Lucrările Seminarului de
CREATIVITATE MATEMATICĂ
Vol.5 (1995-1996), 109-114

Dedicat Centenarului Gazetei Matematice (1985-1995)

ASUPRA UNEI METODE DE COMPRIMARE/DECOMPRIMARE A FIŞIERELOR

1. Introducere

Problema de la care se porneşte este următoarea: un beneficiar achiziţionează un produs soft cu anumite caracteristici (număr de utilizări, tipuri de teste etc.). În timp el şi-a epuizat total sau parţial aceste facilităţi, dar doreşte utilizarea în continuare a programului. Este necesar în acest caz un upgrade pentru acest produs soft, iar acest upgrade trebuie să fie cât mai eficient posibil. Desigur că este posibil ca beneficiarului să i se livreze un nou pachet conţinand programul cu noii parametri solicitaţi, dar această operaţie este costisitoare atât ca timp cât şi ca material, şi în plus este superfluă, programul propriu-zis este acelaşi. Într-o asemenea situaţie beneficiarul are nevoie doar de modificarea "contoarelor" unde sunt păstrate caracteristicile ce trebuiesc modificate. În cazul a numeroase produse soft aceste caracteristici sunt păstrate în cheia de protecţie hard (hardlock), cheie pe care beneficiarul o primeşte în momentul achiziţionării produsului (prima achiziţionare). Revenind, pentru un upgrade de acest fel este nevoie doar de modificarea unor chei din hardlock, programul rămânând nemodificat. Utilizatorul trebuie să trimită distribuitorului programului doar conţinutul cheii de protecţie (sau o parte a sa), distribuitorul face modificările (de obicei incrementări) de rigoare şi returnează noua cheie (noul conţinut). Această operaţie trebuie făcută cu cheltuieli minime, atât din punctul de vedere al beneficiarului cât şi al distribuitorului. Se

va studia în continuare modul în care se va face acest schimb prin modem sau prin telefon.

2. "Schimbul" de informaţie

Se va porni în continuare de la premisa că cheia hard are mărimea de 496 octeţi. Schimbul prin telefon se va face în modul următor:

1) utilizatorul se hotărăşte să facă upgrade-ul

2) selectează opţiunea respectivă din program (eventual poate folosi un program de sine stătător)

3) pe ecran apare conţinutul hardlock-ului comprimat, criptat, codificat; se va numi acest conţinut **şir de intrare**

4) utilizatorul telefonează distribuitorului căruia îi comunică şirul de intrare şi noii parametri solicitaţi

5) distribuitorul lansează un program special care cere ca date de intrare şirul de intrare şi noii parametri

6) distribuitorul introduce datele de intrare

7) programul returnează un **şir de ieşire** (tot comprimat, criptat), şir care va fi comunicat beneficiarului

8) beneficiarul tastează acest şir ca răspuns la opţiunea selectată

9) din acest moment programul va funcţiona cu noile contoare.

2.1. Observaţii

1) Este de dorit ca atât şirul de intrare cât şi cel de ieşire să fie cât mai scurt posibil.

2) Alfabetul folosit pentru scrierea acestor şiruri să fie cât mai inteligibil, având în vedere că şirurile se transmit prin viu grai.

2.2. Schimbul via modem

În cazul transmiterii datelor prin modem nu mai este necesară codificarea cheii comprimate, care implicit înseamnă creşterea dimensiunii şirului de intrare. Ambiguităţile care pot apărea în cazul schimbului prin telefon aici nu mai există. Acest schimb este mai avantajos din multe puncte de vedere, dar presupune că utilizatorul are modem!

3. Comprimarea propriu-zisă

Problema care se pune este comprimarea unui fişier de 496 octeţi. Acest fişier de intrare are (sau ar trebui să aibă) anumite caracteristici, de exemplu conţine multe secvenţe FFFF şi 0000. Pornind de la această observaţie, ca prim pas în comprimare, se va efectua o compresie cu pattern pe 16 biţi.

