítések jelzése) szempontjából tudományos értékei vitathatatlanok. De ezek az előnyök nem tudják ellensúlyozni hihetetlenül hatalmas társszükségletét. A program, a négy mágneslemez file és az alapige-szótár együttes terjedelme meghaladja a 150 K byte-ot (1 K byte = 1024 byte), bár ebből az operatív tárban állandó jelleggel csak kb. 20 K byte-nyi rész tartózkodik, a többi valamely háttértárolón. De az igeragozáshoz szükséges adatok, a DISELT, DISMIN és DISRAG file-ok még így is kb. 40 K byte-nyi mágneslemez-helyet igényelnek.

**5.** Az előbb ismertetett két rendszer után úgy tűnhet, hogy azok közé tartozom, akik mindenek elé helyezik a tárterület maximális kihasználásának az elvét. Pedig ez az elv nem hagyható figyelmen kívül. Ugyanis nemcsak a nyelvészek bizonyos része tartozik a gépi fordítás ellentáborába, hanem a számítógépes szakemberek egy része is, akik a gépi fordítást a hatalmas tárterület-igény miatt tartják kilátástalannak. Ez utóbbiak állásfoglalása könnyebben megváltoztatható, ugyanis a számítógépek korszerűsödése egyre inkább legyőzi ezt a gondot, de a továbblépés szempontjából ezek a gazdaságossági kérdések sem elhanyagolhatók.

Ezek után összefoglaljuk a vektorszorzás módszerén alapuló program főbb tulajdonságait, amelyek úgy véljük, egyben előnyei is: A program nem tárolja a ragokat teljes kiírt alakjukban, csupán betűsorok formájában. Így mindössze 114 betűt kell tárolni, és a betűválasztó vektorok a választóvektorokkal együtt sem hosszabbak, mint kb. 450 byte. Maga a program, amennyiben önálló szubrutin lesz, szótár nélkül, beleértve a kb. 575 byte-os adathalmazt is, kb. 3 K byte nagyságú. A jelenlegi kis szótár 36 szavas, a programban levő változata a rendhagyó alakok összes adataival együtt is csak 514 byte-os. Így egy igére átlagosan 15 byte esik, ha az ÉrtSz. kb. 7500-nyi igekötő nélküli igéjére egyenként 20 byte-os hosszúságot számítunk, a 150 000 byte terjedelem kb. 147 K byte-ot jelent. De minden valószínűség szerint ez a szám még tovább csökkenthető, hiszen a gépi fordításban előreláthatólag sokkal kevesebb igével is megelégedhetünk.

Mielőtt dolgozatomat befejezném, egy megjegyzés a táblázat oktális számadataihoz: A többi adatformától eltérően, ebben a szó elején vannak

a betűválasztó vektorok, és az 1., ill. 2. szó végén az utolsó 8 bit van fenntartva a választóvektor számára. Mivel oktális felírásmódot a szavas gépeknél használnak, ezekben a választóvektor hátulrahelyezése látszik célszerűbbnek, a könnyebb kezelhetőség érdekében.

A 2. táblázat áttekinthetősége érdekében a kettes számrendszerű adatokat nem a szakirodalomban szokásos 1—0, ill. L—0 alakban írtam, hanem 0 helyett . (pontot) használva.

Remélem, hogy evvel az ismertetéssel sikerült némiképp hozzájárulni mind elméleti, mind gyakorlati vonatkozásban a témakör munkálataihoz.[16]

VÁSÁRHELYI S. ISTVÁN

[16]A hivatkozott táblázatokat alább közöljük.

