LINGUISTICA

SERIES A

STUDIA ET DISSERTATIONES, 4.

PAJZS JÚLIA

SZÁMÍTÓGÉP ÉS LEXIKOGRÁFIA

A MAGYAR TUDOMÁNYOS AKADÉMIA NYELVTUDOMÁNYI INTÉZETE
INSTITUTUM LINGUISTICUM ACADEMIAE SCIENTIARUM HUNGARICAE

1990

LINGUISTICA
SERIES A
STUDIA ET DISSERTATIONES, 4.

SZÁMÍTÓGÉP ÉS LEXIKOGRÁFIA

LINGUISTICA

SERIES A

STUDIA ET DISSERTATIONES, 4.

PAJZS JÚLIA

SZÁMÍTÓGÉP ÉS LEXIKOGRÁFIA

A MAGYAR TUDOMÁNYOS AKADÉMIA NYELVTUDOMÁNYI INTÉZETE
INSTITUTUM LINGUISTICUM ACADEMIAE SCIENTIARUM HUNGARICAE

1990

Lektorálta: Kálmán László

ISBN 963 8461 40 3
ISSN 0238 8642

MAGEX
Computer typeset by Heckenast

Felelős kiadó: Herman József
Hozott anyagról sokszorosítva

9019250 MTA Sokszorosító, Budapest. F. v.: dr. Héczey Lászlóné

Tartalom

1. Bevezetés 7

2. Számítógépes szótári adatbázisok 9

2.1. Nyomtatott szótárból készült szótári adatbázisok 10

2.1.1. A Debreceni Tezaurusz 10

2.1.2. Merriam-Webster 11

2.1.3. Az új Van Dale szótár 12

2.1.4. Az Új Oxford English Dictionary 13

2.1.5. A Longman szótár 18

2.2. Új szótárak készítése számítógép segítségével 20

2.2.1. A Trésor de la langue française 20

2.2.2. Dictionary of Old English 26

2.2.3. A COBUILD szótár 27

2.2.4. New York Times Everyday Dictionary 28

2.2.5. Német lexikográfiai adatbázis 29

2.3. A történeti szótárak és a számítógép 30

3. A számítógépes szótárak előállítása 33

3.1. Adatgyűjtés és rögzítés 33

3.2. Lemmatizálás 34

3.3. Konkordanciák 35

3.4. Szócikkírás 39

4. A magyar irodalmi és köznyelv nagyszótára 41

4.1. Történeti előzmények 41

4.2. A nagyszótár új koncepciója 43

4.3. Az adatgyűjtés számítógépes vonatkozásai 46

4.3.1. Az adatrögzítés legfontosabb konvenciói 46

4.3.2. Forrásnyilvántartás 48

4.3.3. Előfeldolgozás 50

4.3.4. Morfológiai elemzés 51

4.4. Az adatok feldolgozása 63

4.4.1. Index előállítás 64

4.4.2. Logikai egységek kijelölése 66

4.4.3. A kereső-program 67

4.4.4. A címszójegyzék összeállítását segítő program 70

4.4.5. A szótár szerkesztését támogató programrendszer 72

4.5. Zárómegjegyzések 73

Mellékletek 75

Irodalom 79

1. Bevezetés

AZ ELSŐ SZÁMÍTÓGÉPES SZÓTÁRAK még a 60-as években készültek, ennek ellenére a számítógépes lexikográfia a 80-as években kezd önálló tudományággá válni. Míg az első szótárak főként a fényszedés melléktermékeként kerültek számítógépre, a 70-es évek végén, 80-as évek elején mind több olyan projektum indult meg, amelynek fő célja egy lehetőleg sokoldalúan felhasználható szótári adatbázis, avagy nyelvi korpusz előállítása, és csak mintegy melléktermékként áll elő ebből a nyomtatott szótár. A számítógépes lexikográfia mint önálló tudományág jelen pillanatban van születőben; jól jelzik ezt a sűrűsödő számítógépes lexikográfiai konferenciák, szimpóziumok és „workshop"-ok, illetőleg a lexikográfiai konferenciák számítógépes szekciói. E tudományág kialakulását elsősorban a számítógépek technikai fejlődése tette lehetővé, és a korábbi számítógépes szótárak kedvező és kedvezőtlen tapasztalatai tették sürgetővé.

A számítógépes szótárak legfőbb előnye a nyomtatottakkal szemben az, hogy míg a nyomtatott szótárakban kizárólag a címszavak ábécérendjében tudunk keresni, a számítógépes szótári adatbázisban — az adatbázishoz használt szoftver minőségétől függően — számtalan különböző szempont szerint kereshetünk. További nagy előnyük, hogy viszonylag könnyen javíthatók, aktualizálhatók, pótkötetek készítése helyett a kiegészítések, javítások beágyazhatók az egységes anyagba. Ha egyszer létrehozunk egy szótárat számítógéppel olvasható formában, utána a szótár újabb, módosított kiadásai könnyen elkészíthetők, ráadásul minimális az esélye annak, hogy a javítás során újabb hibákat vigyünk az adatbázisba. Kellő körültekintéssel készíthető olyan számítógépes szótári adatbázis is, amely egyszerre több nyomtatott szótár anyagát tartalmazza. Például egy szótári család három tagja, a nagyszótár, kéziszótár és zsebszótár szócikkei közös adatbázisban tárolhatók, ha speciális jelekkel megkülönböztetjük, hogy melyik címszó szerepel mindháromban, melyik csak a nagyszótárban, a szócikkekben pedig szintén elkülönítjük a közös, és csak az egyik vagy másik változatban publikálandó részeket, jelentéseket stb. Mindezen túl, bármely számítógépes szótár tekinthető úgy, mint az adott nyelv egy korpusza, amely sok szempontból érdekesebb információkat tartalmaz, mint egy folyó szövegekből összeállított korpusz, hiszen számos grammatikai, esetleg nyelvtörténeti adat is található benne. Különösen izgalmas kutatásokat folytathatunk akkor, ha egy nyelvről folyó szövegekből álló korpuszunk és szótárunk is van számítógépesített formában: összehasonlíthatjuk a kétféle adatbázisból kapott adatokat. Sőt, a szótárból eleve legalább háromféle adatot kaphatunk ugyanarra a nyelvi jelenségre, attól függően, hogy a címszavak között,

az értelmezések között vagy az idézetek között keresgélünk: a címszóállományból például a szókincsről kaphatunk információkat, az idézetekből korok szerint csoportosított szó/szókapcsolat előfordulásokat kerestethetünk, míg az értelmezésekből a szótárírók által használt nyelvről nyerhetünk adatokat. A fenti szempontokat figyelembe véve döntött úgy az MTA Elnöksége, hogy a Magyar irodalmi és köznyelv nagyszótárát — amely már régi adóssága a magyar lexikográfiának — számítógép segítségével kell előállítani. A szótár forrásanyagául számítógépre rögzített folyamatos szövegek szolgálnak, és maga a szótár is számítógépen készül majd. A jelen dolgozat célja e munkálat számítógépes vonatkozásainak ismertetése a nemzetközi és magyar számítógépes lexikográfiai tapasztalatok tükrében. A dolgozat első felében a legfontosabb szótári adatbázisokat ismertetem, és összegzem a számítógépes történeti szótárak szerkesztésének leglényegesebb tanulságait (2.). A 3. fejezetben a számítógépes szótárírás munkafázisait, a 4-ben a magyar nagyszótár munkálatait ismertetem.

2. Számítógépes szótári adatbázisok

A SZÁMÍTÓGÉPES SZÓTÁRAKAT többféleképpen csoportosíthatjuk. Megkülönböztethetjük a számítógéppel olvasható szótárakat (machine-readable dictionaries) és a számítógépesített szótárakat (computerized dictionaries) (KAZMAN 1986). A különbség a kettő között az, hogy a számítógéppel olvasható szótár rendszerint csupán a fényszedés melléktermékeként jön létre, ezért csak a fényszedésnél használt speciális jeleket tartalmazza, és nem tükrözi megfelelően a szótár belső struktúráját, így nem támogatja a szótárban való több szempontú keresést. A számítógépesített szótár ezzel szemben olyan adatbázis, amely jól tükrözi a szótár belső szerkezetét, speciális adatbáziskezelő szoftverrel van ellátva, amelynek segítségével hatékonyan kereshetünk a szótár bármely részében. A számítógéppel olvasható szótárak egy részét fokozatosan átalakítják számítógépesített formára. Mivel a strukturálatlan, csupán számítógéppel olvasható szótárak meglehetősen érdektelenek, a továbbiakban csak a számítógépesített szótárakkal foglalkozom, és az egyszerűség kedvéért ezentúl csak ezeket nevezem számítógépes szótárnak. Ezek csoportosíthatók funkciójuk szerint is: vannak olyanok, amelyeket gépi fordítás vagy természetes nyelvű interfész[1] tőtáraként használnak (ezeket sokszor „lexical database"-ként emlegeti az angol szakirodalom, szembe állítva a „computerized dictionary"-val), vagy olyanok, amelyek nyomtatott szótárak alapanyagául szolgálnak és így tovább. Az alábbiakban ismertetendő számítógépes szótárakat létrehozásuk körülményei alapján csoportosítottam. A 2.1. pontban ismertetem a nyomtatott szótárakból készülteket: ezek jelentős része lényegében számítógéppel olvasható szótárból hoz létre szótári adatbázist (lexical database-t). A 2.2-ben olyan szótárakat sorolok fel, amelyeket számítógépes korpuszból, gépi segítséggel készítettek. Ezek a típusok azonban rendszerint nem tiszták, a projektumok nagy része egyszerre fel is dolgozza már elkészült szótárak adatait, de ugyanakkor újabb nyomtatott szótár kiadását is célul tűzi ki, a számítógépes adatbázis többféle felhasználásának egyikeként. A munkálatok kimerítő felsorolására azonban nem törekszem, hiszen egyrészt angolul elérhetőek teljességre törekvő beszámolók (AMSLER 1984, KIPFER 1984, KEITZ 1982, WARWICK 1986),[2] másrészt tanulságosabbnak tűnik néhány jellemző projektum részletesebb ismertetése, mint az összes vázlatos felsorolása.

[1] Interfész (vagy interface): kapcsolat szoftver ill. hardver eszközök között. Természetes nyelvű interfésznek az olyan szoftvereket nevezik, amelyek a felhasználó és a számítógép közötti természetes nyelvű (magyar, angol, stb.) kommunikációt teszik lehetővé.

[2] Amsler tanulmánya a számítógépes szótárakról jó kiindulópont a téma tanulmányozásához, elsősorban

Nem foglalkozom gyakorisági szótárakkal és a természetes nyelvi interfészekhez, fordítórendszerekhez készített szótárakkal, mivel ezek meghaladnák a jelen dolgozat kereteit. Részletesebben foglakozom két kiemelkedő jelentőségű munkálattal: az Oxford English Dictionary számítógépes változatának előállításával, és a Trésor de la langue française-zel, részben, mert ezek a szótárak hasonlítanak céljaikban leginkább a tervezett magyar NSz-ra, másrészt, mivel ezekkel — tanulmányútjaimnak köszönhetően — alaposabban megismerkedhettem.

2.1. Nyomtatott szótárból készült szótári adatbázisok

2.1.1. A Debreceni Tezaurusz

A NEMZETKÖZI viszonylatban is úttörő jellegű munkálatok a 60-as évek elején indultak meg. Az MTA Matematikai nyelvészeti munkabizottságának 1962. évi ülésén döntés született arról, hogy el kell készíteni a magyar nyelv szóvégmutató szótárát számítógép segítségével. A munkálat irányitását a Debreceni Matematikai és Alkalmazott Nyelvészeti munkacsoport vezetőjére, Papp Ferencre bízták, aki fokozatosan ismerte fel, hogy nem szabad megelégednie a bizottság által kitűzött céllal: nem elegendő elkészíteni a szóvégmutató szótárat (VégSz.), hanem meg kell ragadni az alkalmat, hogy egy mai értelemben vett számítógépes szótári adatbázist hozzanak létre. (Noha ez a fogalom akkoriban még aligha létezett.) Korai publikációinak már a címe is mutatja, hogy eleinte benne sem tudatosult: nem szóvégmutató szótár a lényeg, hanem egy lexikális adatbázis létrehozása. Az e pontban alább említendő nemzetközi kutatásoktól eltérően, amelyek mind a szótárra mint szótárra irányultak, Papp kezdettől fogva a szókincset magát, az egyes szavakban megtestesülő grammatikát kívánta géppel vizsgálni, ehhez a szótár csak mint eszköz jelent meg (benne vannak összegyűjtve a szavak). Ezért csak azt rögzítette a szótárból, ami *objektíve*, kvázi *grammatice* az egyes szavakra vonatkozik (alaktani, mondattani stb. ismérvek) és minden egyéb lexikográfiai adatot figyelmen kívül hagyott (jelentésekre bontás, idézetek, az egyes jelentések magyarázata). Teljes szótárak rögzítésekor, amint ez a későbbi nemzetközi kutatásokban történt, esetenként mintha szem elől tévesztődne a lényeg, a nyelv, annak szókincse, alaktani szerkezete, az ezen belül fellelhető globális törvényszerűségek. „Össze tudjuk gyűjteni az összes Shakespeare idézetet — de minek, amikor engem mint nyelvészt inkább az érdekel, miért ***ablaka*** de ***barackja***, miért ***szépen*** de ***jól***, mely szavakban van *baleket–balekot*-féle ingadozás, mikor

---

azért, mert kimerítő bibliográfia egészíti ki. A tanulmány egyébként rengeteg adatot tartalmaz meglehetősen eklektikus elrendezésben, sok, a témához nem szorosan kapcsolodó részt és kevés eredeti gondolatot. Viszont a témában nem járatos érdeklődönek képet ad a számítógépes lexikográfia jelenlegi állásáról.

Kipfer tanulmánya lényegesen értékesebb, jól szerkesztett, ennélfogva több hasznos információt tartalmaz, bibliográfiája kevésbé részletes. Nem törekszik arra, hogy az összes számítógépes szótárat ismertesse, de nagyon használható felbontásban ismerteti egyes kutatások részeredményeit.

Keitz csupán felsorolja a European Science Foundation által támogatott összes számítógépes szótári projektumot, országonként csoportosítva. Ez az összeállítás azért nagyon hasznos, mert a teljesség igényével készült, megadja az egyes témák legfontosabb adatait, a munkálatot végző intézmény nevét és címét.

Warwick arra törekszik, hogy egy általános bevezető után ismertesse és értékelje az összes európai, nem angol projektumot. Hiánypótló munka.

gyárak és mikor *kalapok* stb. Ilyen jellegű kérdések persze az angol kiindulási agyban fel sem vetődnek, mert nincsenek; talán ezért *is* szóródhatott szét a figyelem a szótár egészére, a szavakon belül meg olyan jelenségekre, mint egymásra rímelő szavak." (PAPP 1988) Ezért nem csupán azokat az adatokat vitték gépre, amelyek a VégSz. előállításához feltétlenül szükségesek voltak, hanem egy sor olyan további információt is, amelyek a későbbiek során számos értékes megfigyelés, kutatási eredmény alapjául szolgáltak. A szótár címszóállományát az ÉrtSz. címszavai alkották, és a kódok nagy részéhez is az ebben található információk szolgáltak alapul. Először is minden egyes címszót külön lyukkártyára rögzítettek (az ékezetes betűk helyett részben számokat használva), balra és jobbra igazítva (ez utóbbi változatra az a-tergo rendezéshez volt szükség). Ezen túlmenően kódolták a szóra vonatkozó grammatikai, stilisztikai és etimológiai információkat is. A grammatikai és stilisztikai kódokat az ÉrtSz. adataiból, az etimológiára vonatkozókat a Bárczi-féle Szófejtő szótárból vették. A grammatikai kódok közül különösen értékesek a főnév ragozási típusára vonatkozók: részben ezen adatokból kiindulva születhetett meg többek közt egy új főnévszintézis modell (PAPP 1966, PAPP 1975). A stilisztikai minősítések kódolása lehetővé tette érdekes összesítések készítését (vö. KORNAI 1986), az etimológiai kódoknak köszönhetően pedig új megvilágításba kerülhetett számos, az anyagból nyert puszta statisztikai eredmény. Egyebek közt kimutatták, hogy az ÉrtSz. legtöbb jelentéssel bíró címszavai között a formaszavakon kívül számos ige is található, és valamennyi ilyen ige magyar ill. finnugor eredetű (vö. PAPP 1969B). Egy másik összesítésből kiviláglott, hogy a finnugor eredetű szavak mássalhangzósabbak, mint a szláv eredetű jövevényszavak. Ezekből a példákból is jól látszik, miért nevezték el ezt az adatbázist tezaurusznak, „kincsestárnak": az eredetileg egy szótár alapanyagául szánt adatbázisból a szóvégmutató szótáron kívül számos összesítés, tanulmány stb. született. Az anyag külön érdekessége az ún. „debreceni kód": minden olyan címszót, amely összetett szó vagy igekötős ige volt, a gyűjtők saját nyelvérzékük alapján maguk láttak el grammatikai kódokkal. Egy szerencsés véletlen folytán (vö. KORNAI 1986) a régi, lyukkártyára rögzített anyag megmenekült az enyészettől, sikerült mágneslemezre konvertálni. Jelenleg az MTA SZTAKI nagy IBM gépén érhető el a teljes anyag, a címszavak és a kódok egy része megtalálható az MTA Nyelvtudományi Intézet személyi számítógépein is. Ezt az adatbázist ma is több célra használjuk, egyebek között, mint később látni fogjuk, a NSz. tőtárának alapanyagául szolgál.

2.1.2. Merriam-Webster

AZ ANGOL SZÓTÁRAK közül az első, amelyet számítógépen is tároltak a *Merriam-Webster Seventh New Collegiate Dictionary* (W7), és a *Merriam-Webster New Pocket Dictionary* volt (MPD) (AMSLER 1984). Miután elkészültek a nyomtatott szótárak, kutatást indítottak a számítógépes változat kidolgozására. Az erre irányuló projektum 1966–68-ig tartott, Olney és Ziff vezetésével. Először is lyukkártyára gépelték a két szótár teljes anyagát, ami nem volt könnyű, mivel a lyukkártyák karakterkészlete jóval kisebb, mint a nyomtatott szótáré. A felvitel után programokat írtak a szócikkek egy részének elemzésére, konkordanciát készítettek az értelmezésekre, morfológiai elemző és generáló programokat fejlesztettek, és leválogatták a szócikkek egy részét további kutatás céljára.

Ez a projektum a számítógépes lexikográfia „hőskorának" terméke. Az adatbázis jelenleg is a kutatók rendelkezésére áll, számos — elsősorban nyelvészeti — kutatás alpanyagául használták.

Az egyik ilyen, a W7-et alapanyagául használó kutatás során (BYRD 1984, 1985) a szótárhoz egy olyan adatbáziskezelőt készítettek, amellyel 3 „dimenzióban" lehet kérdéseket feltenni. A WORDSMITH elnevezésű szótári adatbáziskezelő használatakor a képernyő közepén állandóan az aktuális szócikket látjuk egy ablakban,[3] az ablak körül csillag alakban kiírathatók a kapcsolódó címszavak. A dimenzió itt az anyag háromféle szempontból való rendezettségét jelenti; eszerint az első dimenzió a címszavak ábécérendje, a második dimenzió a rímelő szavak listája, a harmadik dimenzió az a-tergo rendezés. Ez a két utóbbi csak látszatra tűnhet azonosnak, hiszen míg az a-tergo az íráskép szerint azonos végződésű szavakra van rendezve, a rímelő szavak listájában a kiejtés alapján hasonló végződésű szavak találhatók egymás mellett. Azaz minden aktuális szócikkhez kiírathatók a szótárban előtte-utána lévő címszavak, az ezzel rímelő szavak, a megegyező végződésű szavak. Mivel az adatbázisba beleépítették Roget Thesaurusának anyagát is, egy további dimenzióban az egyes szavak szinonimáit is kérhetjük. Később a Longman Dictionary of Contemporary English (1978) szócikkállományát szintén hozzáillesztették az adatbázishoz, és folyamatosan újabb és újabb kérdés-dimenziókat adnak a rendszerhez. Ilyen pl. a kiejtés-dimenzió, amely az aktuális címszó összes lehetséges kiejtését listázza ki. Ez a projektum jó példa arra, hogyan lehet és kell fokozatosan továbbfejleszteni egy szótári adatbázist, részben újabb és újabb szótárak hozzáadásával, részben pedig a szoftver eszközök állandó fejlesztésével.

2.1.3. Az új Van Dale szótár

A VAN DALE SZÓTÁRT (Nieuw Woordeboek der Nederlandsche Taal) először 1872-ben adták ki, ezt újabb és újabb átdolgozott kiadások követték (STERKENBURG 1981). A szócikkek a *Woordenboek der Nederlandsche Taal* (A holland nyelv szótára) alapján készültek, amely az 1500–1920 közötti szókincsállományt öleli fel. 1976-ban a Van Dale korábbi kiadóvállalata (Kluwer) létrehozta a Van Dale Projektum leányvállalatot, abból a célból, hogy elkészítsék a Van Dale-nek egy olyan módosított változatát, amelynek címszóállománya többnyelvű szótárak alapanyagául szolgálhat. Első lépésként a jelenlegi holland szókincs feltérképezését tűzték ki célul. Mivel a kiadó viszonylag gyors eredményt várt, nem törekedhettek arra, hogy sok nyomtatott szöveget vigyenek számítógépre, és ebből próbálják előállítani az aktuális szókincset. Inkább a Van Dale címszóállományát vitték gépre, csupán néhány népszerű folyóirat és magazin anyagával egészítették ki. Ugyanakkor egy csoport hagyományos módon gyűjtötte az új szavakat/jelentésárnyalatokat, és az ezekre talált idézeteket szintén hozzáadták az adatbázishoz.

A Van Dale mintegy 220 000 címszavát körülbelül 90 000-re kellett csökkenteni, részben azért, hogy csak a szinkrón szókincset tartalmazza, részben azért, hogy kevesebb

[3] Ablakrendszer: olyan szoftver eszköz, amelynek segítségével számítógépünk képernyőjét több, egymástól jól elkülöníthető részre oszthatjuk, az egyes részeket egymástól keretekkel választja el a programrendszer. Az egy-egy keretben lévő részt nevezik ablaknak.

helyet foglaljon el. Ezért a Van Dale utolsó kiadását teljes terjedelmében mágnesszalagra vitték, majd a lexikográfusok minden címszó minden jelentésárnyalatát ellátták egy „szinkrónia kóddal", amely azt jelölte, felvegyék-e a címszót/jelentést az új címszóállományba. (A kód 0-tól 5-ig terjedő érték volt.) A kód felhasználásával a gép válogatta ki az új címszólállományt.

A címszóállomány összeállítása után megtervezték az új szótárak szerkezetét. Ehhez segítségképpen felhasználták egy felmérés eredményeit. A felmérést 1979-ben piackutatással együtt végezték: megkérdezték a várható szótárfelhasználókat, szerintük milyen adatoknak kell a szótárakban lenniük. Az általános vélemény szerint az egynyelvű értelmező szótárban elsősorban a szó helyesírásának és értelmezésének kell szerepelnie, de szeretnék, ha a régies szavak és a szinonimák is benne lennének. Az etimológiát, tabu szavakat, antonimákat kevésbé tartják fontosnak.

Elhatározták, hogy külön, tömör nyelvtani részt készítenek, a szócikkekben pedig erre a nyelvtanra fognak hivatkozni. A kétnyelvű szótárakban összevető nyelvtan lesz, példákkal.

Mivel a teljes szótári adatbázist számítógépen fogják tárolni, az általános szócikkszerkezetet és az alkalmazott kódokat úgy alakították ki, hogy az egyértelmű gépi tárolást és keresést biztosítsák. Külön kód jelzi a címszó elején, hogy az adott szócikk melyik szótárban legyen benne (nagy-, közép-, zseb-), utána nyelvtani szempontból szétbontva következnek a jelentések, példákkal.

Az így kialakított adatbázis legfőbb előnye az, hogy a három nyelv (francia, német, angol) kétnyelvű szótárai ugyanazt a szókincset fogják tartalmazni, a szerkezetük szintén egyforma lesz, így aki az egyik szótárat megtanulja kezelni, az összes többit is tudja használni. Az egységes grammatikai-szemantikai kódolás biztosítja a számítógépes változatban való keresést.

2.1.4. Az Új Oxford English Dictionary

Az Oxford English Dictionary (Murray 1884–1928) (OED) a legnagyobb történeti angol szótár. A XIX. század közepén kezdődött meg az anyaggyűjtés, 1857-ben határozta el a londoni Philological Society a szótári munkálatok megindítását. A gyűjtést kezdetben Jonh Furniwall irányította; a szótár első főszerkesztője James A. H. Murray volt, és noha nem érhette meg a teljes szótár megjelenését, a mű szembetűnő belső egysége, szemlélete elsősorban neki köszönhető. Az első füzet 1884-ben jelent meg, az utolsó 1928-ban, eredetileg 125 papír kötésű vékony füzetben. Ezt egy pótkötet kiadása követte 1933-ban, amit a szótár első és utolsó kötetének megjelenése közt eltelt 44 év tett szükségessé. Az ötvenes évek végén az Oxford University Press (a továbbiakban OUP) Robert Burchfield irányításával hozzákezdett egy újabb pótkötet készítéséhez, amely az 1933 óta keletkezett új szavakat ill. új jelentésárnyalatokat tartalmazta, vagy csupán új idézeteket adott a szócikkekben szereplő jelentésekhez. Ez a kiegészítés végül is négy kötetes lett, az első 1972-ben az utolsó 1986-ban jelent meg.

Az OED szócikkeinek számát a szerkesztők 414 825-ben jelölték meg (Országh 1966), ezzel szemben a legfrisebb számítógéppel készített statisztika szerint valójában mindössze 252 259 szócikket tartalmaz, a pótkötetek anyagát is figyelembe véve pedig

321 631-et! (UW Centre for the New OED Newsletter No.15. 1987, No.17. 1988). A szótár elvben az angol nyelv teljes szókészletét tartalmazza, a gyűjtés során legalábbis ezt tűzték célul maguk elé. A szócikkek szerkezete szigorúan a történetiség elvét követi: előbb vannak a szó bizonyíthatóan legrégibb jelentései, a megfelelő idézetekkel, ez után következnek a később kialakult jelentésváltozatok. A kihalt szavak vagy jelentésárnyalatok esetén az utolsó előfordulást is megadják. Az idézetek válogatásakor arra törekedtek, hogy minden jelentésárnyalatot, minden századból legalább egy adattal dokumentálni tudjanak.

Az utolsó pótkötet készítése idején az OUP lexikográfusai gondolkozni kezdtek, vajon mi történjék a szótárral ezután. Reménytelen és értelmetlen vállalkozásnak tűnt a régi módszerekkel folytatni az új szavak jelentésének összegyűjtését, és ezeket újabb és újabb pótkötetekben kiadni, hiszen ha minden szót 3–4 helyen kell megnézni ahhoz, hogy összeálljon egy teljes szócikk, akkor a szótár lassanként használhatatlanná válik. Ugyanakkor esedékessé vált a teljes szótár újbóli kiadása is, amely a régi nyomólemezek felhasználásával már nem lett volna lehetséges, így a fényszedéssel való újbóli kiadás mellett döntöttek. Ezért 1984-ben, az OED első kötete megjelenésének 100. évfordulóján elhatározták, hogy a teljes OED-t és a négy pótkötetet számítógépre viszik, s a pótkötetek anyagát — amennyire lehet, automatizálva — összeolvasztják az OED eredeti szócikkállományával, és egyúttal a 80-as években keletkezett új szavakat is hozzáadják a szótárhoz. A szótári adatbázis anyagát teljes egészében kiadják nyomtatásban, a továbbiakban pedig a szótári anyag kiegészítését mindig a számítógépes adatbázison végzik el.

Az adatbázis nemcsak a szótár kiadását teszi gyorsabbá, korszerűbbé, s a XXI. században is használhatóvá, hanem eddig elképzelhetetlen távlatokat nyit az angol nyelv kutatói számára. Hiszen míg a nyomtatott szótárban az egyetlen keresési mód a címszavak ábécérendjében való keresés, addig a számítógépen tárolt verzióban közvetlenül kereshetünk számos más szempont szerint: pl. kikereshetjük az összes Shakespeare idézetet, vagy az összes görög eredetű szót stb. Az OED számítógépesítése két fő munkafázisból áll, az első fázisban számítógépre viszik a szótár és a pótkötetek teljes szövegét, a pótkötetek anyagát összeolvasztják a szótárral, és némi javítás után kiadják a teljes szótárat. Az új szótár várhatóan 1989-ben fog megjelenni. A második fázis a számítógépes szótári adatbázis kialakítása oly módon, hogy az egyrészt segítse a lexikográfusok munkáját, másrészt támogassa a legváltozatosabb kutatói igényeket. A program megvalósítására több intézmény fogott össze: a munkálatok irányítását, a lexikográfiai és a kiadással kapcsolatos feladatokat az OUP végzi, az IBM UK számítógép hardver és szoftver adományokkal, továbbá számítástechnikai szakértők kölcsönzésével támogatja a témát. Az adatrögzítést az ICC (International Computaprint Corporation) végezte. A kanadai University of Waterloo pedig a szótári adatbázis számítástechnikai realizálására és a számítógépes szótárakkal kapcsolatos kutatások támogatására kutatóközpontot hozott létre (UW Centre for the New OED).

Az első fázis munkálatai

Mielőtt a rögzítéshez hozzákezdtek volna, kipróbáltak egy Kurzweill optikai olvasó berendezést, ez azonban nem vált be. Mivel a szótárban túl gyakran váltakoznak a különböző

karaktertípusok, az optikai olvasó az elvárásokhoz képest nagyon lassan tudta csak bevinni a szöveget. A kísérletek alapján úgy tűnt, mintegy 12,5 év alatt lehetne felvinni az adatokat ezzel a géppel. Az ICC ezzel szemben 18 hónap alatt rögzítette a mintegy 500 millió karakternyi szótár anyagát! (120 adatrögzítő dolgozott rajta egyidejűleg!) Egyéb kísérletek is arra vallanak, hogy az optikai olvasót elsősorban olyan esetekben lehet hatékonyan alkalmazni, amikor egy-egy könyv teljes szövegét kell felvinni, és a szövegben nem váltakoznak túl gyakran a betűtípusok. A rögzítéssel kapcsolatos legnagyobb probléma az volt, hogyan őrizzék meg a szótár eredeti belső struktúráját, amit elsősorban a tipográfiával fejeztek ki a szerkesztők. Végül is kidolgoztak egy speciális kódrendszert, amely átmenet a tipográfiai kódolás és a strukturális kódolás között. Elsősorban az volt a cél, hogy a rögzítők számára könnyen érthetőek és alkalmazhatóak legyenek a jelek, de ugyanakkor semmilyen információ ne vesszen el az eredeti szótárból. Szemléltetésül néhány tétel a kódtáblázatból:

+ET etimológia kezdete
+FB matematikai formula kezdete
+FE matematikai formula vége
+FT idegen szöveg
+G gót betű
+GB vastag gót betű

A rögzített szövegeket mind az ICC, mind az OUP többszörösen ellenőrizte (az OUP által végzett ellenőrzés előtt az átlagos hibaarány kb. 4 hiba volt 10 000 leütésben, a többszöri ellenőrzés után ez kb. 1/250 000-re csökkent). Ahhoz azonban, hogy az íly módon rögzített szövegből létre tudjanak hozni egy olyan adatbázist, amelynek struktúrája tükrözi az eredeti szótár struktúráját, és ugyanakkor biztosítja a különböző szempontok szerinti keresést, egy külön elemző programot kellett írniuk. A program egy erre a célra fejlesztett mesterséges nyelv használatával működik. A nyelv szerepe elsősorban az volt, hogy explicitté tegye azt a struktúrát, amelyet a kódolt változat csak implicite tartalmazott. Az ICC által alkalmazott kódrendszer legfőbb hiányossága az volt ugyanis, hogy csak a különböző szerkezeti elemek, ill. betűtípusok kezdetét jelölték, a végét nem, ami bonyolult szerkezetek esetén megnehezítette az adatok visszakeresését. A másik nehézség, hogy a kódok jelentős része csak a tipográfiára utal, ám a különböző betűtípusok, attól függően, hogy a szócikk melyik részében fordulnak elő, mást és mást jelölhetnek. Ezért volt szükség egy olyan változat előállítására, amely pontosan tartalmazza az egyes szótári egységek elejét ill. végét.

