

KRIPTOGRAFIJA

Zadaća 1.130

Filip Nikšić

25. ožujka 2007.

Zadatci i rješenja

1. Afinom šifrom s ključem K je iz otvorenog teksta na hrvatskom jeziku dobiven sljedeći šifrat:

```
BWLDU HNROK BSCOY WXYOW XGCOF USGJR OFOXM XUKUD  
GJWFB GROCB GMXWG XBWRO UMWYM XUKUD GJWYW XOYWX  
GSWBO WJXGS COKBS COFBG YWBHU ROLWJ GYWNW KOUYO  
XBGRW HBURO TOMIF BUISW TWNGC WUMTG YKOUY OXBGR  
MCOYR OBO
```

Navedite pet najfrekventnijih slova, te pet najfrekventnijih bigrama u ovom šifratu.

Odredite ključ $K = (a, b)$ i otvoreni tekst (dekriptirajte šifrat)!

- Rj.** Frekvencijska analiza napravljena je priloženim programom *frequency*. Relevantan dio izlaza:

Slova:	Bigrami:
[22] O	[7] RO
[20] W	[6] YW
[15] G	[5] BG CO
[14] B	[4] OY WX
[13] U	

Vidimo da je bigram RO najčešći. Slovo O je na samom vrhu po broju pojavljivanja, a R se uopće ne nalazi među prvih 5 najčešćih slova. Stoga se čini razumnim pretpostaviti da je bigram RO upravo najčešći bigram hrvatskog jezika JE.

Kad uvrstimo odgovarajuće numeričke reprezentante slova, ta pretpostavka vodi nas na sljedeći sustav jednadžbi:

$$\begin{aligned} 9a + b &= 17 \\ 4a + b &= 14 \end{aligned}$$

Oduzimanjem druge jednadžbe od prve dobivamo $5a = 3$ pa je $a = 11$ (operacije vršimo u prstenu \mathbb{Z}_{26}). Lako dobijemo uvrštavanjem a u neku od jednadžbi da je $b = 22$.

Priloženi program *affine* konvertira šifrat u otvoreni tekst uz prepostavku ključa $K = (a, b)$ (programu dajemo parametre $a^{-1} = 19$ i $b = 22$). Evo izlaza programa:

```
RAZDO BLJEG RCKEM ATEMA TIKEP OCINJ EPETS TOGOD
INAPR IJEKR ISTAI TRAJE OSAMS TOGOD INAMA TEMAT
ICARE ANTIC KEGRC KEPRI MARNO JEZAN IMALA GEOME
TRIJA BROJE VESUP ROUCA VALIK AOSVI MGEOF ETRIJ
SKEMJ ERE
```

Zaključujemo da je prepostavka bila ispravna. Tekst je šifriran ključem $K = (11, 22)$, a lješe napisan, glasi:

Razdoblje grčke matematike počinje petsto godina prije Krista i traje osamsto godina. Matematičare antičke Grčke primarno je zanimala geometrija. Brojeve su proučavali kao svim geometrijske mjere.

(Moram naglasiti da sam nekoliko puta ručno provjerio i uvijek se pojavi ta riječ *svim* u zadnjoj rečenici koja se čini suvišnom.)

2. Dekriptirajte šifrat

```
XHYKH ETAYO RJMES QEYKY GHMGQ NYTMO RNRZY KMGQD
NSQZS QPRSH IERKQ APQEM IYYMR TZHSH ESQWN HZERW
WPHLH OPQLT SQLEY ARLY SHORL EYSQI YRTIH ZAMYM
GEHAA RDOPR DYTPO PQLNH GQYGJ RPQGH APHSR ZQWRL
YEQ
```

dobiven supstitucijskom šifrom, i to Cezarovom šifrom s ključnom riječi. Poznato je da je otvoreni tekst na hrvatskom jeziku, te da je ključna riječ izraz (fraza ili riječ) na hrvatskom jeziku.

Rj. Frekvencijska analiza programom *frequency* dala je sljedeće rezultate:

Slova:	Bigrami:	Trigrami:
[16] H Q Y	[6] SQ	[2] ESQ HOP LEY
[15] R	[4] PQ	MGQ OPQ PQL
[11] E	[3] EY GQ HO MG	
[10] S	OP OR QL RL	
[8] M P	SH YK	
[7] A G L	[2] AP AR ER ES	
[6] O T Z	GH HA HE HS	
[5] N	HZ IY LE NH	
[4] I K W	PH PR QE QW	
[3] D J	RD RT RZ YG	
[1] X	YM YS YT	

(Bigrame i trigrame koji se javljaju samo jednom nisam ovdje napisao jer ih je previše, a nisam ih koristio ni za jednu prepostavku.)