*1. táblázat*

A programban használt igék szótára

| Nyelvtani információk | | | | | | | | | Tőalakok | Rendhagyó tő esetvektorai (hexadec.) |
|---|---|---|---|---|---|---|---|---|---|---|
| HR | IK | J2 | J3 | THG | FSZT | SZL | KH | TVLT | | |
| 00 | 0 | 11 | 0 | 00 | 011 | 01 | 0 | 00 | *akadaalyoz* | |
| 00 | 1 | 11 | 0 | 11 | 000 | 00 | 0 | 00 | *alkud* | |
| 00 | 0 | 00 | 0 | 00 | 000 | 00 | 0 | 00 | *akar* | |
| 00 | 0 | 01 | 1 | 00 | 110 | 11 | 1 | 00 | *aart* | |
| 01 | 0 | 01 | 1 | 00 | 100 | 11 | 1 | 00 | *eebreszt* | |
| 01 | 0 | 01 | 1 | 00 | 011 | 01 | 1 | 11 | *edz* | |
| | | | | | | | | | *edd* | 9C C1 C2 C3 C4 C5 C6 C9 CB CC CD CE CF |
| | | | | | | | | | *ed(d)z(e)d* | 4A |
| 11 | 1 | 11 | 0 | 11 | 000 | 01 | 0 | 00 | *eskued* | |
| 01 | 1 | 00 | 0 | 11 | 000 | 01 | 1 | 01 | *fek#d* | |
| | | | | | | | | | *fekve* | 50 |
| 01 | 0 | 01 | 1 | 00 | 110 | 11 | 1 | 01 | *fest* | |
| | | | | | | | | | *fes* | C1 C2 C3 C4 C5 C6 C9 CA CB CC CD CE CF |
| 11 | 0 | 01 | 1 | 00 | 110 | 11 | 1 | 00 | *gyuejt* | |
| 00 | 0 | 01 | 1 | 00 | 000 | 11 | 1 | 01 | *hall* | |
| | | | | | | | | | *hallak* | 1F |
| 00 | 1 | 01 | 1 | 00 | 011 | 01 | 1 | 00 | *hangwz* | |
| 01 | 0 | 00 | 0 | 11 | 000 | 01 | 0 | 11 | *hiq* | |
| | | | | | | | | | *higq* | C1 C2 C3 C4 C5 C6 C9 CB CC CD CE CF |
| | | | | | | | | | *higgyed(hidd)* | 4A |
| 00 | 0 | 01 | 1 | 00 | 110 | 11 | 1 | 00 | *javiit* | |
| 11 | 0 | 00 | 0 | 11 | 000 | 00 | 0 | 11 | *joeq* | |
| | | | | | | | | | *joev* | 91 94 CO |
| | | | | | | | | | *joessz* | 12 |
| | | | | | | | | | *joen* | 13 96 |
| | | | | | | | | | *joettoek* | 15 |
| | | | | | | | | | *joej* | C1 C2 C3 C4 C5 C6 |
| 00 | 1 | 01 | 1 | 00 | 111 | 01 | 1 | 00 | *laatsz* | |
| 01 | 0 | 00 | 0 | 00 | 000 | 00 | 0 | 11 | *men* | |
| | | | | | | | | | *megy* | 91 13 94 |
| | | | | | | | | | *meesz* | 12 |
| 01 | 0 | 01 | 1 | 00 | 111 | 01 | 1 | 00 | *metsz* | |
| 00 | 0 | 11 | 0 | 00 | 001 | 01 | 0 | 00 | *mos* | |
| 00 | 0 | 11 | 0 | 00 | 000 | 00 | 0 | 00 | *mozwg* | |
| 00 | 0 | 00 | 0 | 00 | 000 | 00 | 0 | 00 | *ro** | |
| 11 | 0 | 00 | 0 | 00 | 000 | 00 | 0 | 00 | *szoe** | |
| 01 | 0 | 00 | 0 | 11 | 000 | 01 | 0 | 01 | *teq* | |
| | | | | | | | | | *tegyed(tedd)* | 4A |
| 11 | 0 | 00 | 0 | 00 | 000 | 00 | 0 | 00 | *toerwl* | |
| 11 | 0 | 01 | 0 | 00 | 101 | 00 | 0 | 00 | *uet* | |
| 01 | 0 | 00 | 0 | 11 | 000 | 01 | 0 | 01 | *veg* | |
| | | | | | | | | | *vegyed(vedd)* | 4A |
| 01 | 0 | 01 | 0 | 00 | 101 | 00 | 0 | 00 | *vet* | |
| 00 | 1 | 00 | 0 | 11 | 000 | 01 | 1 | 01 | *al#d* | |
| | | | | | | | | | *alva* | 50 |
| 00 | 1 | 11 | 0 | 11 | 000 | 01 | 1 | 00 | *harag\|d* | |
| 00 | 1 | 11 | 0 | 11 | 000 | 01 | 1 | 00 | *nyugxd* | |
| 01 | 1 | 11 | 0 | 01 | 011 | 01 | 1 | 00 | *emleek\|z* | |
| 01 | 1 | 11 | 0 | 11 | 011 | 01 | 1 | 00 | *igyek\|z* | |
| 01 | 1 | 11 | 0 | 01 | 000 | 01 | 1 | 00 | *mosak\|d* | |

*2/a. táblázat*

A kijelentő mód jelen idő képzése
(Bemenő adat: 1 1 J K)