A pótkötetek anyagát csak részben lehet automatikusan összefésülni az eredeti szótár anyagával, ezért egy speciális interaktív programot készítettek ennek a munkafolyamatnak a támogatására. A pótkötetekből származó szócikkek egyértelmű információt tartalmaznak arról, milyen műveletet kell velük végezni (egy régi értelmezést egy újjal helyettesíteni, vagy csak idézeteket kell hozzáadni a megfelelő jelentéshez, esetleg egy új címszót felvenni a megfelelő helyre). Az összeolvasztás után az eddigi utalások jelentős része helytelen lesz, ezért ezeket — amennyire lehet automatikusan — módosítani kell az OED teljes szövegében. Az OED-ben Murray által kifejlesztett speciális fonetikai jeleket

alkalmaztak, ezeket most helyettesítik az International Phonetic Alphabet jeleivel. Az új kiadás előtt a szócikkek egy részét is átdolgozzák ill. kijavítják, felvesznek egy-két új jelentést is, mindez azonban csak néhány címszót fog érinteni. A szócikkek komolyabb átdolgozását csak egy későbbi kiadás előtt tervezik, mivel túlságosan nagy feladat lenne ezt az összefésüléssel egyidejűleg megoldani. (Az esetleges átdolgozással kapcsolatos szerkesztői problémákról részletesebben l. SIMPSON 1985A.) A szótári adatbázist jelenleg egy standard IBM adatbáziskezelő nyelv, az SQL segítségével érik el. Az SQL olyan relációs adatbáziskezelő szoftver, amelyet bármilyen programnyelvből el lehet érni, CMS alatt használható, és változó hosszúságú mezők hatékony kezelésére is alkalmas. Mindazonáltal a szótári projektum második fázisában kifejlesztendő adatbázis nem az SQL-t fogja használni, hanem külön szótárra orientált adatbáziskezelő rendszert írnak erre a célra. Az SQL fölé az OUP-ben kifejlesztettek egy szövegszerkesztőt, kifejezetten a fentebb ismertetett feladatokra specializálva. (Oxford English Dictionary Integration, Proofreading, and Updating System: OEDIPUS)

A szótári adatbázisból fényszedéssel fogják előállítani a nyomtatott szótárat, a kiadás várható időpontja 1989. Mintegy 22 kötetet terveznek, az ára várhatóan 1500 angol font lesz. A nyomtatott formában való kiadáson túl az optikai lemezen való terjesztést is tervezik. Már 1987-ben kiadták az eredeti OED szövegét a Supplement nélkül optikai lemezen, főként abból a célból, hogy tapasztalatokat szerezzenek a számítógépes változat használatával kapcsolatban. Ezen tapasztalatok felhasználásával akarják kifejleszteni azt a szoftvert, amelyet majd a teljes szótárhoz fognak árusítani.

A második fázis munkálatai

Az ún. második fázis — a szótári adatbázis kiadáson túli, kutatási célokra való továbbfejlesztése — a valóságban párhuzamosan kezdődött az elsővel, és valószínűleg több egymást követő fázisból fog állni. A szótári adatbázisokkal kapcsolatos kutatói és szoftverfejlesztői tevékenységet a University of Waterloo Centre for the New OED vállalta magára. A kutatóközpont — szoros együttműködésben az OUP-vel — részt vett az elemző program kidolgozásában, és hozzákezdett a szöveges adatbázisok számára leghatékonyabb adatbáziskezelő-rendszer kialakításához. A szoftverfejlesztést megelőzően az elképzelhető felhasználók körében felmérést végeztek arról, hogy várhatóan milyen típusú kérdésekre akarnak majd választ kapni a kutatók. A megkérdezettek olyan kutatók, könyvtárosok, írók és újságírók közül kerültek ki, akiknek valószínűleg lesz lehetőségük és igényük arra, hogy használják az OED számítógépes változatát. A felmérés egyik legfontosabb eredménye az volt, hogy míg a nyomtatott szótárakban gyakorlatilag csak a címszó alapján lehet keresni, a számítógépes változatban csak az esetek 28%-ában szeretnének egy-egy adott szócikkel megismerkedni, 54%-ban viszont előre nem ismert címszavak vagy szócikkek összekeresését várják eredményként. (pl. írd ki az összes olyan címszót, amely magyar eredetű.) Talán a kérdőívre adott válaszok világítottak rá először arra, hogy a *vagy* nyomtatott szótár *vagy* számítógépes lexikográfiai adatbázis kérdésfeltevés helytelen: mind a kettő kell, más-más célra. Míg a nyomtatott szótárban elsősorban az egyes címszavakhoz tartozó értelmezéseket akarják kikeresni, a számítógépes változatban az idézeteket ill. a hozzájuk tartozó címszavakat akarják a leggyakrabban kiíratni. (Míg a nyomtatott szó-

tár, annak használata stb. évtizedek óta ismert, a számítógépes lexikográfiai adatbázis használatában viszont nincs tapasztalat — így lehet, hogy itt jobban fognak változni az igények, teljesen újak fognak megjelenni és így tovább.) Részben ebből a felmérésből, részben a gyakorlati tapasztalatokból azt a következtetést vonták le, hogy a szótári adatbáziskezelő szoftver fejlesztésekor nem arra kell törekedniük, hogy a nyomtatott szótár helyett használják a számítógépes változatot, hanem arra, hogy olyan kérdésekre tudjanak választ adni, amelyekre számítógép nélkül egyáltalán nem, vagy csak igen nehezen lehetne válaszolni. Már csak azért is indokoltnak látszik ez a törekvés, mert ha az embernek egyaránt lehetősége van a nyomtatott és a számítógépes változat használatára, lényegesen egyszerűbb a könyvben fellapozni a keresett címszót, mint begépelni a számítógépbe, előzőleg beindítani a megfelelő kereső programot stb. Kétségtelen azonban az is, hogy néhány éven/évtizeden belül valószínűleg lényegesen olcsóbb lesz megvenni a szótárt optikai lemezen, mint nyomtatott formában, ez esetben nyilván több olyan felhasználó lesz, aki a nyomtatott szótár helyett is a számítógépes változatot akarja használni.

Az adatbevitellel párhuzamosan megkezdődött a hatékony szövegkezelő programok fejlesztése. Ez a folyamat feltehetőleg számos programrendszer elkészítéséből fog állni, s a gyakorlati tapasztalatok alapján döntik majd el, hogy melyeket érdemes leginkább használni. Azok az eredmények, amelyekről beszámolhatok, csupán a kísérletezések részeredményei, távolról sem biztos, hogy a jelenleg meglévő programok közül bármelyiket is fel fogják használni az adatbázis kezelésére. Először is kifejlesztettek egy olyan hatékony szövegkereső programot, amely hatalmas szövegfile-okban[4] képes szavakat, szópárokat, karaktersorozatokat keresni (GONNET 1987). A program egyik legelőnyösebb tulajdonsága, hogy kezelése rendkívül könnyen elsajátítható, még a számítástechnikai tapasztalatokkal nem rendelkezők is egy nap alatt begyakorolják használatát. Alapfunkciója, hogy bármilyen szövegállományból szavakat, vagy szópárokat keressünk, ezek gyakoriságát és előfordulásuk tetszőleges méretű szövegkörnyezetét kiírathassuk. A különböző utasítások kombinációjával például kilistáztathatjuk a *child* kezdetű szavak összes előfordulását, vagy az összes olyat, ahol ez előtt/után előfordul az, hogy *mother* stb. Szűkíthetjük a keresés helyét különböző szempontok szerint, például megadhatjuk, hogy csak az idézetek szövegéből írja ki a keresett szót, vagy csak a legkorábbi előfordulásra vonatkozó idézetből. A programnak számítástechnikai szempontból az a rendkívül vonzó tulajdonsága, hogy a keresés segédeszközéül használt ún. indexfile-ok viszonylag kevés helyet foglalnak el (az eredeti szöveg méretének kb. felét), a keresés rendkívül gyors, mivel az indexben faszerkezetben tárolják az egyes karaktersorozatok előfordulásának kezdőpozícióját. A kereséskor a program a „fából" keresi ki az előfordulás számát és helyét, és csupán olyankor fordul az eredeti szövegfile-hoz, amikor konkordancia sorokat akarunk kinyomtatni. A program további előnye, hogy lehetőséget biztosít a leggyakoribb szavak — rendszerint a formaszavak — keresésből való kizárására, íly módon az indexfile mérete tovább csökkenthető. (Ez nem azt jelenti, hogy a szövegből hagyja ki a formaszavakat, csak azt, hogy nem írja föl az indexfile-ba, mely karakterpozíciókon fordultak elő ezek a szavak.)

[4] A file szó szabványos magyar fordítása: adatállomány, állomány. Használata a számítástechnikában nem terjedt el, nyelvhasználati szempontból is kifogásolható, így inkább az elterjedtebb idegen szakszót használom. Jelentése: logikailag összetartozó adategyüttes, „dosszié".

Rugalmasan kezeli a lehetséges írásjeleket, és megadhatjuk, hogy a kis- és nagybetűket a keresés szempontjából azonosnak tekintse-e.

Ennél intelligensebb, de lényegesen nehézkesebben használható a másik általuk kifejlesztett programnyelv, a GOEDEL. A nyelv célja, hogy a New OED adatbázisában a szótári struktúrát használva biztosítson hatékony keresést. Előnye a fent ismertetett programhoz képest, hogy lehetőséget biztosít a keresett adatok megőrzésére és kinyomtatására, továbbá bonyolult kérdések feltételére. A GOEDEL segítségével változatos táblázatokat állíthatunk össze, például olyat, amely az összes angol képző történeti-statisztikai adatait tartalmazza: mely képző milyen időszakban volt a legtermékenyebb, mikor vált elavulttá stb. Ebben a rendszerben arra is van lehetőség, hogy egyes részeredményeket változókban tároljunk (mondjuk az összes képzett szót), és utána ezen a kiválogatott adatállományon végezzünk különböző műveleteket, esetleg összehasonlítsuk más részeredményekkel stb. A GOEDEL-en belül is használhatjuk a korábban körvonalazott szövegkereső műveleteket (szavak, szópárok szövegkörnyezetének keresése, gyakoriságuk kilistázása), itt azonban összetettebb kérdéseket is feltehetünk. Ez a programnyelv azonban rendkívül bonyolult, a dokumentáció hiánya miatt jelen pillantban csaknem használhatatlan. Mivel a rendszert most fejlesztik, remélhető, hogy belátható időn belül lesz egy hatékony, dokumentált, használható változata. A szótári adatbázissal folytatott kísérletezések során a programfejlesztők arra a következtetésre jutottak, hogy a szöveges adatbázisok alapvetően különböznek az egyéb típusú — pl. üzleti célú, vagy statisztikai nyilvántartásra szolgáló — adatbázisoktól (GONNET–TOMPA 1987). Ha például egy bibliográfiai tételt megpróbálunk relációs adatbáziskezelőben ábrázolni, az adatokat és a közöttük lévő kapcsolatokat megfelelően tudjuk tárolni, csak éppen a szöveg eredeti formáját veszítjük el. Ennek a hiányosságnak a kiküszöbölésére tervezték a nyelvtan által vezérelt szöveg-orientált adatbázismodellt. Amennyiben a bibliográfiai tételt a nyelvtannal elemezzük, egy fát kapunk eredményül, amelynek gyökere a „tétel", levelei pedig a bibliográfia-tétel karakterei. Az így kapott elemzett karaktersorozat az ún. p-string (parsed string), amellyel azután különféle műveletek végezhetők. Most készítik a GOEDEL programnyelvnek egy olyan változatát, amely p-stringekkel dolgozik. A kezdeti eredmények biztatóak, úgy tűnik, ez az új megközelítés jobban megfelel a szöveg-orientált adatbázisok kezelésére, mint a hagyományos relációs adatbáziskezelők. Mindenesetre érdekes kísérletnek tűnik olyan új szemléletű szöveges adatbáziskezelő rendszer kilakítására, amely rendelkezik a relációs adatbáziskezelők előnyös tulajdonságaival, de közben a szöveg megtartja eredeti, kötetlen formáját. Így megoldható, hogy ne kelljen fix rekordméretű és struktúrájú adatbázist létrehozni, a mezők nagyrésze lényegében szabadon kitölthető vagy elhagyható, mégis bármilyen szempont szerint csoportosíthatjuk, válogathatjuk az adatokat, valamint táblázatokat és listákat készíttethetünk.

2.1.5. A Longman szótár

A LONGMAN DICTIONARY OF CONTEMPRORARY ENGLISH (LDCE) (PROCTER, 1978) az első olyan számítógép segítségével készült nyomtatott szótár, amelynél a számítógépet nem pusztán a fényszedéshez használták, hanem a szócikkek kidolgozásához is. Az LDCE legfőbb újítása, hogy az értelmezéseket egy 2000 szónyi alapszókincsállományra

vezeti vissza. A definíciókban csak olyan szavak szerepelhetnek, amelyek vagy ennek az alapszókincsnek elemei, vagy az alapszavak segítségével definiálva vannak. Ez többek közt az angolt idegen nyelvként tanulók számára lehet nagyon hasznos: mivel egynyelvű értelmező szótárat rendszerint nem teljesen kezdők használnak, ha már elsajátították az alapszókincset, viszonylag könnyen eligazodnak a szótárban. Lexikográfiai szempontból is régi és logikus törekvés a körkörös definíciók kiszűrése. Noha egyes lexikográfusok szerint az így készült értelmezések nem feltétlenül egyszerűbbek és használhatóbbak, mint az e megszorítás nélkül készült egynyelvű szótárakéi (HERBST 1986), a számítógépes szótárkészítésben mégis óriási jelentősége van ennek a műnek.

A korlátozott szókincsállomány használata lehetővé tette a körkörös definiciók kiküszöbölését, így a címszavak definíciói jól formalizálhatók. A számítógépes változatot kiegészítették egy szemantikai kódrendszerrel (MICHIELS 1981). A szemantikai kódok egy része hierarchikus (a főneveknek pl. ilyen kódjai lehetnek: +Animal, +Female stb.), más része a címszó használatára vonatkozó információ (az igéknél például a lehetséges alany és tárgy sajátosságait jellemzik). Grammatikai kódrendszerrel is ellátták a címszavakat, ill. a különböző jelentésárnyalatokat (pl. a *claim* egyik jelentésében C5 kódot kap, annak jelzésére, hogy megszámlálható, és *that*-tel kezdődő mellékmondatot vonz, a másik jelentésében C3-at kap, megszámlálható és *to*-val kezdődő infinitivuszt vonz stb.). Az egyes jelentésárnyalatokat a számítógépes verzióban tárgyszavakkal is ellátták (pl. a *hammer* szónak a tőzsdei csőd jelentésárnyalatához a *gazdaság*, *tőzsde* tárgyszavakat rendelték).

Számos projektum használta és használja ki a LDCE számítógépes formájának előnyös tulajdonságait. Tekintettel a korlátozott alapszókincsre és a fent említett kódrenszerre, jó alapanyagnak bizonyult különböző célú szótári adatbázisok elkészítéséhez. A Longman–Liege projektum például az LDCE és a Longman Dictionary of English Idioms (LDEI) (LONG 1979) számítógépes változatát felhasználva alakít ki egy olyan szótári adatbázist, amelyet gépi fordítórendszerek szintakiai-szemantikai elemző-generáló moduljaihoz akarnak majd felhasználni. Tekintettel arra, hogy az LDCE korlátozott definiáló szókincset használ, és a fent röviden jellemzett kódrendszerrel van ellátva, ideális alapanyagnak tűnik a gépi fordítás számára. A kutatócsoport 1981-ben részben a kódrendszer továbbfeljesztésén dolgozott, részben pedig automatikus szintaktikai-szemantikai elemzőt írt az angol nyelvre. Az elemző fő újdonsága, hogy elsősorban a lexikonban lévő információk irányítják az elemzést. A grammatikai kódok alapján az elemző először megjósolja, milyen szintaktikai szerkezet következik; ha azonosítani tudja a keresett szerkezetet, sikeres az elemzés. Az elemző segítségével megvizsgálják a címszavak értelmezéseit, és feljegyzik, hogy a grammatikai kódok és a ténylegesen előfordult szövegkörnyezetek összhangban vannak-e. Ennek segítségével elkészítenek egy annotált szótári adatbázist, amiben szerepel az LDCE-beli homográfkód, az értelmezés száma, a választott grammatikai kód, és a fenti elemzés eredménye. Egy interaktív programmal feljegyzik az anaforákat és az operátorok hatókörét is. Az adatbázis felépítésére és lekérdezésére az IBM STAIRS adatbáziskezelőt használják.

Az IBM Thomas J. Watson Research Center-ben (USA, New York) — részben a fentebb említett W7-re alapuló kutatások folytatásaként — elsősorban az LDCE számítógépes változatára támaszkodva fejlesztenek egy olyan adatbázist, amely természetes

nyelvű rendszerek lexikonjaként fog működni (KLAVANS 1988). Arra törekszenek, hogy a szótárból a rendszerhez szükséges információkat a lehető legnagyobb mértékben automatizálva vonják ki. Hasonló célú kísérleteket végeznek más kutatóközpontokban is (WILKS ÉS TSAI. 1988)

2.2. Új szótárak készítése számítógép segítségével

2.2.1. A Trésor de la langue française

A FRANCIA TRÉSOR talán a legambiciózusabb számítógépes történeti szótári projektum. 1957-ben egy strassbourgi kerekasztal konferencián vetődött fel először, hogy szükség lenne egy modern francia szótárra, amely az OED hírnevével vetekszik. Az új szótár végül is egy szótársorozat lesz, amelynek első része az 1789–1960-ig terjedő időszak szókincsét öleli fel. Ennek elkészülte után kezdik feldolgozni a korábbi évszázadok szókincsét. A projektum 1960–61-ben indult el, amikor a Centre National de la Recherce Scientifique Nancyban létrehozta a szótár előállításával foglalkozó kutatólaboratóriumot. A kutatás vezetésével Paul Imbset bízták meg.

A szótár forrásanyagát számítógép segítségével gyűjtötték össze. Összesen mintegy 90 millió szövegszónyi korpuszt vittek számítógépre, az ebből készült konkordanciák felhasználásával szerkesztik meg a szócikkeket. A szótár első kötete már 1971-ben elkészült, a közeljövőben jelenik meg a 13. kötet. Várhatóan 1991–92-ben készül el a 17., befejező kötet, amelyet majd egy pótkötet és egy kiegészítés követ: a pótkötetben szerepelnek majd az újabb adatok, a kiegészítő kötetben pedig a módosítások, javítások.

A forrásanyag kiválasztása és számítógépre vitele

Mivel a szótár elsősorban a művelt irodalmi nyelv szókincsét kívánja leírni, a forrásokat kizárólag híres írók jelentős műveiből válogatták ki. Arra törekedtek, hogy a kiválasztás minél objektívebb legyen, amit úgy próbáltak elérni, hogy a legfontosabb irodalomtörténeti kézikönyvek alapján listát készítettek azokról a művekről, amelyeket XVII–XIX. századi művek esetén legalább 4, XX. századi művek esetén legalább 2 kézikönyvben megemlítettek. Az így összeállított forrásjegyzéket elküldték a megfelelő korszakok szakértőinek, akik elbírálták őket. (Részlet egy szakértői bírálatból, Sartre neve mellett: „Sartre! Valóban???".) Természetesen ez a viszonylag objektív lista sem nélkülözi a szubjektív előítéleteket, így aztán a Trésor bírálói között akadnak olyanok, akik a felhasznált forrásanyag hiányosságai, esetleges aránytalanságai miatt tartják használhatatlannak a szótárat.

A kiválasztott műveket teljes egészében számítógépre vitték (vagy a legelső, vagy a kritikai kiadást használták fel), az összes rögzített szöveget megőrizték a könyvtárban. A rögzítést az előszerkesztést követően kezdik meg. Az előszerkesztés során speciális kódokat írnak a címlapra és az oldalak végére. Színművek esetén bejelölik a felvonások, jelenetek határát, megkülönböztetik az egyes szereplők megszólalásait, és a színpadi utalásokat. Kihagyják a szövegből a hosszú idézeteket, a tulajdonneveket pedig egy csillaggal jelölik meg.

A szépirodalmi anyag összesen mintegy 70 millió szövegszó terjedelmű, amely 1000 kötetből származik. Ezen kívül szakirodalmi szövegrészleteket is hozzáadtak a korpuszhoz. Ezeket meglehetősen ad hoc módon válogatták össze: a laboratórium dolgozói saját hobbi-könyvtárukból jelöltek meg érdekesnek ígérkező részleteket, amelyeket gépre rögzítettek. Ez az anyagrész mintegy 20 millió szövegszóból áll, körülbelül 500 különböző műrészletből származik.

A szöveges korpuszon kívül számítógépre vitték az *összes* egynyelvű francia szótár címszóállományát is, így számítógépes kimutatást tudnak készíteni arról, mely szótárakban szerepel egy adott címszó.

Az adatokat lyukkártyára vitték, a rögzítés csaknem tíz évig tartott, 38 adatrögzítő dolgozott rajta egyidejűleg! (Ami azt jelenti, hogy egy rögzítőjük teljesítménye kb. felekétharmada a mieinknek, miközben teljesítményarányos fizetésük pontosan annyi frank, ahány forint a mieinké.)

A számítógépes korpuszon túl több cédulagyűjteményt is felhasználnak: mintegy 6 millió cédulát örököltek Mario Rognes-től (Inventoire Général de la Langue Française), és további cédulákat gyűjtenek a hiányosnak tűnő szócikkekhez. Felhasználják a neologizmusok adatbázisát is, amelyet hagyományos módszerrel folyamatosan bővítenek. Az adatokat elsősorban folyóiratokból, magazinokból gyűjtik: kijelölik a releváns szövegkörnyezetet és csak azt viszik fel számítógépre — természetesen a forrás pontos megjelölésével.

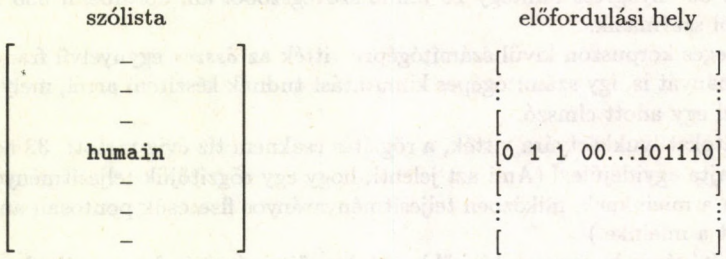
A teljes korpusz rögzítése után félautomatikusan szófaji kategóriákba sorolták a szövegszavakat, azonban a homográfokat így nem sikerült szétválasztaniuk. Ezért a szófaji besorolás körülbelül 70%-ban helyes. A szövegszavakat nem lemmatizálták ugyan, az igékre azonban készítettek egy olyan eljárást, amellyel az összes lehetséges toldalékolt forma kikereshető. Az eljárás lényege, hogy a főnévi igenevekből előállították a teljes igei paradigmákat. Ezeket egy adatbázisban tárolják, amelyből lekérdezhetjük az egyes formákat. Ezután valamennyi toldalékolt alak konkordanciáját ki tudjuk íratni.

A konkordanciákat nem pusztán ábécérendben írják ki, hanem a kulcsszót közvetlenül megelőző illetőleg követő szavak grammatikai kategóriái szerint csoportosítva. (pl. ha a *szeret* lexéma összes előfordulásának konkordancialistáját nézzük, először láthatjuk azt a csoportot, ahol utána közvetlenül főnév szerepel a szövegben, utána azt, ahol melléknév stb. Majd az összes konkordanciasor megismétlődik, a szó előtt lévő szó szófaji kategóriája szerint csoportosítva.) A csoportokon belül az adatok ábécérendben követik egymást. Így a szócikkíró két azonos tartalmú, de különböző rendezettségű konkordancialista alapján dolgozik.

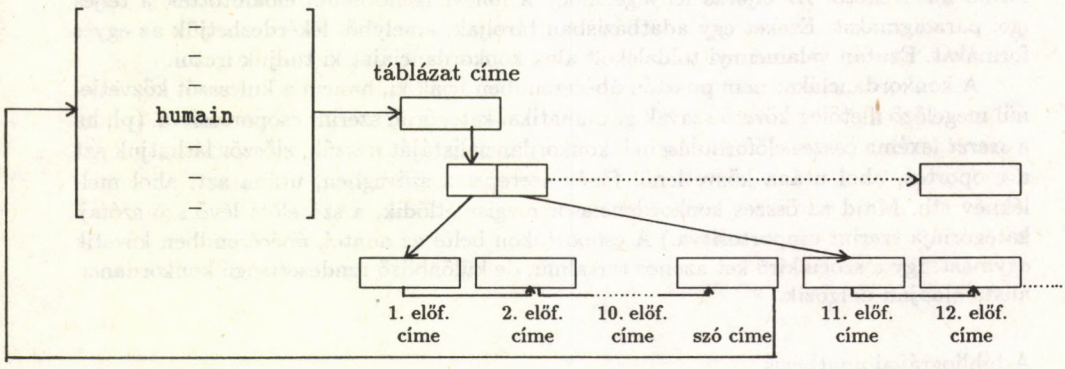
A bibliográfiai adatbázis

A kutatóközpont egyik fő feladata a francia lexikográfia és lexikológia kimerítő bibliográfiájának elkészítése és folyamatos karbantartása. A bibliográfia célja a szócikkszerkesztők munkájának megkönnyítése volt, ezért minden címszó teljes bibliográfiáját összegyűjtötték. Így nemcsak a szócikkírók munkáját könnyítették meg, de lehetővé vált, hogy magában a szócikkben is feltüntessék azokat a műveket, amelyek a szóról korábban íródtak.

Kezdetben a bibliográfiát csak cédulákra gyűjtötték, majd a katalógus anyagát elkezdték nyomtatott formában is kiadni. 1969 óta negyedévenként jelenik meg a *Bulletin*



1. ábra
durva index



2. ábra
finom index

Analitique de Linguistique Française, amely felöleli az aktuális időszak francia nyelvészeti szakirodalmának bibliográfiai adatait, tárgykör, címszó valamint szerző szerint csoportosítva. A folyóirat anyaga számítógépen is megtalálható, és különböző szempontok szerint visszakereshető.

A szócikkírás

A szócikkek készítésekor így háromféle adatra támaszkodhatnak a szerzők: a korábbi szótárakra, a szóról előzőleg írt tanulmányokra és végül, de nem utolsósorban a számítógépes korpuszból készült konkordanciákra. Minden szócikket két lexikográfus ír: az egyik az értelmezést, jelentésekre bontást készíti, és kiválogatja a megfelelő idézeteket; a másik a történeti-etimológiai rész szerzője. A lexikográfus minden szócikkről kap egy dossziét, amelynek első lapja egy számítógéppel készített táblázat a szó legfontosabb statisztikai adatairól: hányszor fordul elő összesen a korpuszban, ebből mennyi a szépirodalmi és szakirodalmi előfordulások száma, mely értelmező szótárakban található meg, és hány szakszótárban szerepel az adott szó. Az előfordulás száma és az eloszlás alapján a számítógép megadja, hogy legfeljebb hány idézetet célszerű kiválogatni (ez csak irányszám, ennél rendszerint kevesebbet használnak fel). A következő oldalak tartalmazzák a címszó teljes bibliográfiáját, majd részletek következnek a neologizmusok gyűjteményéből. Ezután az összes értelmező és szakszótár megfelelő szócikkeinek fénymásolatait találhatjuk meg a feldolgozandó szó különböző adatait tartalmazó dossziéban. A lexikográfus mindezek tüzetes tanulmányozásával kezdi meg a munkát. (Ezek átolvasása egyszerűbb szavak esetén 1–2 napot vesz igénybe.) Elsősorban az értelmező szótárak szócikkeit használja fel, a szakszótárak anyagát csak átfutja. Ezután lát hozzá a konkordanciák tanulmányozásához. Rendszerint minden egyes idézetet elolvas, és megjelöli azt a néhányat, amely érdekesnek ígérkezik. Arra törekszik, hogy minden jelentést több korszakból származó idézettel illusztráljon. Mivel a teljes konkordancialista egysoros, ennek alapján nem dönthető el, hogy megfelelő idézetet talált-e. A jónak ígérkezőket nagyobb szövegkörnyezettel cédulákra írattatja (8-8 sor a szó előtt és után), majd a cédulákból válogatja ki a végleges idézeteket. Amennyiben úgy érzi, nem talált elegendő vagy megfelelő adatot a konkordanciában, megnézi a cédulagyűjteményt is, és ha még mindig elégedetlen, ő maga is kereshet idézeteket hagyományos módon (egyszerűen olvasva az irodalmat).

A történeti rész írói elsősorban a régebbi és a történeti szótárak anyagára, illetve a szótörténeti tanulmányokra támaszkodnak. Természetesen a konkordancia nekik is segítséget jelent, érdekes módon azonban ők is ugyanazokat a konkordancialistákat kapják meg, amelyek nem a keletkezés évére, hanem a környező szavak grammatikai kategóriáira vannak rendezve. (Mindent összevéve úgy érzem, a projektum nem használja ki kellő mértékben a számítógép nyújtotta lehetőségeket. Ennek az is oka lehet, hogy csupán 1985 óta van on-line formában a korpusz, addig csak papíron írhatták ki a konkordanciákat, meglehetősen nehézkesen. Mivel ekkor már több mint 15 éve folyt a szótárírás, addigra kialakultak a kevéssé rugalmas környezethez alkalmazkodó munkamódszerek.) A fent vázolt munkamódszerrel kb. 1 hét alatt készül el egy egyszerűbb szócikk, a bonyolultabbak azonban több hónapi munkát is igényelhetnek. Ebben az ütemben a központ 40 lexikográfusa körülbelül 25 év alatt tudja megírni a szótárat. (Eszerint úgy becsül-

hetjük, hogy az utóbbi két évszázad szókincsét felölelő szótár kb. 1000 „lexikográfus-év" munkával készíthető el, számítógépes korpuszt használva. Azaz, ha a Trésorhoz hasonlóan először csak a magyar nyelvújítástól napjainkig terjedő kor szótárát készítenénk el, akkor is kb. 25 lexikográfus 40 évi munkájára lenne szükség ennek befejezéséhez!)

