Bidirectional Adaptive Compression

Aharon Fruchtman
Shmuel T. Klein
Dana Shapira

Data Compression

- Static
  - The model - the distribution of the encoded elements
    - Given in advance
    - Gathered in a first scan
- Adaptive
  - The model - learned incrementally.

Data Compression

- Statistical
  - Huffman
  - Arithmetic
- Dictionary Based
  - Lempel - Ziv.

Adaptive Algorithms

- Backward looking:
  - Base the current model on what has already been seen.
  - *The past is a good approximation of the future*
- Forward looking:
  - Exact statistics
  - Uses the model's knowledge of what is still to come.

Differences

- Backward:
  - Increments the frequency
  - “Selfish” behavior
- Forward:
  - “Altruistic” approach
  - Decrements the frequency

Backward Looking Example

- Vitter's dynamic Huffman variant
- NYT - Not Yet Transmitted

Backward

T = BANANAS

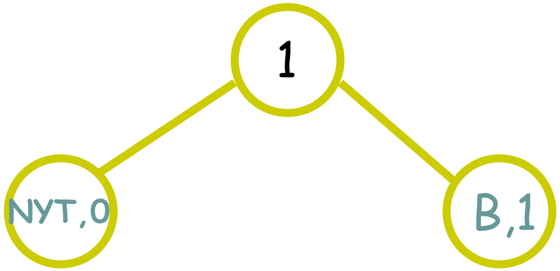
Backward

$T = \text{BANANAS}$

$$\mathcal{E}(T) = \underset{\text{ASCII(B)}}{\overset{\text{ASCII(B)}}{01000010}}$$

Backward

$T = \text{BANANAS}$

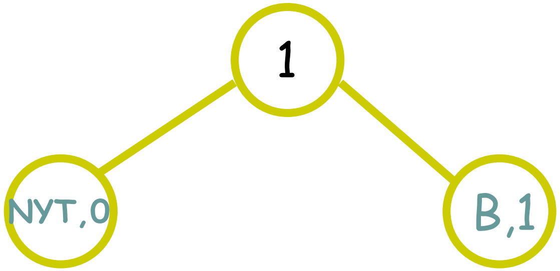


ASCII(B)

$\mathcal{E}(T) = 01000010$

Backward

$T = \text{BANANAS}$

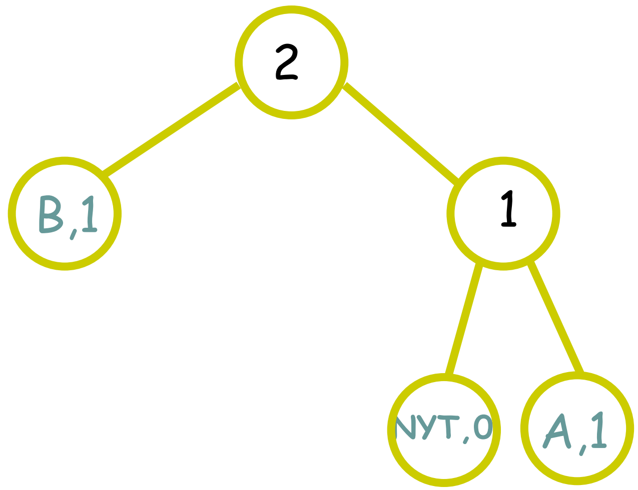


NYT ASCII(A)

$\mathcal{E}(T) = 01000010\ 0\ 01000001$

Backward

$T = \text{BANANAS}$

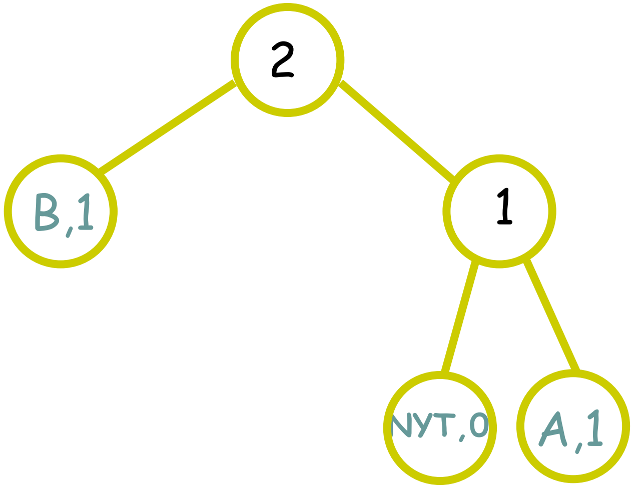


NYT ASCII(A)

