

KAMİLƏ VƏLİYEVƏ*
LALƏ MURADOVA*

Ə.DƏMİRÇİZADƏNİN FONOLOJİ TƏDQIQATLARININ TƏTBİQİ MƏSƏLƏLƏRDƏ ROLU

XÜLASƏ

Verilənləri kompüterin yaddaşına daxil edildikdən sonra tərtib edilmiş alqoritmə əsasən qoyulmuş məsələnin (ixtiyari söz-formaları hecalara ayırmaq) proqramı qurulur. Proqram üzrə ilk mərhələdə söz-formalardan sait və samit qrafemlər axtarılır. Sonra sait və samit birləşməli qrafemlər kompüterin yaddaşında yerləşdirilən heca modelləri ilə müqayisə edilir. Üst-üstə düşən birləşmələr heca hesab edilir. Əks halda proses davam edir.

Açar sözlər: *alqoritm, proqram, qrafem, heca, model.*

Azərbaycan dilçiliyinə zəngin əsərləri ilə böyük töhfə verən Ə.Dəmirçizadənin elmi yaradıcılığından onun yetirmələri – elm aləmində tanınmış alimlər: Q.Kazımov, F.Şahbazlı, B.Xəlilov və başqaları geniş məlumat vermişlər. Biz yalnız alimin əsərlərinin praktikada fayda verdiyi tətbiqi məsələlərinə toxunacağıq.

Bu baxımdan məqalə Azərbaycan dilçiliyində çoxşaxəli əsərləri ilə tanınmış görkəmli dilçi alim Ə.Dəmirçizadənin fonologiya sahəsində qurduğu heca modellərinə əsasən onların avtomatik yolla müəyyənləşdirilməsinə həsr olunmuşdur.

Məlumdur ki, hecaların formalaşmasında əsas göstərici (meyar) saittir.

Müasir Azərbaycan dilində söz-formaları məntədən avtomatik yolla hecalara ayırmaq üçün ilk növbədə saitlər çoxluğunu:

a, e, ə, i, ı, o, ö, u, ü – {A₁}

samitlər çoxluğunu: {A₂}

b, c, ç, d, f, g, ğ, h, x, j, k, q, l, m, n, p, z, s, ş, t, v, y, z – nəzərə alınmalıyıq.

Qeyd etməliyik ki, Ə.Dəmirçizadə tamamilə haqlı olaraq vurğulayır: “Heca fonetik axının kiçik parçası olsa da, bir sıra xüsusiyyətlərinə görə çox mühüm nitq vahididir və nitqin fonetik zəncirində ən çox yer tutan həlqəcikdir. Belə həlqəcik, yəni heca ya bircə səsdən, ya da bir neçə səs qovuşmasından ibarət olur [Dəmirçizadə: 1972, s.100]. Bu fikir əsas tutaraq Azərbaycan dilindəki heca modelləri aşağıdakı şəkildə qruplaşdırılmışdır:

Heca növləri

Sıra №-si	Qrafemin sayı	Model	Nümunələr
1	2	3	4
1	1	0	a-ta, ü-tü, ö-zü, i-ki
2	2	01	al, ot, uç, üz, öz, üç, əl, el
3	3	011	alt, üst, əzm, eşq, ilk, elm

* AMEA Nəsimi adına Dilçilik İnstitutu, filologiya elmləri doktoru, dosent.

* AMEA Nəsimi adına Dilçilik İnstitutu, lalamuradova2017@gmail.com

5	2	10	bu, ba-la, nə-nə, ye-ni, bo-ru
6	3	110	qra-fik, sta-kan, pro-letar
7	4	1110	skle-ra, kny-a-ginya
8	3	101	baş, qaz, qız, qol, dur, gəl, gör
9	4	1011	cəld, yurd, hərf, mart, turş
10	4	1101	plan, qran, xrom, stul, plov
11	5	11011	blank, şlanq, spirt, sport
12	5	10111	filtr
13	5	11101	knyaz, plyaj, ştrix
14	6	111011	sprint
15	6	110111	spektr
16	7	1110111	Bryansk
17	6	101111	Voljsk və s.