A FRANTEXT

Amikor gépre vitték a Trésor alapanyagául szolgáló 90 millió szövegszónyi korpuszt, nem hagyták abba a forrásanyagok gyűjtését, hanem tovább bővítették a XVII–XVIII. századból származó szövegekkel, illetve a neologizmusok gyűjteményével. Ílymódon ma már mintegy 160 millió szövegszót tárolnak számítógépen, a jelenlegi szótárhoz azonban csak az eredeti 90 milliót és a neologizmusok gyűjteményét használják fel. Az 1789 előtti szövegek majd a következő szótárhoz szolgálnak alapanyagul, az érdeklődő kutatók azonban már most is használhatják on-line számítógépes archívum formájában.

A korpusz lekérdezésére szolgáló programrendszer, az ún. STELLA (Systéme de Textes en Ligne en Libre Accés) 1985-ben készült el. Interaktív, könnyen kezelhető, hatékony program, amellyel elfogadható idő alatt (max. néhány perc) kikereshetjük, hogy az adott szó hány műben fordul elő a teljes korpuszban, majd kiszámíttathatjuk a teljes előfordulási számot. Kérésre kiírja a képernyőre a konkordanciákat is, vagy ezeknek egy részét. Lehetőségünk van arra is, hogy több szó együttes előfordulását keressük, vagy csak bizonyos művekből írassuk ki a szót (pl. Zola műveiből, vagy az 1870-ben írt művekből stb.). A szavakra vonatkozó információkon kívül számos speciális kérdésre kaphatunk választ: pl. „Ki beszél másodszor a Tartuffe II. felvonásának első jelenetében?", „Ki mondja először azt a szót, hogy *aime*?" stb.

A lekérdező program mindössze néhány egyszerű függvény felhasználásával válaszol a legkülönbözőbb kérdésekre. A szövegeket művenként 1-1 file-ban tárolják. Minden 48 fileból készítenek egy „csomagot", amelyekhez egy-egy indexfile tartozik. Ez a durva index, amelyben megtalálható az összes szövegszó, amely a „csomagban" előfordult, a szavak mellett lévő vektor pedig megmutatja, hogy a csomag hányadik művében fordult elő a szó. Amikor megkérdezzük, hogy egy szó hány műben fordult elő, a program először a durva indexeken szalad végig, és a vektorok alapján összeszámolja, hány műben fordult elő a szó.(Az előfordulásszámot azonban még nem tudja megmondani.) Ha tovább vizsgáljuk a szót, a program egyenként „belemegy" azokba a szövegfile-okba, amelyekben a durva index szerint a szó megtalálható. Minden szövegfile-hoz tartozik egy finom index, amely a szövegszavak összes előfordulásának helyét tárolja. A finom index végeredményben egy sajátos pointer-lánc, amely először megmutatja, hol találjuk az első előfordulást, majd innen mutat a következőre, és így tovább. Ahhoz azonban, hogy ne csak az előfordulást tudjuk megkeresni, hanem azt is meg lehessen mondani, hogy pl. a szövegfile 137. eleme milyen szó, vissza is kell mutatniuk a pointereknek a szóra. Ezért a pointerlánc utolsó eleme mindig a szövegszóra mutat. Mivel azonban ez azt jelentené, hogy egy bizonyos előfordulási hely visszakeresésekor végig kellene mennie a programnak a teljes láncon ahhoz, hogy visszaérjen a szövegszóhoz, a pointereket 10-esével csoportosították. Minden 10-es csoport utolsó eleme a szövegszóra mutat, a 10-es csoportok elejére pedig egy közbeiktatott táblázat mutat. (Lásd 2. ábra)

Így például tegyük fel, hogy a *humain* szó 53. előfordulását akarjuk kiíratni egy szövegből. Ha nem lennének 10-esével csoportosítva a pointerek, egyesével végig kellene mennünk a pointer láncon, míg megtaláljuk a keresett szót. Így viszont az 51. elemnél kezdhetjük a keresést.

A program által használt alapfüggvények:

Legyen

$r_n \in \Re$	a rang (az adott szó sorszáma a szövegen belül)
$s_1 \ldots s_n, f$	a szöveg szavai az írásjeleket, kódokat stb. beleértve
$clair(i)$	az i-edik szó
$prem(f)$	az f szó első előfordulása
$suiv(f)$	az f szó következő előfordulása
$vgauche(f,r)$	hol fordul elő az f szó az r-edik helytől balra
$vdroite(f,r)$	hol fordul elő az f szó az r-edik helytől jobbra
$nba(f)$	az f szó előfordulási száma

E néhány függvény segítségével a program gyakorlatilag bármely kérdésre választ tud adni.

Pl: „Írd ki az összes olyan szövegkörnyezetet, ahol a *férfi* és a *nő* egyszerre fordul elő, legfeljebb 10 szónyira egymástól, 10-10 szavas szövegkörnyezettel!"

$x = prem(férfi)$
csináld, amíg $suiv(férfi) \neq 0$:
$z = vdroit(nő, x)$; $y = vgauche(nő, x)$
ha $|x - z| \leq 10$ vagy $|x - y| \leq 10$ akkor
$clair(x-10)clair(x-9)\ldots clair(x)clair(x+1)\ldots clair(x+10)$
$x = suiv(férfi)$

Az algoritmus gyorsítása érdekében érdemes az első lépésben megvizsgálni, melyik szó a kevésbé gyakori, és azzal kezdeni a keresést. Ezzel a módszerrel természetesen akárhány szó illetve elem együttes előfordulását kereshetjük, gyakorlatilag bármekkora szövegkörnyezetben.

Az elkészült kötetek szócikkeinek szerkezete jól áttekinthető. Míg az OED-ben a jelentésekre bontás bizonyos esetekben rendkívül bonyolult, itt jól elkülönülnek az összefüggő és távolabb eső aljelentések. Történeti szótárban újdonság, hogy a címszavak után az előfordulás gyakorisága is szerepel, ez a melléktermékként előállított gyakorisági szótárnak köszönhető. Az értelmezések rendkívül rövidek, csak a legfontosabb információkat tartalmazzák, az idézetek hivatottak a jelentés pontosabb megvilágítására. Az idézeteket jelentésárnyalatonként mindig szigorúan a forrás időrendjében közlik. Egy-egy jelentéshez 5–10 idézetet válogattak ki az archívumból. A szó stilisztikai értékét, használati körét is megjelölik. A szótár normatív. Külön tárgyalják a címszó részletesebb történetét, és külön a szorosan vett etimológiát. A szócikkek végén felsoroják azoknak a munkáknak a bibliográfiai adatait, amelyek az adott címszóval kapcsolatban készültek.

2.2.2. *Dictionary of Old English*

A DICTIONARY OF OLD ENGLISH (DOE) adatgyűjtését 1970-ben kezdték meg a torontói egyetemen. A szótár első főszerkesztője Angus Cameron volt, számítástechnikai tanácsadója Richard Venezky.

A DOE alapanyaga az összes 750 és 1200 között keletkezett és fennmaradt angol kézirat. Első lépésben fénymásolat formájában összegyűjtötték az összes forrásanyagot, az eredeti és publikált verziókat egyaránt. (Az anyaggyűjtés 5 évet vett igénybe.) A teljes korpusz mintegy 3 millió szövegszó terjedelmű, az egészet számítógépre vitték (AMOS 1984).

A számítógépre vitelt, megfelelő hardver hiányában, elképesztően bonyolultan oldották meg. Először egy gépírónő legépelte az összes szöveget olyan speciális írógépen, amelynek karaktereit a rendelkezésükre álló optikai olvasó le tudta olvasni. Ezt az ellenőrzés és javítás után olvasta be az optikai olvasó, természetesen újabb hibákat téve a szövegbe. Mivel akkoriban a munkacsoportnak nem volt saját számítógépe, a mágnesszalagon tárolt szöveget kinyomtatták, ellenőrizték, visszaküldték javításra Madisonba, újra kinyomtatták stb.

Az adatgyűjtéssel párhuzamosan készültek a feldolgozó programok (VENEZKY 1971). A LEXICO elnevezésű programrendszer a konkordancia készítést támogatta. Ehhez először is lemmatizálni kellett a teljes szöveget: a program lehetővé tette az interaktív, nem automatikus lemmatizálást. A lemmatizálással párhuzamosan az idézeteket besorolták az egyes címszavak alá, és a géppel cédulákat készítettek. (Erre azért volt szükségük, mert hardver hiányában a lexikográfusok nem tudtak közvetlenül a számítógépen dolgozni.)

Amint elkészültek a forrásanyag felvitelével, a teljes konkordanciát kiadták mikrofilmen. A teljes konkordancia kétsoros, lexémára rendezett. Külön táblázat tartalmazza a címszavak listáját és indexét, továbbá a gyakoriságra vonatkozó adatokat. A konkordancia és a gép által készített cédulák felhasználásával készítik a szócikkeket. Eddig az A és C betű anyagát tudták kiadni, 1987 tavaszán kezdték meg a B betű szerkesztését.

Az utóbbi években saját számítógépet kaptak: egy VAX 11/730 minigépen dolgoznak, két 121 Mbyte-os[5] lemezen tárolják a szótár anyagát. XEROX Dandelion munkaállomásokat csatlakoztattak hozzá, amelynek nagy grafikus képernyője, saját memóriája (1,5 Mbyte), és 43 Mbyte-os fix lemeze van. Jelenleg fejlesztik azokat a programokat, amelyek az on-line (azaz közvetlenül a gépen történő) szócikkírást támogatják; a program kikeresi a konkordanciaállományból a kijelölt azonosítójú idézetet, és bemásolja a szócikkek megfelelő részébe, ezen kívül segít a nyomtatási formátum kialakításában. Mivel a képernyő grafikus, a teljes karakterkészlet megjeleníthető rajta: a szócikket úgy látjuk, ahogy a nyomtatott szótárban meg fog jelenni. A szócikkeket azután lézer printeren nyomtatják ki, a szerkesztők ellenőrzik és javítják.

---

[5] Byte: a legkisebb címezhető egység. Az egyszerűség kedvéért úgy képzelhetjük el, hogy minden egyes karakter (betű vagy számjegy) 1 byte-ot foglal el a számítógépben. 1 kilobyte=1024 byte, 1 Megabyte (Mbyte)=1024×1024 byte≈1 millió byte, azaz 1 millió karakter, 1 Gigabyte (Gbyte)=1024 Mbyte≈1000 millió karakter.

Csak most kísérleteznek azzal, hogy a lexikográfusok számítógépen szerkesszék a szócikkeket (jelenleg papírra írnak mindent, a szócikkeket gépírónő rögzíti). Olyan programot készítenek, amellyel a címszóhoz tartozó összes konkordancia kikereshető és a képernyőre íratható, vagy válogatások ill. minták készíthetők az idézetekből. Becslésük szerint a szócikkírás várhatóan újabb 15 évet fog igénybe venni. Tapasztalataik szerint a számítógép segítségével is azoknak a szócikkeknek a kidolgozása nehéz, amelyre túl sok vagy túl kevés adat van.

2.2.3. A COBUILD szótár

A BRIT COBUILD az egyik leglendületesebb számítógépes szótári projektum eredménye. Alapanyaga az ún. COBUILD korpusz, amelyet a birminghami egyetem angol tanszékének irányításával gyűjtöttek számítógépre. A korpusz jelenleg is folyamatosan nő, 1984-ben mintegy 12 millió szövegszót tartalmazott. Túlnyomórészt írott, kisebb részben beszélt nyelvi szövegekből áll. A projektum célja kezdettől fogva kettős: a korpuszból generált konkordanciák segítségével szerkesztették meg a nyomtatott szótárat, ugyanakkor a teljes korpusz on-line formában folyamatosan a kutatók rendelkezésére áll, a legkülönbözőbb elemzések céljára.

Mivel a nyomtatott szótárat elsősorban nem angol anyanyelvűeknek szánták, a forrásokat úgy válogatták össze, hogy a mai standard angol szókincset jól reprezentálja. Csak 1970 után megjelent műveket vittek fel, kivételként, nagyhatású művek esetén valamivel korábban megjelent könyveket is (pl. Golding *Lord of the Flies* című regényét). Mindig a teljes művet dolgozzák föl. A szövegszavak kb. 65–70%-a a brit, 25–30%-a az amerikai és 5%-a az egyéb nyelvváltozatokat reperezentálja. Túlnyomórészt szépprózai, részben egyéb műfajokból válogattak (hetilapok, magazinok stb.). A beszélt nyelvi szövegek elsősorban interjúk átiratai.

1981-ben készítettek először a lexikográfusok számára egy minta konkordanciát az addig rögzített mintegy 6 millió szövegszónyi anyagból. A konkordanciát papíron kinyomtatva archiválták és mikrofilmen kiadták. Ezt az anyagot később kiegészítették a további gyűjtésekből származó konkordanciákkal. A teljes konkordanciából mindig kihagyják a leggyakoribb 50 szót — mivel ez az anyag kb. egyharmadát lefedi —, és ezekből külön válogatott konkordanciát készítenek (nem az összes előfordulásukat írják ki, csak a szócikkészítéshez szükséges mennyiséget). Minthogy időközben a projektum kapott egy nagyobb számítógépet (az első konkordanciákat egy ICL 1906A gépen a COCOA nevű konkordanciaprogrammal állították elő, meglehetősen nehézkesen), lehetővé vált a konkordanciák interaktív lekérdezése is. Az interaktív konkordanciaprogramban megadható, hogy milyen méretű szövegkörnyezetet szeretnénk kiíratni, s hogy legfeljebb hány előfordulást írjon ki a gép. Ezzel a programmal nemcsak egyes szavak, hanem több szó együttes előfordulása is kiíratható. Így a lexikográfusok dolgozhatnak közvetlenül a képernyőn is — lekérdezve az éppen szerkesztés alatt álló szócikkhez tartozó adatokat — de válogathatnak akár a mikrofilmen akár a papíron tárolt konkordancia példaanyagából is.

Az így készített szótár (SINCLAIR 1987) bizonyos szempontból a jelenleg leghasználhatóbb angol értelmező kéziszótár. Az értelmezések — noha (vagy éppen mert) nem korlátozott szókincs felhasználásával íródtak — rendkívül egyszerűek, érthetőek, tömö-

rek. Az értelmezésekben mindig szerepel az értelemezendő címszó is, így rögtön látunk egy példát az adott jelentés legjellemzőbb vonzataira. Noha az értelmezéseket természetesen lexikográfusok írják, a számítógép gondoskodik arról, hogy ezek szerkezete egységes legyen: felhasználva a megadott kódokat (pl. főnév, megszámlálható) a szócikkíró által javasolt értelmezéshez a program automatikusan olyan főmondatot generál, amelyben a helyes használatára utaló formában szerepel az adott címszó. (pl. a tranzitív igék mögé egy megfelelő tárgyat tesz.) Minden jelentésárnyalatot legalább egy, de rendszerint több példával illusztrálnak, amelyeket természetesen a számítógépes korpuszból választottak ki. Sajnálatos módon az idézeteknél forrásmegjelölés nem szerepel. Míg az értelmezés nyelvezete egyszerű, közérthető, az idézetek valamivel nehezebbek, tekintve, hogy valódi irodalmi szövegekből választották ki őket. A szócikkek margóján megjegyzésként láthatók az adott jelentés legjellemzőbb szintaktikai környezetei kódolt formában, illetőleg az esetleges szinonimák, antonimák stb. A szótárban használt valamennyi kód magyarázata megtalálható a szócikkek között, kiemelve. A számítógépes korpuszból vett adatok érdekes hatással voltak a szócikkek szerkezetére. Noha a szerkesztők arra törekedtek, hogy az első helyen általában a konkrét jelentés szerepeljen, pl. a *long* szócikk esetében első helyen a *long time* és az ezzel kapcsolatos kifejezések szerepelnek, és csupán 11. jelentésként van felsorolva a térbeli hosszúság, amely pedig nyilván konkrétabb, mint az időtartam hosszúsága. (Egyébként az összes korábbi szótár a térbeli hosszúságot tekinti első jelentésnek.) Ennek az elrendezésnek az az oka, hogy az időbeli hosszúságra vonatkozó *long* nyilván elsöprő gyakorisággal szerepelt a korpuszban. Az pedig rendkívül használhatóvá tesz egy szótárat, ha a jelentésárnyalatok — legalább hozzávetőlegesen — előfordulási gyakoriságuk sorrendjében következnek egymás után. Még érdekesebb lenne a szótárat forgatni, ha a címszavak előfordulásának számát is feltüntették volna, ami számítógépes korpuszból készített szótáraknál általában bevett gyakorlat.

2.2.4. New York Times Everyday Dictionary.

A NEW YORK TIMES EVERYDAY DICTIONARYT 1982-ben adták ki. A szótár címszóanyagát a New York Times fényszedőszalagjairól készített konkordanciákból állították össze (PAIKEDAY). A konkordanciákat a mikroszámítógépek hőskorában készítették, egy icipici számítógépen. (TRS-80 Model I. A gépnek 48K memóriája volt, óra frekvenciája 2 MHz!)

A Lexicon elnevezésű program először is összefűzte a szövegállományokat egy logikai állományba. Az összefűzött szövegeken különféle kereséseket lehet végrehajtani: a `find` parancs hatására a program egy szó teljes képernyőnyi szövegkörnyezetét, a `phrase` hatására szavak–karakterek együttes előfordulását írja ki. (Lényegileg csupán egy átlagos szövegszerkesztő képességeivel rendelkezik.) Ezen kívül lehetőség van a szövegben talált szövegszavak mennyiségének kiíratására, továbbá az egyes szavak egysoros konkordanciáinak előállítására.

Hősi erőfeszítésnek tűnik ilyen kis számítógépen szótári programokat fejleszteni, a szerző azonban, úgy tűnik, sosem alkalmazta 200 000 szövegszónál nagyobb anyagra a programot, ami egy szótár készítésénél meglehetősen kis adatmennyiségnek tűnik. Mindenesetre úgy látszik, hasznos volt a számítógép a címszóállomány kiválasztásában és

a jelentések definiálásában, de azért ne higgyünk a szerzőnek, aki szerint a Merriam-Webster vagy az OED sok milliós cédulaállománya mikroszámítógépre tehető és azon hatékonyan kezelhető. (Ezen a gépen a 200 000 szónyi anyagban egy szó kikeresése átlagosan 8 perc volt, ugyanez a New OED SUN gépén néhány század másodperc.)

2.2.5. Német lexikográfiai adatbázis

A MANNHEIMI Institut für deutsche Sprache számítógépes nyelvészeti osztálya elhatározta, hogy létrehoznak egy olyan német szótári adatbázist, amely folyamatos szövegmintákból készül, de az ezekből kapott adatokat összevetik a korábbi szótárak szócikkállományával (TEUBERT 1984).

Első lépésként összegyűjtötték az összes számítógépes adathordozón lévő német nyelvű korpuszt, amelyek természetesen mind különböző konvenciók szerint voltak rögzítve, ezért ezeket gépi úton egyformára alakították. Mintegy 7 millió szövegszónyi korpuszt sikerült így összeállítaniuk. Ezt optikai olvasó segítségével további szöveges korpusszal fogják kiegészíteni.

Ahhoz, hogy a meglévő szótárakkal összehasonlíthassák a szövegekből kapott címszóállományt, egy szótári adatbankot is létre kell hozniuk, amit elsősorban a Bonnlex Kumuliertes Lexikon szótári adatbázis felhasználásával remélnek megoldani.

A Bonnlex (BRUSTKERN 1981, 1982) szótári adatbázist a bonni egyetem Institut for Kommunikationsforschung und Phonetik intézetében hozták létre, abból a célból, hogy az összes német számítógépes szótárat egy helyen, azonos formátumban tárolják. Tizenegy, más-más célra készült számítógépes szótárat fésültek össze egy adatbázisba, ezért a bennük található információ tartalma és struktúrája igen sokféle. Miután felmérték, melyek azok az információk, amelyek minden szótárban szükségesek, megtervezték az új adatbázis szerkezetetét és az előállításához szükséges szoftvereket. Az így létrejövő szótár elsődleges funkciója a természetes nyelvű interfészek és fordítórendszerek kiszolgálása lesz, de azt remélik, hogy az itt tárolt adatokat a lexikográfusok is fel tudják majd használni. A központi szótár lehetővé teszi, hogy ne kelljen minden projektumnak elölről kezdenie a szótárkészítést, elég, ha kiegészítik a meglévő alap-szótárat azokkal a specifikus információkkal, amelyekre az adott kutatásnak szüksége lehet. Az egyes szócikkek általános szerkezete formálisan:
A szótár:

$$W = LE_1, LE_2 \ldots LE_m$$

szótári címszavak halmaza, ahol LE_i szótári címszó. Minden szótári címszó n információt tartalmaz:

$$LE_i = (I_{i,1}, I_{i,2}, I_{i,3}, \ldots I_{i,n}); I_{i,j} K_j$$

Minden $I_{i,j}$ információ egy lexikológiai információ osztályba tartozik, ezeket az osztályokat így definiálták:

K_1: a címszó grafikus reprezentációja, az írásváltozatokkal együtt
K_2: fonetika, fonológia, szótagolás, hangsúly
K_3: szófaj
K_4: morfológia: információ a ragozásról, képzésről stb., a szótő megadása
K_5: szintaktikai információk: vonzatok, lehetséges szintaktikai szerkezetek, mélyszerkezetek stb.
K_6: szemantikai információk: szemantikai primitívek, szemantikai mezők, releváns szövegkörnyezetek, jelentés definició
K_7: pragmatikai megjegyzések: stílusérték, használat stb.

Az így előállított szótári és szöveges adatbázisokat felhasználva, a lexikográfusok speciális szótárszerkesztői munkaállomások segítségével írják az új szócikkeket. Minden szócikket külön állományban helyeznek el, ezeket módosítják, javítják stb. Külön szoftver segíti az egységes stílusú szócikkírást.

2.3. A történeti szótárak és a számítógép

ANNAK érdekében, hogy reálisan meg tudjuk becsülni, mennyit segíthet a számítógép az újonnan készülő történeti szótárak szerkesztésében, szükségesnek láttam külön kiemelni a számítógépes történeti szótárkészítés eddigi tapasztalatait. Amikor ugyanis a 60-as évek végén, a 70-es évek elején az első számítógéppel támogatott szótári projektumok megindultak (a francia Trésor, a DOE), a főszerkesztők rendkívül optimisták voltak: úgy gondolták, a számítógépes gyűjtés töredékére csökkentheti a szótárkészítésre fordítandó időt. A DOE-t 15 év alatt remélték befejezni — ezzel szemben 15 év után kezdődött meg a tényleges szócikkírás, és most úgy gondolják, jó, ha újabb 15 év elég lesz a szótár elkészítéséhez. A Trésor főszerkesztője 1971-ben úgy nyilatkozott, hogy várhatóan 6–7 éven belül a teljes szótárat kiadják — a máig eltelt 16 év alatt kb. a szótár felét-kétharmadát sikerült kiadniuk.

Tekintélyes lexikográfusok is azt állították, hogy a számítógép forradalmasítani fogja a történeti szótárkészítést. Zgusta szerint: „It is also quite possible that large academic dictionaries will not be published any more. The point is that even the academic dictionaries which consist of ten, twenty, or any number of volumes, do not and cannot present the whole material contained in the archive... then why publish a twenty-volume reduction of the material if a one, two or four-volume reduction could suffice for the first information, which must be eventually followed by the archive search, in any case?" (ZGUSTA 1971).[6] Aitken is ugyanerre a következtetésre jutott: „the existence of computer archives would often seem to remove the need to burden library shelves with still larger dictionaries filled with still more detailed information of interest to only a few people" (AITKEN 1971).[7]

A gyakorlat azonban nem igazolja ezt a derűlátást. Egyrészt bármilyen hasznos is egy nagy számítógépen tárolt szövegarchívum, nem helyettesíti a lexikográfusok által

[6] pp. 354–355
[7] p. 16

kiválogatott, jelentésárnyalatok szerint rendezett szócikkeket. A lexikográfusok több száz idézetből válogatják ki azt a néhányat, amely véleményük szerint legjobban reprezentálja az adott jelentésárnyalatot, s az átlagos szótárfelhasználó számára sokkal hasznosabb az így készített összefoglalás, mint az archívumból kiíratható, esetleg több ezer sorból is álló konkordancia. Amsler is azt írja: „Simply obtaining a multimegabyte set of words by grabbing all the machine-readable sources available and merging them together hardly provides the basis for scientfic observation of the nature of the language as a whole. Thus ten million words of newspaper stories can be less useful than one million words of *carefully sampled* text taken from a variety of sources" (AMSLER 1982).[8] Ha pedig a több ezer konkordanciából véletlenszerűen íratunk ki néhányat, semmi biztosíték sem lesz arra, hogy az összes jelentésre kapunk példát, a jelentésárnyalatok kialakulása pedig végképp nem követhető a puszta konkordanciákból. Legcélszerűbbnek az látszik, hogy párhuzamosan hozzáférhetővé tegyük a szótárkészítésre használt archívumot és a szótári adatbázist egyaránt.

Ami a számítógéppel segített adatgyűjtést illeti, mint a Trésor és a DOE példáján láthatjuk, nem számíthatunk arra, hogy a számítógép alkalmazása önmagában lényegesen gyorsabbá teszi a szótárírást. (Tekintve, hogy az OED 44 év alatt készült el, és csaknem egy évezred szókincsét öleli fel, a Trésor és a DOE pedig mintegy két évszázad szókincsének leírására vállalkozik, és mindkettő várhatóan kb. 30 év alatt fog elkészülni.) A gyorsaság és eredményesség inkább a számítógépen kívüli tényezők függvénye marad továbbra is: kiforrott koncepció már az adatgyűjtés időszakában, határozott főszerkesztő, aki tűzön-vízen át harcol a koncepció következetes megvalósításáért, megfelelő pénzügyi háttér és munkaerő. Mindezekre a hagyományos szótárkészítésnel is szükség volt, ezen túl azonban, a számítógépes szótárhoz még kellő méretű és minőségű számítógéppel és számítógépes szakemberekkel is rendelkezni kell. Igaz ugyan, hogy ha rögzítettük a korpuszt, a gép nagyon gyorsan tud szólistát készíteni, ábécébe rendezni, konkordanciát listázni. A baj éppen az, hogy „túl jól" csinálja (OAKMAN 1980).[9] Ha nagy a korpuszunk, bizonyos szavakról kétségbeejtő mennyiségű idézetünk lesz, ezeket egyenként megvizsgálni, eldönteni, melyiket érdemes felhasználni, időnként csaknem lehetetlen. Ez a művelet gyorsítható ugyan azzal, ha a gyakori szavakról csak bizonyos számú véletlenszerűen kiválasztott mintát kérünk, ilyenkor azonban nem biztos, hogy a kapott mintában még mindig benne lesz a szó összes jelentésárnyalata. Lehetséges ugyan, hogy lassan kialakul egy új, kifejezetten számítógépes szótáríró módszer, ami talán valóban gyorsítani fogja a szócikkírást, a valószínűbb azonban az, hogy változatlanul több évtizedig fog készülni egy-egy történeti szótár, feltéve, hogy a történeti szótártól ugyanazt várjuk el ezentúl is, amit Grimm vagy Murray nyújtott az első igazi történeti szótárakban. Ahhoz, hogy minden szó összes jelentésárnyalatának legkorábbi előfordulását megadjuk, a nyelv *teljes* írott korpuszát számítógépre kellene vinni, ami lehetetlen, vagy kombinálni kell a régi gyűjtési módszert és a számítógépes gyűjtést.

Mivel teljes korpuszt csak a holt nyelvek esetében vihetünk számítógépre, az élő nyelvek szótárainak készítésekor törekedhetünk a teljesség helyett arra is, hogy egy jó

[8] p. 661
[9] p. 1

mintánk legyen az adott nyelv szókincséről. Megtehetjük, hogy egy valamilyen elv szerint kiválogatott zárt korpuszt viszünk gépre, ennek szinkrón vagy diakrón leírását készítjük el és publikáljuk, leszögezve, hogy az így készült szótár milyen zárt korpusz leírására vállalkozik. Az így készülő diakrón szótár ugyan nem azonos a hagyományos történeti szótárral, de többé-kevésbé pótolhatja azt.

A történeti szótárkészítés ugyan, mint láttuk, a számítógép alkalmazásától nem gyorsul fel a remélt mértékben, mégis célszerűbbnek látszik a jelenleg induló projektumok esetén a gépi gyűjtés használata. Egyrészt számos mechanikus műveletet takaríthatunk meg (ábécébe rendezés, kor szerinti rendezés stb.), másrészt a számítógépen tárolt adatok sokkal többféleképpen elérhetőek, könnyen módosíthatók, kiegészíthetők. Számolnunk kell azonban azzal, hogy a számítógépes szótár sokkal alaposabb tervezést igényel, mint a hagyományosan készülő, különösen akkor, ha olyan szótári adatbázist akarunk készíteni, amely nemcsak egy célra használható fel; márpedig számítógépen éppen sokcélú adatbázist érdemes létrehozni. A tervezéskor tehát gondolnunk kell a későbbi revíziókra, rövidített kiadásokra és a szótár egyes részeinek leválaszthatóságára is, ami a tervezés időszakában még nagyon távolinak tűnik. A DOE és a Trésor viszonylagos lassúsága valószínűleg annak is köszönhető, hogy még csak most ismerkednek a számítógép felhasználási lehetőségeivel a szótárírásban, számos olyan problémát kell megoldani, amellyel nagy lexikográfus elődeinknek nem kellett szembe nézniük. (pl. a számítógépes szótárban megsérülhet a lemez, elveszhet rengeteg adat, külön tudomány, hogy hogyan lehet visszaállítani az elveszett adatokat. Murray céduláinak egy részét viszont az egerekették meg.) Röviden szólva, a számítógépes lexikográfus dolga nem könnyebb és nem nehezebb, mint elődeié volt: más.

Lényegesen meggyorsíthatja azonban a számítógép-használat az új, egynyelvű, hétköznapi használatra készülő értelmező szótárak írását. Erre a New York Times Everyday Dictionary nagyon jó példa, hiszen ha kész fényszedő szalagokat használhatunk fel a konkordanciakészítéshez, valóban töredékére csökkenhet az adatgyűjtésre fordítandó idő. A COBUILD szótár szintén meghökkentően rövid idő alatt készült el, ahhoz képest, hogy milyen hatalmas forrásanyagot használt fel. (1980-ban döntöttek a mintaszövegek kiválasztásának alapelveiről, 1987-ben megjelent a kész szótár, amely 12 millió szövegszónyi forrásnyagot használt fel!) A forrásanyag kiválasztása is lényegesen egyszerűbb, ha az aktuális nyelvállapot leggyakoribb jelenségeit szeretnénk leírni. Az újabban kiadott szövegek géprevitelét az optikai olvasó alkalmazásának lehetősége is jelentősen lerövidítheti, ami a történet szövegeknél nem, vagy csak töredékesen merülhet fel.