A választóvektor bitjeinek jelentése:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| HR | IK | J2 | J3 | FSZR | — | SZL | KH |

| Bemenő adat J K | Típus | Választó-vektor | | Betűsor és betűválasztó vektorok | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1234 | 5678 | AEJS | ZJSZ | NIIU | LAEN | AOOE | TOOE | KKML | D |
| 1 1 AR | | 11.. | .... | .... | .... | .... | .... | .... | .111 | 1.1. | . |
| 2 | | 1... | ..1. | 11.. | ..11 | .... | .... | .... | .111 | ...1 | . |
| 3 | | .1.. | .... | .... | .... | .1.. | .... | .... | .... | .1.. | . |
| 4 | | 1... | .... | .... | .... | ...1 | ..11 | .... | .... | .1.. | . |
| 5 | | 1... | ...1 | .... | .... | .... | .... | .111 | 1111 | .1.. | . |
| 6 | | 1... | ...1 | 11.. | .... | 1... | .11. | .... | .... | .1.. | . |
| 2 1 TR | | 1... | .... | .... | .... | .... | .... | .... | .111 | ..1. | . |
| 2 | | 1... | .... | .... | .... | .... | .... | .... | .111 | .... | 1 |
| 3 | | 1... | 1... | ..11 | 1... | ..1. | .1.. | .... | .... | .... | . |
| 4 | | 1... | 1... | .... | .111 | ...1 | ..1. | .... | .... | .1.. | . |
| 5 | | 1... | 1... | ..11 | 1... | ..1. | .1.. | 1... | 11.1 | .1.. | . |
| 6 | | 1... | 1... | ..11 | 1... | ..1. | .1.. | 1... | .... | .1.. | . |
| 7 IMPL | | 1... | ...1 | 11.. | .... | .... | 111. | .... | .... | .1.. | . |
| 3 1 FNINR | | 1..1 | .... | 11.. | .... | 1... | .... | .... | .111 | ..1. | . |
| 2 | | 1..1 | .... | 11.. | .... | 1... | .... | .... | .111 | .... | 1 |
| 3 | | 1..1 | .... | 11.. | .... | 11.. | .11. | .... | .... | .... | . |
| 4 | | 1..1 | .... | 11.. | .... | 1..1 | ..11 | .... | .... | .1.. | . |
| 5 | | 1..1 | .... | 11.. | .... | 1... | .... | .111 | 1111 | .1.. | . |
| 6 | | 1..1 | .... | 11.. | .... | 11.1 | ..1. | .... | .... | .1.. | . |
| | HR = 00 | | | 1.11 | 1111 | 11.1 | 11.1 | 11.. | 11.. | 1111 | 1 |
| | 01 | | | .1.. | .111 | 1111 | 1.11 | ...1 | 1..1 | 1111 | 1 |
| | 11 | | | .1.. | .111 | 1111 | 1.11 | ..11 | ..11 | 1111 | 1 |
| | IK = 0 | | | .... | .... | .... | .... | .... | .111 | 1... | . |
| | 1 | | | .... | .... | .1.. | .... | .... | .111 | .11. | . |
| | J3 = 0 | | | .... | .... | 11.1 | .111 | .111 | 1111 | .11. | 1 |
| | 1 | | | 11.. | .... | 11.1 | .111 | .1.1 | 11.1 | .11. | 1 |
| | FSZT = 000 | | | ..1. | .1.. | ..11 | .11. | 1... | 11.1 | .1.. | . |
| | 001 | | | ...1 | ..1. | ..11 | .11. | 1... | 11.1 | .1.. | . |
| | 010 | | | ...1 | 1.11 | ..11 | .11. | 1... | 11.1 | .1.. | . |
| | 011 | | | .... | 1..1 | ..11 | .11. | 1... | 11.1 | .1.. | . |
| | 100 | | | ..1. | .1.. | ..11 | .11. | 1... | 11.1 | .1.. | . |
| | 101 | | | ..1. | .1.. | ..11 | .11. | 1... | 11.1 | .1.. | . |
| | 110 | | | ..1. | .1.. | ..11 | .11. | 1... | 11.1 | .1.. | . |
| | 111 | | | ...1 | 1.11 | ..11 | .11. | 1... | 11.1 | .1.. | . |
| | SZL = 00 | | | .... | ..11 | .... | .... | .... | .... | .... | . |
| | 01 | | | .... | .... | .... | .... | .... | .111 | ...1 | . |
| | 11 | | | 11.. | ..11 | .... | .... | .... | .... | .... | . |
| | KH = 0 | | | .... | .... | 1... | 111. | .... | 1111 | .1.. | . |
| | 1 | | | 11.. | .... | 1... | 1111 | .111 | 1111 | .1.. | . |