$\mathcal{E}(T) = 01000010\ 0\ 01000001$

Backward

$T = \text{BANANAS}$

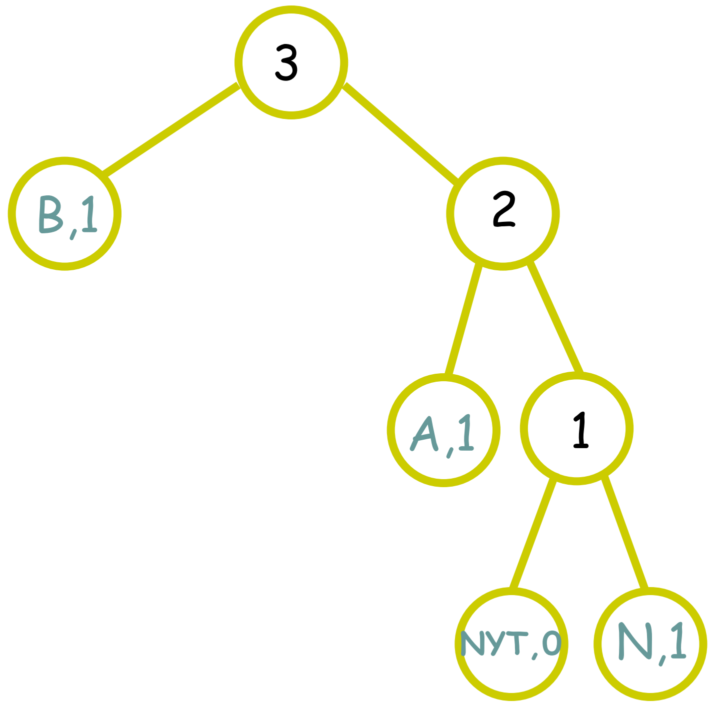


NYT ASCII(N)

$\mathcal{E}(T) = 01000010\ 0\ 01000001\ 10\ 01001110$

Backward

$T = $ BANANAS

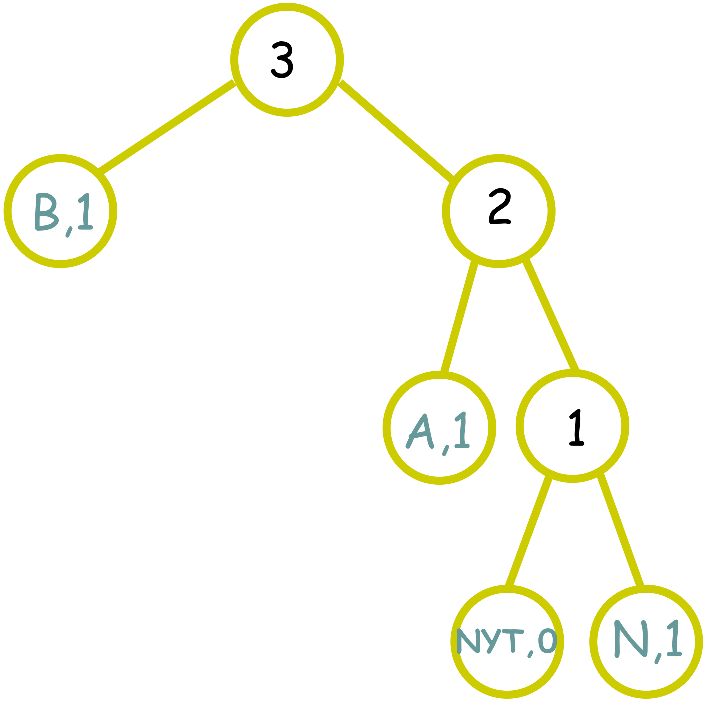


NYT ASCII(N)