3. A számítógépes szótárak előállítása

3.1. Adatgyűjtés és rögzítés

A SZÓTÁRKÉSZÍTÉS első munkafázisa a gyűjtendő anyag kijelölése. Ehhez először is tisztázni kell, milyen korszakból, milyen jellegű adatokat várunk, ezek után jelölhető ki nagy vonalakban, hogy milyen mennyiségű és minőségű szöveget kell gépre vinnünk. Ez a számítógépes szótárkészítés egyik sarkalatos pontja: kellő tapasztalat hiányában nagyon könnyen rossz eredményre juthatunk. Először is semmiféle mebízható adat nincs még arra, hogy adott típusú szótár esetén mi az optimális gyűjtendő szövegszómennyiség. A COBUILD példájából arra következtethetünk, hogy egy adott évtized nyelvállapotának leírásához elegendő kb. 12 millió szövegszónyi forrásanyag, helyenként azonban nekik is használniuk kellet kézi gyűjtésből származó adatokat. A Trésor 150 millió szövegszónyi korpusza már-már riasztó példának tűnik, különösen annak tudatában, hogy ezt még kézi gyűjtéssel és nyomtatott szótárak szókincsállományával is ki kellett egészíteni. Quemada tapasztalatai szerint a gyűjtött anyag mennyiségének növelése egy idő után már nem befolyásolja jelentős mértékben az egy szóra vonatkozó adatok minőségét, sajnos azonban arra nem készültek megbízható mérések és kísérletek, hogy milyen szövegszómennyiségnél érdemes áttérni a gépi gyűjtésről a kézi válogatásra. Az optimális mennyiség nyilván nagymértékben függ a kiválasztott anyagoktól is, feltehetően kisebb szövegszómennyiség is értékesebb adatokat szolgáltathat, ha több kisebb részletből áll össze, mintha kevesebb teljes műből válogatták. A NSz-i munkacsoport tervei között szerepel, hogy — szoros együttműködésben a Trésor munkatársaival — kisérleteket végezzünk arra, hogy történeti szótár készítéséhez milyen méretű és összetételű korpusz a legoptimálisabb.

A számítógépes tervezés és gyűjtés csak a forrásanyag kijelölése után, annak ismeretében kezdődhet meg. A számítógépre vitelnek számos módja van. A régebbi szótárak egy részét lyukkártyára, ill. lyukszalagra rögzítették, ma már ezt a technikát nemigen alkalmazzák. Felhasználhatók korpuszkészítésre a nyomdai fényszedő szalagok is, így gyűjtötte alapanyagát a New York Times Everyday Dictionary. A legfejlettebb adatbeviteli mód az intelligens optikai karakterolvasó használata, ez azonban egyelőre még a fejlett országokban sem terjedt el. Az OED szövegét például — mint láttuk — olyan géppel (Kurzweil) próbálták meg beolvastatni, amely számos betűtípus felismerésére megtanítható. Ez a gép azonban csak akkor használható hatékonyan, ha folyamatosan egy betűtípust ol-

vastatunk be vele. Így például megfelelően működött a COBUILD korpusz túlnyomó részének felvitelekor, hiszen általában új kiadású, teljes könyveket vittek fel. Problémát itt csak a papírkötésű, rossz nyomdatechnikával előállított könyvek okoztak, valamint az újságok és magazinok.

Nem nevezhető hatékonynak az optikai olvasásnak az a módja, amelyet a DOE munkatársai voltak kénytelenek alkalmazni. Ha az optikai olvasó csak egy fajta betűtípust ismer fel, és ezért azzal a betűtípussal külön le kell gépelni a szöveget, jobb nem használni, és inkább egyből valamilyen számítógépes adathordózóra gépelni az anyagot.

A legelterjedtebb rögzítési mód a terminálokon vagy személyi számítógépeken való adatrögzítés. Legfőbb előnye, hogy könnyen és gyorsan javítható, és jobbnál jobb szövegszerkesztő programok állnak rendelkezésre, amelyek megkönnyítik az adatbevitelt. A rögzített adatokat mágneses adathordozón tárolják, innen bármikor lemásolhatók, kinyomtathatók ill. lekérdezhetők.

A legtöbb szótári projektumnak problémát okoz a számítógépek szűkös karakterkészlete. Rendszerint valamilyen kódkombinációval vagy helyettesítéssel oldják meg az összes szükséges karakter ábrázolását. Az Old English Dictionarynél elegendő volt a 256 karakter, így a behelyettesítést választották. Külön szoftverjük gondoskodik arról, hogy a szócikkíráskor visszahelyettesítsék a megfelelő karaktereket. Mivel grafikus képernyőt használnak, az összes *old English* karaktert meg tudják jeleníteni a képernyőn és a lézer printeren egyaránt. A New OED-ben speciális tagoló kódokkal jelölték a különböző betűtípusokat.

3.2. Lemmatizálás

A LEXICO, mint láttuk, nem vállalkozik automatikus lemmatizálásra. Vannak azonban olyan szótári projektumok is, amelyek megkíséreltek automatikus morfológiai elemzőt készíteni. A pisai Istituto di linguistica computazionale-ban többféle morfológiai elemzőeljárást is kidolgoztak (ZAMPOLLI 1983).

A legegyszerűbb automatikus morfológiai elemző a lemmatizáláshoz tulajdonképpen egy morfológiai szintetizáló programot használ fel. Először egy szótár felhasználásával létrehozzák az összes lehetséges toldalékolt alakot, és mindegyiket ellátják a megfelelő morfológiai kódokkal. Az elemzéskor a szövegben talált szóalakokat ezekhez illesztik; ha megtalálják a szövegszót a szótárban, kész az „elemzés". Gyakorlatilag ugyanez történt a Trésor esetében is: itt csupán az igék lemmatizálását végezték el ezzel a módszerrel, illetve a szövegszavak grammatikai kategóriákba sorolására tettek így kísérletet. Becsléseik szerint a teljes korpusz mintegy 60–70%-át sikerült így megfelelő grammatikai kategóriába sorolni.

Egy másik elemző program, amelyet a spanyol nyelvre fejlesztettek ki (CATARZI 1982), már valóban szegmentálja a szövegszavakat. Mielőtt hozzákezdenének az elemzéshez, egy előfeldolgozó program a *Juillard* gyakorisági szótár első 750 szavának felhasználásával kiszűri a szövegből a leggyakoribb szavakat és állandósult szókapcsolatokat. Így a szöveg 50%-át az első lépésben kiszűrik. A tulajdonképpeni elemző egy tőtárat és egy toldaléktárat felhasználva működik; minden tő- és toldalékmorféma morfológiai kódokkal van ellátva. Amennyiben nem sikerül elemezni a szóalakot, a program jelzi a lexikográ-

fusnak a hibát. A homográf alakokat a szövegkörnyezet vizsgálatával próbálja azonosítani a program. Ezzel a módszerrel az esetek mintegy 80%-ában helyes elemzéshez jutnak.

3.3. Konkordanciák

A SZÁMÍTÓGÉPES szótáríráshoz általában csak az egyszerű konkordanciákat használják. A konkordanciák rendszerint minden szó 1–2 sornyi szövegkörnyezetét és az előfordulás pontos helyét tartalmazzák. A konkordancia-programok túlnyomó része lehetőséget ad arra, hogy a leggyakoribb szavakat — rendszerint a formaszavakat — kizárjuk a konkordanciából, mivel ezek általában az anyag 30–40%-át teszik ki. Néhány közismert, készen megvásárolható konkordancia program:

A debreceni kokordancia program

Ez a program a számítógépes nyelvészeti munkacsoport által készített írói szótárakhoz és egyéb konkordancia listát igénylő munkálatokhoz készült. Általában egysoros konkordanciát készíttethetünk vele, akár a kulcsszóra és annak jobboldali környezetére ábécében rendezve, akár a-tergo sorrendben (ekkor természetsen a kulcsszó baloldali környezetére rendezve). Hátránya, hogy a konkordancia készítést megelőzően a szöveget fix, egy-egy szót tartalmazó rekordokra darabolja, íly módon feleslegesen jelentősen megnövelve a szövegfile méretét. Ezzel a programmal a konkordanciákat csak papírra írathatjuk ki, és nincs lehetőség az adatok interaktív lekérdezésére és valamilyen szempont szerinti válogatására sem.

Oxford Concordance Program (OCP)

Az OCP parancsnyelve egyszerű angol szavakból áll, könnyen elsajátítható. A programot ANSI Fortranban írták, így számos géptípuson futtatható (IBM, CDC, Digital, ICL, Univac, Burroughs, Honeywell, Prime). Meghatározhatjuk azokat a szavakat, amelyeket nem akarunk a konkordanciába kiíratni: ha csak néhány szóról akarunk listát kapni, ezeket is megadhatjuk. A kulcsszavakat rendeztethetjük jobbról vagy balról ábécé rendbe, illetve gyakoriság vagy hosszúság szerint csökkenő vagy növekvő sorrendbe. A listák főbb típusai:

— szóalak lista, gyakorisággal
— index: a szavak előfordulási helyei
— konkordancia: szólista szövegkörnyezettel, előfordulási hellyel és gyakorisággal
— statisztikák

A konkordancia-sorok hosszúsága szükség szerint változtatható.

A pisai konkordancia program

Ez a konkordancia program (ZAMPOLLI 1983) szintén lehetővé teszi, hogy felsoroljuk azokat a szavakat, amelyeknek a konkordanciájára nem vagyunk kíváncsiak. A nagyon gyakori szavak esetén a program automatikusan csak az előfordulás számát írja ki, bizo-

nyos esetekben pedig a teljes konkordancia helyett csak mintát ír ki az előfordulás száma mellett.

Változtatható a kiírandó szövegkörnyezet mérete, de a keresett szó mindig a konkordancia közepén van. A felhasználó megadhatja a szöveghatárok jeleit (versszak vége, bekezdés vége stb.), továbbá a szóhatárt jelölő írásjeleket. Grammatikai címkékkel ellátott szöveg esetén a keresett szó grammatikai tulajdoságai maghatározhatják a kontextus méretét.

A konkordanciát többféleképpen rendeztethetjük: a címszóra, és ezen belül a morfológiai kódokra, a szó után vagy előtt lévő szövegre, a szöveg eredeti sorrendjére, vagy időrendben.

A Lexicon

Ez a konkordancia program (PAIKEDAY 1983) alig több, mint egy átlagos szövegszerkesztő. Mindazonáltal ezzel is kiírathatjuk kisebb szövegfile-ok konkordanciáit, de csak egy-egy szóalakot tud keresni, szókapcsolatok konkordanciáját nem tudja előállítani. A konkordancia-sor mérete nem változtatható rugalmasan.

A Lexico

Ezt a programot Venezky készítette a DOE-hez. Nem pusztán konkordancia-program, hanem egy összetett szótárírást segítő programrendszer. Fő funkciói: a szöveg tárolása, szerkesztése, konkordancia készítése és lemmatizálás. A szerkesztés során a programmal kikerestethetjük az összes olyan idézetet, amelyben az éppen szerkesztés alatt álló címszó előfordult. Az idézeteket azonosító szerint is kereshetjük, és besorolhatjuk őket a megfelelő címszavak alá, azaz lemmatizálhatjuk őket. A lemmatizált idézetállomány konkordanciáját ezután újra elkészíttethetjük a programmal: ekkor már címszavakra rendezett konkordanciát kapunk.

A Wordcruncher

A konkordancia-készítésen túl számos szövegkeresési műveletre is alkalmazható (HUGHES 1987). A program nem szótári projektum keretében készült, IBM XT-n és ezzel kompatibilis gépeken futtatható felhasználó-orientált szövegkezelő rendszer.

Mivel mikroszámítógépre készült, elsősorban kisebb szövegállományokban való hatékony keresésre alkalmazható, lehetőség van azonban arra, hogy a kisebb szövegeket összefűzzük, és a keresést az összefűzött állományon végeztessük. Egy-egy kis szövegállományban 13–15 000 szó lehet, és 50 kis szöveget fűzhetünk össze. Így maximálisan kb. 300 000 szó lehet az összefűzött állományban. Egy szó hossza legfeljebb 31 karakter lehet. A program futtatásához legalább 512K memória, két floppy meghajtó vagy egy fix lemezes meghajtó, valamint DOS 2.1 vagy 3.2 operációs rendszer szükséges. A programot az Electronic Text Corporation terjeszti (Provo, Utah, USA).

A rendszer három fő programból áll: az *IndexETC* hozza létre a kötetlen formátumú szövegekből az indexállományt; a *ViewETC*-vel kereshetünk az indexelt állományban, a

BYU concordance-szal készíthetjük el a konkordanciákat. Az index-készítés előtt megadhatjuk a nem-indexelendő szavak listáját, a szóéget jelölő írásjeleket, a kívánt rendezési szempontot, és a szövegekben használt elhatároló jeleket (bekezdés, versszak stb.). A *ViewETC*-vel kiválaszthatunk egy szövegrészt, a szóalakokat kiírathatjuk ábécérenben, gyakoriságukkal együtt. Kereshetünk szavakat vagy szókapcsolatokat, logikai műveleteket használva. Változtathatjuk a kiírandó szövegkörnyezet méretét, a kiírás formátumát. Kérhetjük azt is, hogy mindig csak egy előfordulást, vagy hogy csak bizonyos kóddal rendelkező szavakat írjon ki.

A nem angol nyelvű szövegekben való keresést is támogatja, német, francia és spanyol szöveghez kaphatók kész interfészek, grafikus képernyő használata esetén görög, orosz, és héber karaktereket is tud kezelni.

Felhasználó interfésze rendkívül jó, könnyen elsajátítható. A funkcióbillentyűk használata és a Help-menük nagyon megkönnyítik a rendszerrel való ismerkedést.

A PAT

A PAT olyan szövegkereső program, amely hatalmas szövegfile-okban képes szavakat, szópárokat, karaktersorozatokat keresni. A programot elsősorban az Oxford English Dictionary számítógépes adatbázisában való hatékony keresésre fejlesztették ki. Interaktív keresésre alkalmas program. Mivel ez, vagy egy ehhez nagyon hasonló tulajdonságokkal bíró program látszik a NSz. céljaira is a legalkalmasabbnak, ezt a többinél részletesebben kívánom ismertetni.

A felhasználó a terminálon keresztül írja be kérdéseit, és a képernyőn azonnal megjelenik a válasz. Amennyiben ki akarjuk nyomtatni az eredményt, a programot használhatjuk úgy, hogy az eredményt a képernyő helyett egy file-ba írja, ezt a file-t pedig kinyomtathatjuk.

Háromféle keresési mód van a PATben: kereshetünk szavakat, karakatereket és hasonló kiejtésű karaktersorozatokat (soundex mode, amely csak angol szövegekre van értelmezve). A programot leggyakrabban szavak keresésére használják. A program először index-file-t készít a szöveghez úgy, hogy minden egyes szó kezdőpozícióját megjegyzi. Amikor valamilyen szót vagy szókezdő karaktersorozatot akarunk keresni, a program az index-file-ból keresi ki az adatokat. Eredményül kiírja, hányszor találta meg a keresett karaktersorozatot, ezután az eredmény mennyiségétől függően kiírathatjuk az összes előfordulást vagy csak több-kevesebb mintát közülük.

Ha a karakteres keresési módot használjuk, nemcsak a szókezdeteket vizsgálhatjuk, hanem a szavak belsejében előforduló bármely karakter(eke)t is. A program ilyenkor valamennyi karakter előfordulási helyét tárolja az index-file-ban, ezért tudunk a szavakon belül is keresni. Ennek következtében azonban az index-file-unk hatalmas nagy lesz; ezt a keresési módot ezért többnyire csak kisebb szövegfile-okon érdemes alkalmazni. Esetleg kivételesen indokolt esetben nagyobb szövegekre is használhatjuk, ilyenkor azonban számolnunk kell azzal, hogy az index-file mérete meg fogja haladni az eredeti szövegfile méretét.

A keresések előtt előkészítő műveleteket kell végeztetnünk, és ehhez meg kell adnunk néhány adatot. Először is meg kell neveznünk, milyen keresési módhoz akarunk

index-file-t készíteni. Ezután megadhatjuk azoknak a karaktereknek a listáját, amelyeket a keresésnél azonosnak szeretnénk tekinteni. Általában a kis- és nagybetűk között a keresés szempontjából nem akarunk különbséget tenni, ezeket tehát itt felsorolhatjuk így: `(CharMappings Aa Bb Cc... )`. Kizárhatjuk az indexelésből — és így a kereséből is — a leggyakoribb szavakat, pl. a névelőket, kötőszókat. Ezáltal az index-file-unk lényegesen kisebb lesz; de ne feledjük el, hogy ebben az esetben ezeket a szavakat nem fogjuk tudni megkeresni! (Viszont az index-file mérete legalább 20%-kal kisebb lesz!) Megtehetjük, hogy két index-file-t hozunk létre ugyanahhoz a szövegfile-hoz, egy kisebbet, amelyből kihagytuk a leggyakoribb szavakat, és egy nagyobbat, amelyet csak olyankor használunk, amikor a kihagyott szavakat akarjuk megkeresni.

A keresést az index-file elkészítése után kezdhetjük meg. A legegyszerűbb kereső parancs, ha beírjuk a keresett karaktersorozatot. Válaszul megkapjuk, hogy ez a karaktersorozat hányszor fordult elő. A `pr` parancs hatására a program kiírja az összes előfordulást, a `pr sample` hatására 10 véletlenszerűen kiválasztott mintát ír ki, a `pr sample (40)` hatására 40 véletlenszerűen kiválasztott sort ír ki. A konkordanciasorok elején megadja, hogy a teljes szövegfile hanyadik karakterén kezdődött a szó; ennek segítségével kiírathatunk a szót megelőző szövegből annyit, amennyit akarunk. A szám után a szót megelőző 14 karakter következik, majd a keresett szó, és az utána lévő 46 karakternyi szöveg.

A szó *után* kiírandó karakterek száma többféleképpen is módosítható; a `(Print-Length 100)` parancs hatására pl. az összes további printparancs 100 karakternyi szöveget fog kiírni a megtalált szó után. Ha csak ideiglenesen akarjuk módosítani a kiírandó szöveg hosszát, a `print` parancsba írt szám segítségével tehetjük meg. pl. `pr 1,200` hatására a szó szövegkörnyezeteit úgy fogja kiírni, hogy a szó után mindig 200 karaktert ír ki, az eredeti 46 helyett. Sajnálatos módon a szó *előtt* kiírandó karakterek száma egyelőre nem módosítható ilyen egyszerűen. Egyetlen lehetőségünk, hogy megjegyezzük a sor elején lévő karakterszámot. Ebből kivonva az általunk kívánatosnak ítélt karakterszámot, a `pr` utasítás segítségével kiírhatunk tetszőleges méretű környezetet: pl. a `pr (1024), 300` parancs hatására a szöveg-file 1024. karakterétől kezdődően 300 karakternyi folyamatos szöveget ír ki a program.

Ha bizonyos szavak együttes előfordulására vagyunk kíváncsiak, ezeket a `nxt` és `fby` parancsok segítségével kereshetjük meg. A `fby` parancs a szó jobboldali, a `nxt` a szó jobb és baloldali környezetében egyaránt keres. Beállíthatjuk a figyelembe veendő környezet hosszát is: pl. a `meg nxt (10) mond` azokat a sorokat fogja kiírni, ahol a *meg* előtt vagy után 10 karakteren belül előfordult a *mond* szó. A megfelelő keresési intervallumot általában csak kísérletezéssel tudjuk meghatározni.

Megtudhatjuk, mi volt a leghosszabb megegyező karaktersorozat a szövegünkben. Ha paraméter nélkül adjuk ki a parancsot, az egész file-ban talált leghosszabb ismétlődő szöveget írja ki; ha a parancs után valamilyen karaktersorozatot írunk, a leghosszabb, ezekkel a karakterekkel kezdődő egyforma szósorozatokat írja ki. Gyakorisági szótár ugyan nem készíthető automatikusan ezzel a programmal, azt azonban megtudhatjuk, hogy melyek az adott karakterrel vagy karaktersorozattal kezdődő leggyakoribb szavak.

A programot feltétlenül érdemes továbbfejleszteni. A legsürgősebbnek tűnő kiegészítések:

— lehetőség a locuskódok kijelölésére és kiíratására, közvetlenül a konkordanciasorok elején,
— részletek kinyomtatása a válaszokból,
— rugalmas kiíratási intervallum megadásának lehetősége a keresett szó előtt és után egyaránt,
— a `signif` utasítás használatakor szükség lenne egy olyan változatra, ahol a keresett karaktersorozat beírása helyett meg lehetne adni egy korábbi kérdés számát (ezt egyébként az összes többi utasításnál meg lehet tenni).

A fenti kiegészítésekkel ellátott programot magyar szövegekre csak lemmatizált szövegállományokon lehetne hatékonyan használni.

A program erényei

Gyors keresést biztosít, tizedmásodpercek alatt keresi ki a 15 millió karakternyi szövegből a szavakat. (Ugyanez, mint láttuk, a FRANTEXT-hez készült STELLA programnál akár 10 percig is eltarthat!) Nagyon kevés helyet igényelnek a kereséshez szükséges segédeszközök: az index-file mérete a kizárt szavak számától függően az eredeti file méretének 40–60%-a, a szövegfile méretét nem kell megnövelni a hatékony kereséshez. A rendszer parancsai nagyon egyszerűek, könnyen megtanulhatók és használhatók, a rendelkezésre álló dokumentáció alapján egy nap alatt mindent meg lehet róla tudni. Használata semmiféle számítástechnikai ismeretet nem igényel.

3.4. Szócikkírás

A SZÓCIKKÍRÁSBAN a számítógép csak kifejezetten e célra kifejlesztett szerkesztőprogramokkal tud valamelyes segítséget nyújtani. Jelenleg a Lexico az egyik legjobb ilyen program. A szócikkszerkesztő program a képernyőn ablakokat használ. Az egyik ablakban láthatjuk a billentyűzetet, rajta a speciális karakterek képével; a fő ablakban jelenik meg a kitöltetlen szócikk. A megfelelő helyre bemásolhatjuk azokat az adatokat, amelyek a gépen is tárolva vannak (gyakoriság, legkorábbi előfordulás); ezt kiegészíthetjük új adatokkal is (értelmezés). A lexikográfusok által bejelölt azonosítójú idézeteket lehívhatjuk a konkordanciaállományból: ezeket a program a kívánt helyre írja be. Végül a képernyőn lévő billentyűzetet felhasználva a szócikk-file-ban kicseréltethetjük azokat a speciális karaktereket, amelyeket a konkordanciában valamilyen más karakterrel helyettesítettek. A képernyőn úgy látjuk a szócikket, ahogy majd a szótárban fog megjelenni (dőlt betűk, félkövér betűk stb.). Most fejlesztenek ehhez olyan programot, amely szócikkírás közben a képernyőre írja a konkordanciát, és a lexikográfus rögtön bejelölheti ill. átmásolhatja a jónak tartott idézeteket.

A pisai számítógépes nyelvészeti központban szintén fejlesztenek sokoldalú lexikográfiai munkaállomásokat. Mivel itt egyszerre több számítógépes szótári munkálat is fo-

lyamatban van[10] (ZAMPOLLI 1983B), a munkaállomások sokkal több lehetőséget biztosítanak a szócikkírónak, mint a fent vázolt LEXICO. Szintén ablakrendszer használatával segítik a szerkesztők munkáját, a szócikkek közvetlenül a képernyőn is írhatók, a szükséges konkordanciák a képernyőn is megtekinthetőek. Az adatbázisokat kompakt lemezen tárolják, a szerkesztők pedig IBM személyi számítógépeken dolgoznak: a gyorsabb és rugalmasabb hozzáférés érdekében az általuk használt korpusz illetőleg adatbázis részletét számítógépük fix lemezére másolják.

[10] Gépre visznek meglevő egy és többnyelvű nyomtatott szótárakat, de újakat is készítenek, jórészt fényszedő szalagokról átalakított korpuszokból. Az így nyert hatalmas anyagból számos adatbázist is készítenek, amely természetes nyelvű interfészek és fordítórendszerek lexikonja lehet.

4. A magyar irodalmi és köznyelv nagyszótára

4.1. Történeti előzmények

1898-BAN SZÜLETETT MEG az Akadémiai Nagyszótár terve, a szótári anyaggyűjtés azonban már ezt megelőzően elkezdődött. A munkálat első irányítója Zolnai Gyula volt. Eleinte a Nyelvtörténeti Szótár kiegészítése, folytatása volt a cél: az 1750-től 1900-ig terjedő teljes szókincset kívánták feltérképezni. Noha az anyaggyűjtés jó fél évszázadon át gyakorlatilag változatlan ütemben folyt, és mintegy ötmillió cédula gyűlt össze, a szótár máig sem készült el. A kudarc fő oka abban keresendő, hogy a szótárnak sosem volt egységes, világos koncepciója. Mivel a szótári gyűjtés először spontán módon indult meg, eleve szó sem lehetett kikristályosodott koncepcióról. Zolnai Gyula *Tájszó-tarlózat*ában a következőképpen körvonalazza elképzeleseit „Jókai, Kemény Zsigmond, Petelei, Mikszáth, Kazár Emil, újabban Vereskövi (Szalóczy Bertalan) s számos egyéb írónk munkáiból időközönként minden nyelvünk ügye iránt érdeklődő olvasó följegyezhet egy-két tájszóadatot, sőt bárminő szót és szóalakot, s «több szem többet lát» igazságánál fogva a nyelvbúvárlatnak addig is sok becses anyagot bocsáthatnak rendelkezésére, míg a magyar szógyűjtés betetőző munkája, nyelvünknek második teljes szótára elkészülhet" (Nyr. 22, 219).[11] Simonyi Zsigmond ugyanekkor csupán a XIX. század irodalmi nyelvének szótárát kívánja elkészíteni. „Úgyis ideje már, hogy a befejezéshez közeledő Nyelvtörténeti Szótár után, melynek anyaga a nyelvújításig terjed, immár a nyelvújítás és az újabb irodalom szókincse is hasonló tárgyalásban részesüljön, vagy legalább hogy foglalkozzunk az ehhez való előkészületekkel. Természetesen az anyaggyűjtésen kell kezdeni, s e tekintetben a Nyelvőr dolgozótársai sokat tehetnének. Földolgozhatna mindenki, aszerint, a mint kedve vagy ideje van, vagy a mint tanulmányai adnak rá alkalmat — egy-egy kisebb vagy nagyobb munkáját a Kisfaludyaknak, Csokonainak, Kölcseynek, Vörösmartynak, Aranynak, Jósikának, Brassainak, Mikszáthnak, sat, lehetőleg egybevetve a nyelvtörténeti szótár adataival. Efféle munkához hasznos előkészületül kinálkozik Lehr Albert Toldi magyarázatának figyelmes olvasása" (Nyr. 20, 59). E két koncepció-féle alapján már 1891 táján megindult a spontán adatgyűjtés, természetesen ki-ki saját elképzelése szerint cédulázgatott. 1897-ben hozták létre a Szótári Bizottságot, amelynek fő feladata

[11] Az e pontban szereplő valamennyi idézet forrása: HUTÁS 1974.

„az ún. Nagy Szótár szerkesztése, folytonos javítása és új kiadása" volt. (vö. HUTÁS 1974) A Szótári Bizottság tagjait úgy válogatták össze, hogy lehetőleg minden tudományágat jeles tudósok képviseljenek. 1898-ban Simonyi Zsigmond felhívást tett közzé „az új Nagy Szótár munkálataiban való részvételre", ezt követően a gyűjtés a korábbinál több munkaerővel, de változatlanul koncepció nélkül folyt. 1899-ben Szily Kálmán összeállított egy címszójegyzéket, amelyet kézirat gyanánt terjesztettek a gyűjtők között, noha mind Zolnai, mind Simonyi ellenezte a címszójegyzék alapján való gyűjtést. Végre 1899-ben megjelenik a gyűjtési utasítás, amelyben körvonalazzák a szótár jellegét és célját: „A M.T. Akadémia Új Nagy Szótára a magyar nyelv összes szókincsének történeti alapon készült tudományos szótára kíván lenni." Fő célnak azonban az irodalmi és társalgási nyelv feldolgozását tekintették, a régi adatokat csak kiegészítésként, magyarázatként akarták felvenni a szótárba. A mindennapi szavakról csak akkor gyűjtöttek anyagot, ha azok új jelentésárnyalatban fordultak elő. (Ezzel szemben az OED anyaggyűjtői már kezdettől tudatában voltak annak, hogy a mindennapi szavak jellemző használatára vonatkozó idézetek éppoly fontosak, mint amelyek különös jelentést illusztrálnak. Talán ez is szerepet játszott abban, hogy az OED-ből szótár lett.)

Zolnai Gyula kidolgozott néhány próbaszócikket is, amelyet mintaként terjesztett a gyűjtők között. Ezek is azt bizonyítják, hogy Zolnai teljes szótárat kívánt szerkeszteni, amellyel mind a régi, mind az új szóhasználatot bemutatja. Eközben azonban önmagával is ellentmondásba került, mivel egy szótárba akarta belesűríteni az aktuális köznyelv és a szótörténet leírását. 1909-ben már maga Zolnai is kénytelen volt belátni az addigi gyűjtés kudarcát. A szótár főszerkesztését időközben Tolnai Vilmos vette át (1907). 1911-re mintegy 1,4 millió cédulányi anyag gyűlt össze. Mivel a XVIII. századi forrásokat már feldolgozták, megkezdték a XIX. századi anyag gyűjtését is. (Azonban máig is a XVII. századból származó anyag a leggazdagabb.) A XIX. századi források tekintetében nagy egyenetlenség mutatkozott. A szótár koncepciója ismét módosult: annak érdekében, hogy a szótár mielőbb elkészülhessen, Tolnai lemondott a teljesség elvéről: elsősorban olyan íróktól kívánt anyagot gyűjteni, akik a népnyelvből és a beszélt nyelvből merítettek. 1925-től kezdve a szótár főszerkesztői sűrűn váltották egymást, s velük a koncepció is újra meg újra módosult. 1928-ban újra felmerült a teljesség igénye: „A szótár terve szerint föl kell használni a Nyelvtörténeti Szótárban összehordott és a Nyelvtörténeti Szótár kiegészítésére közzétett nyelvtörténeti anyagot, föl kell dolgozni a nyelvújítás óta napjainkig terjedő irodalmat, az élő nyelvet, főkép a művelt társalgás nyelvét s a népnyelv egész szókincsét, föl kell venni a szótárba a művészetek, tudományok és mesterségek magyar műszavait és kifejezéseit, továbbá a szólásokon kívül a közmondásokat is; az idegen szók közül csak a meghonosodottak kerülhetnek bele a szótárba, a tulajdonnevek közül mint címszók csupán a keresztnevek vehetők fel, ezek közül is csak a meghonosodottak, a helynevek mellőzendők, de a magyar szókból képzetteknek legrégibb adatai az illető köznevek alatt esetleg idézhetők."[12]

1938-ban Ságí István vette át a munkálatok vezetését. Ekkor már két éve folyt az összegyűjtött cédulák betűrendbe szedése, ezzel párhuzamosan tovább folytatódott az

[12] Gombocz Zoltán, a Szótári Bizottság 1928. évi jegyzőkönyvében.

anyaggyűjtés is. 1944-ben a gyűjtés eljutott 1900-ig, majd a háború miatt a munka félbeszakadt. (kb. 4 millió cédulányi anyag gyűlt össze eddig.)

1950-ben Szabó Dénes javaslatára a készülő Értelmező Szótárhoz kiemeltek a gyűjteményből mintegy 450 ezer cédulányi XIX. századi klasszikus íróktól származó anyagot. A nagyszótári munkálatokat ebben az időszakban Gáldi László irányította; folytatták az anyaggyűjtést és a korábbi adatok szoros betűrendbe szedését. A gyűjtés a korábbiaktól eltérően nem csupán a ritka, kivételes szavakra terjedt ki, a cédulákon a korábbinál lényegesen hosszabb — és használhatóbb — szövegkörnyezetet tüntettek fel. Ekkor egyértelműen történeti szótár készítése volt a cél: az 1772 (nyelvújítás) és 1972 közötti időszak szókincsének történeti szempontú leírását kívánták elkészíteni. Néhány próbaszócikket is közzétettek (GÁLDI–WACHA 1957A, GÁLDI–WACHA 1957B, GÁLDI 1960), ezek visszhangja azonban korántsem volt kedvező. Egyrészt némely szócikk reménytelenül terjengős (l. „fa"), másrészt a próbaszócikkeket a rendelkezésre álló cédulákból csak igen nehezen lehetett elkészíteni, mivel ahhoz, hogy megfelelő idézetet tudjanak találni az esetek legnagyobb részében vissza kellet menni az eredeti forráshoz. 1960-ban Gáldi László vezetésével megindultak a Petőfi Szótár szerkesztésének munkálatai is, és a nagyszótár munkatársai fokozatosan erre a munkára tértek át. A 60-as évek végén Kelemen József készítette el a nagyszótár hosszú ideig utolsó tervezetét, amelyben mintegy 33 millió forintot kért a munkálat teljes befejezésére. Talán ez volt „az utolsó csepp a pohárban", a nagyszótár kérdése hosszú időre lekerült a napirendről, ezután csupán állagmegőrzés, anyagrendezés folyt.