*2/b. táblázat*

**A kijelentő mód múlt idő és a feltételes mód jelen idő képzése (Bemenő adatok: 1 2 J K, ill. 1 3 J K)**

| Bemenő adat J K | Típus | Betűválasztó vektorok | | | | | |
|---|---|---|---|---|---|---|---|
| KM betűsor | | OEOE | TTTA | AUEE | TOEM | LAED | NK |
| 1 1 AR | | 11.. | 11.1 | ..1. | ...1 | .... | .. |
| 2 | | 11.. | 11.1 | 1.11 | .... | 1... | .. |
| 3 | | 1111 | 111. | .... | .... | .... | .. |
| 4 | | 11.. | 11.. | .11. | .... | .... | 11 |
| 5 | | 11.. | 11.1 | ..1. | 111. | .... | .1 |
| 6 | | 11.. | 11.1 | ..1. | .... | .... | .1 |
| 2 1 TR | | 11.. | 11.1 | ..1. | ...1 | .... | .. |
| 2 | | 11.. | 11.1 | ..1. | .... | ...1 | .. |
| 3 | | 11.. | 11.1 | ..1. | .... | .... | .. |
| 4 | | 11.. | 11.. | .11. | .... | .... | .1 |
| 5 | | 11.. | 11.1 | 1.11 | 111. | .... | .1 |
| 6 | | 11.. | 11.1 | 1.11 | .... | .... | .1 |
| 7 IMPL | | 11.. | 11.1 | ..1. | .... | 111. | .1 |
| | HR = 00 | 1.1. | 1111 | 11.. | 11.1 | 11.1 | 11 |
| | 01 | .1.1 | 1111 | 1111 | 1.11 | 1.11 | 11 |
| | 11 | 1111 | 1111 | .111 | 1.11 | 1.11 | 11 |
| | J2 = 00 | .... | 1..1 | 1111 | 1111 | 1111 | 11 |
| | 01 | 11.. | 11.1 | 1111 | 1111 | 1111 | 11 |
| | 11 | ..11 | .111 | 1111 | 1111 | 1111 | 11 |

| FTJ betűsor | | AENA | AEEE | ETNA | OELA | EKMD | |
|---|---|---|---|---|---|---|---|
| 1 1 AR | | 111. | ...1 | 1... | .... | .1.. | |
| 2 | | 1111 | 111. | .... | ..1. | .... | |
| 3 | | 1111 | .1.. | .... | .... | .... | |
| 4 | | 1111 | 111. | ..1. | .... | .1.. | |
| 5 | | 1111 | 111. | .1.. | 11.. | .1.. | |
| 6 | | 1111 | 111. | ..11 | .1.. | .1.. | |
| 2 1 TR | | 1111 | 111. | .... | .... | ..1. | |
| 2 | | 1111 | 111. | .... | .... | ...1 | |
| 3 | | 1111 | 111. | .... | .... | .... | |
| 4 | | 1111 | 111. | ..1. | .... | .1.. | |
| 5 | | 1111 | 111. | .1.. | 11.. | .1.. | |
| 6 | | 1111 | 111. | .... | .... | .1.. | |
| 7 IMPL | | 1111 | 111. | .... | ..11 | 11.. | |
| | HR = 00 | 1.11 | 1..1 | 1111 | 1.11 | .111 | |
| | 01 | .11. | .111 | 111. | .11. | 1111 | |
| | 11 | .11. | .111 | 111. | .11. | 1111 | |
| | J3 = 0 | ..11 | 1111 | 1111 | 1111 | 1111 | |
| | 1 | 1111 | 1111 | 1111 | 1111 | 1111 | |

*2/c. táblázat*

A felszólító mód és az igenevek képzése (Bemenő adatok: 1 4 K J ill. 0 LI 0 0)