Nakon niza neuspjeha i pogrešnih prepostavki, evo kako sam došao do rješenja:

Naslućujemo da su najfrekventnija slova H, Q, Y, R samoglasnici A, I, E, O jer se često pojavljuju i sudjeluju u mnogim bigramima. Samo je pitanje koje slovo je koji samoglasnik.

Zanimljivo je za primijetiti da se u šifratu uopće ne pojavljuju slova B, C, F, U, V. Četiri od pet tih slova morala bi zamjenjivati slova Q, W, X, Y, kojih nema u hrvatskom jeziku. Poznata je činjenica da se radi o Cezarovoj šifri s ključnom riječi pa je logično pretpostaviti da BCF ne pripada ključnoj riječi i upravo zamjenjuje WXY. To se onda lijepo slaže s pretpostavkom da G zamjenjuje Z, a najfrekventnije slovo H zamjenjuje samoglasnik A.

Najčešći bigram SQ mogao bi biti JE, dakle, još jedan samoglasnik bi mogao biti otkriven.

Primijetimo sad da je X najrjeđe slovo (pojavljuje se samo jednom), a Y se pojavljuje čak 16 puta. U abecedi šifrata ta dva slova nedvojbeno stoje jedno do drugog jer se nijedno ne koristi za ključnu riječ koja je na hrvatskom. U hrvatskom jeziku slova H i I stoje jedno do drugog u abecedi. H spada među najrjeđa slova, a I među najčešća. Pretpostavimo da X zamjenjuje H, a Y zamjenjuje I. Pretpostavimo i da W zamjenjuje G jer u abecedi šifrata W sigurno stoji lijevo od X.

Konačno, R bi mogao biti posljednji od neotkrivenih čestih samoglasnika—O, a E, kao posljednje slovo s dvoznamenkastim brojem pojavljivanja za koje još ništa nismo pretpostavili, bi moglo zamjenjivati N. Pogledajmo trenutne pretpostavke:

```
ABCDEFGHIJKLMNPQRSTUVWXYZ  
h...q.wxys...er.....bcfg
```

Program *change* učitava navedena dva retka iz konfiguracijske datoteke *change.conf* i zamjenjuje slova iz šifrata s odgovarajućim slovima otvorenog teksta:

```
hai.a n..i. o..nj eni.i za.ze .i... o.o.i ..ze.  
.je.j e.oja .no.e ..en. .ii.o ..aja njeg. a.nog  
g.a.a ...e.. je.ni .o..i ja.o. nije. io..a ...i.  
zna.. o...o .i.a. .e..a zeiz. o.eza ..ajo .ego.  
ine
```

Na kraju drugog i početku trećeg razaznaje se nekakvo *osvajanje glavnog grada*, a posljednja riječ u tekstu opako podsjeća na *godine*. Klupko se počelo odmatati. Jednostavan iterativni proces pogadanja riječi u otvorenom tekstu i nadopunjavanja abecede šifrata nakon nekoliko iteracija daje sljedeći otvoreni tekst:

```
haica nskip obunj enici zauze lisup olovi cuzem  
ljevj eroja tnoce krenu tiiuo svaja njegl avnog  
grada preds jedni kodbi japod nijet iosta vkuiu  
znakk ompro misap redla zeizb oreza krajo vegod  
ine
```

Abeceda šifrata izgleda ovako:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
hjkqlq.wxysandero.ptimzbcfg

Konačan udarac zadajemo pogađanjem ključne fraze: *Sanaderov optimizam.*

ABCDEFGHIJKLMNOPQRSTUVWXYZ
hjkqlquwxysanderovptimzbcfg

Ljepše napisan otvoreni tekst:

Haićanski pobunjenici zauzeli su polovicu zemlje. Vjerljivo će krenuti i u osvajanje glavnog grada. Predsjednik odbija podnijeti ostavku i u znak kompromisa predlaže izbore za kraj ove godine.