4.2. A nagyszótár új koncepciója

A NYELVTUDOMÁNYI INTÉZET előterjesztése alapján az MTA Elnöksége 1984-ben határozta el, hogy — gyökeresen új módszerekkel — újra elindítja a nagyszótári munkálatokat. A szótárnak tartalmaznia kell a könyvnyomtatástól napjainkig terjedő magyar irodalmi és köznyelv szókincsét; a szótár forrásanyagát számítógép segítségével kell összegyűjteni. A tervezet szerint mintegy 8 millió szövegszónyi folyamatos szöveget kell számítógépre vinni, és az ezekből készített konkordanciák szolgálnak majd a szótár forrásanyagául. Tekintettel arra, hogy a folyamatos magyar szövegeknek mintegy 30 százalékát az ún. formaszók teszik ki, ahhoz, hogy körülbelül 8 millió szövegszónyi értékes adatunk legyen, *legalább* 13 millió szónyi szöveget kell gépre vinnünk. Kétséges azonban, hogy még ez a megemelt mennyiség is elegendő lesz-e a szótár elkészítéséhez. A folyamatosan — gyakorlatilag válogatás nélkül — begépelt szövegek ugyanis minden valószínűség szerint összehasonlíthatatlanul redundánsabbak lesznek, mint a hagyományos módszerekkel, célirányosan gyűjtött „cédulák". Sajnálatos azonban, hogy semmilyen pontos adat nincs arra, hogy mi az az optimális szövegszó mennyiség, amely egyrészt elegendő a szótár szócikkeinek elkészítéséhez, másrészt nem sokszorosa a feltétlen szükségesnek. Nyilvánvalónak tűnik, hogy nem csupán a rögzített szöveg mennyisége a döntő, hanem annak minősége is. Az a két szótári projektum, amely az anyaggyűjtés módszerében a magyar nagyszótárhoz leginkább hasonló (COBUILD és Trésor), kizárólag teljes műveket vitt számítógépre, és kifejezetten arra törekedtek, hogy a leghíresebb, legolvasottabb műveket vegyék fel a korpuszba. A mi nagyszótári korpuszunkba

ezzel szemben csak kivételes esetekben viszünk fel teljes, hosszabb lélegzetű műveket, és noha a leghíresebb írók legfontosabb műveiből hosszabb részleteket rögzítünk, kifejezetten jelentéktelen szerzőktől is viszünk fel kisebb terjedelmű szövegrészleteket. Bár remélhető, hogy az így gyűjtött anyag változatosságánál fogva relatíve információgazdagabb lesz, mint a COBUILD vagy a FRANTEXT korpusz, mennyisége valószínűleg reménytelenül kicsi lesz a kitűzött feladathoz képest. Azonban egyelőre még megbecsülni sem tudjuk, hogy milyen mennyiségű és minőségű anyaggal kell majd kiegészíteni korpuszunkat ahhoz, hogy ebből kiindulva hagyományos értelemben vett történeti szótárat tudjunk készíteni. Reméljük azonban, hogy a FRANTEXT korpuszon végzendő kísérletek — amelyek a korpusz méretének és összetételnének optimalizálására fognak irányulni — segítséget nyújtanak majd abban, hogy már a 13 milliós korpusz géprevitele és áttekintése előtt meghatározzuk, milyen mértékben és módon kell korpuszunkat bővítenünk.

A 13 millió szövegszónyi anyag a feldolgozandó századok között nem egyenletesen oszlik meg. Tekintettel arra, hogy a könynyomtatás kezdetétől napjainkig a megjelent művek száma fokozatosan növekszik, az egyes századokból hozzávetőlegesen a következő szövegszó mennyiségeket rögzítjük:

XVI. század	1 millió szövegszó
XVII. század	1 millió szövegszó
XVIII. század	2 millió szövegszó
XIX. század	4 millió szövegszó
XX. század	5 millió szövegszó

Ezek a számok természetesen távolról sem pontosak: a források kijelölőit arra kértük, inkább ennél több, mint kevesebb szövegszónyi anyagot jelöljenek ki. A források kiválogatásánál az egyes századok szakértői arra törekedtek, hogy — figyelembe véve a sok szerző — sok kisebb műrészlet alapelvet — az egyes korszakokra legjellemzőbb műrészleteket válasszák ki. Hosszas viták után abban állapodtunk meg, hogy nem célszerű előzetesen megszabni sem az egyes műfajok, sem az egyes szerzők századonkénti arányát. A teljes forrásjegyzék összeállításán túl külön problémát jelentett a legjobb, leginkább szöveghű kiadás kiválasztása. Végül is az tűnt a legmegbízhatóbbnak, ha vagy a szerző életében megjelent utolsó kiadást, vagy a kritikai kiadást vesszük alapul. A régi századok szakértői amellett kardoskodtak, hogy a csupán kéziratban terjesztett művek közül is vigyük fel azokat, amelyek fontosak, elterjedtek voltak. Erről azonban le kellett mondanunk, mivel ez a nagyszótári adatbevitelt beláthatatlanul megnehezítette volna. Természetesen felviszünk olyan műveket, amelyek később nyomtatásban megjelentek.

Az elkészült forrásjegyzékeket szakértőkkel lektoráltattuk. A szépirodalmi forrásokon túl néhány szaktudományból is válogattattunk szemelvényeket, ezek összegyűjtésére az illető szakterület legjobb ismerőinek adtunk megbízást. Az általuk összeállított anyagot a szépirodalmi forrásjegyzékek készítői bírálják el, és ők jelölik ki azokat a részleteket, amelyeket végül számítógépre viszünk.

Valamennyi kiválasztott forrásrészletről az Országos Széchényi Könyvtár készít számunkra fénymásolatot. Ezeket azután a lehető legnagyobb mértékben szöveghűen szá-

mítógépre rögzítjük. A rögzítés megkezdése előtt számos problémára kellett megoldást találnunk. Először is, hogyan reprezentáljuk a számítógépen a teljes magyar karakterkészletet (a történeti karaktereket is beleértve), úgy, hogy a rögzített szöveg egyértelmű és könnyen kezelhető legyen (rendezés, adatátvitel, konvertálás stb.). Végül is „csúnya", de a célra megfelelőnek tűnő megoldást választottunk: az ékezetes és történeti karaktereket egy betű és egy szám kombinációjával jelöljük. Így pl á=a1; é=e1; ö=o2... stb. Azt is el kellett határoznunk, mennyi és milyen kódot használjunk a szöveg rögzítésekor. Végül egyéves próbarögzítés után úgy határoztunk, minimalizáljuk az alkalmazandó kódokat, csupán a legszükségesebb jellemzőket jelöljük meg (pl. idegen szó, tulajdonnév, idézet stb.; ezekről később részletesebben szólunk). A rögzített szövegeket kétszeresen ellenőrizzük és javítjuk, majd több példányban mind mágnesszalagon, mind floppy lemezeken archiváljuk.

A kétszeresen javított szövegeket egy automatikus morfológiai elemző program segítségével lemmatizálni fogjuk. Az elemző program a szövegszavakat tőre és toldalékmorfémákra próbálja felbontani, valamint ahol tudja, bejelöli a morfémahatárokat és címkékkel lát el minden egyes morfémát. A címkék a tövek esetén a címszó szófaját, és esetleges homonimakódját tartalmazzák, s ammenyiben a tő aktuális formája eltér a címszótól, a szótári címszót is. Toldalékok esetén a címke a toldalék kódja lesz. Mindez arra szolgál, hogy ne csupán szövegszóra, hanem lexémára rendezett konkordanciákat készíthessünk. Ráadásul az elemzett szövegekben sokkal többféle szempont szerint kereshetünk, például csoportosíthatjuk az előfordulásokat vonzatok szerint, kereshetünk jellegzetes szintaktikai szerkezettípusokat stb. Mivel a gépi elemzést minden esetben kézi ellenőrzés és javítás követi, ez újabb alkalmat ad az esetleg még megmaradt rögzítési hibák kiszűrésére.

Az elemzett-lemmatizált korpuszt egy rugalmas konkordancia program segítségével kívánjuk kezelni. Ez lehetővé teszi majd egy lexéma vagy szövegszó teljes konkordanciájának kilistázását, akár a szövegkörnyezetre ábécében, akár a keletkezés éve szerint rendezve. Csoportosíthatjuk majd a konkordanciát a kulcsszó előtt/után előforduló elemek szerint is, és válogatott konkordanciát is készíttethetünk adott korszakra, szerzőre stb. A konkordancia programon túl egy kifejezetten szótárszerkesztést, szócikkírást segítő programrendszert is szeretnénk készíteni, amely lehetővé tenné, hogy a lexikográfusok közvetlenül a képernyőn írhassák meg a szócikkeket, ott válogassák ki a korpuszból a legmegfelelőbbnek látszó idézeteket, és akár közvetlenül átmásolhassák azokat a szócikk megfelelő részébe.

A próbaszócikkek írása akkor kezdődhet meg, ha a kijelölt szövegrészleteknek legalább 80%-a rögzítve és elemezve a számítógépen rendelkezésünkre áll, és legalább a konkordancia programok elkészültek. (Ehhez persze nagy teljesítményű számítógép is szükséges.) A próbaszócikkek elkészítése után dönthetjük el, milyen módon kell még kiegészítenünk a forrásanyagot ahhoz, hogy valódi történeti szótárat készíthessünk. Mindamellet célszerűnek tűnik a teljes korpusz és a hozzá tartozó konkordancia program mielőbbi kiadása optikai lemezen, így az érdeklődő kutatók már a szótár készítése alatt is tanulmányozhatnák, hasznosíthatnák ezt az értékes forrásanyagot. Ez annál is inkább célszerűnek tűnik, mivel maga a szótár még a legoptimistább becslések szerint is aligha készülhet el egy emberöltőn belül.

A továbbiakban a nagyszótár számítógépes megvalósításának rendszertervét ismertetem, részletesebben kitérve a már folyamatban lévő munkákra (adatrögzítés, forrásnyilvántartás, előfeldolgozás: morfológiai elemzés), és vázlatosan ismertetve a később készítendő illetőleg megvásárolandó programrendszert (index generálás, konkordancia és címszólista készítés, szócikkírást támogató programrendszer).

4.3. Az adatgyűjtés számítógépes vonatkozásai

4.3.1. Az adatrögzítés legfontosabb konvenciói

AZ ADATRÖGZÍTÉST Varyter XT és Commodore 64 számítógépeken végezzük, nem használva ki a szövegszerkesztőknek és a számítógépeknek egyetlen specifikus tulajdonságát sem. Az adatokat floppy lemezeken tároljuk, a Commodore 64-en rögzített szövegeket a kétszeri ellenőrzés és javítás után Varyter XT floppykra másoljuk át. Időnként minden kétszeresen javított Varyter XT lemezt mágnesszalagon két példányban archiválunk, standard IBM formátumban. A mágnesszalagokat két helyen őrizzük: az egyik példányt az intézet szalagtároló fémszekrényében, a másikat az MTA SZTAKI Victor Hugo utcai számítóközpontjában.

Minden szövegrészlet külön file-ba kerül. A file neve a szerző nevéből és a mű számkódjából kialakított, legfeljebb 8 karakteres név. A xeroxon minden műrészlet nevét megadjuk: az első három adat jelöli a forráskódot, a negyedik adat a file-név. Az egyes file-ok azonos formátumúak, valamennyi file első rekordja a címrekord. Ennek hossza változó. Az első 10 pozíción szerepel a mű forráskódja , a század, a szerző, a mű összetett kódja. A 11. pozíciótól rögzítjük a szerző teljes nevét, a név után kettőspont következik, majd a mű (részlet) címe. A cím mindig a mű címe, nem a kötet címe. Amennyiben idegen nyelvből fordított művet rögzítünk, a szerző neve helyett a fordító nevét írjuk be. Ha többen fordítottak egy művet, vagy a bekarikázott nevet, vagy az elsőként szereplő fordító nevét írjuk be. Előfordul az is, hogy nem a szerző, hanem a szerkesztő neve alapján azonosítunk egy művet, ezt a xeroxon szintén bejelöljük; több szerkesztő esetén ugyanúgy járunk el, mint a fordításoknál. Az utolsó négy pozíción vezető nullákkal kiegészítve feltüntetjük a szövegrészlet kezdő oldalszámát.

A címrekord után folyamatosan rögzítjük a szöveget. Minden szövegrészlet ␣#/#␣[13] jellel kezdődik, és ugyanezzel fejeződik be. Az alcímeket, ha a kijelölt szövegrészletbe esnek, szintén rögzítjük. A szöveg többi részétől az `<alcim>...` `...` `<alcim v>` jelekkel választjuk el. A kijelölt szövegrészlet első értelmes mondatánál, ill. gondolatánál kezdjük mindig a rögzítést, és az utolsó értelmes mondatnál illetőleg gondolatnál fejezzük be.

Az írásjelek után (esetleg előtt) legfeljebb egy szóközt rögzítünk. A bekezdéseket nem kezdjük új sorban, a bekezdés végét, ill. kezdetét ␣#␣-kal jelöljük. Ha a forrásszövegben egy oldal végére érünk, akkor az itt szereplő utolsó szó vagy írásjel után ␣*****pppp␣-ot rögzítünk, ahol *pppp* a következő oldal száma, vezető nullákkal kiegé-

[13] A továbbiakban ␣-val jelöljük a kötelező szüneteket (space).

szítve. A versszakok határát u#/u jellel, a verssorok határát u/u-lel jelöljük. Ha prózai illetőleg verses szövegekben verses vagy prózai betét fordul elő, azt attól függően, hogy ugyanannak a szerzőnek vagy másnak a művéből idézett részletről van-e szó, többféleképpen jelöljük:

u#1u jelöli a verses betét elejét, ill. végét, ha a szerző szövegét saját versben folytatja.
u#2u jelöli a szerző által másoktól idézett verses betétek elejét, ill. végét.
u#3u jelöli a más szerzőtől származó prózai idézetek elejét, ill. végét.
u#4u jelöli a prózai betétek elejét, ill. végét olyankor, amikor a szerző verses szövegét saját prózájával folytatja.

Drámai művek rögzítésekor a beszélők neve után kettőspontot és szünetet írunk, akár szerepel kettőspont az eredeti drámában, akár nem. A szereplők nevét, akár csupa nagybetűvel, akár kiemelt írásmóddal van szedve, u(1u ... u)1u-gyel jelöljük, a kettőspontot a)1 után tesszük. Nem versszakokra tagolódó verses drámáknál a megszólalások végén csak a bekezdés vége jelet használjuk.

A kijelölt szövegrészletben előforduló valamennyi jegyzetet rögzítjük, akkor is, ha maga a jegyzet szövege nem a kijelölt oldalon van, de a jegyzettel bővített törzsszöveg igen. A jegyzetek rögzítését, függetlenül attól, hogy az eredeti kiadványban a jegyzet szövege hol helyezkedik el, a következő módon végezzük: az után a szó után, amelynél a jegyzetre utaló bármilyen jel (szám, csillag stb.) megjelent, egy szóköz után egy u(9u-et írunk, és ide írjuk jegyzet szövegét. A jegyzet után, ha megállapítható, odaírjuk, hogy a kiadó vagy a szerző jegyzete volt-e, majd u)9u-el bezárjuk a jegyzetrészt, és folytatjuk a törzsszöveg rögzítését. Nem tüntetjük fel a jegyzetre utaló indexet, csillagot stb., számunkra a u(9u u)9u zárójelpár fogja egyértelműen jelölni, hogy jegyzetről van szó.

Ahol a szövegben legfeljebb egy mondatnyi idegen szövegrész fordul elő, a szót vagy szövegrészt a u(4u u)4u zárójelpárral jelöljük meg. Ha egy mondatnál hosszabb idegen szövegrész van, kihagyjuk a teljes idegen részt, és a kihagyást a következő módon jelöljük: u(4u.....u)4u. Ahol kétséges, nem jelöljük idegenként a szót. Általában arra törekszünk, hogy a lehető legkevesebb szót jelöljük idegenként, abból azonban komoly probléma nem származik, ha ugyanazt a szót az esetek egy részében idegenként, máskor magyarként jelöljük, hiszen ez a kategória sohasem lehet egyértelmű.

A rögzítők az általuk esetleg hibásnak ítélt szót is változatlanul rögzítik le. Az ellenőrök döntik el, hogy sajtóhibáról van-e szó. Ilyenkor is meghagyjuk az eredeti szót, csak a javítás fázisában egy u<!>u jelet írunk mögé, hogy ezzel jelezzük a későbbi felhasználóknak: feltehetően sajtóhibáról van szó.

A grafikus jeleket, kottákat, matematikai és egyéb képleteket általában elhagyjuk. A kihagyást u(2u ... u)2u-vel jelöljük. A négyszögöl jelzésére szolgáló négyzet helyett leírjuk a négyszögöl szót, és a u(2u u)2u jelekkel jelöljük, hogy a grafikus jelet változtattuk át szövegre. Ugyanezt a jelölést használjuk táblázatok rögzítése esetén is, amennyiben a táblázatnak van olyan része, amely rögzítésre ki van jelölve. Az adatelőkészítés során a szövegrészletben szereplő táblázatokat általában ceruzával áthúzzuk, hogy a kihagyást je-

lezzük, előfordulhat azonban, hogy egy táblázatot, vagy annak részeit érdemes rögzíteni; ezeket a részeket a xeroxlapon ceruzával bekarikázzuk.

Bármely, a többitől eltérő módon szedett szót vagy szövegrészt (dőlt betű, vastag betű stb.) u(1u u)1u-el jelölünk. A nyitó zárójelet a kiemelt írású szövegrész elé, a zárót mögé tesszük. Ha nem az egész szó van kiemelt írással szedve, akkor is a szó vége után tesszük a záró zárójelet. Ha idegen szó vagy szövegrész van kiemelt írással szedve, akkor mindkét számozott zárójelet alkalmazzuk, tetszőleges sorrendben. Ilyenkor a kétféle számozott zárójel közé nem kell feltétlenül szóközt tenni.

Nem rögzítjük a mottókat és a kritikai kiadásokban szereplő, az eredeti oldalszámra utaló jelet.

Minden ékezetes karaktert az ékezet elhagyásával, és egy szám kombinációjával jelölünk. Minden ékezettípusnak egy számot feleltetünk meg. Az egy vesszőből álló ékezetet az 1-es, a két pontból állót a 2-es, a két vesszőből állót a 3-as jelöli:

á = a1	é = e1	í = i1
ó = o1	ö = o2	ő = o3
ú = u1	ü = u2	ű = u3

A többi ékezetes betűt és speciális jelet is hasonlóan jelöljük, ezek kódját a 2. mellékletben külön táblázatban közöljük. Régebbi, rossz minőségű nyomtatványoknál előfordul, hogy a karakterek egy része nem azonosítható egyértelműen. Ilyenkor „azt írjuk, amit látunk", és a file elején a címrekord után egy megjegyzés rovatba beírjuk, melyek voltak azok a karakterek, amelyeknek azonosítása kétséges. A megjegyzés szövege itt ilyesmi lehet: Nem egyértelműen azonosítható karakterek: o18, o2, u16, u2, u3.

4.3.2. Forrásnyilvántartás

A FORRÁSNYILVÁNTARTÁS KÉT FŐ CÉLJA a forráskódok feloldása (ahol forráskódon a továbbiakban a század, szerző, mű összevont kódját értjük), és a szövegrészletekkel kapcsolatos különböző adatok nyilvántartása. Így a forrásnyilvántartó-file a következő adatokat tartalmazza:

1. A feldolgozott mű keletkezésének évszázada
2. A szerző nevének azonosítója (századon belül egyedi)
3. A feldolgozott műrészlet számkódja (szerzőn belül egyedi)
4. A rögzített szöveget tartalmazó Varyter XT file neve
5. A felhasznált mű műfaj kódja
6. Megjegyzés a műfajhoz (pl: fordítás)
7. A feldolgozott műrészlet kezdő oldalszáma
8. A feldolgozott műrészlet befejező oldalának száma (4 jegyű szám)
9. A szerző teljes neve betűkkel
10. A szövegrészlet címe
11. A feldolgozott mű keletkezésének éve. Ha a keletkezés éve bizonytalan, itt jelöljük. pl: 1787 vagy ?

12. A felhasznált kiadvány megjelenésének helye.
13. A felhasznált kiadvány kiadójának megnevezése.
14. A felhasznált kiadvány megjelenésének éve.
15. A felhasznált kiadvány címe.
16. A feldolgozott szövegrészlet kezdő sora. Régi kiadványoknál van jelentősége, és mindenütt, ahol a mű címe nem azonosítja egyértelműen a művet. (Ez nem az általunk feldolgozott részlet első sora, hanem a teljes mű kezdő sora!).
17. A szövegrészlet karaktereinek száma. Az összes leütések száma, beleértve az üres leütést, segédjeleket, stb. Ez az adatrögzítők és ellenőrök munkájának mennyiségi nyilvántartására szolgáló szám.
18. Az adatrögzítő nevének kódja, a rögzítés befejezésének dátuma.
19. Az ellenőr nevének rövidítése, az ellenőrzés befejezésének dátuma.
20. A javítást végző személy nevének rövidítése, a javítás befejezésének dátuma.
21. A második ellenőr nevének rövidítése, a második ellenőrzés befejezésének dátuma.
22. A második javítást végző személy nevének rövidítése, a második javítás befejezésének dátuma.
23. Az egy file-ban tárolt szövegrészlet szavainak száma.

A forrásnyilvántartó-file egy DBASE III programcsomaggal előállított adatfile. Az adatok nagy részét egy DBASE III alatt fejlesztett adatbevivő/módosító program segítségével kézzel visszük fel a VARYTER XT-re. Az 1–22. adatokat a forrásokat nyilvántartó személy tölti ki, a szószámmezőt pedig az előfeldolgozó program sikeres lefutása után maga a program.

Ha egy műből több, egymástól független részletet rögzítünk, minden részlet külön forráskódot kap.

A program főbb funkciói:

a. Karbantartás:
- Kikeresi a megadott forráskódú rekordot. Ha még nincs ilyen, egy üres rekordot jelenít meg, amelyet feltölthetünk adatokkal. Folyamatos adatbevitelnél nem üres rekordot ad, hanem az utoljára felvitt rekord adatait jeleníti meg. (Ez azért kényelmes, mert ha például egy kötetből több verset akarunk felvinni, nem kell minden adatot újra kitölteni.) A rekord lemezre másolása előtt a program ellenőrzi, nincsenek-e benne hibás vagy hiányzó adatok.
- Rendezés a forráskódra (század, szerző, mű).
- Újraindexelések.

b. Betekintés:
- A képernyőn vagy a nyomtatón megjeleníthetjük a már felvitt adatokat. Szükség esetén áttérhetünk az adatok módosítására. Lehetőségünk van arra is, hogy ne a teljes rekordot, hanem csak a főbb adatokat jelenítsük meg.

c. Listázások:

— Egyes szerzők összes művének listázása akár a szerző kódja, akár a szerző neve alapján (képernyőn vagy nyomtatón)

— A szerzők és a műcímek kilistázása századonként csoportosítva, a kezdő- és befejező oldal továbbá a feldolgozott szavak számának feltüntetésével. Rendezettsége: szerzők szerint ábécében (képernyőn vagy nyomtatón).

— A szerzők és művek listázása századonként és ezen belül műfajonként csoportosítva (képernyőn vagy nyomtatón).

— A forrásnyilvántartó-file aktuális tartalmának teljes listázása nyomtatóra.

d. Lekérdezések:

— Egyes adatrögzítők teljesítménye tetszőleges intervallumban.

— Egyes ellenőrök teljesítménye tetszőleges intervallumban.

— Adatrögzítők, ill. ellenőrök összesített teljesítménye tetszőleges intervallumban.

— Feldolgozott szavak száma századonkénti bontásban és összesítve.

— Adott forráskódhoz tartozó összes adat lekérdezése.

— Állománynév alapján az összes adat megjelenítése.

— Adott forráskódhoz tartozó file-név megjelenítése.

4.3.3. Előfeldolgozás

A MÓDOSÍTOTT SZÖVEGFILE előállítására szolgáló program először beolvassa az egyes rögzített szövegfile-ok első, ún. címrekordját. Az itt szereplő forráskód alapján megkeresi a forrásnyilvántartó-file-ban a megfelelő bibliográfiai tételt. Ha talál azonos forráskódú tételt, ellenőrzi a kezdő oldalszám értékét, és ha ez is azonos, megkezdi a feldolgozást. A program a feldogozást a következő lépésekben végzi:

a. Különböző, a későbbi felhasználás számára hasznos segédjeleket illeszt a szövegbe. A szövegfile a segédjelekkel együtt az alábbi formátumú:

`<eleje><kod>` *forráskód* `<kod v><szerzo>` *a szerző neve* `<szerzo v><mufaj>` *a műfaj megnevezése, az esetleges megjegyzéssel együtt* `<mufaj v><kelet>` *keletkezés éve* `<kelet v>*****pppp` *(oldal)* `<szoveg>` *itt következik a teljes szövegfile* `<szoveg v><vege>`

Ezeknek a jeleknek (a továbbiakban tagolójelek) a segítségével egyrészt pontosan meg tudjuk határozni az egyes dokumentumok elejét, ill. végét, másrészt gyors keresést tudunk biztosítani akár szerző, akár század, akár műfaj szerint. A tagolójelekben az ékezeteket jobb nem jelölni, mert így könnyen megkülönböztethetjük őket a szövegszavaktól.

b. Elkülöníti a szövegfile-ból a jegyzetek szövegét és a jegyzetfile-hoz illeszti őket.

c. Ellenőrzi az oldalszámozást.

d. Beírja a szövegszavak számát a forrásnyilvántartó-file megfelelő mezőjébe.

4.3.4. Morfológiai elemzés

MIÉRT VAN SZÜKSÉG a folyamatos szövegek morfológiai elemzésére? Amint a 2. és 3. fejezetben láttuk, a számítógépes szótári projektumok rendszerint nem vállalkoznak a gyűjtött szövegek automatikus lemmatizálására, annak ellenére, hogy a konkordanciákat lehetőleg lexémákra rendezve szeretnék megkapni. A COBUILD szótárnál feltehetőleg nem volt igazán szükség elemzésre, hiszen a modern angolban alig fordulnak elő ragok. A DOE-nek ugyan nyilván nagyon hasznos lett volna egy legalább részben automatikus lemmatizáló eljárás, de ez az alakváltozatok nagy száma miatt valószínűleg rendkívül nehezen és gazdaságtalanul lett volna csak megoldható. Mint láttuk, a Trésor munkatársai valódi lemmatizálás helyett megelégedtek azzal a megoldással, hogy az összes ige teljes paradigmáját generálták, és ennek alapján keresték ki a korpuszból a lexémához tartozó valamennyi szövegszót. A magyar nagyszótár megtervezésekor úgy láttuk jónak, ha nem elégszünk meg a szövegszavakra rendezett konkordanciákkal, hanem megkísérlünk kidolgozni egy olyan eljárást, amellyel legalábbis nagy részben automatikussá tehető a lexémák és a toldalékok határának megállapítása, illetve az egyes morfémák azonosítása. A puszta folyamatos szövegből ugyanis még egy jó konkordanciaprogram segítségével is rendkívül nehéz a hasonlóan kezdődő szövegszavak tengeréből kiválogatni egy-egy lexéma összes előfordulását. Az általunk használt elemző eljárás Prószéky Gábor tanulmányának felhasználásával készült (PRÓSZÉKY 1985). Mielőtt erről bővebben szólnék, röviden ismertetek néhány jellegzetes morfológiai elemző módszert, egyúttal azt is megindokolva, hogy miért éppen ezt az algoritmust választottuk. Mivel Prószéky fenti tanulmányában részletesen ír a morfológiai elemzés és szintetizálás legelterjedtebb módjairól, ezeket itt nem kívánom megismételni, csupán a legfontosabbnak tűnőket ragadom ki.

Morfológiai elemző eljárások

a. Kétszintes morfológiai elemző és szintetizáló módszer

Ezt a modellt Koskenniemi dolgozta ki (KOSKENNIEMI 1983A, 1983B, 1984). Amellett, hogy maximálisan kihasználja a számítógépek új genererációjá által nyújtott párhuzamos processzálás lehetőségét, nyelvészetileg is adekvát modellnek bizonyult. További előnye, hogy az elemzést és szintetizálást egységes módon kezeli. A modell tövekből, toldalékokból és ún. nem-fonológiai jellegű alternatív mintákból álló szótárból és párhuzamosan alklamazandó fonológiai jellegű szabályok halmazából áll. Szemben a generatív elképzelés szukcesszív szabályalkalmazásával, itt a szabályokat párhuzamosan (egyidejűleg) kell alkalmazni, ebből ered a rendszer elnevezése is: csupán két szint van, a szótári és a felszíni szint, közbülső állapotok nincsenek.

A szótárban csak a szótöveket tárolja, tőalternánsok nélkül, de a szótári szóalakok tartalmazhatnak morfofonémákat és ezek felszíni realizációját irányító szabályokat is. A szabályok nem folyamatot, hanem valamiféle statikus egyenlőséget írnak le a két szint, a lexikon és a felszín között.

A kétszintes modellt számos nyelv morfológiai leírásával tesztelték (finn, svéd, görög, arab, ószláv, lapp, japán, román, angol, francia), és ezek mind hatékony működéséről

adtak számot. Célszerűnek tűnne egy magyar nyelvű szabályrendszer elkészítése és kipróbálása is.

b. Szabály-vezérelt elemző

A B. Brodda által kifejlesztett elemző, az ún. BETA rendszer egy Turing-gép és egy környezetfüggő szabályrendszer ötvözete. Az automatában egy kurzor jelzi, hogy a rendszer „éppen hol áll a szalagon." A kurzortól balra levő részt az újraírószabályok által meghatározott módon egy másik füzérrel próbáljuk helyettesíteni. A teljes szabályrendszer három részből áll: a kivételek listájából, a tulajdonképpeni szabályrendszerből és egy metaszabály-halmazból. A kivételek listája teljes szóalak-elemzéseket, elemezhetetlen alakokat és nem-elemezendő toldalékokat tartalmaz. Az elemzés három fő lépésben történik: először a gép megkeresi a főhangsúly helyét (az első magánhangzót), és egy aposztroffal jelöli meg. Ha a kurzor szóvégi helyzetben elemezhető végződést talál, elemzi és balra húzódik az előbb beszúrt aposztrofig. Ezt kitörli, majd ismét jobbra indul. Az utolsó lépésben kitörli az esetleg bennmaradt + jeleket, amelyek az elemzést irányították. Ezután a rendszer kezdőállapotba kerül, és várja a soron következő szót. A BETA rendszert eddig a finn és svéd nyelv morfológiai elemzésére használták fel.

c. Toldaléktár által vezérelt elemzők

A toldaléktárat használó eljárások alapvetően kétféleképpen működhetnek: vagy jobbról kezdik az elemzést, és a feltételezett toldalékok levágása után ellenőrzik, hogy a kapott morféma valóban tő-e, vagy balról kezdik az elemzést, és a feltételezett tő levágása után keresik a megmaradt toldalékot a toldaléktárban. Ennek megfelelően az első esetben toldaléktár-vezérelt, a második esetben tőtár-vezérelt eljárásról beszélhetünk. Mindkét módon hatékony elemzők készíthetők: elsősorban az elemzés célja dönti el, melyik eljárást célszerű használni. Egyebek között két magyar morfológiai elemző program is készült toldaléktárból kiindulva.