A választóvektor bitjeinek jelentése:
1 2 3 4 5 6 7 8
HR IK J2 J3 FSZT — SZL KH

| Bemenő adat | | Típus | Választó vektorok | | Betűválasztó vektorok | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1234 | 5678 | | | | | | | | |
| J | K | FSZ betűsor | | | (JS) | (Z)( | AUEN | AETO | OEL) | AENM | DKEE | K |
| 1 | 1 AR | | 11.. | 1... | .11. | 111. | 1.1. | .... | .... | ...1 | .... | . |
| | 2 | | 1... | 1... | .11. | 1111 | 1.1. | 11.. | ..11 | .... | .... | . |
| | 3 | | 11.. | 1... | .11. | 111. | .... | ...1 | 11.. | ..1. | ..11 | 1 |
| | 4 | | 1... | 1... | .11. | 111. | .111 | .... | .... | .... | .1.. | . |
| | 5 | | 1... | 1... | .11. | 111. | 1.1. | ..11 | .1.. | .... | .1.. | . |
| | 6 | | 1... | 1... | .11. | 111. | ..11 | 11.. | .... | .... | .1.. | . |
| 2 | 1 TR | | 1... | 1... | .11. | 111. | 1.1. | .... | ...1 | .... | .... | . |
| | 2 | | 1... | 1... | 1111 | 1111 | 1.1. | .... | ...1 | .... | 1... | . |
| | 3 | | 1... | 1... | .11. | 111. | 1.1. | .... | .... | .... | .... | . |
| | 4 | | 1... | 1... | .11. | 111. | .11. | .... | .... | .... | .1.. | . |
| | 5 | | 1... | 1... | .11. | 111. | 1.1. | 1111 | .1.. | .... | .1.. | . |
| | 6 | | 1... | 1... | .11. | 111. | 1.1. | 11.. | .... | .... | .1.. | . |
| | 7 IMPL | | 1... | 1... | .11. | 111. | 1.1. | 11.. | ..1. | 11.. | .1.. | . |
| | | HR = 00 | | | 1111 | 1111 | 11.1 | 1.11 | ..11 | 1.11 | 1111 | 1 |
| | | 01 | | | 1111 | 1111 | .111 | .11. | .111 | .111 | 1111 | 1 |
| | | 11 | | | 1111 | 1111 | .111 | .11. | 1111 | .111 | 1111 | 1 |
| | | IK = 0 | | | .11. | 111. | 1.1. | ...1 | 11.. | ..1. | .1.. | . |
| | | 1 | | | .11. | 111. | 1.1. | .... | .... | ...1 | ..11 | 1 |
| | | FSZT = 000 | | | 11.. | ...1 | 1111 | 1111 | 1111 | 1111 | 1111 | 1 |
| | | 001 | | | 1.1. | ...1 | 1111 | 1111 | 1111 | 1111 | 1111 | 1 |
| | | 010 | | | 1.11 | .1.1 | 1111 | 1111 | 1111 | 1111 | 1111 | 1 |
| | | 011 | | | 1... | .1.1 | 1111 | 1111 | 1111 | 1111 | 1111 | 1 |
| | | 100 | | | 1.11 | .1.1 | 1111 | 1111 | 1111 | 1111 | 1111 | 1 |
| | | 101 | | | 1.1. | ...1 | 1111 | 1111 | 1111 | 1111 | 1111 | 1 |
| | | 110 | | | ..1. | ...1 | 1111 | 1111 | 1111 | 1111 | 1111 | 1 |
| | | 111 | | | 1.11 | 1111 | 1111 | 1111 | 1111 | 1111 | 1111 | 1 |

| LI | IN betűsor | | | VAEA | ENID | OOOE | TT | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 1..1 | .... | ...1 | 111. | .... | .. | | | | |
| 2 | | 1... | .... | .... | .... | 11.1 | .. | | | | |
| 3 | | 1.1. | .... | .... | .... | 1.11 | 11 | | | | |
| 4 | | 1... | .... | ...1 | 11.1 | 11.1 | .. | | | | |
| 5 | | 1... | .... | 1..1 | 1... | .... | .. | | | | |
| 6 | | 1... | .... | 1111 | 11.. | .... | .. | | | | |
| | HR = 00 | | | 11.1 | .111 | 1... | 11 | | | | |
| | 01 | | | 1.1. | 1111 | .1.1 | 11 | | | | |
| | 11 | | | 1.1. | 1111 | .111 | 11 | | | | |
| | J2 = 00 | | | .... | .... | .... | .1 | | | | |
| | 01 | | | .... | .... | 1.11 | 11 | | | | |
| | 11 | | | .... | .... | 1.11 | 11 | | | | |
| | J3 = 0 | | | .... | .11. | .... | .. | | | | |
| | 1 | | | ...1 | 111. | .... | .. | | | | |