$\mathcal{E}(T) = 01000010\ 0\ 01000001\ 10\ 01001110$

Backward

$T = \text{BANANAS}$

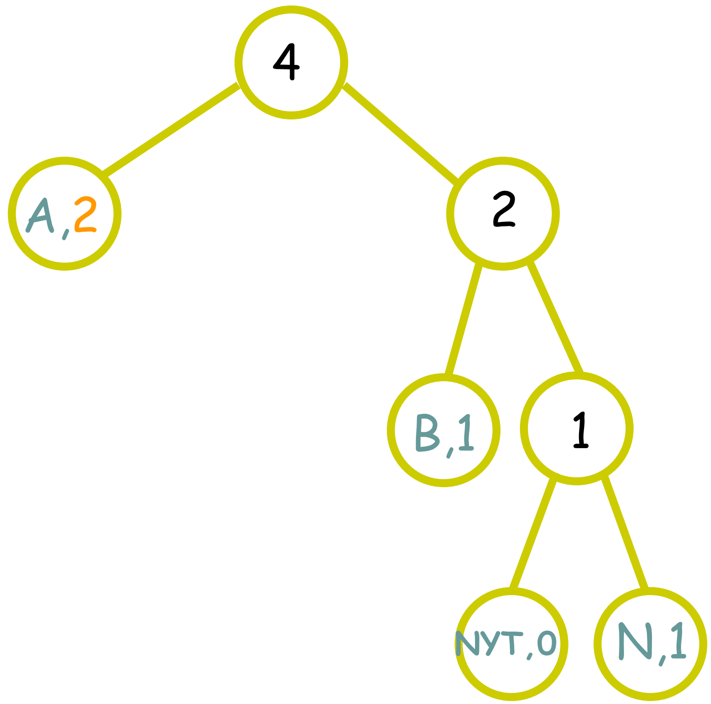


$\mathcal{E}(A)$

$\mathcal{E}(T) = 01000010\ 0\ 01000001\ 10\ 01001110\ 10$

Backward

$T = \text{BANANAS}$

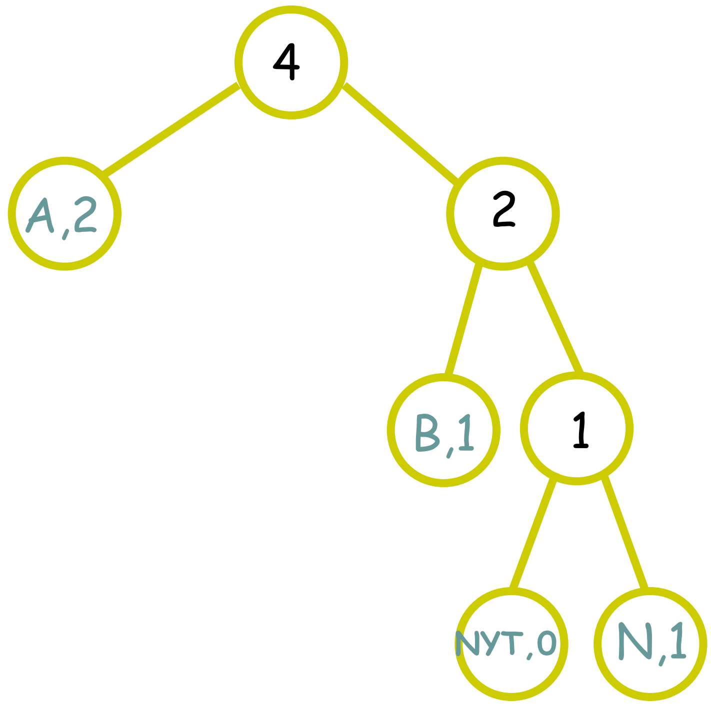


$\mathcal{E}(A)$

$\mathcal{E}(T) = 01000010\ 0\ 01000001\ 10\ 01001110\ 10$

Backward

T = BANANAS

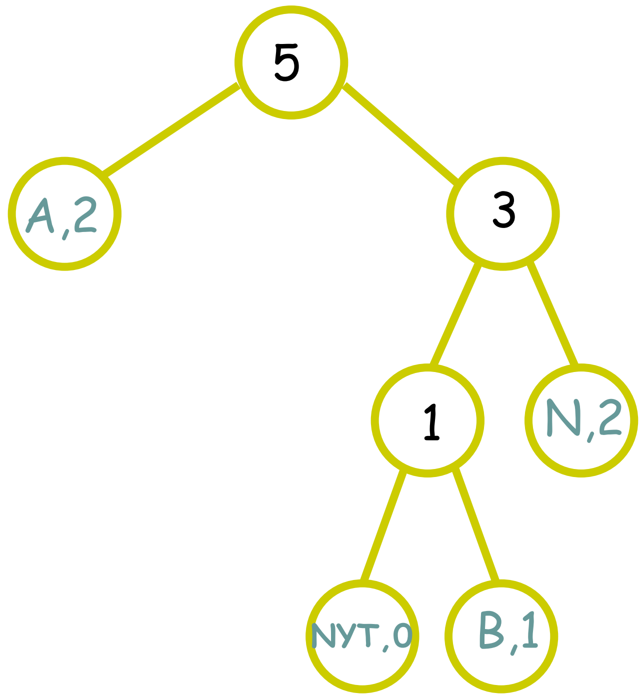


$\mathcal{E}(N)$

$\mathcal{E}(T) = 01000010\ 0\ 01000001\ 10\ 01001110\ 10\ 111$

Backward

$T = $ BANANAS

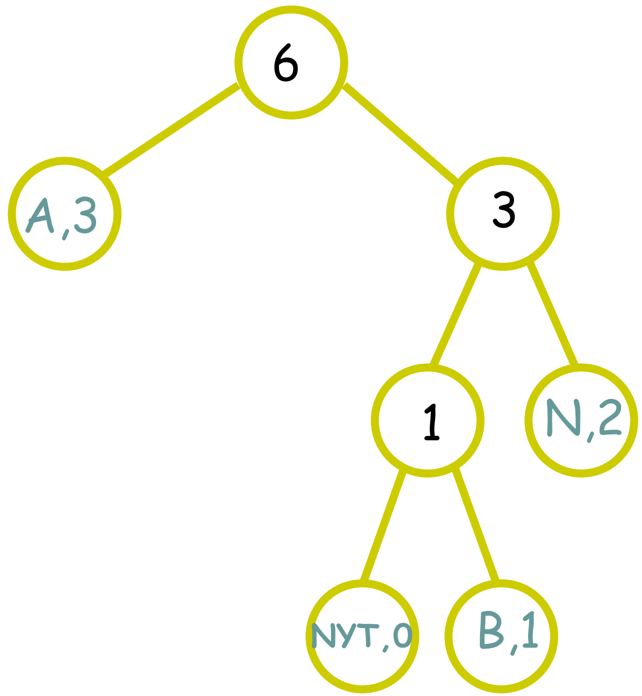