[Dəmirçizadə: 1972, s.105]

Heca növləri cədvəlindən modelləri ayıraraq aşağıdakı çoxluqda qruplaşdırmaq:

Qrafemin sayı	Qrafemin modeli
1	2
1	0
2	01
3	011
4	0111
2	10
3	110
4	1110
3	101
4	1011
4	1101
5	11011
5	10111
5	11101
6	111011
6	110111
7	1110111
6	101111

Nəzərdə tutulan sonuncu iki nəzəri modelə görə nümunə tapılmamışdır. Azərbaycan dilindəki ixtiyari məndən götürülmüş mətnlərin tərkibindəki söz-formalarının hecaya ayrılması üçün ilk növbədə saitlərin (hecanın əsas meyarı olan saitin) avtomatik yolla axtarışı başlanmalıdır. Məsələnin bu mərhələsi əvvəlki tədqiqatlarımızda qeyd etdiyimiz kimi, yerinə yetirilmişdir (bax: tədq.3).

Sadalanan modellərə əsasən Azərbaycan dilindəki mətnlərin avtomatik yolla hecaya ayrılması prosesi iki mərhələ üzrə aparılır: maşından qabaqkı mərhələ və sonrakı mərhələ. Maşından qabaqkı mərhələdə kompüterin yaddaşına prosesdə iştirak edən, linqvistik bazaya daxil olan lazımi verilənlər: Azərbaycan dilinin əlifbası, sait və samitlər qrupu, Dəmirçizadənin tərtib etdiyi heca modelləri çoxluğu (çoxluq {D} ilə işarə edək) daxil edilir. Sonrakı mərhələdə qoyulmuş məsələnin alqoritmi tərtib olunur. Alqoritmə əsasən hecaya ayırmaq proqramı qurulur. Burada saitlərin axtarış proqramına müraciət olunduğundan (proqramı Muradova Lalə əvvəlki tədqiqatlarda hazırlamışdır) kompüterin yaddaşına riyazi təminat bazası kimi daxil edilmişdir. Xatırlatmalıyıq ki, yuxarıda qoyulmuş məsələnin həlli üçün onun alqoritmini və blok-sxemini qurmalıyıq. Burada nəzərinizə çatdırmalıyıq ki, belə səpkili məsələlərin həlli üçün “C++” alqoritmik dilin seçilməsi daha məqsədə uyğundur [Vəliyeva: 2018, s.20]. Bu məqsədlə ixtiyari mətndən seçilmiş bir parça (fragment) götürək və əməliyyatı onun üzərində aparaq. Məs.: “kompüter dilçiliyi məsələlərinin həlli üçün təkə linqvistik biliklər kifayət etmir” [Vəliyeva: 2022, s.3].

Yuxarıda qeyd etdiyimiz kimi, məsələnin həlli üçün onun alqoritmini və blok-sxemini quraq:

1. Mətn daxil edilir.
2. Mətdən cümlələr ayrılır. Mətn bitmişdirmi? Hə – keç 11-ə. Yox – keç 3-ə.
3. Cümlədən başlanğıc söz ayrılır. Söz bitmişdirmi? Hə – keç 2-yə. Yox –
4. Cümlə bitmişdirmi? Hə – keç 5-ə. Yox – keç 6-ya.
5. Əgər bitibsə, mətdən sonrakı cümlə ayrılır.
6. Sözü başlanğıc hərfi ayrılır. Keç 7-yə.
7. Ayrılmış qrafem {A₅} ilə müqayisə edilir. Simvol {A₅}-dədirmi? Hə – keç 8-ə. Yox – keç 9-a.
8. Simvol çap olunur. Keç 2-yə.
9. Ayrılmış qrafem {A₂} ilə müqayisə edilir. Simvol {A₂}-dədirmi? Hə – keç 10-a. Yox – keç 11-ə.
10. Simvol çap olunur. Keç 2-yə.
11. Proqramı dayandır.
12. Cümlə, sözlərin və modellərin sayını çap edin.