*3/a. táblázat*

## A kijelentő mód jelen idő vektorainak hexadecimális és oktális alakja

| Bemenő adat J K | Önálló alak esetvektora (hex.) | Ragozandó alak esetvektora (hex.) | Hexadecimális alak (5 ill. 4 byte) | Oktális alak (2 szó) |
|---|---|---|---|---|
| 1 1 AR | 11 | 91 | C0 00 00 07 A0 | 00000007 50000300 |
| 2 | 12 | 92 | 82 C3 00 07 10 | 60600007 04000202 |
| 3 | 13 | 93 | 40 00 40 00 40 | 00040000 20000100 |
| 4 | 14 | 94 | 80 00 13 00 40 | 00011400 20000200 |
| 5 | 15 | 95 | 81 00 00 7F 40 | 00000177 20000201 |
| 6 | 16 | 96 | 81 C0 86 00 40 | 60103000 20000201 |
| 2 1 TR | 19 | 99 | 80 00 00 07 20 | 00000007 10000200 |
| 2 | 1A | 9A | 80 00 00 07 08 | 00000007 02000200 |
| 3 | 1B | 9B | 88 38 24 00 00 | 16022000 00000210 |
| 4 | 1C | 9C | 88 07 12 00 40 | 01611000 20000210 |
| 5 | 1D | 9D | 88 38 24 8D 40 | 16022215 20000210 |
| 6 | 1E | 9E | 88 38 24 80 40 | 16022200 20000210 |
| 7 IMPL | 1F | 9F | 81 C0 0E 00 40 | 60007000 20000201 |
| 3 1 FNINR | | | 90 C0 80 07 20 | 60100007 10000220 |
| 2 | | | 90 C0 80 07 08 | 60100007 02000220 |
| 3 | | | 90 C0 C6 00 00 | 60143000 00000220 |
| 4 | | | 90 C0 93 00 40 | 60111400 20000220 |
| 5 | | | 90 C0 80 7F 40 | 60100177 20000220 |
| 6 | | | 90 C0 D2 00 40 | 60151000 20000220 |
| HR = 00 | | | BF DD CC F8 | 57756714 76000000 |
| 01 | | | 47 FB 19 F8 | 21775431 76000000 |
| 11 | | | 47 FB 33 F8 | 21775463 76000000 |
| IK = 0 | | | 00 00 07 80 | 00000007 40000000 |
| 1 | | | 00 40 07 60 | 00040007 30000000 |
| J3 = 0 | | | 00 D7 7F 68 | 00153577 32000000 |
| 1 | | | C0 D7 5D 68 | 60153535 32000000 |
| FSZT =000 | | | 24 36 8D 40 | 11033215 20000000 |
| 001 | | | 12 36 8D 40 | 04433215 20000000 |
| 010 | | | 1B 36 8D 40 | 06633215 20000000 |
| 011 | | | 09 36 8D 40 | 02233215 20000000 |
| 100 | | | 24 36 8D 40 | 11033215 20000000 |
| 101 | | | 24 36 8D 40 | 11033215 20000000 |
| 110 | | | 24 36 8D 40 | 11033215 20000000 |
| 111 | | | 1B 36 8D 40 | 06633215 20000000 |
| SZL = 00 | | | 03 00 00 00 | 00600000 00000000 |
| 01 | | | 00 00 07 10 | 00000007 04000000 |
| 11 | | | C3 00 00 00 | 60600000 00000000 |
| KH = 0 | | | 00 8E 0F 40 | 00107017 20000000 |
| 1 | | | C0 8F 7F 40 | 60107577 20000000 |

*3/b. táblázat*

**A kijelentő mód múlt idő és a feltételes mód jelen idő vektorainak hexadecimális és oktális alakja**

| Bemenő adat J K | Önálló | Ragozandó | Hexadecimális alak (3 byte) | Oktális alak (1 szó) |
|---|---|---|---|---|
| | alak esetvektora (hex.) | | | |
| KM 1 1 AR | 21 | A1 | CD 21 00 | 6322 0400 |
| 2 | 22 | A2 | CD B0 80 | 6333 0200 |
| 3 | 23 | A3 | FE 00 00 | 7740 0000 |
| 4 | 24 | A4 | CC 60 0C | 6306 0014 |
| 5 | 25 | A5 | CD 2E 04 | 6322 7004 |
| 6 | 26 | A6 | CD 20 04 | 6322 0004 |
| 2 1 TR | 29 | A9 | CD 21 00 | 6322 0400 |
| 2 | 2A | AA | CD 20 10 | 6322 0020 |
| 3 | 2B | AB | CD 20 00 | 6322 0000 |
| 4 | 2C | AC | CC 60 04 | 6306 0004 |
| 5 | 2D | AD | CD BE 04 | 6333 7004 |
| 6 | 2E | AE | CD B0 04 | 6333 0004 |
| 7 | 2F | AF | CD 20 E4 | 6322 0344 |
| HR=00 | | | AF CD DC | 5374 6734 |
| 01 | | | 5F FB BC | 2777 5674 |
| 11 | | | FF 7B BC | 7764 5674 |
| J2 = 00 | | | 09 FF FC | 0237 7774 |
| 01 | | | CD FF FC | 6337 7774 |
| 11 | | | 37 FF FC | 1577 7774 |
| FTJ 1 1 AR | 31 | B1 | E1 80 40 | 7030 0100 |
| 2 | 32 | B2 | FE 02 00 | 7740 1000 |
| 3 | 33 | B3 | F4 00 00 | 7500 0000 |
| 4 | 34 | B4 | FE 20 40 | 7742 0100 |
| 5 | 35 | B5 | FE 4C 40 | 7744 6100 |
| 6 | 36 | B6 | FE 34 40 | 7743 2100 |
| 2 1 TR | 39 | B9 | FE 00 20 | 7740 0040 |
| 2 | 3A | BA | FE 00 10 | 7740 0020 |
| 3 | 3B | BB | FE 00 00 | 7740 0000 |
| 4 | 3C | BC | FE 20 40 | 7742 0100 |
| 5 | 3D | BD | FE 4C 40 | 7744 6100 |
| 6 | 3E | BE | FE 00 40 | 7740 6100 |
| 7 IMPL | 3F | BF | FE 03 C0 | 7740 1700 |
| HR =00 | | | B9 FB 70 | 5637 5560 |
| 01 | | | 67 E6 F0 | 3176 3360 |
| 11 | | | 67 E6 F0 | 3176 3360 |
| J3 = 0 | | | 3F FF F0 | 1777 7760 |
| 1 | | | FF FF F0 | 7777 7760 |