Az ún. GÁZOLAJ rendszer (KISS ÉS TSAI. 1979, PRÓSZÉKY ÉS TSAI. 1982) jobbról kezdi a szóalakok elemzését, a toldalékokat nem karakteresen, hanem szimbolikusan illeszti. Az algoritmus nemcsak a további balra levő toldalékokra és lehetséges tövekre állít fel hipotéziseket, hanem ezek morfológiai tulajdonságaira is. Az elemzés úgy indul, hogy a program ellenőrzi, illetve feltételezi, hogy a szóalak tőszó, majd megkeresi az első olyan szabályt, mely a toldaléktárban tárolt valamely szimbólumot (karakterfüzért) le tudja választani a szóalak végéről. A toldalék leválasztása után az elemző azon állapotok halmazába lép, amelyek a toldalékot megelőzhetik. Ilyenkor az elemző szűkíti a levágható toldalékok halmazát. Ha nem talál leválasztható toldalékot, a zérus morfémát „vágja le."

Az MTA SZTAKI-ban kifejlesztett morfológiai elemző (BACH ÉS TSAI. 1987) szintén jobbról kezdi a „szeletelést," és az összes feltételezett toldalék levágása után ellenőrzi, hogy a kapott szóalak tő-e. A szótár nem pusztán szótöveket tartalmaz, hanem összetett és képzett szavakat is, illetőleg néhány speciális toldalékolt alakot (pl. *annak*). A szón kívül a lexéma ragozási típusának kódját és szófaját is tárolják. Az esetek egy részében a tővariánsokat külön feltüntetik (pl. *teher*, *terh-*), máskor csak a toldalékok előtt előfor-

duló tövet tüntetik fel (pl. *kutyá*, amikor majdnem minden toldalék előtt ez a kötelező alak). A toldaléktár elemeit osztályonként csoportosítva tárolják, és a toldalék előtt esetleg előforduló tőváltozásokat is kódolják. Az igeragokat összevontan kezelik (egy elemnek tekintve pl. a *mondottátok*-ból az *-ottátok* toldaléktömböt). Az elemző a toldalékolás sorrendi szabályainak megfelelően az adott szó végéről levágja a lehetséges végződéseket, az elejéről pedig leválasztja az előtagokat, miközben figyelembe veszi a lehetséges tőváltozások inverzét is. Az így kapott szótövet esetleg felbontja szóösszetételként, majd a szótöveket azonosítja a szótárral. Szóosztályegyezés esetén a szótövet, az előtagot és a toldalékot adja eredményül. Egy szövegszó több megoldást is adhat, egyrészt mert a tő és a végződés is lehet többértelmű, másrészt mert a szó többféleképpen vágható szét tőre és toldalékokra. Az elemző minden lehetséges megoldást felsorol. A program egy magyar nyelvű interfész morfológiai moduljaként készült, jelenleg néhány ezer szónyi mintaanyagon működik, igen hatékonyan.

Hasonló elven működik a 3.2-ben már említett spanyol nyelvre készült elemző (CATARZI 1982). A rendszer háromféle alapadatot használ: a prefixumok, a tövek és a toldalékok adatbázisát. Az előfeldolgozás után, amelynek során azonosítják a leggyakoribb szavakat és állandósult szókapcsolatokat, majd — balról kezdve az elemzést — a megmaradt szövegszavakról először levágják a lehetséges prefixumokat. A második lépésben jobbról illesztik a szóhoz a lehetséges toldalékokat, míg végül eljutnak a feltételezett szótőhöz. Ezután megvizsgálják, hogy az így kapott szótő megtalálható-e a tövek adatbázisában. Homográfok esetén megvizsgálják a szó szövegkörnyezetét, és ennek alapján próbálják eldönteni, mi a helyes elemzés. Néhány jellemző szabály, amelynek segítségével a homográfok közül választanak:

— névelő vagy személyes névmás?
  ha a következő szó egyértelműen főnév, akkor névelő
  ha a következő szó egyértelműen ige, akkor személyes névmás
  ha az előző szó egyértelműen ige, akkor névelő
  ha a vizsgált és az azt követő szó személyben és számban nem egyezik, akkor személyes névmás

— ige vagy főnév?
  ha az előző szó személyes névmás, a következő pedig nem ige, vagy pedig olyan ige, amely személyben és számban nem egyezik a személyes névmással, akkor ige
  ha az előző szó névelő, amely számban egyezik a vizsgált szóval, akkor főnév

Ezeket a meglehetősen egyszerű szabályokat használva végül is mintegy 80 százalékban helyes elemzéshez jutnak.

d. Tőtár által vezérelt elemzők

A tőtárból kiinduló eljárások az első lépésben rendszerint levágják a szóalakra balról illeszthető leghosszabb szótövet, majd a megmaradó részt (ha van) toldalékként kísérlik meg elemezni. Ilyen például a Siemens cégnél készült általános, nyelvfüggetlen morfológiai elemző algoritmus (MEYA 1983, THURMAIR 1983) és ennek német nyelvre fejlesztett

speciális változata is (GEHRKE ÉS BLOCK 1986). Ez utóbbit természetes nyelvű interfész morfológiai elemző moduljaként készítették. Alkalmazkodva a német nyelv sajátosságaihoz, az elemzéshez morfématárat használnak. Ez azt jelenti, hogy csak az elemi töveket, a képzőket és a ragokat tárolják a lexikonban, az összetett és képzett szavakat nem. A lexikon minden elemét kiegészítik a morféma típusának kódjával (pl. igető, főnévképző stb.). Az elemzéskor a szövegszóhoz balról illesztik a leghosszabb lehetséges szótövet, majd a megmaradó részt tovább keresik a lexikonban. (Mivel egy lexikonban vannak a tövek és végződések, egységesen kezelhetők.) A feltételezett szegmentálások előtt ellenőrzik, hogy ez lehetséges felbontás-e. Eredményként mindig megadják az összes lehetséges felbontást: első helyen a balról leghosszabban illeszthető megoldásokat közlik. A lexikonban fastruktúra segítségével kereshetünk, a fa levelei pointerek az úton leírt karaktersorozat helyéhez. Így egy lépésben megkapjuk a leghosszabb és az összes rövidebb balról illeszthető karakterfüzér helyét. A lexikai és inflexiós analízist egy véges állapotú automata végzi, amely képzők esetén „megjegyzi," hogy az adott képző milyen hatással van a továbbiakban lehetséges toldalékokra.

A NSz. számára készített elemző program

Ennek a programnak elsődleges célja az, hogy a későbbiekben lehetővé tegye lexémára rendezett konkordancialisták készítését, továbbá, hogy megkönnyítse a korpuszból való több szempontú lekérdezéseket. Az eljárással szemben támasztott legfontosabb követelmények az alábbiakban foglalhatók össze:

— Helyesen lemmatizáljon, tekintetbe véve a tőváltozatokat és alakváltozatokat is.
— A lemmatizált illetőleg szegmentált változaton túl őrizze meg az eredeti szövegszót is.
— Az elemzett szöveg az eredetivel együtt is a lehető legkevesebb helyet igényelje, ugyanakkor legyen egyértelmű megfeleltetés a szövegszavak és elemzett változataik között.
— A szegmentálás, amennyire lehet, automatikus legyen, ezt követően azonban minden esetben szükség van emberi erővel történő ellenőrzésre és javításra. Az eredményül kapott elemzett szövegnek gyakorlatilag hibátlannak kell lennie!

Figyelembe véve ezeket az elvárásokat, olyan algoritmus megvalósítása mellett döntöttünk, amely tőtárból kiindulva végzi az elemzést. Így ugyanis lehetővé vált, hogy minden esetben meg tudjuk adni a szótári szóalakot, még akkor is, ha az eltér a szótőtől. A helyes elemzéshez a szófaj meghatározására is szükség van, ezt az információt is a tőtárból kapjuk. Képzett szavak és összetételek esetén, ha a szót nem találjuk meg a tőtárban, de az elemeit igen, a program felismeri a szótövek és képzők határát. Mivel azonban első lépésként mindig a leghosszabb balról illeszthető szótövet azonosítjuk, a képzett és összetett szavak egy részét megtalálja a program a tőtárban, ezért ilyenkor csak akkor jelöljük meg a képzés, illetőleg összetétel határát, ha a tőtárban szerepel valamilyen morfémahatár-jel.

A homonimák, illetőleg homográfok helyes elemzését — egyelőre — nem tűztük ki célul. Ezeket az utólagos ellenőrzéskor kell majd kiszűrni. A program ehhez mindössze

annyi segítséget ad, hogy *-gal jelöl meg minden többféleképpen elemezhető alakot, ezzel hívja fel az ellenőr figyelmét a problémás esetekre.

Az elemző program főbb lépései:

Megvizsgáljuk, hogy a szövegszó nem azonos-e valamely lexémával.

Ha igen, a szófajkóddal és — ha van — a homonimakóddal együtt kiírjuk a lexémát. Homonimáknál *-ot teszünk annak jelzésére, hogy bizonytalan az elemzés.

Ha a tőtárban nem találunk olyan lexémát, amely az adott szövegszónak töve lehetne, az egész szövegszót kiírjuk. Mellette *! jelzi, hogy új szótőt vagy hibás szót találtunk.

Ha találtunk megfelelő tövet (amennyiben többet találtunk, kiválasztjuk a leghosszabbat), levágjuk a szövegszóról, és a megmaradó részt megpróbáljuk megkeresni a toldaléktárban. Ha az egész toldaléktömböt megtaláljuk, megvizsgáljuk, hogy ez a toldalék állhat-e a megtalált szótő után. (Egyelőre csak a szófajt vizsgáljuk, nem ellenőrizzük, hogy helyes toldalékváltozatot használtunk-e.) Ha megfelelő toldalékot találtunk, kiírjuk a szótövet. Ezt speciális zárójel-pár közé téve követik a kódok, ill. a lexéma, amennyiben nem egyezik meg a szótővel. A program kiírja a szófajkódot, a homonimakódot, majd — ha van — a toldalék és annak kódja következik. A kódok után mindig bezárjuk a speciális zárójelet (pl. az „ablakot" elemzett változata: `ablak≪FN≫ot≪ACC≫`; a „lovaknak" elemzett változata: `lov≪ló/FN≫ak≪PL≫nak≪DAT≫`).

Ha a toldaléktárban nem találjuk meg a teljes toldaléktömböt megkeressük a leghosszabb jobbról illeszthető toldalékot, és levágjuk a toldaléktömbről. Az ezután megmaradó toldalékot tovább keressük a toldaléktárban azok között a toldalékok között, amelyek a már megtalált toldalék előtt elvileg előfordulhatnak. Ezt mindaddig ismételjük, amíg a toldaléktömb hossza zérus nem lesz. A megtalált toldalékkódokat közben egy változóba gyűjtjük. Amikor a toldaléktömb hossza zérus, megvizsgáljuk, hogy a tárolt toldalékkódok következhetnek-e a szótő után. Ha nem, elölről kezdjük az elemzést úgy, hogy a toldaléktömböt másképpen próbáljuk felosztani. Ha így sem sikerül helyesen elemeznünk, megpróbálunk másik lehetséges tövet keresni a tőtárban, és erre elölről kezdeni az elemzést. Sikertelen elemzési kísérlet esetén a teljes szövegszót írjuk ki, *! jellel.

Az eredményül kapott file tartalma csaknem azonos a módosított szövegfile tartalmával, csupán — a morfológiai elemzés eredményeként — kibővül a tő és toldaléktömb határán elhelyezett szófajkóddal, esetleg homonimakóddal, és ha nem zérus morfémával szerepeltek a szövegben, toldalékkódokkal, esetenként a lexémával. Így tehát továbbra is rendelkezünk mindazokkal az információkkal, amelyet az eredeti szöveg tartalmazott, csupán további — a lexéma szerinti keresést, válogatást, rendezést megkönnyítő — adatokat nyertünk. A program által előállított, „nyers", elemzett szövegfile-t átvizsgáljuk, a csillaggal megjelölt részeken beírjuk a helyes elemzést, ill. a megfelelő szófaj- és homonimakódot. A *!-lel jelölt szavakról eldöntjük, valóban új lexémák-e, vagy esetleg egy már tárolt lexéma alakváltozatának tekintsük őket, és mint ilyeneket vegyük fel a tőtárba. A megfelelő új lexémát mind a tőtárba, mind az elemzett szövegfile-ba beírjuk. Esetenként előfordulhat, hogy az újnak látszó lexéma csupán rögzítési hiba eredménye. Ilyenkor a hibát kijavítjuk, és a helyes elemzést kézzel írjuk be.

Igekötők, összetételek

Az igekötők és az igék, valamint az összetett szavak tagjainak határát is a fent leírt módon, speciális zárójelpár közé tett szófajkóddal fogjuk jelölni, ezért az egybe- vagy különírt változat elemzett formája közt a különbség csupán annyi, hogy a különírt változatban az elemzett morfémák között szünet lesz, míg az egybeírt változatban a zárójelek között szereplő kódok és a morfémahatárt jelölő + jelek választják el egymástól a morfémákat. A + jelet csak olyankor tesszük ki, amikor nem jelek vagy ragok határát jelöljük. Ezzel különböztetjük meg a ragos, jeles szóalakokat az olyan, több morfémából álló szövegszavaktól, amelyek feltehetőleg új címszók lesznek (képzett, összetett szavak). Például, ha a eredeti szöveg:

```
Nem mondhatom el senkinek, elmondom hát mindenkinek...
```

akkor az elemzett szöveg:

```
Nem<MO> mond<IG+>hat<HAT>om<Te1> el<IK> senki<NM>nek<DAT>,
  el<IK+>mond<IG>om<Te1> hát<KO> mindenki<NM>nek<DAT>...
```

A kereső program a `mond nxt el<IK>` parancs hatására (l. 4.4.3) az *elmond* összes alakváltozatát kiírja, tekintet nélkül arra, hogy külön- vagy egybeírva szerepeltek-e. Ugyanígy kezeljük az összetett szavakat, ha külön vannak írva, úgy hagyjuk, majd a `nxt` vagy a `fby` parancsokkal megkeressük (ezeket lásd a 4.4.3 pontban). Hasonló módon oldható meg a régi szövegek néhány egybe- vagy különírási problémája: Mikes pl. előszeretettel írja egybe a névelőt az utána következő névszóval. Ilyenkor az elemzett változatban a két szót szünet helyett megint csak a zárójelek közé írt szófajkóddal és + jellel választjuk el.

Az így elemzett szövegben természetesen kereshetjük a szövegszavakat és környezetüket úgy is, mintha nem is elemeztük volna a szöveget. Az elemzett szövegből azonban lexémára rendezett konkordanciákat is készíttethetünk, ami nyilván nagy segítséget fog jelenteni a szócikkek írásakor.

Az elemző nagy vonalakban történt áttekintése után lássuk a megvalósítás részletkérdéseit.

A tőtár

A tőtár alapjául a Debreceni Tezaurusz és a *Szépprózai gyakorisági szótár* összefésült címszóanyaga szolgál. (Erről részletesebben l. KORNAI 1986.) Mivel ez az adatbázis csak szótári szóalakokat tartalmaz, ezekből az összes lehetséges tőváltozatot is elő kell állítanunk ahhoz, hogy a program felismerhesse a toldalékolt alakokat. Ennek érdekében a címszójegyzéket kiegészítjük az Elekfi László *Szókincsünk nyelvtani alakrendszere* című kéziratos munkájában található, a szavak ragozási típusára utaló kódokkal. E kódokat felhasználva egy tőgeneráló program állítja elő az összes lehetséges tőalternánst, majd elhelyezi őket a tőtárban. A tőtár ezen kívül tartalmazza a szavak szófaj- és homonima-kódját is. Azonban a homonimák jelentős részét a program úgysem tudja helyesen szétválasz-

tani, ezért a tőtárnak olyan „kivonatolt" változata is lesz, amelyben bizonyos homonimák közül csak az egyik szerepel; mellette csillag jelzi, hogy ez a szó többféleképpen is elemezhető, tehát utólag kell a megfelelő elemzést beírni. Ez a megoldás azért tűnt célszerűnek, mert ha a program mindig megvizsgálná az összes lehetséges homonimát — olyankor is, amikor végül nem tudja eldönteni, melyik alakról van szó —, rengeteg időt fecsérelne el, mert pl. az *a* névelő minden előfordulásánál végigolvasná mind az öt homonim alakot, majd megállapítaná: bizonytalan az elemzés. Ehelyett a kivonatolt tőtárban csak egy, csillaggal megjelölt *a* szerepel majd, névelő szófaji minősítéssel. A program minden, a szövegben előforduló *a* szóról azt tételezi majd fel, hogy névelő, de csillaggal jelöli meg, hogy felhívja az ellenőr figyelmét az esetleges hibás elemzésre. A tőtárnak eszerint két változata lesz: a teljes tőtárban megtalálható lesz minden címszó, amely vagy az Értelmező Szótárban, vagy az Értelemző Kéziszótárban, vagy a Gyakorisági Szótárban címszóként szerepelt (külön kóddal jelöljük meg, melyikben szerepelt, melyikben nem), homonimák esetén annyiszor, ahány homonim alak van. Ebben a változatban megőrizzük mindhárom szótár szófaji besorolását is, és az új, általunk használt, egyszerűsített szófajkódokat is. (Ezeket l. a mellékletben.) Itt tároljuk a *Szókincsünk nyelvtani alakrendszeré*ből és az Elekfi László által készített, megjelenés alatt lévő *Ragozási szótár*ból származó kódokat is, továbbá az előbbiben használt különböző elválasztó jeleket (pl. összetétel határ, igekötő stb.). Külön mezőkben tároljuk a kódok segítségével előállított tővariánsokat, illetve a tőtől eltérő szótári szóalakokat. Két rendezési segédmezővel gondoskodunk arról, hogy a címszavakat a magyar helyesírási szabályzatban lefektetett ábécérend szerint tudjuk rendezni. Egy megjegyzés rovatban tároljuk azokat a kiegészítő információkat, amelyek az elemzés kézi javítása során hasznosak lehetnek (pl. a homonimák jelentése). Megőrizzük továbbá a *Gyakorisági Szótár*ból származó ún. F-kódot, amely a szó előfordulási gyakoriságára utal (l. 1. melléklet), ennek segítségével ugyanis mindig könnyen készíthetünk kisebb, csak a leggyakoribb szavakat tartalmazó tőállományokat.

A kivonatolt tőtárállományban csak azok az adatok szerepelnek, amelyekre az elemző programnak szüksége lehet. Ezek: a címszó, a tőváltozatok és alakváltozatok, szófajkód, homonimakód és esetenként az ezt kiegészítő csillag. A csillaggal megjelölt homonimák itt csak egyszer fordulnak elő: egyelőre nem másoljuk át ide azokat a címszavakat, amelyek a *Gyakorisági Szótár*ból származnak, de másutt nem szerepeltek.

A toldaléktár

A toldaléktár az összes lehetséges igei és névszói ragon és jelen kívül számos képzőt is tartalmaz. A ragokat és jeleket minden esetben „levágjuk" a szövegszóról (pontosabban bejelöljük a tő és a toldalékok határát), a képzőket azonban csak akkor, ha a képzett forma nem fordult elő a tőtárban. (Ezentúl, ha csak külön nem jelezzük, tőtáron mindig a kivonatolt tőtárat értjük.) Az igeragokat — a hatékonyabb működés érdekében — egy tömbként kezeljük, azaz nem választjuk szét az időre, személyre, számra és módra utaló morfémákat. Az ún. kötőhangokat az egyszerűség kedvéért a toldalék részének tekintjük. A tőtárban találhatunk „fiktív" alakokat is, például a *vá/vé* toldalékot *á/é* alakban is tároljuk, azzal a megszorítással, hogy előtte kettős mássalhangzónak kell állnia, méghozzá úgy, hogy a kettős mássalhangzó második tagját elhagyva az elemzés folytatható legyen.

(Ezt — a toldaléktárban lévő kódot felhasználva — a program ellenőrzi.) Szintén fiktívek az *a/e/ja/je* változataként felvett *á/é/já/jé* alakok is, ezek után valamilyen ragnak kell következnie.

A toldaléktárban minden egyes alakot külön rekordban tárolunk, a toldalék kódjával ellátva. Annak érdekében, hogy a toldalékokat jobbról balra illesztve kereshessük, a toldalékokat a-tergo formában is tároljuk, az adatbázis erre a mezőre van indexelve. Az elemzendő toldalékok listája a következő:

igeragok:

```
e1 = <ok,ek,ök>
e2 = <sz,asz,esz,ol,el,öl>
e3 = <0,ik>
t1 = <unk,ünk>
t2 = <tok,tek,tök,otok,etek,ötök>
t3 = <nak,nek anak, enek>
Te1 = <om,em,öm>
Te2 = <od,ed,öd>
Te3 = <ja,i>
Tt1 = <juk,jük,°uk,°ük,>
Tt2 = <játok,°itek>
Tt3 = <ják,°ik>
Me1 = IINe1 = <tam,tem,ttam,ttem,ottam,ettem,öttem>
Me2 = IINe2 = <tad,ted,ttad,tted,ottad,etted,ötted>
Me3 = IINe3 = <ta,te,tta,tte,otta,ette,ötte>
Mt1 = IINt1 = <tuk,tük,ttuk,ttük,ottuk,ettük,öttük>
Mt2 = IINt2 = <tátok,tétek,ttátok,ttétek,ottátok,ettétek,
              öttétek>
Mt3 = IINt3 = <ták,ték,tták,tték,ották,ették,ötték>
Fe1 = <nék,anék,enék,nám,ném>
Fe2 = <nál,nél,anál,enél>
Fe3 = <na,ne,ana,ene,nék>
Ft1 = <nánk,nénk,anánk,enénk>
Ft2 = <nátok,nétek,anátok,enétek>
Ft3 = <nának,nének,ananak,enének>
Pe1 = <jak,jek,°ak,°ek>
Pe2 = <jál,jél,°ál,°él,j,°0>
Pe3 = <jon,jen,jön,°on,°en,°ön,ék>
Pt1 = <junk,jünk,°unk,°ünk>
Pt2 = <jatok,jetek,°atok,°etek>
Pt3 = <janak,jenek,°anak,°enek>
TFe1 = <nám,ném,anám,eném>
TFe2 = <nád,néd,anád,enéd>
TFe3 = <ná,né,aná,ené>
TFt1 = <nánk,nénk,anánk,enénk>
TFt2 = <nátok,nétek,anátok,enétek>
```

```
TFt3 = <nák,nék,anák,enék>
TPe1 = <jam,jem,°am,°em>
TPe2 = <jad,jed,°ad,°ed>
TPe3 = <ja,je,°a,°e>
TPt1 = <juk,jük,°uk,°ük>
TPt2 = <játok,jétek,°átok,°étek>
TPt3 = <ják,jék,°ák,°ék>
 Ie1 = <lak,lek,alak,elek>
IMe1 = <talak,telek,ttalak,ttelek,ottalak,öttelek,
       ettelek>
IFe1 = <nálak,nélek,análak,enélek>
IPe1 = <jalak,jelek,°alak,°elek>
```

Igenévképzők:

```
MIF = <ó,ő>
MIA = <andó,endő>
MIB = <t,tt,ott,ett,ött>
HIN = <va,ve,ván,vén>
 FI = <ni>
```

Főnévi igenév személyragos alakjai:

```
INRe1 = <nom,nem,nöm,anom,enem,önöm>
INRe2 = <nod,ned,nöd,anod,ened,önöd>
INRe3 = <nia,nie,ania,enie>
INRt1 = <nunk,nünk,anunk,enünk>
INRt2 = <notok,netek,nötök,anotok,enetek,önötök>
INRt3 = <niuk,niük,aniuk,eniük,niok,niök>
```

Melléknév jelek és ragok:

```
(FFOK)+MN+(KFOK)
(FFOK)+MN+(KFOK)+(PERS)+(POSS)+CAS
(FFOK)+MN+(KFOK)+(PL)+(POSS)+CAS
```

Főnév jelek és ragok:

```
FN+(PL)+(POSS)+CAS
FN+(PERS)+(POSS)+CAS
```

Ahol:

```
   PL = <k,ak,ek,ok,ök>
  POS = <é,éi>
PERSe1 = <m,am,em,om,öm>
PERSe2 = <d,ad,od,ed,öd>
PERSe3 = <a,e,ja,je,á,é,já,jé>
PERSt1 = <nk,unk,ünk>
PERSt2 = <tok,tek,tök,atok,etek,ötök>
PERSt3 = <°uk,°ük,juk,jük>
```

```
PERSe1i = <im,aim,eim>
PERSe2i = <id,aid,eid>
PERSe3i = <i,ai,jai,jei>
PERSt1i = <ink,aink,eink,jaink,jeink>
PERSt2i = <itok,itek,jaitok,jeitek>
PERSt3i = <ik,aik,eik,jaik,jeik>
CAS:
    NOM = <0>
    FOR = <ként,képp,képpen>
    TEM = <kor>
    CAU = <ért>
    TER = <ig>
    DAT = <nak,nek>
    SUB = <ra,re>
    DEL = <ról,ről>
    INE = <ban,ben>
    ELA = <ból,ből>
    ILL = <ba,be>
    ADE = <nál,nél>
    ABL = <tól,től>
    INS = <val,vel,°al,°el>
    SOC = <stul,stül>
    FAC = <vá,vé,°á,°é>
    ALL = <hoz,hez,höz>
    SUP = <n,on,en,ön>
    ACC = <t,at,et,ot,öt>
  FFOK = <leg,leges,legesleg,legisleg>
  KFOK = <bb,abb,obb,ebb,öbb>
```

Képzők, amelyeket esetenként — amikor a tőtárban nincs meg a képzett szó — fel kell ismernünk:

Főnevek után:

```
BELI   = <beli,fajta,féle>
FNI    = <ás,és>
FFOSZT = <tlan,tlen,atlan,etlen,talan,telen>
FELE   = <féle>
EK     = <ék>
SAG    = <ság, ség>
```

Igék után:

```
MUV    = <at,et,tat,tet>
GYAK   = <gat,get,ogat,eget,öget>
HAT    = <hat,het>
VISSZ  = <ód, ed>
IFOSZT = <atlan, etlen>
IKEP   = <i>
```

```
SKEP    = <s, as, os, es, ös>
UKEP    = <ü, ú, jú, jü>
```

Melléknevek után:

```
MN+KFOK+KIEM KIEM = <ik>
ESS               = <ul,ül,n,an,en>
```

Számnevek után:

```
MAGA  = <maga>
RESZ  = <rész>
ANNYI = <annyi>
A     = <a,e>
MUL   = <szor,szer,ször>
TORT  = <ad,od,ed,öd>
MOD   = <lag,leg>
SZAM  = <szám, számra>
DIST  = <nta, nte, anta, onta, ente, önte>
DIS   = <nként, anként, onként, enként, önként>
```

Az elemző program működése

A program jelenlegi — kísérleti — változata Turbo Pascalban készült (3.0-ás verzió), az adatbáziskezelő funkciókat a DACCESS programrendszerrel oldottam meg. A program tőtárul egyelőre a Gyakorisági Szótár mintegy 6000 leggyakoribb címszavából készített leszűkített tőfile-t használja, amelyben a tővariánsok és homoníma kódok nem szerepelnek (mivel sajnálatos módon az ezek alapjául szolgáló ún. Elekfi-kódoknak még csak töredékét sikerült a tőtári adatbázisba felvinni.). A toldaléktár azonban gyakorlatilag teljes, bár a használat során még kiderülhet, hogy további toldalékokat is célszerű felvennünk. Mind a tőtárat, mind a toldaléktárat DBASE III formátumú file-ban tároljuk, a könnyű kezelhetőség érdekében.

A program, miután beolvasta az elemzendő szövegszót, megkisérli megtalálni az egész szót a tőtárban. Ha nem találja, a szövegszó első három karakterét keresi. Azokat a töveket, amelyek a szövegszóra balról illeszthetőek, elhelyezi egy tömbben, amikor már nem talál több illeszthető tövet, előveszi a tömb leghosszabb elemét. Ezt levágja a szövegszó bal oldaláról, majd a maradékot megfordítja, és ezt próbálja megtalálni a toldaléktárban. Ha a teljes maradékot nem sikerül egy lépésben értelmezni, mindig a leghosszabb illeszthető toldalékot választja a lehetségesek közül, az elemzés itt jobbról balra halad. A program attól függően, hogy milyen toldalékot választott le, mindig új állapotba kerül, az állapot-kóddal tartjuk nyilván, hogy melyek az ezután következő lehetséges állapotok. (Azaz, ha az *alhatnak* szó végén álló *-nak* ragot esetragként értelmezte, a *-hat* igeképzőt nem fogja ezelőtt elfogadni. Így kénytelen lesz visszalépni, és előveszi a toldaléktömbből a *-nak* igeragot. Ekkor már az előtte álló igeképzőt fel fogja ismerni.) Ha az automata végállapotba ér, és a toldaléktömb hossza nem nulla, a program jelenlegi verziójában hibás elemzést jelez. Később azonban továbbfejlesztem oly módon, hogy a megmaradó karaktersorozatot ilyenkor próbálja megkeresni a tőtárban. Így módunk lesz a tőtárban nem található összetett szavak elemzésére is. Ha a toldaléktömb hossza nulla az elemzés végén, a program ellenőrzi, hogy a szótő szófaja összhagban van-e az utoljára felismert

toldalékkal. Ezt az ellenőrzést azért kellett utoljára hagynunk, mert a képzők számos esetben megváltoztatják a lexéma szófaját, így amikor pl. a *csinálóknak* szót elemzi a program, és első lépésként megtalálja a *csinál* igét a tőtárban, még nem dönthető el, hogy a *-nak* igei vagy névszói toldalékként való értelmezése lesz-e a helyes, mindaddig, amíg a teljes elemzést végre nem hajtottuk. Ezt a problémát megkerülhetnénk oly módon, hogy a toldalékokat nem balról jobbra, hanem jobbról balra haladva keresnénk a toldaléktárban. Ebben az esetben azonban nehezebben tudnánk megoldani az összetett szavak elemzését. (Akkor ugyanis minden egyes karaktersorozatról először feltételeznünk kellene, hogy összetett szó eleme, és csak miután ezt a lehetőséget elvetettük, kereshetnénk a toldaléktárban. Mivel a szavak többsége nem összetett, de toldalékolt, ez rendkívül sok felesleges szótárhoz fordulással járna.)