Burada bazaya aşağıdakı çoxluqlar siyahısına daxil etmişik:

$A_1 = \{A, I, O, U, E, İ, Ö, Ü, a, ı, o, u, e, i, ö, ü\}$

$A_2 = \{\text{samitlər}\} \{b, c, ç, d, f, g, ğ, h, x, j, k, q, l, m, n, p, z, s, ş, t, v, y\}$

$A_3 = \{‘.’, ‘?’ , ‘!’\}$

$A_4 = \{‘.’, ‘?’ , ‘!’ , ‘;’ , ‘:’ , ‘;’ , ‘;’\}$

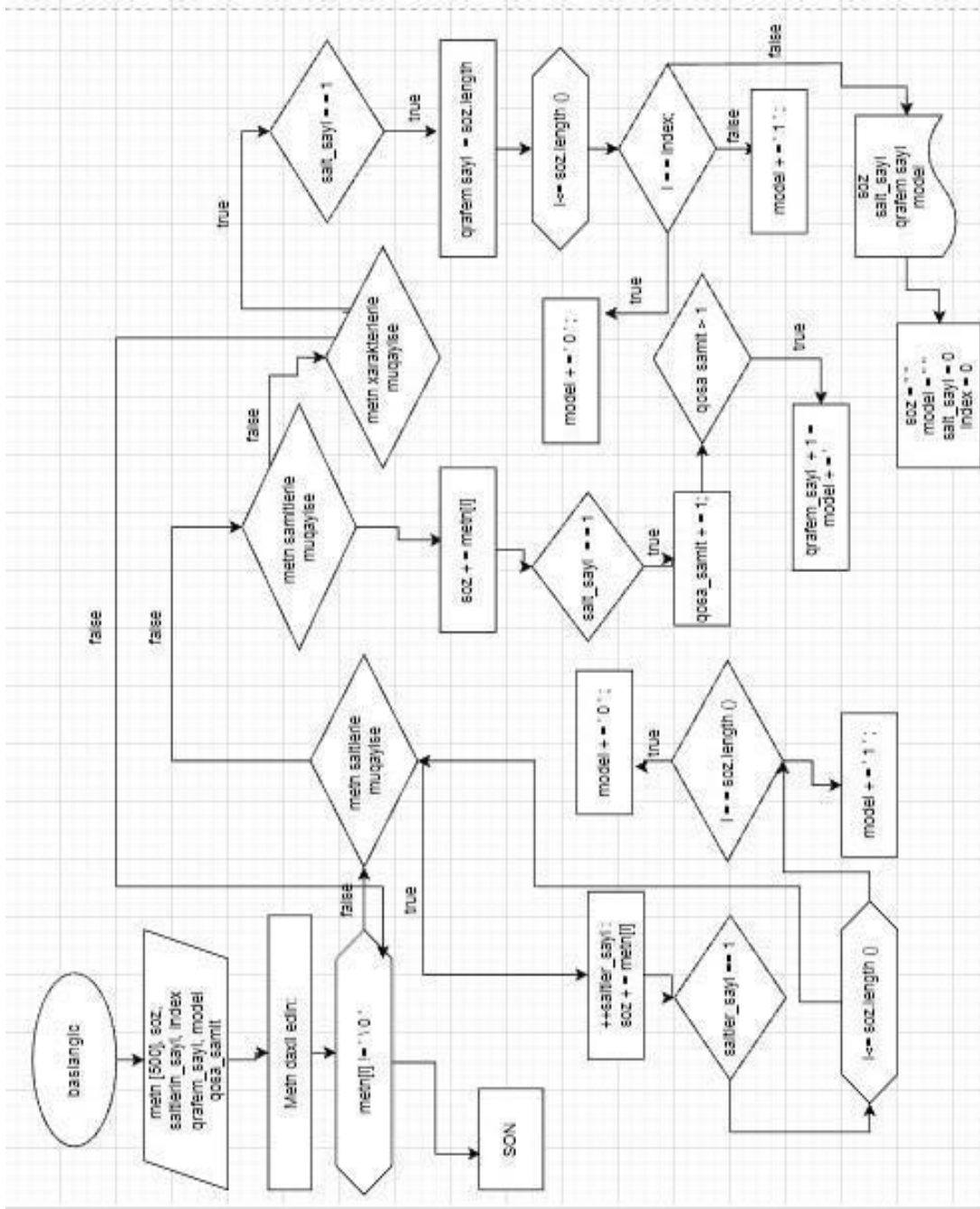
$A_5 = \{\text{heca tipləri: } 01, 011, 0111, 10, 100, 110, 1110, 101, 1011, 1101, 11011, 10111, 11101, 111011, 110111\}$