$\mathcal{E}(N)$

$$\mathcal{E}(T) = 01000010\ 0\ 01000001\ 10\ 01001110\ 10\ 111$$

Backward

$T = \text{BANANAS}$

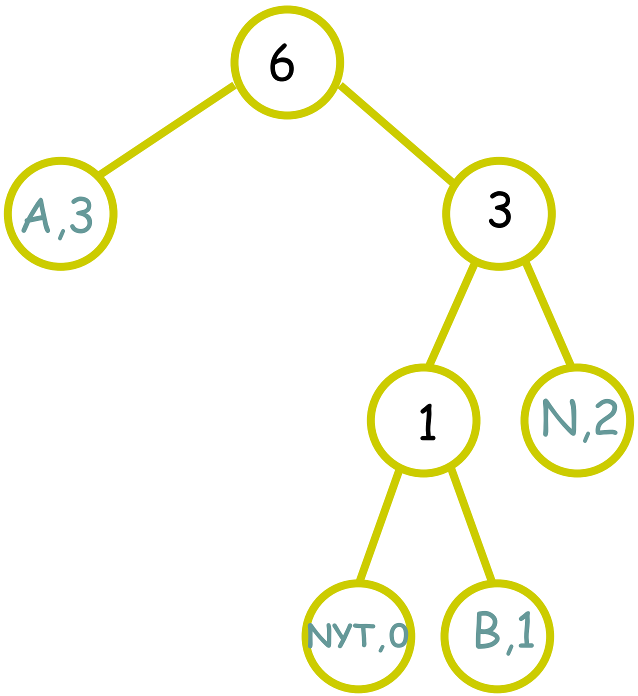


$\mathcal{E}(A)$

$\mathcal{E}(T) = 01000010\ 0\ 01000001\ 10\ 01001110\ 10\ 111\ 0$

Backward

T = BANANAS

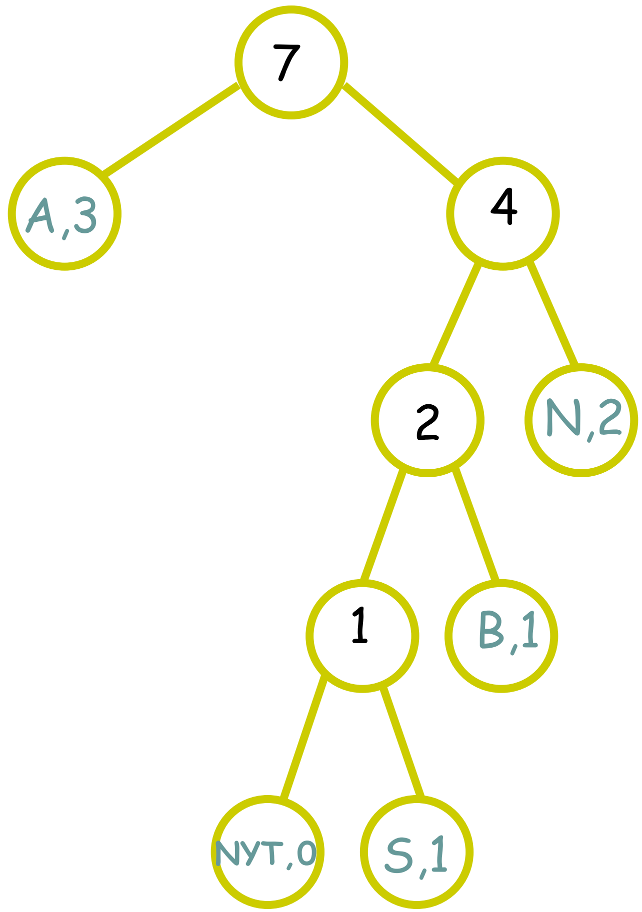


NYT ASCII(S)

$\mathcal{E}(T) = 01000010\ 0\ 01000001\ 10\ 01001110\ 10\ 111\ 0\ 100\ 01010011$

Backward

T = BANANAS



NYT ASCII(S)

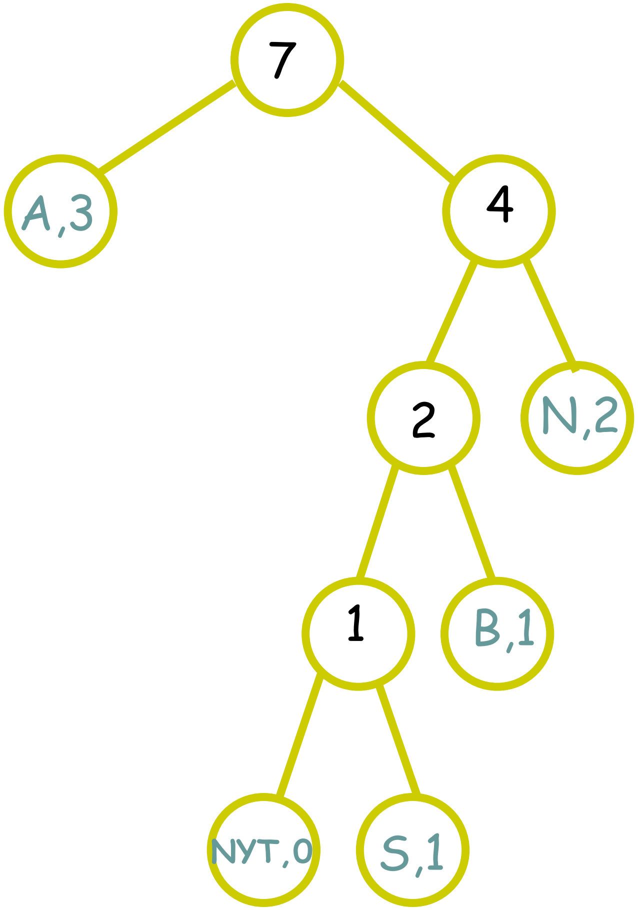
$\mathcal{E}(T) = 01000010\ 0\ 01000001\ 10\ 01001110\ 10\ 111\ 0\ 100\ 01010011$