*3/c. táblázat*

## A felszólító mód és az igenevek vektorainak hexadecimális és oktális alakja

| Bemenő adat | | Önálló alak esetvektora (hex.) | Ragozandó alak esetvektora (hex.) | Hexadecimális alak (5 ill. 4 byte) | Oktális alak (2 szó) |
|---|---|---|---|---|---|
| FSZ | J K | | | | |
| 1 | 1 | 41 | C1 | C8 6E A0 01 00 | 33520001 00000310 |
| | 2 | 42 | C2 | 88 6F AC 30 00 | 33726060 00000210 |
| | 3 | 43 | C3 | C8 6E 01 C2 38 | 33400702 16000310 |
| | 4 | 44 | C4 | 88 6E 70 00 40 | 33470000 20000210 |
| | 5 | 45 | C5 | 88 6E A3 40 40 | 33521500 20000210 |
| | 6 | 46 | C6 | 88 6E 3C 00 40 | 33436000 20000210 |
| 2 | 1 | 49 | C9 | 88 6E A0 10 00 | 33520020 00000210 |
| | 2 | 4A | CA | 88 FF A0 10 80 | 77720020 40000210 |
| | 3 | 4B | CB | 88 6E A0 00 00 | 33520000 00000210 |
| | 4 | 4C | CC | 88 6E 60 00 40 | 33460000 20000210 |
| | 5 | 4D | CD | 88 6E AF 40 40 | 33527500 20000210 |
| | 6 | 4E | CE | 88 6E AC 00 40 | 33526000 20000210 |
| | 7 | 4F | CF | 88 6E AC 2C 40 | 33526054 20000210 |
| HR | = 00 | | | FF DB 3B F8 | 77755473 76000000 |
| | 01 | | | FF 76 77 F8 | 77673167 76000000 |
| | 11 | | | FF 76 F7 F8 | 77673367 76000000 |
| IK | = 0 | | | 6E A1 C2 40 | 33520702 20000000 |
| | 1 | | | 6E A0 01 38 | 33520001 16000000 |
| FSZT | = 000 | | | C1 FF FF F8 | 60377777 76000000 |
| | 001 | | | A1 FF FF F8 | 50377777 76000000 |
| | 010 | | | B5 FF FF F8 | 55377777 76000000 |
| | 011 | | | 85 FF FF F8 | 41377777 76000000 |
| | 100 | | | B5 FF FF F8 | 55377777 76000000 |
| | 101 | | | A1 FF FF F8 | 50377777 76000000 |
| | 110 | | | 21 FF FF F8 | 10377777 76000000 |
| | 111 | | | BF FF FF F8 | 57777777 76000000 |
| LI — Igenevek — | | | | (3 ill. 2 byte) | (1 szó) |
| 1 | | 10 | 90 | 90 1E 00 | 0740 0220 |
| 2 | | 20 | A0 | 80 00 D0 | 0015 0200 |
| 3 | | 30 | B0 | A0 00 BC | 0013 6240 |
| 4 | | 40 | C0 | 80 1D D0 | 0735 0200 |
| 5 | | 50 | D0 | 80 98 00 | 4600 0200 |
| 6 | | 60 | E0 | 80 FC 00 | 7700 0200 |
| HR | = 00 | | | D7 8C | 6570 6000 |
| | 01 | | | AF 5C | 5365 6000 |
| | 11 | | | AF 7C | 5367 6000 |
| J2 | = 00 | | | 00 04 | 0000 2000 |
| | 01 | | | 00 BC | 0013 6000 |
| | 11 | | | 00 BC | 0013 6000 |
| J3 | = 0 | | | 06 00 | 0140 0000 |
| | 1 | | | 1E 00 | 0740 0000 |