Izvorni kôdovi

frequency.c

```
#include <stdio.h>
#include <string.h>

#define nalphabet 19683 /* Koristim engl. alfabet + praznina */

typedef struct {
    int word;
    int freq;
} words;

words hash_table[nalphabet];

int compare(const void *a,const void *b) {
    return (*(words *)b).freq-(*words *)a.freq;
}

int main(int argc, char **argv) {
    FILE *in;
    int i,last;
    char c,pre=0,prepre=0;
    char rijec[4];

    if (argc<2 || (in=fopen(argv[1],"r"))==NULL) {
        printf("Couldn't open input file!\n");
        return 0;
    }

    for (i=0;i<nalphabet;i++) hash_table[i].word=i;
```

```

while (!feof(in)) {
    fscanf(in,"%c",&c);
    if (c>='a' && c<='z') c-=’a’-’A’;
    if (c>='A' && c<='Z') {
        c-=’A’-1;
        hash_table[c].freq++;
        if (pre) hash_table[pre*27+c].freq++;
        if (prepre) hash_table[prepre*27*27+pre*27+c].freq++;
        prepre=pre;
        pre=c;
    }
}

qsort(hash_table,27,sizeof(words),compare);
qsort(hash_table+27,27*27-27,sizeof(words),compare);
qsort(hash_table+27*27,27*27*27-27*27,sizeof(words),compare);

printf("Slova:");
for (i=0,last=0;i<27 && hash_table[i].freq>0;i++) {
    if (!hash_table[i].word) continue;
    if (hash_table[i].freq!=last)
        printf("\n[%2d]",last=hash_table[i].freq);
    printf(" %c",hash_table[i].word+’A’-1);
}
printf("\n\nBigrami:");
for (i=27,last=0;i<27*27 && hash_table[i].freq>0;i++) {
    rijec[0]=(hash_table[i].word)/27+’A’-1;
    rijec[1]=(hash_table[i].word)%27+’A’-1;
    rijec[2]=’\0’;
    if (rijec[0]==’A’-1 || rijec[1]==’A’-1) continue;
    if (hash_table[i].freq!=last)
        printf("\n[%2d]",last=hash_table[i].freq);
    printf(" %s",rijec);
}
printf("\n\nTrigrami:");
for (i=27*27,last=0;i<27*27*27 && hash_table[i].freq>0;i++) {
    rijec[0]=(hash_table[i].word)/(27*27)+’A’-1;
    rijec[1]=(hash_table[i].word)/27%27+’A’-1;
    rijec[2]=(hash_table[i].word)%27+’A’-1;
    rijec[3]=’\0’;
    if (rijec[0]==’A’-1 || rijec[1]==’A’-1
        || rijec[2]==’A’-1) continue;
    if (hash_table[i].freq!=last)
        printf("\n[%2d]",last=hash_table[i].freq);
    printf(" %s",rijec);
}
printf("\n");

fclose(in);

```

```
        return 0;
}
```

affine.c

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc,char **argv) {
    FILE *in;
    int a,b,c;
    char cc;

    if (argc<4) {
        printf("Usage: affine <a^(-1)> <b> <file>\n");
        return 0;
    }

    if ((in=fopen(argv[3],"r"))==NULL) {
        printf("Couldn't open input file.\n");
        return 0;
    }

    a=strtol(argv[1],NULL,10);
    b=strtol(argv[2],NULL,10);

    while (!feof(in)) {
        fscanf(in,"%c",&cc);
        if (cc>='a' && cc<='z') cc-=='a'-'A';
        if (cc>='A' && cc<='Z') {
            cc-=='A';
            /* Za kalkulaciju koristim int da
             * izbjegnem overflow kod char-a. */
            c=a*(cc-b);
            /* Klasican % jednostavno ne radi dobro
             * za negativne integere */
            while (c>=26) c-=26;
            while (c<0) c+=26;
            cc=c+'A';
        }
        printf("%c",cc);
    }

    fclose(in);
    return 0;
}
```

change.c

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

```

int main(int argc, char **argv) {
    FILE *in,*fsups;
    char buffer[30],sups[26];
    char c;
    int i;

    in=fopen(argv[1],"r");
    fsups=fopen("change.conf","r");

    fscanf(fsups,"%s",buffer); /* Prvu liniju ignoriramo */
    fscanf(fsups,"%s",buffer);

    for (i=0;i<26;i++) sups[i]='.';
    for (i=0;i<30 && buffer[i];i++) {
        if (buffer[i]>='a' && buffer[i]<='z') buffer[i]-='a'-'A';
        if (buffer[i]>='A' && buffer[i]<='Z')
            sups[buffer[i]-'A']=i+'a';
    }

    while (!feof(in)) {
        fscanf(in,"%c",&c);
        if (c>='a' && c<='z') c-= 'a'-'A';
        if (c>='A' && c<='Z') printf("%c",sups[c-'A']);
        else printf("%c",c);
    }

    fclose(in);
    fclose(fsups);

    return 0;
}

```