Forward – previous results

- Best known bound for dynamic is $\leq n$ bits + Static
- For a given distribution of frequencies, the average codeword length of FORWARD is at least as good as the average codeword length of STATIC
- Classic might produce a file twice the size of Forward

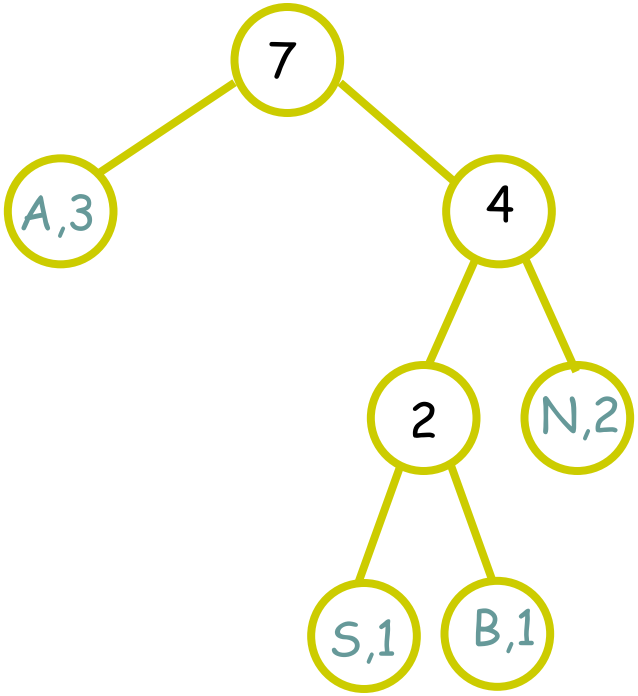
Backward

T = BANANAS



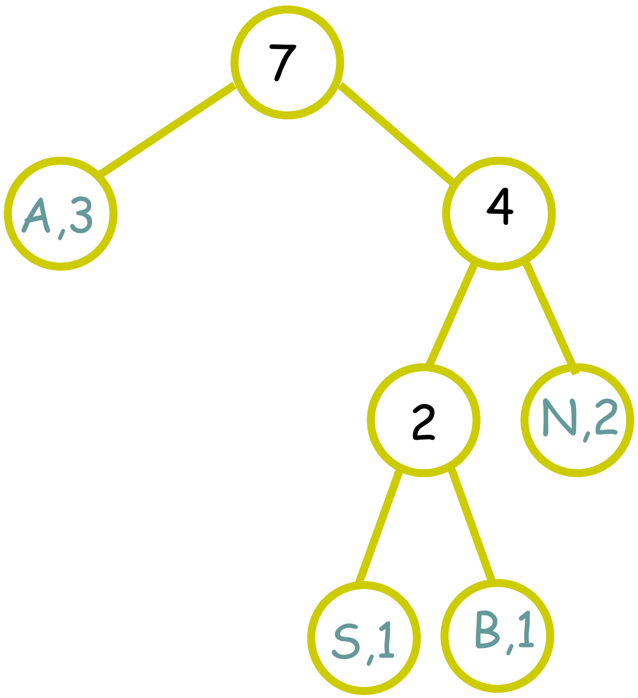
Forward

T = BANANAS



Forward:

$T =$ BANANAS

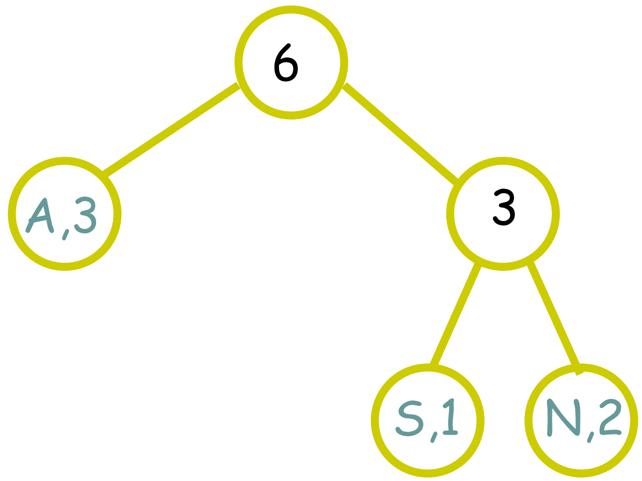


$\mathcal{E}(T) = 101$

$\mathcal{E}(B)$

Forward:

$T = \text{BANANAS}$

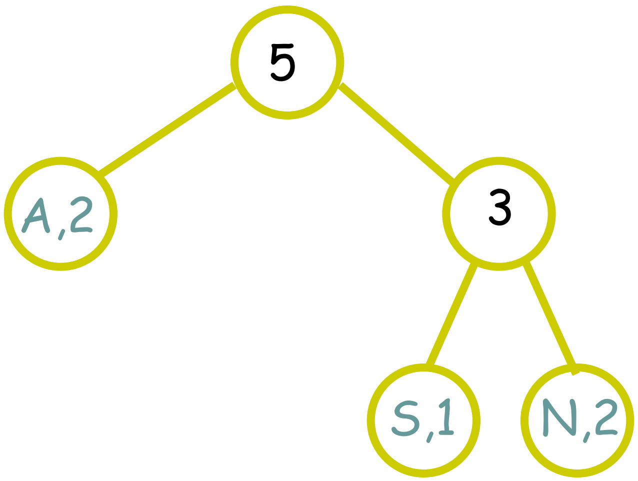


$\mathcal{E}(T) = 101\ 0$

$\mathcal{E}(A)$

Forward:

T = BANANAS

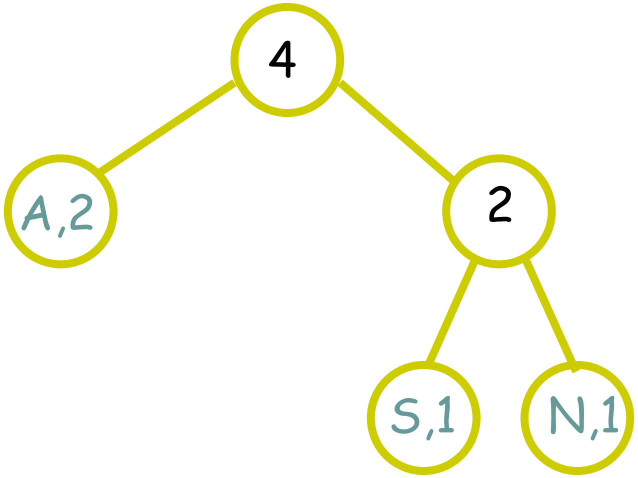


$\mathcal{E}(T) = 101\ 0\ 11$

$\mathcal{E}(N)$

Forward:

$T = \text{BANANAS}$

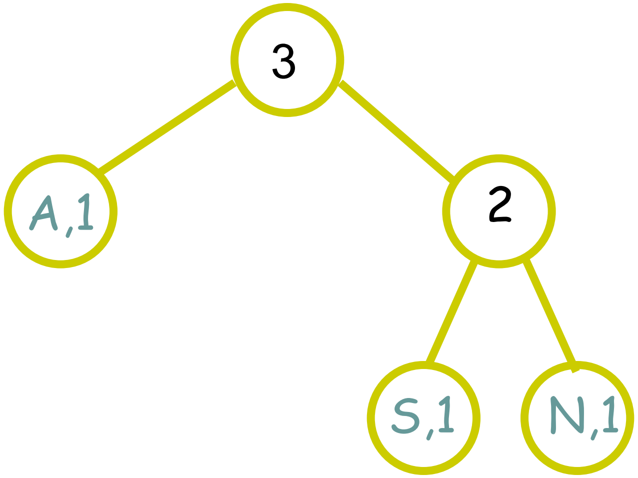


$\mathcal{E}(T) = 101\ 0\ 11\ 0$

$\mathcal{E}(A)$

Forward:

$T = \text{BANANAS}$

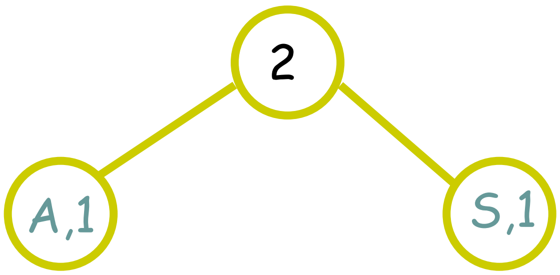


$\mathcal{E}(T) = 101\ 0\ 11\ 0\ 11$

$\mathcal{E}(N)$

Forward:

$T =$ BANANAS



$\mathcal{E}(T) = 101\ 0\ 11\ 0\ 11\ 0$

$\mathcal{E}(A)$

Forward:

S,1

$T =$ BANANAS

$\mathcal{E}(T) = 101\ 0\ 11\ 0\ 11\ 0$

No Need to transmit S

Drawbacks of current methods

- Backward and Forward – use information about the distribution which isn't necessarily needed.
  - Static – frequencies of the characters in the entire text.
  - String of characters {a, b, … , z} followed by numbers {0,…,9}.

A new Hybrid coding

- NYT - Not Yet Transmitted
- Encoding the model
  - Forward - exact frequencies at the beginning of the process
  - Backward - incrementally