3.1. Comprimarea cu pattern pe 16 biţi

Fie un şir de pornire de dublu octeţi. Se va construi alfabetul asociat acestui şir:

$s_1, s_2, \ldots, s_k$ - şirul de dublu octeţi (nu neapărat distincţi)

$a_1, a_2, \ldots, a_t$ - alfabetul asociat acestui şir

$p_1, p_2, \ldots, p_t$ - frecvenţele elementelor alfabetului.

Atunci: a) $1 \le k$

b) $\sum_{i=1}^{t} p_i = k$

c) lungimea şirului de intrare are 16xk biţi.

Se va comprima şirul s (de intrare) la şirul z (de ieşire) astfel:

a) z este un şir de octeţi

b) în z_1 şi z_2 se va păstra lungimea în biţi a şirului de intrare s (16xk)

c) în z_3 şi z_4 se va păstra dublu octetul care apare cel mai frecvent (acel a_i pentru care $p_i = \max\{p_j, j=\overline{1,t}\}$)

d) se creează o mască de biţi astfel:

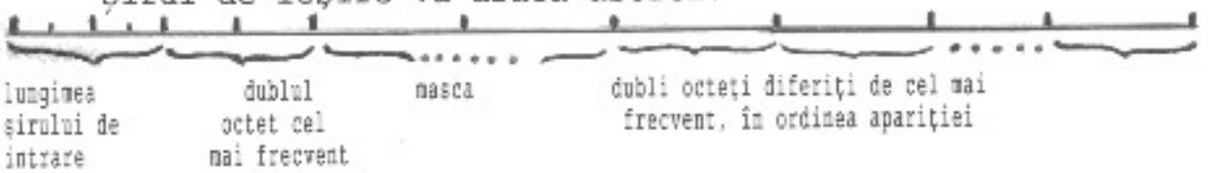
- bit pe 1 dacă dublul octet (caracterul) respectiv este a_i
- bit pe 0 dacă dublul octet (caracterul) respectiv nu este a_i
- această mască se scrie în $z_5, z_6, \ldots, z_t$
- lungimea acestui subşir este t-5+1 octeţi şi

$$t-5+1 = \begin{cases} \frac{k}{8} & \text{dacă 8 } \textit{divide } k \\ \left[\frac{k}{8}\right]+1 & \text{dacă 8 } \textit{nu divide } k \end{cases}$$

- dacă 8 nu divide pe k se completează ultimii biţi (nesemnificativi) din z_c cu 0.

e) se completează z cu acei dubli octeţi din s care sunt diferiţi de a_i în ordinea apariţiei lor în z cu repetiţie (dacă se repetă).

Şirul de ieşire va arăta astfel:



La decomprimare se procedează astfel:

a) se citeşte lungimea şirului de intrare

b) se citeşte dublul octet cel mai frecvent

c) se parcurge masca bit cu bit (mai puţin eventualii biţi nesemnificativi din ultimul octet din mască; aceştia se cunosc pentru că se cunoaşte lungimea şirului) şi pentru fiecare bit 1 se scrie în şirul de ieşire dublul octet cel mai frecvent, iar pentru fiecare bit 0 se scrie dublul octet următor din secvenţa de după mască.

Problema care se pune este următoarea: este şirul de ieşire z mai scurt decât cel de intrare s? Evident, pentru a se realiza o compresie propriu-zisă răspunsul trebuie să fie afirmativ.

Fie l(s) şi l(z) lungimile celor două şiruri.

Atunci $l(z) = 2+2+t-4+2\cdot(p_1+\ldots+p_{i-1}+p_{i+1}+\ldots+p_t) = k+2\cdot(k-p_i)$.

Pentru cazul considerat k= 496/2= 248. Fie $x=\frac{p_i}{8}$ procentul în care cel mai frecvent dublu octet apare în şirul de intrare.

Lungimea şirului de ieşire devine $l(z) = 35+496\cdot(1-x)$. Lungimea şirului de intrare este l(s)= 496(octeţi).