Néhány példa a program által készített elemzésre:

szövszó = csinálóknak	tő = csinál	kódok = +MIF+PL+DAT
szövszó = csinálnak	tő = csinál	kódok = +t3
szövszó = hatodikai	tő = hatodik	kódok = +PSe3i !
szövszó = hatodikos	tő = hatodik	kódok = +SKEP
szövszó = tizedikei	nincs ilyen szó a tőtárban!	
szövszó = tízedikei	tő = tíz	kódok = +TORT+KIEM+A+IKEP !
szövszó = negyedikeseket	tő = negyedik	kódok = +SKEP+PL+ACC
szövszó = házambeliek	tő = ház	kódok = +PSe1+BELI+PL
szövszó = falumbelieket	tő = falu	kódok = +PSe1+BELI+PL+ACC
szövszó = lányomék	tő = lány	kódok = +PSe1+EK
szövszó = látott	tő = lát	kódok = +Me3
szövszó = nézett	tő = nézet	kódok = +ACC !
szövszó = lássalak	nincs ilyen szó a tőtárban!	
szövszó = láthatjuk	tő = lát	kódok = +HAT+TPt1 !
szövszó = nézzük	tő = néz	kódok = +TPt1
szövszó = néztem	tő = néz	kódok = +TMe1
szövszó = kézzel	tő = kéz	kódok = +INS
szövszó = házzá	tő = ház	kódok = +FAC
szövszó = magasabbak	tő = magas	kódok = +KFOK+PL
szövszó = látottakat	tő = lát	kódok = +MIB+PL+ACC
szövszó = láthatóbbakat	tő = látható	kódok = +KFOK+PL+ACC
szövszó = csinálhatóbbakat	tő = csinál	kódok = +HAT+MIF+KFOK+PL+ACC

Ezzel a néhány példával jól szemléltethetjük a program előnyeit és hátrányait egyaránt. Előnye, hogy rendkívül bonyolult alakokat is helyesen ismer fel (pl *csinálhatóbbakat*), a képzett szavakat rugalmasan kezeli (mivel a *látható* szerepelt a tőtárban, csak innen kezdve „szeletel"), meg tudja különböztetni a homonim toldalékokat — legalábbis akkor, ha kölünböző szófajokhoz járulhatnak. Így helyesen ismeri fel, hogy a *csinálóknak* esetében a *-nak* dativus, míg a *csinálnak* esetében igerag, továbbá, hogy a *látott -ott* toldaléka nagy valószínűséggel múlt idejű igerag, míg a *látottak* esetében csak befejezett

melléknévi igenévi toldalék jöhet szóba. A leghosszabb balról, illetőleg jobbról illeszthető karaktersorozat preferálása azonban időnként helytelen eredményre vezet: ilyen a *nézett* és a *hatodikai* helytelen elemzése. Ha további ellenőrzéseket iktatnánk be, ezek a hibák — legalábbis részben — kiszűrhetők lennének. Beépíthetnénk például egy olyan ellenőrzést, amely megállapíthatná, hogy *-t* alakú tárgyrag nem következhet *t*-re végződő szavak után, ez azonban azt jelentené, hogy minden egyes tárgyragot ellenőriznünk kellene. Ugyanígy kiszűrhetnénk a *láthatjuk* rossz elemzését is, ha ellenőriznénk, hogy a *-juk* előtt nem *t* van-e stb. Mindez azonban jelentősen tovább lassítaná a máris elfogadhatatlan sebességű elemzőt. Mivel a tővariánsokat egyelőre nem tudtuk előállítani, nem találja meg sem a *tiz*, sem a *lá(s)* tövet a tőtárban, a helytelen formájú *tízedikei* alakot azonban helyesen elemzi.

A program eszerint több szempontból is továbbfejlesztésre szorul. A jelenlegi verzió már lassúságánál fogva (egy-egy szó elemzése 2–20 sec) sem alkalmas üzemszerű használatra, azonban számos hasznos tapasztalatot nyertünk készítése közben. Az algoritmus „lelke" megfelelőnek látszik, az állapotkód segítségével jól sikerült ellenőrizni az elemzés menetét, működnek a „fiktív" toldalékok felismerésére szolgáló rutinok, a program az esetek többségében felismeri, hogy az azonos alakú toldalékok közül melyik fordult elő az adott esetben. A program gyorsítása érdekében először is újraírom az adatbáziskezelő részeket, a DACCESS helyett a CLIPPER programot használva erre, mivel ennek SOFTSEEK opciójával lehetőségem lesz a leghosszabb balról illeszthető tő, illetve jobbról illeszthető toldalék egy lépésben való keresésére. Mivel a CLIPPER csak C nyelvű szubrutinok hívására ad lehetőséget, a program többi részét C-ben fogom megírni. Célszerűnek tűnik továbbá a toldaléktárat a programfutás elején a memóriába másolni, mivel a lemezen való keresés rendkívül lassúnak bizonyult. A gyorsításon kívül, ki kell egészíteni a programot a tővariánsokat felismerő részekkel is, továbbá a korábban említett összetett szó elemző rutinnal is.

4.4. Az adatok feldolgozása

EBBEN A FEJEZETBEN csak vázlatosan tudom ismertetni a szótári korpusz feldolgozására fejlesztendő programrendszert, elsősorban azért, mert a konkrét megvalósítás annak a függvénye, hogy végül is milyen számítógépen dolgozhatjuk fel az anyagot. A jelenleg rendelkezésünkre álló IBM kompatibilis személyi számítógépek ugyanis csak az adatok rögzítésére és előfeldolgozására alkalmasak. Készíthető természetesen konkordanciaprogram is ilyen típusú számítógépen — a III. 3.-ban ismertetett WORDCRUNCHER ilyen például — azonban a tervezett korpusznak csupán töredéke lenne elfogadható módon kezelhető — lekérdezhető ezeken a számítógépeken. Már az eddig lerögzített mintegy 4 millió szövegszónyi anyag sem férne el nyers állapotban egyetlen 20 Megabyte-os fix lemezen, az elemzett szöveg mérete ennek várhatóan csaknem kétszerese lesz. A teljes rögzített korpusz elemzetlen állapotban pedig több mint négyszerese lesz a jelenleginek. Floppy diszkek százain tárolva az anyag természetesen kezelhetetlen, ezek csupán archiválásra használhatók. Mindent összevéve a szótári korpusz feldolgozására csak akkor vállalkozhatunk, ha megfelelő központi- és háttérmemóriával rendelkező számítógépünk lesz. Mivel a legújabb szövegkezelő programrendszerek általában UNIX operációs rendszer alatt fut-

nak, ilyet használó számítógép beszerzése látszik célszerűnek. Ebben az esetben a III. 3.-ban ismertetett konkordanciaprogramok közül megvásárolhatnánk a leghatékonyabbnak tűnő PAT programot, amelyet továbbfejleszthetnénk saját céljainkra. (Ezzel ráadásul egyúttal nemcsak a korpuszban való hatékony keresést oldhatnánk meg, hanem a készítendő szótári adatbázist is részben ugyanezzel kezelhetnénk.) Természetesen megfelelő méretű és minőségű számítógépes munkacsoporttal magunk is vállalkozhatnánk egy hasonló tulajdonságú programrendszer kifejlesztésére. Az alábbiakban azt ismertetem, hogy egy PAT típusú programrendszer fejlettebb változatával hogyan oldhatnánk meg a nagyszótári feldolgozás terveiben szereplő legfontosabb feladatokat.

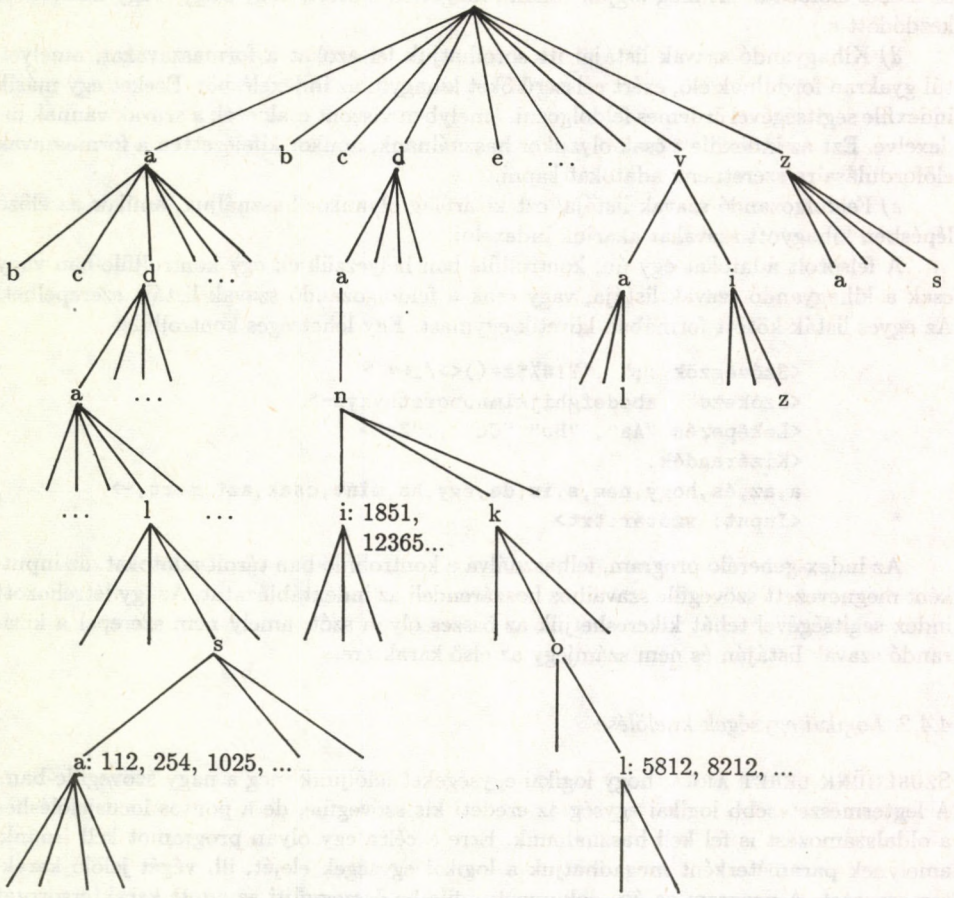
4.4.1. Index előállítás

A HATÉKONY KERESÉSHEZ először is indexfile-okat kell készítenünk. Mint a II.2.1-ben, a Trésor projektumánál láttuk, ez készíthető több lépcsőben is, használhatunk egy durva- és egy finomindex táblázatot. Ekkor azonban egy-egy szó összes előfordulásának kikeresése meglehetősen lassú lenne, igaz ugyan, hogy a szövegfile-ok javítása esetén az indexek újragenerálása viszonylag kevés időt emésztene fel. Mivel azonban a mi korpuszunkat az adatgyűjtés és előfeldolgozás során számtalanszor javítjuk, a feldolgozásra kerülő szövegekben a hibák száma már remélhetőleg elenyésző lesz. Egyébként sem látszik célszerűnek a szövegek folyamatos javítgatása, mivel eközben esetleg újabb hibákat vihetünk fel. Elégségesnek tűnik az is, ha az észlelt hibákat pontos helymegjelöléssel folyamatosan gyűjtjük, és időnként (egy évben egyszer), az összes hibát kijavítjuk, és a karbantartás után a teljes állományt újra indexeljük. Ilyen munkaszervezés esetén kevésbé problematikus, ha az indexkészítés hosszadalmas, mintha rövid ideig indexelünk ugyan, de egy-egy szó összes előfordulását percekig keresi a program. Mindezt figyelembe véve, az indexgenerálás előtt az összes elemzett szövegfile-t egyetlen hatalmas file-ba egymás után másoljuk, az index pedig a szavaknak ebben a file-ban elfoglalt helyére fog utalni.

Maga az index fastrukturájú lesz. A fában leírt utak az egyes szókezdő karaktersorozatok, a fa levelei pointerek a karaktersorozatok szövegbeli előfordulásaihoz (ld. 3. ábra). Így ha keressük például az összes *állam-* kezdetű szót, a kereső program megvizsgálja a fában a megfelelő utat, összegyűjti az ebben a csomópontban és alatta tárolt valamennyi pointert, és azonnal meg tudja mondani, hányszor fordult elő a szó a teljes szövegben. Majd kérésre kiírja az összes előfordulást, vagy ezek közül néhányat. Ahhoz, hogy a program ezt az automatikus indextáblázatot elkészíthesse, meg kell adnunk néhány adatot:

a) A szóvégző karakterek listája (írásjelek és egyéb segédjelek, szünet stb.). Ez többféleképpen is megadható. Készíthetünk olyan indexfile-t, amelyben csak a szóközt és a szóvégző írásjeleket tekintjük szóvégnek, ekkor a teljes elemzett szövegszót egy szóként kereshetjük, és egy olyat is, ahol a morféma elhatároló jeleket is szóvégnek tekintjük, ebben az esetben pedig morfémák szerint csoportosított konkordanciákat készíthetünk.

b) Szókezdő karakterek listája. Itt célszerű megadni az összes betűt és a kötőjelet. Ennek eredményeként egyrészt a számjegyekkel kezdődő szavak (kódok vagy számnevek) nem kerülnek be az indextáblázatba, ami azért hasznos, mert nem valószínű, hogy bárki számjegyeket akarjon keresni, másrészt meg fogjuk találni a kötőjellel írt szava-



3. ábra
A szöveghez tartozó indexfa felépítése

kat (pl. „-e"). Többféle szókezdő állománnyal is készíthetünk természetesen különböző indexeket, akárcsak a szóvégző karakterek esetében.

c) Karakterek leképezése: itt azokat a karakterpárokat adjuk meg, amelyeket a keresés szempontjából azonosnak akarunk tekinteni. Általában a nagybetűket és a kisbetűket nem akarjuk megkülönböztetni, amikor egy szót keresünk, így rendszerint ezeket soroljuk itt fel. (Ettől a nagybetűk nem „változnak át" kisbetűkké, de amikor keresünk egy szót, az összes előfordulását meg fogjuk találni, függetlenül attól, hogy nagy- vagy kisbetűvel kezdődött-e.)

d) Kihagyandó szavak listája: itt sorolhatjuk fel azokat a formaszavakat, amelyek túl gyakran fordulnak elő, ezért célszerű őket kihagyni az indexelésből. Ezeket egy másik indexfile segítségével érdemes feldolgozni, amelyben viszont csak ezek a szavak vannak indexelve. Ezt az indexfile-t csak olyankor használnánk, amikor kifejezetten a formaszavak előfordulásaira szeretnénk adatokat kapni.

e) Feldolgozandó szavak listája: ezt kizárólag olyankor használjuk, amikor az előző lépésben kihagyott szavakat akarjuk indexelni.

A felsorolt adatokat egy ún. kontrollfile-ban helyezzük el; egy kontrollfile-ban vagy csak a kihagyandó szavak listája, vagy csak a feldolgozandó szavak listája szerepelhet. Az egyes listák kötött formában követik egymást. Egy lehetséges kontrollfile:

```
<Szóvégzök :;".,'?!#%$&*()<>/_+= >
<Szókezdök abcdefghijklmnopqrstuvxyz->
<Leképezés "Aa", "Bb" "Cc"..."Zz">
<Kizárandók:
a,az,és,hogy,nem,s,is,de,egy,ha,mint,csak,azt,mert,->
<Input: szótár.txt>
```

Az index-generáló program, felhasználva a kontrollfile-ban tárolt adatokat, az inputként megnevezett szövegfile szavaihoz hozzárendeli az indextáblázatot. Az így létrehozott index segítségével tehát kikereshetjük az összes olyan szót, amely nem szerepel a kizárandó szavak listáján és nem számjegy az első karaktere.

4.4.2. Logikai egységek kijelölése

SZÜSÉGÜNK LEHET ARRA, hogy logikai egységeket jelöljünk meg a nagy szövegfile-ban. A legtermészetesebb logikai egység az eredeti kis szövegfile, de a pontos locusjelöléshez a oldalszámozást is fel kell használnunk. Erre a célra egy olyan programot kell írnunk, amelynek paraméterként megadhatjuk a logikai egységek elejét, ill. végét jelölő karaktersorozatot. A program az ún. dokumentumfile-ba összegyűjti az adott karaktersorozat összes előfordulásának kezdő karakterpozícióit. A korábban ismertetett formával rendelkező szövegekhez olyan dokumentumfile-t készíthetünk, amely tartalmazza a szövegfile-ok kezdetének helyét (az `<eleje>` tagoló jel kezdőpozícióit), és a szövegek végének helyét (a `<vege>` tagoló jel kezdőpozícióit). Ezen kívül célszerű egy másik dokumentumfile-t készítenünk, amely az oldalszám-jelek kezdőpozícióját mutatja. E két dokumentumfile-t felhasználva tudjuk majd a keresőprogram eredménysorában kiírni a forráskódokat, és annak az oldalnak a számát, ahol a keresett szó előfordult.

A dokumentumfile-ok előállítására szolgáló program lényegében ugyanúgy működik, mint a szövegindexelő program, az egyetlen különbség, hogy ez a program egyszerre csak egy vagy két karaktersorozatot indexel (attól függően, hogy különböző jeleket használunk-e a logikai egységek elejének és végének jelölésére).

Ugyanez a program nagyon hasznos lesz majd akkor is, amikor a szócikkeket szerkesztjük, ennek segítségével kijelölhetjük majd a szócikkeken belüli szerkezeti elemek elejét/végét, és így lehetővé tehetjük a struktúrán belüli keresést (nem mindegy pl., hogy az összes *mond-* kezdetű címszót keresem, vagy a *mond* összes előfordulását az idézetekben, vagy az összes előfordulását a jelentésdefiniciókon belül).

4.4.3. A kereső-program

A KERESŐ-PROGRAM fő feladata az lesz, hogy kiírja egy-egy szó, morféma vagy szókapcsolat tetszőleges méretű konkordanciáit. A program úgy működik, hogy miután begépeljük a keresett karaktersorozatot (szót, morfémát, szókezdetet stb.), az index fában a program megkeresi azt az utat, amely az adott karatersorozatra vonatkozik. Összegyűjti a csomóponton és alatta talált pointereket, és válaszként kiírja, összesen hány ilyen pointert talált — azaz mennyi volt a szó, ill. karaktersorozat abszolút gyakorisága. A konkordanciákat az alább ismertetendő `print` vagy `display` hatására írja majd ki, a felhasználó által definiált formátumban. A kereső program fölé készítenünk kell majd egy olyan programot, amely a kiírásnál — ha a felhasználó úgy látja jónak — kihagyja a ≪ ≫ jeleket és a közöttük lévő karaktersorozatokat. Ilyenkor tehát, noha a kereső program az elemzett file kódjait is felhasználja a lexéma szerinti kereséshez, az eredeti szöveget írathatjuk ki, az esetleg zavaró segédjelek nélkül.

Nyomtatás

A számunkra általában megfelelőnek tűnő nyomtatási formátumot egy print-kontrollfile segítségével alakíthatjuk ki. Ebben megadhatjuk a kinyomtatandó konkordanciák hosszát, megadhatjuk, hány karakternyi szöveget írjon ki a keresett szó előtt vagy után, és milyen locus-kód jelölést kérünk. Az alábbiakban egy lehetséges print-kontrollfile-t mutatunk be:

```
<Sorhossz 64>
<Elötte 30>
<Locus 1 := <kod>:10>
<Locus 2 := *****:4>
```

E parancsok hatására konkordanciasoraink formája a következő:

forráskód	*oldalszám*	*szöveg...*	*szövegszó*	*szöveg...*
(10 számjegy a `<kod>` után)	(4 szám a ***** után)	(30 karakter a keresett szó előtt)	(a keresett szó)	

Minden felhasználó definiálhat magának egy saját print-kontrollfile-t, ezt minden munkafolyamat elején aktivizálnia kell. Amennyiben nincsenek speciális igényei, használhatja az alapértelmezés szerinti print-kontrollfile-t, amely lényegében a fenti nyomtatási formátumot adja. Munka közben bármikor változtathatóak lesznek a nyomtatási formátumok, akár úgy, hogy csak néhány sort íratunk ki pl. nagyobb szövegkörnyezettel, akár úgy, hogy újabb nyomtatási formátum utasítás használatáig érvényes legyen a módosított formátum.

Pl.

`<Sorhossz := 200>` — A továbbiakban újabb `<Sorhossz>` utasításig 200 karakteres sorokat ír ki, de a szó előtt még mindig csak 30 karaktert ír.

`<Előtte := 85>` — Ezután 85 karakternyi szöveget ír ki a szövegszó előtt.

`<Utána := 100>` — 100 karaktert ír ki a szövegszó után mindaddig, amíg egy újabb `<Utána>` vagy `<Előtte>` vagy `<Sorhossz>` utasítás nem módosítja. Értelemszerűen, ha a sorhosszat definiáltuk, vagy az előtte, vagy az utána kiirandó karaktereket érdemes csak megadni, a másik már ebből adódik. Ha mégis mindkettőt megadjuk, a később megadott lesz érvényes. A sorhosszban megadott érték felülbírálhatja az előtte/utána kiírható karakterek számát. (Ti. az előtte/utána érték nem lehet nagyobb, mint a sorhossz.)

`<Locus 3 := <mufaj>10>` — Az eredeti szövegfile-ok elején lévő műfajmegnevezéseket is ki fogja ezután írni, a konkordancia sorok elején, az oldalszám után.

Ideiglenes nyomtatási formátum változtatás: lehetőség lesz arra, hogy a szövegfile megadott karakterétől kezdődően bizonyos számú karaktert kinyomtassunk. (Pl. `pr (1024),300`: kinyomtat az file 1024. karakterétől kezdve 300 karaktert, formázás nélkül).

Több szó vagy morféma együttes előfordulása:

Kétféle módon kereshetjük majd több szó együttes előfordulását: a `nxt` utasítás a szó jobb és baloldali környezetében egyaránt fog keresni, a `fby` csak a szó jobboldali környezetében keres. A keresési környezet mérete alapértelmezésben azonos lesz a print-kontrollfile-ban megadott kiíratási környezettel, bármikor megváltoztathatjuk azonban ezeket pl. a `nxt <10>` parancs hatására csak 10 karakternyi környezetben fog keresni.

A megtalált szavak kiírása

A keresésre először csak a szó előfordulásainak számát kapjuk meg válaszul, e szám alapján eldönthetjük, hogy az összes előfordulást ki akarjuk-e íratni, vagy ezek közül csak néhányat. A `print` parancs hatására az összes előfordulást nyomtatóra, a `disp` parancs hatására az összeset képernyőre írja. (A kiírás formáját az aktivizált print-kontrollfile,

ill. a kiadott formátum parancsok szabályozzák.) A `print <100>` hatására 100 véletlenszerűen kiválasztott példát kinyomtat, a `disp <34>` hatására 34 véletlenszerűen kiválasztott konkordancia sort a képernyőre ír. A `print last` hatására az utolsó `disp` utasítás eredményét fogja nyomtatóra írni.

Összetett vagy megszorító műveleti kifejezések

A keresett szavakból egyszerű műveleti kifejezéseket is létrehozhatunk, ha pl. keresni akarjuk a *mond* összes előfordulását, kivéve a *mondat*ot, azt kereshetjük így: `mond - mondat`. Leszűkíthetjük a keresett vagy kiírandó előfordulások helyét is különböző módokon. Pl. a `mond where <szerzo>= 'ARANY JA1NOS'` utasítás csak Arany műveiből keresi majd ki a *mond* előfordulásait, a `mond where <mufaj>= 'dra1ma'` a drámai művekből, a `mond where <kod>= '19'` a 19 századi művekből és így tovább. A program, miközben megvizsgálja a szöveghez tartozó indexfile-ban található pointereket, ezek közül csak azokat gyűjti ki, amelyek benne vannak — a dokumentumfile-ban található adatok szerint — a keresett intervallumban.

Gyakorisági adatok

A NSz. számára gyűjtött korpuszból természetesen gyakorisági szótár is készíthető, ennek megvalósítása azonban körültekintő tervezést igényel, amely mindenképp meghaladná jelen dolgozat kereteit. Arra azonban lehetőségünk lesz, hogy bármely címszó vagy szövegszó előfordulási számát kiírassuk, esetleg a szócikkben feltüntessük. Bármilyen karaktersorozat keresésekor megkapjuk az előfordulás abszolút gyakoriságát. Még egy gyakoriságra vonatkozó adatra lehet szükségünk feldolgozás közben: egy karaktersorozatot tartalmazó leggyakoribb szóra/szavakra. Ezt a `signif` paranccsal oldjuk majd meg. Pl. a `signif m` hatására a program kiírja, hányszor fordult elő a leggyakoribb *m*-mel kezdődő szó, a `disp` paranccsal kiírathatjuk ezt a szót. Amennyiben elkészül a korpuszból készíthető gyakorisági szótár terve, ennek számítógépes megvalósításához külön rendszertervet kell írnunk.

Felhasználói interfész

A felhasználói interfész feladata az, hogy minél jobban megkönnyítse az adatbázis használatát a nyelvészek és a nem számítástechnikai szakemberek számára. A korábbi pontokban is kizárólag könnyen megtanulható, értelemszerűen alkalmazható parancsokat javasoltunk. Ezeket szeretnénk kiegészíteni egy olyan képernyőkezelő programrendszerrel, amely még könnyebbé és hatékonyabbá teszi a nyelvészek munkáját. Ez lényegében egy nagyon egyszerű szövegszerkesztő program lenne, ami arra adna lehetőséget, hogy a munka során kapott, érdekesnek tűnő részeredményeket el lehessen tenni egy munkafile-ba, amit aztán szerkeszteni lehet, ki lehet nyomtatni, ill. át lehet másolni részben vagy egészben más szövegfile-okba. (Ez pl. hasznos lehet tanulmányok készítésénél, de elsősorban a szócikkírást könnyítené meg jelentős mértékben.)

A programnak lényegében két alaputasítása lesz, a `save > file-név` utasítás hatására az utolsó kérdésre kapott választ elteszi a file-név-ben megadott file-ba (ha már van ilyen nevű file, felülírja), a `save >> file-név` hatására úgy tárolja el az utolsó kérdésre adott válaszokat, hogy a meglévő file végére illeszti az új eredményt (nem írja felül a meglévő file-t). Az `edit file-név` utasítás egy szövegszerkesztő programot hív meg, amelynek segítségével a file-unkat szerkeszthetjük. Ez a szerkesztő lehetőleg azonos lesz az adott gépen leggyakrabban használt szerkesztővel, de legalábbis hasonlóan fog működni.

A 4.4.1–4.4.3 pontban ismertetett programrendszer jórészt azonos a waterlooi egyetemen fejlesztett ún. PAT szövegkereső programmal (ld. GONNET, 1987). Ha meg tudnánk vásárolni ezt a programot, csupán néhány olyan utasítással kellene kiegészíteni, amely a program használhatóságát fokozná. (Ilyenek pl. a print-kontrollfile-t létrehozó utasítás, `disp, edit, save,` locus-kód kiíratási lehetősége.) A kiegészítő utasításokat akár a waterlooi egyetem programozóitól megrendelhetnénk, akár együttműködés keretében magunk kifejleszthetnénk.

4.4.4. A címszójegyzék összeállítását segítő program

A NAGYSZÓTÁR CÍMSZÓJEGYZÉKÉT természetesen nem állíthatjuk elő automatikusan (bár ez az igény számtalanszor felmerült), a számítógép csak hathatós segítséget nyújthat a lexikográfusoknak a címszójegyzék összeállításában. Kiindulásként használható lesz majd a korábban említett tőtári adatbázis teljes formája: itt megtalálható lesz három szótár címszóállománya (ÉrtSz., ÉKsz., GyakSz.), mindezek anyaga a morfológiai elemzés során folyamatosan bővül majd az új szótövekkel. Az így kapott tőtár- állomány azonban feltehetőleg nem lesz azonos a NSz. címszóállományával, csupán jó alapul szolgálhat ennek kialakításához. Miért?

Egyrészt: korántsem biztos, hogy a korábbi szótárak címszavai közül mindegyiket célszerű átvinnünk a NSz.-ba is. Hiszen előfordulhatnak ezek között olyan szavak is, amelyek a mi korpuszunkban egyáltalán nem, vagy csak egyszer — kétszer fordultak elő, ilyenkor minden esetben mérlegelnünk kell, hogy ez a korpusz hiányosságának köszönhető-e, vagy valóban olyan ritka, esetleg kérészéletű szó volt, hogy nem érdemes a NSz.-ba felvenni. Mindazonáltal megfontolandó, nem lenne-e célszerű párhuzamosan összegyűjteni egy minél teljesebb magyar címszókészletet, ebben bejelölve, hogy a címszó mely szótáraknak része (NSz., ÉrtSz., KSz.,), de az is előfordulhat, hogy egyiknek sem, csupán a számítógépes archívumban tartjuk nyilván. Másrészt, a korpuszból kaphatunk olyan új szavakat is, amelyeket felveszünk ugyan az elemzéskor a tőtárba, mivel azonban csak egy-két helyen fordulnak elő, később a lexikográfusok úgy dönthetnek, hogy ezeket nem célszerű önálló címszóként kezelni. Számos probléma adódhat majd az alakváltozatokból is: bizonyos szavakat az elemzéskor esetleg alakváltozatnak tekintünk, és mint ilyet vesszük fel a tőtárba, később esetleg a kapott konkordanciák alapján a szócikkíró úgy dönthet, mégis külön címszóként kellene valamelyik alakváltozatot felvenni, stb. Nem is beszélve a képzett szavakról és az összetételekről: ezeknek egy részét az elemző „szétvágja" (amelyek együttesen nem szerepeltek a tőtárban), más részét együtt kezeli — mivel megtalálta a teljes alakot a tőtárban. Ha az így

előálló tőtárat automatikusan a NSz. címszóállományává akarnánk tenni, az azt jelentené, hogy korábbi szótáraink esetleges következetlenségeit örökítenénk át az új szótárra is.

Mindazonáltal első közelítésben természetesen a tőtárból nyerhető adatokból kell majd kiindulni, ide majd az elemzés során azt is felvisszük, mely címszavakra volt példa a korpuszban. Ennek alapján az előfordult címszavakra vonatkozó összes adatot kikereshetjük majd a fenti keresőprogrammal. Első közelítésben talán elegendő lesz a címszó előfordulási számát kiíratnunk, és esetleg futólag betekinteni a konkordanciába, hogy eldönthessük, javasoljuk-e az adott címszó felvételét a NSz.-ba. Kérdéses esetekben — különösen a képzett és összetett szavaknál — speciális kereső-listázó programmal segíthetjük a döntést. E program legfontosabb funkciói:

a) Kiírja az összes címszónak tekinthető szót, azaz minden olyan morfémát, amely szóközzel, szókezdő írásjellel vagy ≪IK≫ jellel kezdődik, szóközzel, szóvégző írásjellel vagy ≪SZF (ahol SZF valamilyen megengedett szófajkód érték) végződik. Ezen kívül megadhatjuk majd, hogy ne csak az ilyen szavakat, hanem a képzetteket is írja ki. Ezeket azonban feltehetőleg más módon lesz érdemes kezelnünk.

b) Csoportosítja a képzett szavakat a képző szerint. Így külön kiíratható lesz pl. minden *-i* képzős melléknév (függetlenül attól, hogy a tőtárban szerepelt-e, vagy nem), minden *-ság*, *-ség* képzős főnév stb. E csoportosított lista valószínűleg nagyon meg fogja könnyíteni a lexikográfusok választását, így ugyanis egyszerre áttekinthetik, mi lenne a hatása egy olyan döntésnek, hogy: „legyen minden *-ság*, *-ség* képzős szó önálló címszó". Mivel az esetek nagy részében nem lesz célszerű ilyen általános döntéseket hozni, a csoportokból feltehetőleg könnyebb lesz kiválogatni, melyek legyenek címszavak és melyek ne, úgy, hogy a döntés a lehető legkevésbé legyen következetlen.

c) Csoportosítja az összetételeket előtag, illetve utótag szerint, így ad teljes listát egy-egy szó valamennyi összetett szóbeli előfordulásáról. Ez is nyilván segítséget jelent majd annak eldöntésében, melyik összetett szó legyen önálló címszó.

d) Teljes listát készít az igekötős igékről, igekötőre és igére rendezve.