$A_5^1 = \{1110, 1011, 1101\}$ – sayı 4 olan

$A_5^2 = \{11011, 11101\}$ – sayı 5 olan

$A_5^3 = \{111011, 110111, 101111\}$ – sayı 6 olan

Burada 0 – sait, 1 – samit ilə işarə edilir.



Məsələnin alqoritmi aşağıdakı ardıcılıqla tərtib olunur:

1. Cümlədən başlanğıc sözü ayırmaq, 2-yə keçməli
2. Cümlə bitmişdirmi? Hə → prosesin sonu; Yox → 3-ə keçməli
3. “Kompüter” (nümunədən ayrılmış söz) saitlərin axtarışı proqramına keçməli. Yəni sözün ilk qrafemini ayırmalı. Söz bitmişdirmi? Hə → 4; Yox → 3-ə keçməli
4. Ayrılmış qrafem saitdirmi? Hə → 5; Yox → 2-yə keçməli
5. Model qruplaşdırılır.

Məsələnin həllini qurulmuş uyğun proqramda izləyək:

Proqrama əvvəlcə char tipində massiv saitlər [18] =

{'a','ı','o','u','e','ə','i','ö','ü','A','I','O','U','E','Ə','İ','Ö','Ü'};

char tipində massiv samitlər [44] = {'b','c','ç','d','f','g','ğ','h','x','j','k','q','l','m','n','p','r','s','ş','t','v','z','B','C','Ç','D','F','G','Ğ','H','X','J','K','Q','L','M','N','P','R','S','Ş','T','V','Z'}; və char tipində massiv şəkildə xarakterlər [7] = {'.',',',';','!','?',':',';'}; daxil edilir. Həmçinin mətn (char tipində massiv), söz (string tipində), saitlərin_sayı (int tipdə), qrafem_sayı (int tipdə), model (string tipində), index (int tipində), qosa_samit (int tipində) adında dəyişənlər yaradılır. Proqramlaşdırma dilində hər bir simvolun ayrı-ayrı “aski” kodu var. Bunun üçün böyük, kiçik hərflər və həmçinin boşluğu da daxil etməliyik. Daha sonra mətn daxil edilir və dövr başlayır. Mətn[i]= '\0' - bu o deməkdir ki, mətnin sonuna qədər dövr davam edir. Dövrün içində ilk şərtimiz dövrə uyğun olaraq, mətnin 1 hərfi ilə saitlərin müqayisəsidir. Əgər hərf saitdirsə, saitlərin sayı artır və “söz” adlı dəyişənə müqayisə olunan hərf əlavə olunur. Heca sayını tapmaq üçün saitlərin sayından istifadə edəcəyik. “söz” dəyişənindən isə, mətndəki sözləri bir-bir ayırmaq üçün istifadə olunacaq. Qrafem sayını tapmaq üçün bir şərt daha verilir, saitlərin sayı 1-ə bərabərdirmi? Əgər “true”-dursa, qrafem sayı və indeks sözün uzunluğuna bərabər olacaq. Bunun səbəbi qrafem sayının mətnin ilk hecasındakı hərf sayına bərabər olaraq hesablanmasıdır. İlk heca isə ilk saitle müəyyən olunur və beləliklə, sait sayının birə bərabər olmağı buna görə yoxlanılır. Modelimiz isə qrafemin sayı qədər olur və saitlər 0, samitlər isə 1 ilə işarələnir. Bu şərtə cədvəldəki sıra nömrəsi 1, 5, 6 və 7 olan qrupdakı sözlərin modeli, hecası və qrafem sayı təyin olunur. “İf”-dən sonra “else if” halına baxılır. Burada samitlər müqayisə olunur. Samitdirsə, yenə “söz” dəyişənində hərfimiz əlavə olunur. Burada isə saitlərin sayı 1-ə bərabər olduqda arxa-arkaya 2 dəfə samit gəlicə, bu zaman “qosa_samit” adında dəyişən sayı 2 olur və bu zaman qrafem sayı 1 rəqəm artır və modelin sonunda 1 samit üçün yer ayrılır. Bu şərt isə kom-püter, kom-bayn və s. qrupunu təyin etməkdir. 3-cü böyük şərtimiz isə qalan digər simvolların müqayisəsidir.