} Hybrid

Hybrid - NYT+ASCII+freq

Hybrid

$T = \text{BANANAS}$

$\mathcal{E}(T) = 01000010\ 1$

ASCII(B) $c_\delta(1)$

Hybrid

NYT, 3

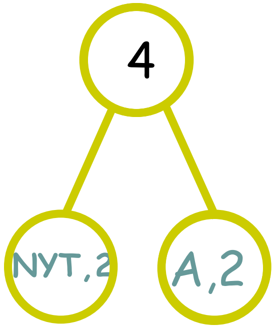
$T = \text{BANANAS}$

$\mathcal{E}(T) = 01000010\ 1\ 01000001\ 0101$

ASCII(A) $c_\delta(3)$

Hybrid

T = BANANAS

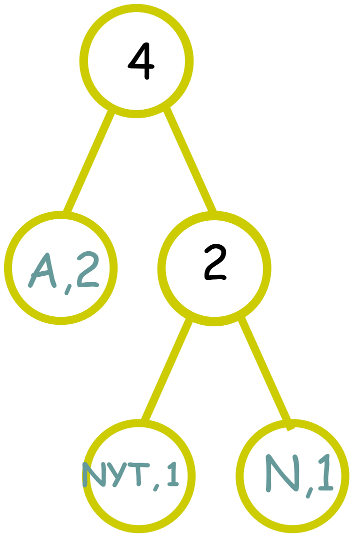


$\mathcal{E}(T)$ = 01000010 1 01000001 0101

ASCII(A) $c_\delta(3)$

Hybrid

T = BANANAS

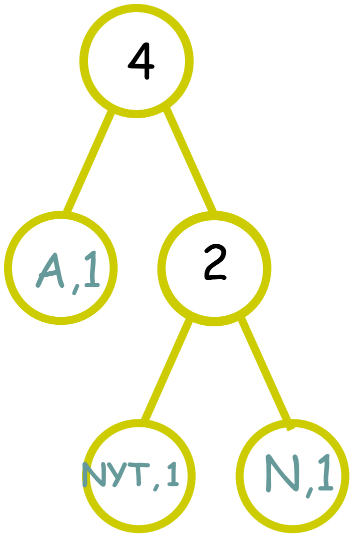


$\mathcal{E}(T) =$ 01000010 1 01000001 0101 0 01001110 0100

NYT ASCII(N) $c_\delta(2)$

Hybrid

$T = $ BANANAS

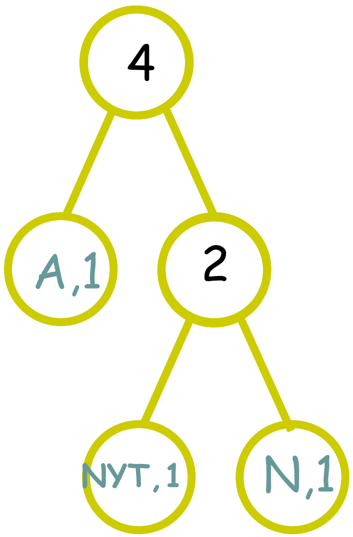


$\mathcal{E}(T) = 01000010\ 1\ 01000001\ 0101\ 0\ 01001110\ 0100\ 0$

$\mathcal{E}(A)$

Hybrid

$T = \text{BANANAS}$

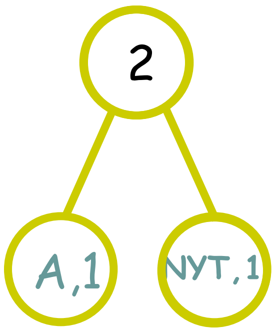


$\mathcal{E}(T) = 01000010\ 1\ 01000001\ 0101\ 0\ 01001110\ 0100\ 0\ 11$

$\mathcal{E}(N)$

Hybrid

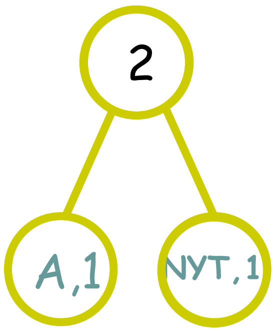
$T =$ BANANAS



$\mathcal{E}(T) = 01000010\ 1\ 01000001\ 0101\ 0\ 01001110\ 0100\ 0\ 11\ 0$

Hybrid

$T = \text{BANANAS}$



$\mathcal{E}(T) = 01000010\ 1\ 01000001\ 0101\ 0\ 01001110\ 0100\ 0\ 11\ 0$

$\mathcal{E}(A)$

Hybrid

T = BANANAS

$\mathcal{E}(T) =$ 01000010 1 01000001 0101 0 01001110 0100 0 11 0 01010011 1

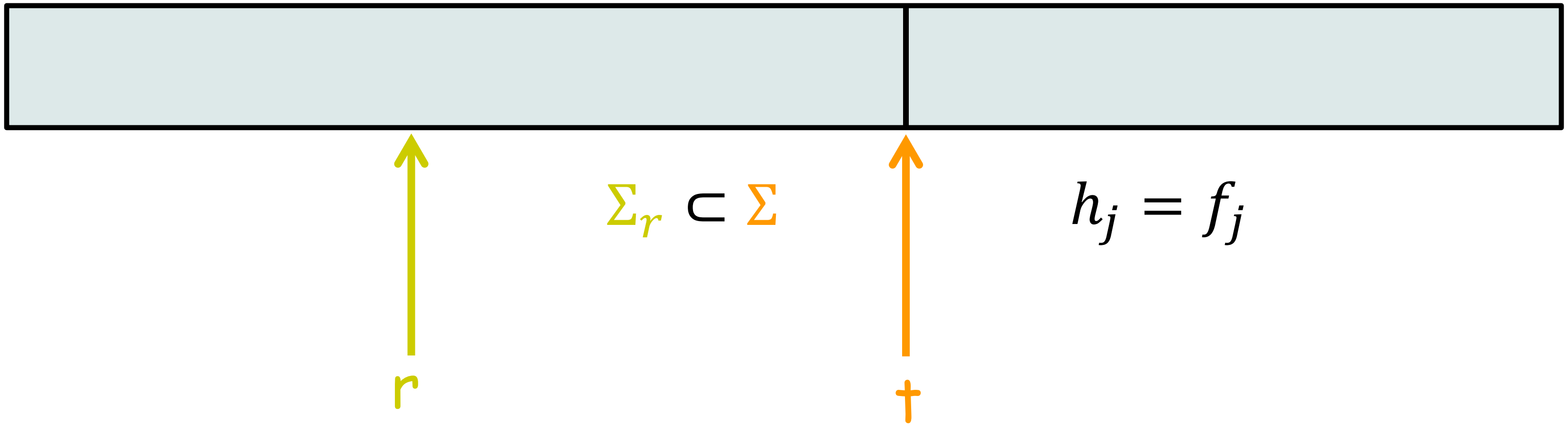
ASCII(S) $c_\delta(1)$

Generic Hybrid-ENCODE($T = x_1 \cdots x_n$)

```
Preprocess T to get freq(σi), ∀σi∈ Σ
Initialize the model with NYT with freq(NYT)← |Σ|
Encode freq(NYT)
for i←1 to n do
