Giải Thuật Nén LZW

Bộ môn : Cấu trúc dữ liệu 2

Giảng viên : Lê Văn Vinh

Nội dung

1. Ý Tưởng
2. Thuật toán nén (Encoding)
3. Thuật toán giải nén (Decoding)
4. Biến thiên độ rộng của mã(Variable-Width codes)
5. Ví dụ
6. Ứng dụng

1.Ý tưởng

Thuật toán LZW được phát triển theo nguyên lý tạo ra 1 dãy mã:

-Mã từ 0 đến 255 miêu tả 1 dãy ký tự thay thế cho ký tự 8-bit tương ứng.

-Mã từ 256 đến 4095 được tạo bên trong 1 từ điển cho trường hợp lặp chuỗi trong dữ liệu.

-Mỗi bước trong khi nén,byte nhập vào được tập hợp lại thành 1 chuỗi cho đến khi ký tự tiếp theo sẽ tạo thành 1 chuỗi chưa tồn tại trong từ điển,và 1 mã mới cho chuỗi được tạo sẽ được thêm vào từ điển,và mã ấy sẽ được xuất ra file output.

2.Nén(Compression)

- Thuật toán mã hóa :
  1. Khởi tạo từ điển chứa tất cã chuỗi có 1 ký tự
  2. Tìm chuỗi W dài nhất trong từ điển đối chiếu với dữ liệu nhập hiện tại
  3. Xuất vị trí từ điển cho W ra file output và xóa W khỏi dữ liệu nhập.
  4. Thêm W và ký tự tiếp theo trong dữ liệu nhập vào từ điển
  5. Đến bước 2.

3.Giải nén

- Đọc giá trị từ dữ liệu nhập đã mã hóa và xuất ra chuỗi tương ứng từ từ điển đã được khởi tạo.
- Tại cùng 1 thời điểm nó thu được giá trị tiếp theo từ dữ liệu nhập,thêm vào từ điển xích chuỗi của chuỗi xuất.Và ký tự đầu tiên của chuỗi nhận được khi mã hóa ký tự tiếp theo.
- Sau đó trình giải nén xử lý giá trị nhập tiếp theo,quá trình lặp cho đến khi dữ liệu nhập không còn,tại thời điểm giá trị nhập cuối cùng được mã hóa không còn bất kỳ giá trị nào thêm vào từ điển.

4.Biến thiên độ rộng. (Variable-width codes)

- Là khoảng rộng của dãy bit chứa các mã đã được mã hóa trong vùng nhớ máy tính
- Trình nén mã sẽ tăng độ rộng từ p lên p+1 khi 1 chuỗi w+s không tồn tại trong từ điển(vì vậy 1 mã phải được thêm vào) nhưng giá trị tiếp theo có thể thêm trong bảng lại là 2^p.

5.Ví dụ

- Chuỗi sau sẽ được mã hóa : “TOBEORNOTTOBEORTOBEORNOT#”

Symbol	Binary	Decimal
#	00000	0
A	00001	1
B	00010	2
C	00011	3
D	00100	4
E	00101	5
F	00110	6
G	00111	7
H	01000	8
I	01001	9
J	01010	10
K	01011	11

Symbol	Binary	Decimal
L	01100	12
M	01101	13
N	01110	14
O	01111	15
P	10000	16
Q	10001	17
R	10010	18
S	10011	19
T	10100	20
U	10101	21
V	10110	22

Symbol	Binary	Decimal
W	10111	23
X	11000	24
Y	11001	25
Z	11010	26

Current Sequence	Next Char	Output	Extended Dictionary	Comment
Null	T			
T	0	20:10100	27:TO	
O	B	15:01111	28:OB	
B	E	2:00010	29:BE	
E	O	5:00101	30:EO	
O	R	15:01111	31:OR	
R	N	18:10010	32:RN	Tăng lên 6bits
N	O	14:001110	33:NO	
O	T	15:001111	34:OT	
T	T	20:010100	35:TT	
TO	B	27:011011	36:TOB	
BE	O	29:011101	37:BEO	
OR	T	31:011111	38:ORT	
TOB	E	36:100100	39:TOBE	
EO	R	30:011110	40:EOR	
RN	O	32:100000	41:RNO	
OT	#	34:100010		#stop
		0:000000		Stop code

- Dung lượng chưa mã hóa:
  - 25 ký tự * 5bits/ký tự = 125 bits.

  Mã hóa:

  - (6 mã * 5bits/mã) + (11 mã * 6bits/mã) = 96 bits

  Vậy sử dụng LZW đã tiết kiệm 29 bits trên tổng số 125 bits,giảm 22%.

Intput	Output Sequence	Từ Điển đầy đủ	Bộ dự trữ	comments
20:10100	T		27:T?	
15:01111	O	27:TO	28:O	
2:00010	B	28:OB	29:B?	
5:00101	E	29:BE	30:E?	
15:01111	O	30:EO	31:O?	
18:10010	R	31:OR	32:R?	
14:001110	N	32:RN	33:N?	
15:001111	O	33:NO	34:O?	
20:010100	T	34:OT	35:T?	
27:011011	TO	35:TT	36:TO?	
29:011101	BE	36:TOB	37:BE?	
31:011111	OR	37:BEO	38:OR?	
36:100100	TOB	38:ORT	39:TOB?	
30:011110	EO	39:TOBE	40:EO?	
32:100000	RN	40:EOR	41:RN?	
34:100010	OT	41:RNO	42:OT?	
0:000000	#			

6.Ứng dụng

- Nén LZW đã trở thành phương thức nén dữ liệu phổ biến trên máy tính.Một file text English có thể được nén thông qua LZW để giảm ½ dung lượng gốc.
- LZW đã được sử dụng trong phần mềm nén mã nguồn mỡ,nó đã trở thành 1 phần không thể thiếu trong HDH UNIX CIRCA 1986.
- LZW đã trở nên phổ biến khi nó được sữ dụng làm 1 phẩn của file GIF năm 1987.Nó cũng có thể được sử dụng trong TIFF và PDF file.

Bài Tập

- Hãy mã hóa chuỗi sau cho ra file output(trên giấy) và tính xem nén được bao nhiêu % dung lượng gốc:
- “TSYUYIROSUYITSONNTTEO#”

Tài liệu Tham Khảo

- Algorithm from WIKIPEDIA
- Code C from stackoverflow.

The end