Condiţia ca compresia să fie eficientă este ca l(s)> l(z), adică $x>35/496 \Leftrightarrow x> 0.07$. În concluzie, dacă procentul în care cel mai frecvent dublu octet este de cel puţin 7%, atunci această metodă este eficientă.

Pentru a se comprima un şir se va folosi această metodă

recursiv. Evident există un număr critic de aplicare a metodei începând de la care dimensiunea şirului nu mai scade. Se pot determine condiţii matematice riguroase pentru aplicarea eficientă recursivă a metodei de un anumit număr de ori. De exemplu, pentru ca aplicarea recursivă de două ori a acestei metode de comprimare asupra unui şir de 512 octeţi, condiţia este $p_2+\frac{p_1}{16}>\frac{1}{16}+\frac{1}{256}$, unde

p_1 şi p_2 sunt procentajele celor mai frecvenţi doi dublu octeţi din şir. Bineînţeles, nu este nevoie să se verifice asemenea condiţii înainte de aplicarea recursivă a metodei de comprimare, este suficient să se aplice metoda atâta timp cât dimensiunea şirului scade.

3.2. Comprimarea cu pattern pe 8 biţi

Este similară cu cea prezentată anterior (pe 16 biţi), diferenţa fiind aceea că elementele şirului de intrare sunt octeţi. Ca şi consecinţă a acestui fapt, dimensiunea măştii se dublează, condiţia de eficienţă fiind mai slabă (x> 13%).

3.3. Comprimare combinată

Pentru unele fişiere (şiruri) se obţin rezultate eficiente dacă aceste metode de comprimare recursivă se aplică alternativ.

4. Codificarea

Odată şirul de intrare comprimat, se pune problema codificării acestuia, în sensul scrierii lui cu ajutorul unui alfabet accesibil (să nu uităm că şirul va fi transmis prin viu grai). Este dificil pentru o persoană să transmită prin telefon un şir format din caractere de genul: # $ & etc. Se va folosi pentru codificare un alfabet format din 64 de caractere {a,..., z, A,..., Z, +, -}. Fiecare grup de 6 biţi se codifică obţinându-se printr-un caracter din acest alfabet. Există în acest caz un dezavantaj şi anume că dimensiunea fişierului codificat creşte de 4/3 ori.

Pentru a realiza compresia, decompresia (pe 8 biţi şi pe 16 biţi), codificarea şi decodificarea s-au scris patru programe în limbajul C, iar în continuare sunt prezentate câteva dintre rezultatele obţinute.

Comprimare:

Fişierul	Tipul compresiei	Lungime iniţială	1	2	3	4	5	6	7
R.HAS	16 biţi	496	121	107	96	94	96		
R.HAS	8 biţi	496	146	113	105	90	82	83	
TEST.HAS	16 biţi	496	137	124	106	97	102		
TEST.HAS	8 biţi	496	168	145	115	105	99	90	94

Codificare:

Fişierul	Lungime necodificat (octeţi)	Lungime codificat (octeţi)
R.HAS	94	127
R.HAS	82	111
TEST.HAS	97	131
TEST.HAS	90	121

BIBLIOGRAFIE

1. Ionescu T.C., Zsako I., Structuri arborescente, Ed.Tehnică, Bucureşti, 1990
2. Knuth D.E., Tratat de programare a calculatoarelor, Algoritmi fundamentali, Ed.Tehnică, Bucureşti, 1976
3. Aho A.V., Hopcroft J.E., Data Structures and Algorithms, Addison Wesley Reading, Massachussets, 1983

ON A COMPRESSION/DECOMPRESSION METHOD

Abstract. In this paper are presented methods for compression, codification, decompression, decodification generally for certain relatively short files that have to be transmitted "by voice" on the phone (usually the contain of a hardlock) in order to upgrade a certain soft product.

Universitatea din Baia Mare
Str. Victoriei nr. 76, 4800 Baia Mare
ROMÂNIA