Xarakterlər yoxlanılır və buna uyğun olaraq sözün sonu tapılır. Burada olan şərt isə hər sözün sonunda yoxlanılır, söz əgər 1 saitdən ibarətdirsə deməli, 1 hecalıdır. Bu zaman qrafem sayı sözün uzunluğuna bərabərdir. Və sonda hər dəfə sözün sonunda bütün dəyişənlər sıfırlanır ki, təzə sözə keçəndə yenisi yazılsın.

Proqramın kodu (C++ dilində):

```
#include <iostream>
#include <conio.h>
#include <stdio.h>
```

```
using namespace std;
```

```

int main() {
    char metn[100];
    cout << "Mətni daxil edin: ";
    cin.getline(metn, 100);

    string soz;
    int saitler_sayi = 0;
    int qrafem_sayi;
    string model;
    int index;
    int qosa_samit = 0;

    for (int i = 0; metn[i] != '\0'; i++)
    {
        if (metn[i] == 'a' || metn[i] == 'ı' || metn[i] == 'o' || metn[i] == 'u' || metn[i] == 'e' || metn[i]
        == 'ə' || metn[i] == 'ı' || metn[i] == 'ö' || metn[i] == 'ü' || metn[i] == 'A' || metn[i] == 'I' ||
        metn[i] == 'O' || metn[i] == 'U' || metn[i] == 'E' || metn[i] == 'Ə' || metn[i] == 'İ' || metn[i] ==
        'Ö' || metn[i] == 'Ü') {

            ++saitler_sayi;
            soz += metn[i];
            if (saitler_sayi == 1) {
                qrafem_sayi = soz.length();
                index = soz.length();
                for (size_t i = 1; i <= soz.length(); i++)
                {
                    if (i == soz.length()) {
                        model += '0';
                    }
                    else {
                        model += '1';
                    }
                }
            }
            else if (metn[i] == 'b' || metn[i] == 'c' || metn[i] == 'ç' || metn[i] == 'd' ||
            metn[i] == 'f' || metn[i] == 'g' || metn[i] == 'ğ' || metn[i] == 'h' || metn[i] == 'x' || metn[i] == 'j' ||
            metn[i] == 'k' || metn[i] == 'q' || metn[i] == 'l' || metn[i] == 'm' || metn[i] == 'n' || metn[i] == 'p'
            || metn[i] == 'r' || metn[i] == 's' || metn[i] == 'ş' || metn[i] == 't' || metn[i] == 'v' || metn[i] == 'z'
            || metn[i] == 'B' || metn[i] == 'C' || metn[i] == 'Ç' || metn[i] == 'D' || metn[i] == 'F' || metn[i]
            == 'G' || metn[i] == 'Ğ' || metn[i] == 'H' || metn[i] == 'X' || metn[i] == 'J' || metn[i] == 'K' ||
            metn[i] == 'Q' || metn[i] == 'L' || metn[i] == 'M' || metn[i] == 'N' || metn[i] == 'P' || metn[i] ==
            'R' || metn[i] == 'S' || metn[i] == 'Ş' || metn[i] == 'T' || metn[i] == 'V' || metn[i] == 'Z') {