   if xi has already appeared earlier then
       encode xi according to current model
       freq(xi) ← freq(xi)-1
   else
       encode NYT according to current model
       freq(NYT) ← freq(NYT)-1
       output ASCII(xi)
       encode freq(xi)
       Update the model with xi, freq(xi) and freq(NYT)
```

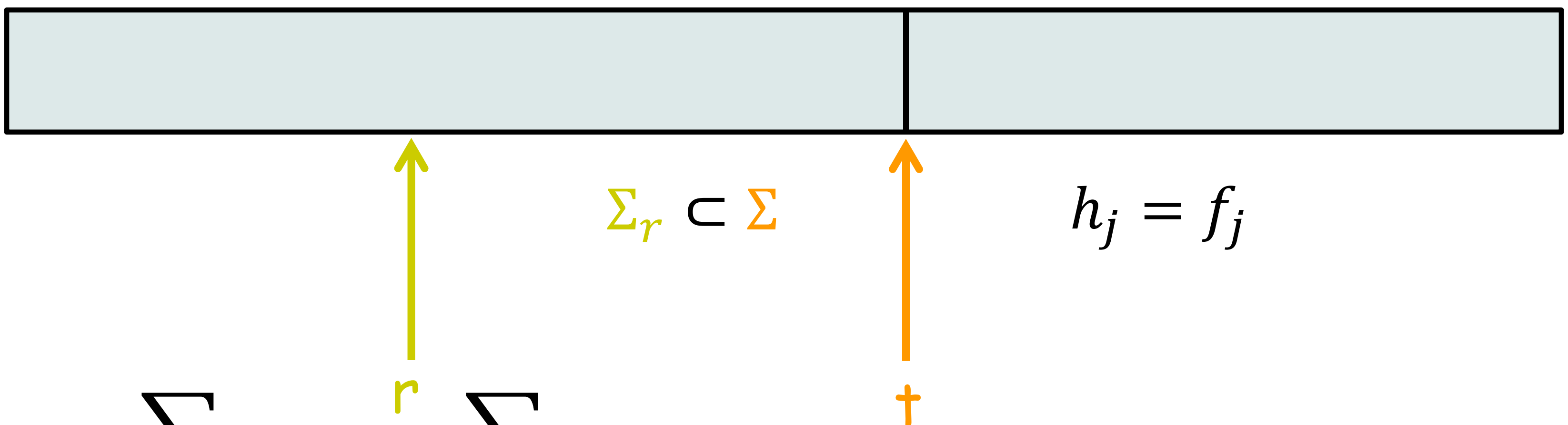
Theorem

- The expected performance of HYBRID is at least as good as FORWARD



Theorem

- The expected performance of HYBRID is at least as good as FORWARD



$$\sum_{\sigma \epsilon \Sigma_r} p_\sigma h_\sigma \leq \sum_{\sigma \epsilon \Sigma_r} p_\sigma f_\sigma$$

Huffman code built for $P = \{p_\sigma | \sigma \epsilon \Sigma_r\}$

Remarks

- Not necessarily true that $h_j \le f_j$ for $j < t$
- Moderate expected savings by using HYBRID instead of FORWARD
  - J and Q appear with probability 0.002

Main contribution: improve a method which already seems better than one considered “optimal”

Empirical Results

File	*Full Size*	*Size of Encoded File*			
		Static	*Adaptive*	*Forward*	*Hybrid*
ebib	3,711,020	1,940,573	1,941,321	1,940,527	**1,940,268**
exe	48,640	31,296	31,851	31,132	**28,930**
ftxt	7,648,930	4,443,525	4,444,660	4,443,419	**4,442,447**
eng	52,428,800	29,914,197	29,915,562	29,914,021	**29,912,644**
dig − ch	3,726,683	1,969,884	1,970,694	1,969,830	**1,945,310**


THANK
YOU


pyright 1996 Randy Glasbergen. www.glasbergen.com
GLASBERGEN
“Never touch the screen while
you’re compressing a file!”