Valamennyi itt felsorolt lista puszta szólista, esetleg az előfordulás számával kiegészítve. Az előfordulások konkordanciáit csupán a korábban ismertetett kereső program segítségével írathatjuk ki. Természetesen, amennyiben újabb igények merülnek fel, számos más célszerűnek tűnő szempont szerint csoportosított szólistákat készíttethetünk a géppel (pl. különböző a-tergo listákat, névutós főnevek listáját stb.), azt azonban, hogy a bármilyen szempont szerint összeállított szólistából mi legyen végül is címszó, csak a lexikográfusok dönthetik el, gondos mérlegelés után. Ehhez a mérlegeléshez az esetek jó részében feltehetőleg a kérdéses szó konkordanciáját is át kell majd tekinteniük, legalább felületesen. A címszóul kiválasztott szavakat egy kifejezetten erre a célra készített adatbázisba fel kell majd vinni. (Esetleg, ha nagy az átfedés a kibővített tőtárral, azt is megtehetjük, hogy az ott megtalálható szavak egy részéhez hozzá írjuk, hogy legyen címszó, az ott nem találhatókat pedig felvesszük, új címszóként.) Ilymódon mindig naprakész listát készíthetünk munka közben is a NSz. címszóállományáról, illetve szükség esetén módosíthatjuk azt.

4.4.5. A szótár szerkesztését támogató programrendszer

JELENLEG ez a leghomályosabb pontja a rendszernek, hiszen senki nem tudja, milyen adatokat fog majd tartalmazni a szótár, hogy fog felépülni egy-egy szócikk stb. Mindössze annyit tudok javasolni, készítsünk majd egy olyan ablak-rendszerrel működő programot, amelynek „fő ablakában" a szócikkíró írja a szócikket. Valószínűleg érdemes lesz célprogramot írni a szócikkszerkesztéshez, olyat, amelyben megjelenik a szócikk szerkezeti váza, a lexikográfusnak pedig „csak" annyi lesz a dolga, hogy ezt a vázat kitöltse. (Hasonló programot fejlesztenek a DOE-nél.)

Pl. egy ilyen „váz" jelenne meg a képernyőn:

```
Címszó:
Előfordulás száma:
Alakváltozatok:
szófaj:
1. jelentés:
idézetek az 1.jelentéshez:
2. jelentés:
stb.
```

A váz kitöltése után egy program automatikusan előállítaná a szócikk nyomtatott formáját, amelyet ellenőrzésre ki lehetne adni, egy másik program pedig eltárolná a szócikket a szócikk adatbázisba, olyan strukturális tagoló jelekkel ellátva, amelyek könnyebbé tennék a szócikk-adatbázisban való szerkezeti elemek szerinti keresést.

```
<szocikk><cimszo> ..... <cimszo v><gyak> ..... <gyak v>

<alakvaltozatok> ..... <alakvaltozatok v>
<1. szofaj><1. jelentes> .....
.....<1. jelentes v>
<1. idezetek:> .....
.....<1. idezetek v>
<2. jelentes:> .....stb.
<2. szofaj> .....stb.
.....<szocikk v>
```

Ehhez természetesen előbb a lexikográfusoknak kellene eldönteniük, milyen szócikkeket akarunk írni, mi legyen benne, mi nem. Azt majd a gyakorlatban lehet kipróbálni, segítséget jelent-e egy ilyen vázlat a szócikkírásban vagy épp ellenkezőleg. Az utóbbi esetben bármely kommersz szerkesztőprogramot használhatjuk szócikkírásra. A fő, szócikkírásra szolgáló ablakon kívül megjeleníthetnénk a képernyő alján egy széles, néhány sornyi ablakban az éppen szerkesztés alatt lévő szócikk konkordanciáit. (A 4.4.3-ban ismertetett konkordanciakészítő illetőleg kereső programot használhatnánk itt.) A kiválasztott

idézeteket a korábban említett szerkesztési lehetőséget alkalmazva, átmásolhatnánk a szócikk megfelelő részébe, a többit pedig csak a jelentésdefiniálás segédeszközéül használnánk. Egy harmadik „oldalablak" szolgálhatna esetenként arra, hogy a már elkészült szócikkeket megnézzük, esetleg átmásoljunk belőlük részeket.

4.5. Zárómegjegyzések

AMENNYIBEN A FENT VÁZOLT RENDSZERTERV[14] megvalósítására alkalmas méretű számítógép a NSz. rendelkezésére állna, elkészíthetnénk a magyar nyelv történeti korpuszát. (Csupán az adatok tárolásához mintegy 1,5–2 Gigabyte-nyi fix lemezterületre, a konkordanciaprogramok megfelelő sebességű működéséhez pedig 32 bites architektúrára és legalább 4 Megabyte-os központi memóriára lenne szükség.) A számítógépes korpuszon túl azonban számos lexikográfus több évtizedes kitartó munkájára is szükség lesz ahhoz, hogy a tervezett nagyszótár álomból valósággá váljék, és unokáink már láthassák ezt a szótárat, akár nyomtatott, akár számítógépen elérhető formában.

[14] A rendszerterv jelenlegi formájának kialakításában számos külső és belső munkatársunk volt segítségemre: a NSz. számítógépes megvalósításának ötletét Herman József vetette fel, neki köszönhető, hogy a munkálat megindulhatott és — a szűkös anyagi körülményekhez képest — lényegében megfelelő ütemben és eszköz-háttérrel folytatható. A munka beindítása és az első, vázlatos koncepció kialakítása Papp Ferenc érdeme, a rendszertervet és a teljes dolgozatot az ő folyamatos szakmai irányításával készítettem. A készülő rendszertervet többször bírálták el mind a munkálat résztvevői (Kiss Lajos, mint a téma vezetője, Csengery Kinga, Elekfi László, Füredi Mihály, Hexendorf Edit, M. Zoller Erzsébet, N. Szederkényi Éva, Pólya Katalin, Prószéky Gábor, Ruzsiczky Éva, Szabó Tiborné, Tüske Margit, Villó Ildikó), valamint felkért külső szakértők (Bakró-Nagy Marianne, Bölcskei András, Kálmán László, Kalmár Tibor, Kiefer Ferenc, Kontra Miklós, Váradi Tamás). Külön köszönettel tartozom a UW Centre for the New OED kutatóinak, (különösen G.H. Gonnet-nek és F. Wm. Tompának, véleményüknek köszönhetően ugyanis az első rendszerterv-vázlatot rendkívül jelentős mértékben módosítottam), továbbá G. Johanessennek és D.L. Bergnek, akik ideális munkakörülményekkel lehetővé tették, hogy ez a dolgozat elkészülhessen. Az ismertetett programok közül a forrásnyilvántartó-program első változata Füredi Mihály nevéhez kötődik, T. Uzsoky Borbála fejlesztette tovább. A morfológiai elemző Prószéky Gábor tanulmánya alapján saját munkám. A feldolgozás rendszervázlatában ismertetett lekérdező programrendszer egy a PAT-hoz hasonló, de annál rugalmasabb rendszer lesz.

Mellékletek

1. sz. melléklet
Az eredeti címszójegyzék kódjai és jelentésük

ENTRYNEV	TIPUS	SZAM	HELY	HOSSZ
1: HOMONIMIA	N	1	0	1
2: FAJOK	C	1	1	3
3: H	N	1	4	1
4: SZF	N	1	5	2
5: F	N	1	7	1
6: SZO1	C	1	8	31

Homonimia kód jelentése

0	NEM HOMONIMA
1	1 HOMONIMA
2	2 HOMONIMA
3	3 HOMONIMA
4	4 HOMONIMA
5	5 HOMONIMA
6	1 ÁLHOMONIMA
7	2 ÁLHOMONIMA
8	3 ÁLHOMONIMA
9	4 ÁLHOMONIMA

***

Fajok kód jelentése (a mező mindhárom karakterén ez)

0	szófaj érthetelen
1	ige
2	főnév
3	melléknév
4	számnév
5	névmás
6	határozószó
7	ikegkötő
8	névutó
9	kötőszó
A	indulatszó
B	egyéb mondatszó
C	névelő
D	igenév
K	kötött morféma
M	csak egyes szám 3. szem.
N	főleg egyes szám 3. szem.
P	felszólító módban nem használatos
Q	felszólító módban ritkán használtos

***

h kód a gyaksz homonimakódja

***

szf kód jelentése

00	idegen szöveg
01	mondatszó
02	halandzsaszó
09	szónál kisebb vagy metanyelvi elem
10	ige
11	segédige
12	főnévi igenév
13	melléknévi igenév
16	határozói igenév
19	állítmányi névszó
20	köznév
21	személynév része
22	egyelemű földrajzi név része, (mozaikszó),utcanév
23	melléknévként használt főnév
26	intézménynév, földrajzi név eleme
29	egyelemű int. név, állatnév, megszem., mozaik
30	melléknév
32	főnévként használt melléknév
36	melléknévi határozó
37	kéttagú melléknév második tagja
40	számjeggyel írt számnév
42	főnévként használt számnév
43	melléknévként használt számnév
44	számnév, ráértéses főnévi használattal
46	határozóragos számnév
52	főnévi névmás
53	melléknévi névmás
54	mn.-i névmás, ráértéses fn.-i haszn.
56	határozói névmás
60	határozószó
62	ráértéses igekötő
63	módosítószó
70	névutó, -szerű viszonyítószó
79	idegen prepozíció
80	névelő
91	mellérendelő kötőszó
92	alárendelő kötőszó
96	többelemű kötőszó eleme
99	több szóból álló rövidítés v. jel

***

f kód jelentése

0	nincs a gyakorisági szótárban
1	ABSZ. GYAK = 1
2	ABSZ. GYAK $\geq$ 2 ÉS MOD GYAK = 0
3	ABSZ = 2 AND FMOD NEM = 0
4	FMOD < 2 AND FMOD > 0
5	FMOD $\leq$ 2 AND FMOD < 4
6	FMOD $\leq$ 4 AND FMOD < 8
7	FMOD $\geq$ 8 AND FMOD < 20
8	FMOD $\geq$ 20

2. sz. melléklet
Speciális karakterek kódolása

ÉKEZET	KÓD	PÉLDA		ÉKEZET	KÓD	PÉLDA		KARAKTER	KÓD	PÉLDA
´	1	á	a1	1)	22	1) a	a22	ȝ	40	z40
¨	2	ö	o2	e/	23	e/ o	o23	ʃ	41	s41
˝	3	ű	u3	¨	24	ö	o24	ȣ	42	us42
`	4	à	a4	¸	25	ş	s25	ɾ	43	z43
°	5	å	a5	.	26	ṇ	n26			
ˇ	6	ď	d6	¸	27	ţ	t27			
~	7	ã	a7	_	28	t̲	t28			
¯	8	ā	a8	–	29	ɵ	o29			
˘	9	ă	a9	~	30	e̦	e30			
˙	10	ȧ	a10	;	31	s;	s31			
ˆ	11	â	a11	/	32	ł	l32			
ˇ	12	č	c12	¸	33	l̨	l33			
˚	13	å	a13	‿	34	c͡t	c34t			
´	14	ḗ	e14							
`	15	ȅ	e15							
ˇ	16	ǧ	g16							
˝	17	ő	o17							
˙	18	ȯ	o18							
^	19	ô	o19							
°	20	ů	o20							

Irodalom

AARTS, J.–MEIJS, W. (1984) *Corpus Linguistics* Recent Development in the Use of Computer Corpora in English Language Research. Amsterdam. Rodopi.

AITKEN, A. J. (1971), Historical dictionaries and the computer. In: *The computer in Literary and Linguistic Reserach.* R.A. WISBEY (ed.): Cambridge. UP.

ALLÉN, S. (1983), Språkdata Lexibase System. An Integrated View of a lexical Project. In: ZAMPOLLI 1983a p 51–63.

AMSLER, R. A. (1984), Machine-Readable Dictionaries. (In: *Annual Rewiev of Information Science*, M.E. William ed., Vol. 19, 1984. p 161–209.)

AMOS, A. (1984), *Computers and Lexicography: The Dictionary of Old Enlgish.* Status Report on the Dictionary of Old English Project. Kézirat.

BACH I. és FARKAS E. és NASZÓDI M. (1987), *A magyar nyelv elemzése számítógéppel* MTA SZTAKI tanulmányok 199.

BENBOW, T. és WEINER, E.(1986a), *Machine-readable Dictionaries for the General Public.* (Workshop on Automating the Lexicon. Second.Grosseto, Italy: European Community, University of Pisa and Istituto di Linguistica Computazionale del CNR. May 19–23, 1986. 10 p.)

BENBOW, T. (1986b), *Status Report on the New OED Project.* Second Annual Conference of the UW Centre for the New OED. University of Waterloo, 9–11 Nov 1986. Kézirat.

BERG, D. L. (1986), *NED to OED to New OED: a Bibliographic Essay.* (Preliminary version) Centre for the New OED. Kézirat

BERG, D. L. (1987), (ed.) *The Uses of Large Text Databases* Proceedings of the Third Annual Conference for the New OED. Nov. 9–10, 1987. Waterloo, Canada

BERG, D. L. és GONNET G.H. és TOMPA, F. Wm. (1988), *The New Oxford English Dictionary Project at the University of Waterloo* OED-88-01

BÖLCSKEI A. (1984), *Szövegfeldolgozás* Egyetemi doktori értekezés. Debrecen.

BRATLEY, P. (1983), Computers and Lexicography: Advances and Tends. (ZAMPOLLI 1983a, 83–95).

BRODDA, B. (1983), An Experiment with Heuristic Parsing of Swedish *Proceeding of the first Conference of ACL European Ch.* Pisa. 66–75.

BRODDA, B. KARLSSON, F. (1981), *An Experiment with Automatical Analysis of Finnish* Univ. of Helsinki, Publ, No 7.

BRUSTKERN, J. és HESS, K. D. (1982), Machine-readable German Dictionaries: from a Comparative Study to an Integration. (CIGNONI 1982, 77–181).

BRUSTKERN, J. és HESS, K. D. (1983), The Bonnlex Lexicon System. (GOETSCHALCKX 1982, 33–40).

BUTLER, S. (1982), Problems with Headwords in Old English. (ZGUSTA 1980, 105–114).

BYRD, R. J. (1984),*DAM — A Dictionary Access Method.* Computer Science Dept. I.B.M. Thomas J. Watson Research Center, Yorktown Heights, New York. Kézirat.

BYRD, R. J. (1985),*WordSmith.* User Guide. Computer Science Dept. I.B.M. Thomas J. Watson Research Center, Yorktown Heights, New York. Kézirat.

CALZOARI, N. és CECOTTI, M. L. és ROVENTINI, A. (1984), Computational Tools for an Analysis of Terminological Data in a General Dictionary. (HARTMANN 1984, 328–332).

CALZOARI, N. és PICHI, E. (1988), Acquisition of semantic information from an on-line dictionary. *COLING 88 Proceedings* Budapest. NJSZT. p. 87–92.

CASSIDY, F. (1980), Computer Maping of Lexical Variants for DARE. (ZGUSTA 1980, 147–162.

CATARZI, M. N.–CAPELLI, G.–RATTI, D.–SABA, A. (1982), A morphosyntactic analyzer for Spanish. (CIGNONI 1982, 115–122).

CIGNONI, L. és PETERS, C. (1982), *Computers in Literary and Linguistic Research.* Proceedings of the VII Internatinal Symposium of the Association for Literary and Linguistic Computing, Pisa, 1982. Linguistica Computazionale vol. III (1983), Supplement.

CSIRIKNÉ C. E. és CSIRIK J. (1986), *Újságnyelvi gyakorisági szótár* I.-II. Magyar pszicholingvisztikai tanulmányok IV. Szeged–Budapest–Debrecen

DENDIEN, J. és GORCY, G. és MARTIN, E. (1986), *Le Trésor général des langues et parlers français de l'institut national de la langue française* Kézirat. INALF.

DOMENIG, M. és SHANN, P. (1986), Towards a Dedicated Databse Management System for Dictionaries. *Proceedings of COLING '86*, Bonn, 1986. p 91–96.

ELEKFI L. *Magyar ragozási szótár*Megjelenés alatt.

ELEKFI L. *Szókincsünk nyelvtani alakrendszere.* Kézirat.

ENGEL, G. és MADSEN, B. N. (1983) From Dictionary to Data-Base. (HARTMANN 1983, 339–344.)

GÁLDI L. (1960), Mutatvány A Magyar Irodalmi Nyelv Nagyszótárából. *Magyar Nyelvőr* 1960/2.

GÁLDI L. és WACHA I.: Mutatvány az Akadémiai Nagyszótárból. I.-II. rész. Az MTA Nyelv- és Irodalomtudományi Osztályának közleményei. XI-XII. kötet 1.-4. szám

GEHRKE, M. és BLOCK, H. U. (1986), Morpheme-based Lexical Analysis. (JOHANNESEN 1986, 1–15.

GOETSCHALCKX, J. és ROLLING, L. (1982) (eds.) *Lexicography in the Electronic Age.* Proceedings of a Symposium held in Luxembourg, 7–9 July, 1981. North-Holland.

GONNET, G. H. (1987a), PAT — *An efficient text searching system.* UW Centre for the New OED. Kézirat.

GONNET, G. H. és TOMPA, F. WM. (1987b), *Mind Your Grammar: a New Approach to Modelling Text.* UW Centre for the New OED.Kézirat

GORCY, G. (1983), Les dérivés d'esprit en français moderne: Méthdologie et esquisses d'articles í paraître dans le trésor de la langue française. *Lessico Intelletuale Europeo IV Colloquio Internazionale.* Roma 7–9 gennaio 1983.

GORCY, G. (1984), L'ordinateur au service de la lexicographie: une experince et ses problemes. (HAMESSE 1985).

GUCKLER, G. (1983), A Computer Based Monolingual Dictionary: a Case study. R. R. K. HARTMANN (ed): *Lexicography: Principles and Practice.* Academic Press 1983. p 198–201.)

HAMESSE, J. és ZAMPOLLI, A. (1985) (ed.) *Computers in Literary and Lingustic Computing.* Proceeding of the Xi. International Conference. Chamion–Slatkine. Paris–Genéve.

HARTMANN, R. R. K. (1983) (ed.): *LEXeter '83 Proceedings.* Papers form the International Conference on Lexicography at Exeter, 9–12 September 1983. Lexicographica Series Maior 1.Max Niemeyer Verlag, Tübingen 1984.

HERBST, T. (1986), Defining With a Controlled Defining Vocabulary in Foreign Learners' Dictionaries. (*LEXICOGRAPHICA.* International Annual for Lexicography. A. KUCERA, A.REY,H.E. WIEGAND,L. ZGUSTA (eds.) 2/1986. Max Niemeyer Verlag Tübingen. p. 101–119.)

HOCKEY, S. (1980), *A Guide to Computer Applications in the Humanities.* Duckworth. London.

HOFLAND, K. és HAUGE, J. H. (1983), A Lemmatized Ibsen Concordance and Potential Consequences for Dictionary Production. (ZAMPOLLI 1983a , 145–151.)

HOWARD, T. H. (1979), *Literary Concordances. A Guide to the Preparation of Manual and Computer Concordances.* Pergamon Press, Oxford–New York–Toronto–Paris–Sydney–Frankfurt.

HUGHES, J. J. (1987), Wordcruncher: High Powered Text-Retrieval Program. *Bits & Bytes Rewiev.* Vol.1 No. 3. Whitefish, Montana.

ICAME News 1978 — Newsletters of the International Computer Archieve of Modern English. Bergen: Norwegian Computing centre for the Humanities.

JOHANSSON, S. (1988), The New Oxford English Dictionary project: A presentation. *ICAME Journal* No. 12. Norwegian Computing Centre for the Humanitites.

JOHANESSEN, G. (1985), (ed):*Information in Data.* First Annual Conference of the UW Centre for the New OED. November 6–7. 1985.Proceedings.

JOHANESSEN, G. (1986), (ed): *Advances in Lexicology.* Second Annual Conference of the UW Centre for the New OED. November 9–11. 1986.Proceedings.

KAY, M. (1983), The Dictionary of the Future and the Future of the Dictionary. (ZAMPOLLI 1983a, 161–174.)

KAZMAN, R. (1986), *Structuring the Text of the Oxford English Dictionary Through Finite State Trasduction.* Master of Mathematics in Computer Science Thesis. University of Waterloo.

KEITZ, W. V. (1982), Projekte zur mashinellen Lexicographie. *Sprach and Patenverb.* Vol 6 (1–2); p 11–22.

KIPFER, B. A. (1984), The Dictionary of the Future: Computer Aplications. *Workbook on Lexicography.* A Course for Dictionary Users with a Glossary of English Lexicographical Terms. Exeter Linguistic Studies. Vol. 8. University of Exeter, p. 161–172.

KISS Z. és PRÓSZÉKY G. és TÓTH L. (1979) Magyar nyelvű szövegek számítógépes morfológiai vizsgálata *SOFTTECH* D 41.

KLAVANS, J. (1988), COMPLEX: a computational lexicon for natural langue processing. *COLING 88 Proceedings* Budapest. NJSZT. p. 815–823.

KNOWLES, F. E. (1983), Dictionaries and Computers. (HARTMANN 1983, 301–314.)

KORNAI A. (1986), Szótári adatbázis az akadémiai nagyszámítógépen. *Műhelymunkák* II. 65–79.

KOSKENNIEMI, K. (1983), Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production. *Univ. of Helsinki, Publ. No. 11.*

KOSKENNIEMI, K. (1984), A General Computational Model for Word-Form Recognition and Production. *COLING 84* 174–177.

KOSKENNIEMI, K. és CHURCH, K. W. (1988) Comlexity, two-level morphology and Finnish. *COLING 88 Proceedings* Budapest. NJSZT. 335–340.

KUCERA, H. és FRANCIS, W. N. (1967), *Computaional Analysis of Present-Day American English.* Brown University Press.

LANDAU, S. I. (1984), Computer Use and the Future of Dictionary Making. *Dictionaries. The Art and Craft of Lexicography.* Charles Scribners's sons. New York. p. 272–294.

LEECH, G.–BEDE, A. : (1984) Computers in English Language Research. *Language Teaching*, Vol. 17. No.3. 216.-229.

LONG, T. H. (1979), *Longman Dictionary of English Idioms.* Burnt Mill, Harlow: Longman.

LUSIGNAN, S. és NORTH, J. S. (1977) (ed.) *Computing in the Humanities.* Proceedings of the third International Conference on Computing in the Humanities. The University of Waterloo Press.

MAKKAI, A. (1980), Theoretical and Practical Aspects of an Associative Lexicon for the 20th Century English. (ZGUSTA 1980, 132a –145.)

MARINONE, N. (1982), A Project for a Latin Lexical Database. (CIGNONI 1982, 175–178.)

MARTIN, E. (1984), Une banque de données sur la langue française. *BRISES. Bulletin de recherches sur l'information en sciences économiques, humaines et sociales.* La Linguistique dans les sytemes documentaires. Avril 1984. No. 4.

MATHIAS, J. (1983), Computer-Aided Processing of Chinese Lexicographic Materials. (HARTMANN 1983, 371–376.)

MERKIN, R. (1983), The historical/academic dictionary. In: R.R.K HARTMANN (ed): *Lexicography: Principles and Practice.* Academic Press. London–New York–Paris stb. p. 123–133.)

MERKIN, R. (1983b) Historical dictionaries and the Computer — Another View. (HARTMANN 1983, 377–384.)

MEYA, M. (1982), Treatment of Suffixes in Automatic Morphological Analysis. (CIGNONI 1982, p. 199–207.)

MICHIELS, A.–MULLENDERS, J.–NOEL, J.(1981),The Longman–LiegeProject.(GOETSCHALCKX 1982, p 201–210.)

MURRAY, J. H.–BRADLEY, H.–RAIGIE, W. A.–ONIONS, C. T. (1882–1928) The Oxford English Dictionary. 1–12 kötet. Oxford: Clarendon Press

NAGAO, M.–TSUJI, J.–UEDA Y.–TAKIYAMA M. (1981), An Attempt to Computerize Dictionary Data Bases. (GOETSCHALCKX 1982, p. 51–73.)

OAKMAN, R. (1980), *Computer Methods for Literary Research.* University of South Carolina Press

ORSZÁGH L. (1966), A mai angol szótárirodalom. in: *Szótártani tanulmányok.* Szerk . Országh L. Tankönyvkiadó Bp.

PAIKEDAY, T. M. (1983) The Joy of Lex. *Creative Computing,* Nov. 1983. p 240–245.

PAJZS J. (1987) *Javaslat a Nagyszótár számítógépes megvalósítására. A gyűjtés és feldolgozás rendszerterve* Kézirat.

PAJZS J. (1988a) Félmillió szó számítógépen. *Computerworld - Számítástechnika* III. évf. 5. p 24–25.

PAJZS J. (1988b) Számítógépes szótárak. *NyK.* Megjelenés alatt.

PAJZS J. (1988c) Dictionary digitalisiert: Oxford Englisch per Knopfdruck. *Computerwelt Östereich* No. 9.

PAPP F. (1964) O gotovjásemszjá obratnom szlovare vengerszkovo jázika. *Computational Linguistics* II. 213–218.

PAPP F. (1965) A Magyar Szóvégmutató Szótár és a vele kapcsolatos munkálatok. *Magyar Nyelv* LXI/2. 187–200.

PAPP F. (1966) A magyar főnévragozás három modellje. *Magyar Nyelv* LXII/2. 194–206.

PAPP F. (1969a) *A magyar nyelv szóvégmutató szótára.* Budapest. Akadémiai Kiadó.

PAPP F. (1969b) Az Értelmező Szótár legtöbb jelentésű szavairól. *Magyar Nyelvőr* 93/1. 94–97.

PAPP F. (1969c) Gépi eredményeink felhasználása szótárírásunkban. *Magyar Nyelvőr* 93/3. 376–381.

PAPP F. (1975a) *A magyar főnév paradigmatikus rendszere.* (Leírás és automatikus szintézis.) Budapest, Akadémiai Kiadó.

PAPP F. (1975b) Konkordancia: irói szótár előkészítése számítógépen. *Magyar Nyelvőr* 99/3. 351–355.

PAPP F. és HEXENDORF E. (1985), Magyar szókincs a könynvnyomtatástól napjainkig — számítógépre tervezve. *Magyar Tudomány* 1985/1.

PAPP F. (1988), *Megjegyzések a hazai számítógépes lexikográfia korai történetéhez.* (Kézirat.)

PETERSEN, P. R. (1983), New Words in Danish 1955–75. A Dictionary Compiled and Worked out in a Traditional Way and Managed and Typed via Computer. (ZAMPOLLI 1983a, p. 179–186.)

PROCTER, P. (1978), Longman Dictionary of Contemporary English. London, Longman.

PRÓSZÉKY G. és KISS Z. és TÓTH L. (1982), Morphological and Morphonolgical Analyssis of Hungarian Word-Forms by Computer *CL & CL* XV. 195–228.

PRÓSZÉKY G. (1985), *Magyar szövegek számítógépes morfológiai elemzése* Kézirat, MTA Nyelvtudományi Intézet.

RABEN, J. és MARKS, G. (1980), (ed.) *Data Bases in the Humanities and Social Sciences.* North Holland, Amsterdam–New York–Oxford.

RAYMOND, D. R. WARBURTON, Y. L. (1987) *Computerization of Lexicographical Activity on the New Oxford English Dictionary* OED-87–03

RENOUF A. (1984), Corpus Development at Birmingham University (ARTS & MEIJS 1984)

R. HUTÁS M. (1974), Az Akadémiai Nagyszótár történetének vázlata. (1898–1952) *Nyelvtudományi Közlemények* 72/2.

SIMPSON, J. (1985), Opening Adress:The New OED Project. (JOHANNESEN 1985, p. 1–6)

STERKENBURG, P. V.–MARTIN, W.–AL, B. (1981), A New Van Dale Project: Bilingual Dictionaries on one and the same Monolingual Basis. (GOETSCHALCKX 1982 p. 221–238.)

SVENSEN B. (1983), A Computerized Concordance Based on a Bilingual Dictionary: a Case Study. In:R.R.K. HARTMANN (ed): Lexicography: *Principles and Practice.* Academic Press. p 202–205.

SINCLAIR, J. és HANKS, P. és FOX, G. és MOON, R, és STOCK, P. (1987), *Collins COBUILD English Language Dictionary.* London & Glasgow: Collins.

TEUBERT (1983), Setting up a Lexicographical Data-Base for German. (HARTMANN 1983 p 426–429.)

THALLER, M. (1982) Recycling the Drudgery. On the Integration of Software Suporting Secondary Analysis of Machine-Readable Texts in a DBMS. (CIGNONI 1982,p 253–268.)

THURMAIR, G. (1982), An Integrated Algorithm for Morpological Decomposition. (CIGNONI 1982,p. 269–277.

TOMPA, F. WM. (1985) *Database Design for the Dictionary of the Future.* UW Centre for the New OED. Kézirat

VENEZKY, R.–RELLES, N.–PRICE, L. (1976), *Man-Machine Integration in a Lexical Processing System.* Madison: University of Wisconsin Computer Center.

VOLLNHALS, O. (1983), Utilization of a Commercial Linguistic Data-Base System for Electronic Storage and Automated Production of Dictionaries. (HARTMANN 1983, p 430–434.)

WARWICK, S. (1986),*A Survey of Lexical Resources in Europe.* Workshop on Automating the Lexicon. Second. Grosseto, Italy. European Commitee, University of Pisa, May 19–23, 1986. Kézirat.

WEBSTER, J. J. (1983), The 'PROLEX' Project. (HARTMANN 1983, p. 435–440.)

WEINER, E. (1984), The New Oxford English Dictionary. *EURALEX Bulletin* 1(2): 31–3

WEINER, E. (1985a), Computerizing the Oxford English Dictionary. *Scholarly Publishing.* April: 240–253.

WEINER, E. (1985b), The New OED: Problems in the Computerization of the Dictionary. *University Computing* 7:66–71.

WILKS, Y. éS FASS, D. éS GUO, C. és MCDONALD, J. E. és PLATE, T. és SLATOR, B. M. (1988), Machine Tractable Dictionaries as Tools and Resources for Natural Language Processing. *COLING 88 Proceedings* Budapest. NJSZT. p. 750–755.

WISBEI, R. A. (1971) (ed.) *The Computer in Literary and Lingustic Research* Papers from a Cambridge Symposium. Cambridge University Press.

ZAMPOLLI, A. és CAPELLI, A. (eds.) (1983a) *Linguistica Computazionale.* Vol III. 1983. The Possibilities and Limits of the Computer in Producing and Publishing Dictionaries. Proceedings of the Eurepean Science Foundation Workshop, Pisa, 1981.

ZAMPOLLI, A. (1983b), Lexicological and Lexicographical Activities at the Istituto di Linguistica Computazionale. (ZAMPOLLI 1983a, p. 237–278.)

ZGUSTA, L. (1971), *Manual of Lexicography.* The Hague: Mouton.

ZGUSTA, L. (1980), (ed.): *Theory and Method in Lexicography.* Hornbeam Press. South Carolina.

ZIMMERMANN, H. H. (1983), Multifunctional Dictionaries. (ZAMPOLLI 1983a, p 279–288.)

Ára: 95,– Ft