CS240 Module 9 Compression

Graham Cooper

July 21st, 2015

Definition: Prefix free:
A $\rightarrow$ 0
B $\rightarrow$ 1
C $\rightarrow$ 01

01 can either be AB or C.

July 23rd lecture:

LZWCompressor

0.1 Going through the algorithm

A B C ... | X Y Z A B ...
w = XYZ $->$ output code $->$ w = A
k = Y

We are checking through each step and checking if the strings we encoutner are in the dictionary, if not we build a dictionary of these new strings.

See slides around slide 24 to see the table/dictionary being built

Slide 27: With the input 133, we do not know what corresponds to 133. So we know that is must be the same as the previous string we decoded, aka AN, then we know that the next character will be ANA with another character added on. That character will be the first prefix of the string we are appending it to.

If we fill up our dictionary there are a few thigns we can do:

- Do nothing
- Replace oldest items

Move to Front (again)

Use move to front to organize our dictionary

Burrows-Wheeler Transform

A = "ABRACADABRA"

becomes
A = "AAAAABBCDRR"

We sort all of the characters, then we can compress very well, similar to compressing binary, blocks of 1's/zeros

Encoding

See slides for example, slide 36

another example:

```
banana$
anana$b
nana$ba
ana$ban
na$bana
a$banan
$banana
```

Sort the above:

FIG D:

```
$banana
a$banan
ana$ban
anana$b
```

```
banana$
na$bana
nana$ba
```

Take the last element of each sorted array to get:
Q = "annb$aa"

The first a in the above array Q, is followed by $.
We write all of these follows out :

```
a $
n a
n a
b a
$ b
a n
a n
```

Notice that the values in the second column are sorted!!

we then transform this to:

```
$ b
a $
a n
a n
b a
n a
n a
```

We now have the first two columns in the sorted FIG D.
Sort and prepend again, we are prepending the array Q from before

```
a $ b
n a $
n a n
b a n
```

```
$ b a
a n a
a n a
```