VHI: MAGYAR IGEALAKOK SZINTEZISE

| SZOTARI TOE: | ADATOK: | SZINTETIZALT IGEALAK: |
|---|---|---|
| AKADAALYOZ | 1 1 1 2 | AKADAALYOZOL |
| ALKUD | 0 2 0 0 | ALKUVO |
| ALKUD | 1 1 1 4 | ALKUSZUNK |
| ALKUD | 1 6 1 5 | ALKUDTATOK VOLNA |
| AKAR | 1 2 1 3 | AKART |
| AART | 1 1 1 6 | AARTANAK |
| EEBRESZT | 1 4 2 4 | EEBRESSZUEK |
| EDZ | 1 4 1 1 | EDDZEK |
| ESKUED | 0 2 0 0 | ESKUEVOE |
| FEST | 1 2 2 7 | FESTETTELEK |
| FEST | 0 4 0 0 | FESTENDOE |
| HALL | 1 1 2 7 | HALLAK |
| HALL | 1 2 2 7 | HALLOTTALAK |
| JAVIIT | 0 6 0 0 | JAVIITVAAN |
| MOS | 1 1 1 3 | MOS |
| MOS | 1 4 1 3 | MOSS(AAL) |
| MOZWG | 1 2 1 3 | MOZGOTT |
| RO* | 0 2 0 0 | ROVO |
| TEQ | 1 1 1 3 | TESZ |
| AL#D | 1 1 1 4 | ALSZUNK |
| AL#D | 1 1 2 4 | ALUSSZUK |
| EMLEEK/Z | 1 1 1 4 | EMLEEKSZUENK |
| MOSAK/D | 1 1 1 4 | MOSAKSZUNK |
| MOSAK/D | 1 5 1 1 | MOSAKODNAANK |

| | | |
|---|---|---|
| GYUEJT | 1 4 1 2 | GYUEJTS(EEL) |
| HANGWZ | 0 1 0 0 | HANGZANI |
| HANGWZ | 1 1 1 3 | HANGZIK |
| HANGWZ | 1 4 1 3 | HANGOZZEEK |
| HIQ | 1 1 1 2 | HISZEL |
| HIQ | 1 4 1 2 | HIGGY(EEL) |
| JOEQ | 0 1 0 0 | JOENNI |
| JOEQ | 0 5 0 0 | JOEEVE |
| LAATSZ | 1 1 1 3 | LAATSZIK |
| LAATSZ | 1 4 1 3 | LAA(T)SS(Z)EEK |
| MEN | 1 5 1 1 | MENNI FOGOK |
| MEN | 1 5 1 4 | MENNI FOGTOK |
| MOZWG | 1 2 1 4 | MOZOGTUNK |
| RO* | 0 5 0 0 | ROTT |
| TEQ | 1 2 1 3 | TETT |
| TEQ | 0 4 0 0 | TEENDOE |
| TEQ | 0 5 0 0 | TEEVE |
| TOERWL | 1 2 2 6 | TOEROELTEEK |
| TOERWL | 0 2 0 0 | TOERLOE |
| UET | 1 1 2 2 | UETOED |
| UET | 1 4 2 2 | UES(S(E)D |
| VEQ | 1 3 2 7 | VENNEELEK |
| AL#D | 1 4 1 4 | ALUDJUNK |
| HARAG/D | 0 2 0 0 | HARAGVO |
| NYUGXD | 0 2 0 0 | NYUGVO |
| NYUGXD | 1 1 1 4 | NYUGSZUNK |
| EMLEEK/Z | 1 6 1 1 | EMLEEKEZTEM VOLNA |
| IGYEK/Z | 0 2 0 0 | IGYEKVOE |
| IGYEK/Z | 1 4 1 4 | IGYEKEZZUENK |

## A Synthesis of Hungarian Verbal Forms through the Help of a Computer (Production of verbal suffixes making use of the product of Boolean vectors)

The author applies in the synthesis of language signs the method employed by Dénes Varga in analyses made up to now. (Dénes Varga: Morphological analysis by help of the method of successive delimination. Computational Linguistics, 1963, 1:223—255). Because of the exceptionally large number and large variety of Hungarian verbal suffixes, large arrays, which uneconomically utilized the memory capacity of the computer, became necessary. It was possible to reduce this number by employing a Boolean vector for each single code type of the machine dictionary; the suffix characters are stored in alphanumeric row matrixes. Two or more vector products contain the value of 1 only where an enumeration of the appropriate characters results in the desired outcome. Without the help of a computer, it can also be demonstrated with visual punched cards.

István Vásárhelyi S.