```

```

soz += metn[i];
if (saitler_sayi == 1) {
    qosa_samit += 1;
    if (qosa_samit > 1) {
        qrafem_sayi += 1;
        model += '1';
    }
}
}
else if(metn[i] == ' ' || metn[i] == '.' || metn[i] == ',' || metn[i] == '!' || metn[i]
== '?' || metn[i] == ':' || metn[i] == ';')
{
    if (saitler_sayi == 1) {
        qrafem_sayi = soz.length();
        model = "";
        for (size_t i = 1; i <= soz.length(); i++)
        {
            if (i == index) {
                model += '0';
            }
            else {
                model += '1';
            }
        }
    }
    cout << soz << " = " << saitler_sayi << " heca" << ", qrafem sayi: "
<< qrafem_sayi << ", model: " << model << endl;
    soz = "";
    model = "";
    saitler_sayi = 0;
    index = 0;
    qosa_samit = 0;
}

}

return 0;
}

```

Beləliklə, avtomatik yolla söz-formaları hecalara ayırmaq mümkün olur.

Nəticə:

```

Microsoft Visual Studio Debug Console
Mətni daxil edin: Kompüter dilçiliyi məsələlərinin həlli üçün təkcə linqvistik biliklər kifayət etmir.
Komputer = 3 heca, qrafem sayı: 3, model: 101
dilçiliyi = 4 heca, qrafem sayı: 3, model: 101
məsələlərinin = 6 heca, qrafem sayı: 2, model: 10
həlli = 2 heca, qrafem sayı: 3, model: 101
Üçün = 2 heca, qrafem sayı: 1, model: 0
təkcə = 2 heca, qrafem sayı: 3, model: 101
linqvistik = 3 heca, qrafem sayı: 4, model: 1011
biliklər = 3 heca, qrafem sayı: 2, model: 10
kifayət = 3 heca, qrafem sayı: 2, model: 10
etmir = 2 heca, qrafem sayı: 2, model: 01
C:\Users\Lenovo\source\repos\Project_Heac\Debug\Project_Heac.exe (process 94840) exited with code 0.
Press any key to close this window . . .

```

Hecaların avtomatik yolla müəyyənləşdirilməsi poliqrafiyada (söz-formaların sətirdən-sətrə keçirilməsində), məşin tərcüməsində (sözün sintezində), korrektor sistemlərində (sözlərin düzgün yazılışını yoxlayan – spelling sistemində) böyük əhəmiyyət kəsb edir.

ƏDƏBİYYAT

1. Dəmirçizadə Ə.M. Müasir Azərbaycan dili (I hissə), Bakı, “Maarif”, 1972.
2. Vəliyeva K.A. Mətnin avtomatik sintaktik təhlili və sintezi, Bakı, “Elm və təhsil”, 1996.
3. Vəliyeva K.A. Kompüter dilçiliyi məsələlərinin həll yolları. Bakı, 2022.
4. Vəliyeva K.A., Muradova L. Kompüter dilçiliyinə dair bəzi məsələlərin proqramlaşdırılması. Tədqiqatlar 3, “Elm və Təhsil”, Bakı, 2018, s.20-28

Kamile VELIYEVA
Lala MURADOVA

THE ROLE OF A.DEMIRCHIZADEH'S PHONOLOGICAL RESEARCH IN THE APPLIED ISSUES

SUMMARY

After the data is entered into the computer's memory, the problem programme (division of random word forms into syllables) is built according to the developed algorithm. In the first stage of the program, vowels and consonant graphemes are searched from word forms. Then graphemes with vowels and consonants are compared with syllable models stored in computer memory. Overlapping combinations are considered syllables. Otherwise, the process continues.

Key words: *algorithm, programme, grapheme, syllable, model.*

Камиля ВЕЛИЕВА
Лала МУРАДОВА

РОЛЬ ФОНОЛОГИЧЕСКИХ ИССЛЕДОВАНИЙ А.ДЕМИРЧИЗАДЕ
В ПРИКЛАДНЫХ ВОПРОСАХ

РЕЗЮМЕ

После ввода данных в память компьютера по разработанному алгоритму строится программа заданной задачи (разложение произвольных словоформ на слоги). На первом этапе программы осуществляется поиск гласных и согласных графем в словоформах. Затем графемы с сочетаниями гласных и согласных сравниваются со слоговыми моделями, хранящимися в памяти компьютера. Совпадающие сочетания считаются слогами. В противном случае процесс продолжается.

Ключевые слова: алгоритм, программа, графема, слог, модель.