Codificação Aritmética

- Apresenta taxas de compressão melhores que o código de Huffman;
- No código de Huffman um símbolo cuja probabilidade, $p(s_i)$, esteja próximo de 1, ou seja, $\log_2 \frac{1}{p(s_i)}$ está próximo de zero, atribuir um bit é muito penalizador.

  (notar que no mínimo, o código de Huffman, atribui um bit ao símbolo mais provável !)
- A codificação aritmética trata a mensagem como uma unidade

- Algoritmo

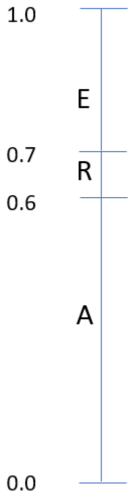
```
low = 0.0
high = 1.0
range = high - low
While (not end)
    s =read symbol
    high = low + range * highrange(s)
    low = low + range * lowrange(s)
    range = high – low
end
write code;       #  low <= code < high
```

Codificação Aritmética décimal

- Mensagem: “AERA”

```
Low = 0.0
High = 1.0
Range  = High - Low
While (not end)
    s = next symbol
    High = Low + Range x High(s)
    Low = Low + Range x Low(s)
    Range  = High - Low
end
output code
```

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

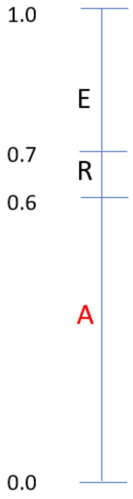


Codificação Aritmética décimal

- Mensagem: “AERA”

Low = 0.0				
High = 1.0				
Range = High - Low				
While (not end)				
s = next symbol	A			
High = Low + Range x High(s)				
Low = Low + Range x Low(s)				
Range = High - Low				
end				
output code				

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

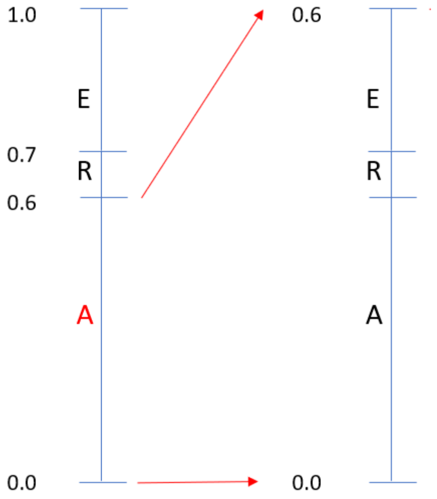


Codificação Aritmética décimal

- Mensagem: "AERA"

Low = 0.0				
High = 1.0				
Range = High - Low				
While (not end)				
s = next symbol	A			
High = Low + Range x High(s)	0.6			
Low = Low + Range x Low(s)	0.0			
Range = High - Low	0.6			
end				
output code				

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

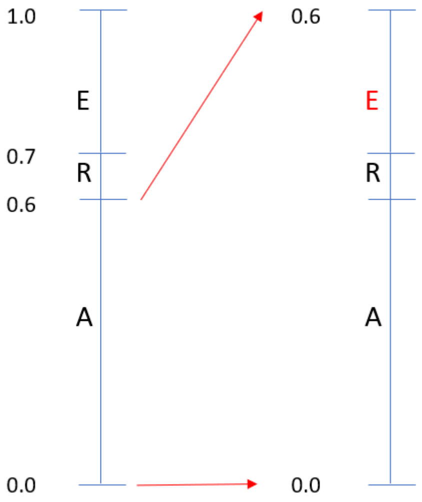


Codificação Aritmética décimal

- Mensagem: “AERA”

Low = 0.0				
High = 1.0				
Range = High - Low				
While (not end)				
s = next symbol	A	E		
High = Low + Range x High(s)	0.6			
Low = Low + Range x Low(s)	0.0			
Range = High - Low	0.6			
end				
output code				

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

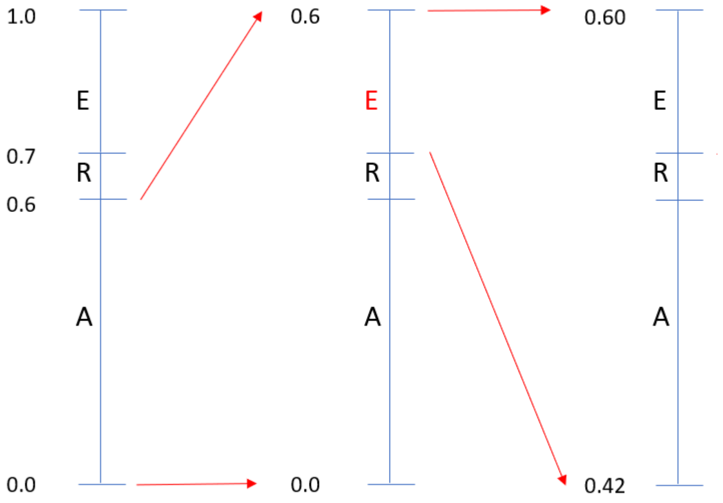


Codificação Aritmética décimal

- Mensagem: “AERA”

Low = 0.0				
High = 1.0				
Range = High - Low				
While (not end)				
s = next symbol	A	E		
High = Low + Range x High(s)	0.6	0.60		
Low = Low + Range x Low(s)	0.0	0.42		
Range = High - Low	0.6	0.18		
end				
output code				

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

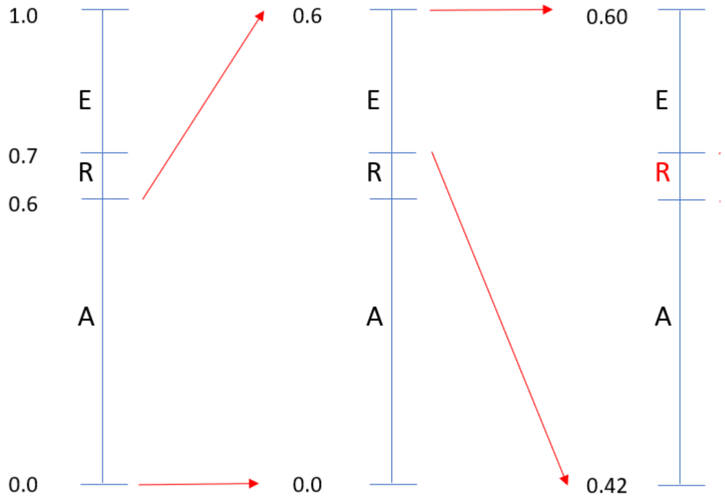


Codificação Aritmética décimal

- Mensagem: “AERA”

Low = 0.0				
High = 1.0				
Range = High - Low				
While (not end)				
s = next symbol	A	E	R	
High = Low + Range x High(s)	0.6	0.60		
Low = Low + Range x Low(s)	0.0	0.42		
Range = High - Low	0.6	0.18		
end				
output code				

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

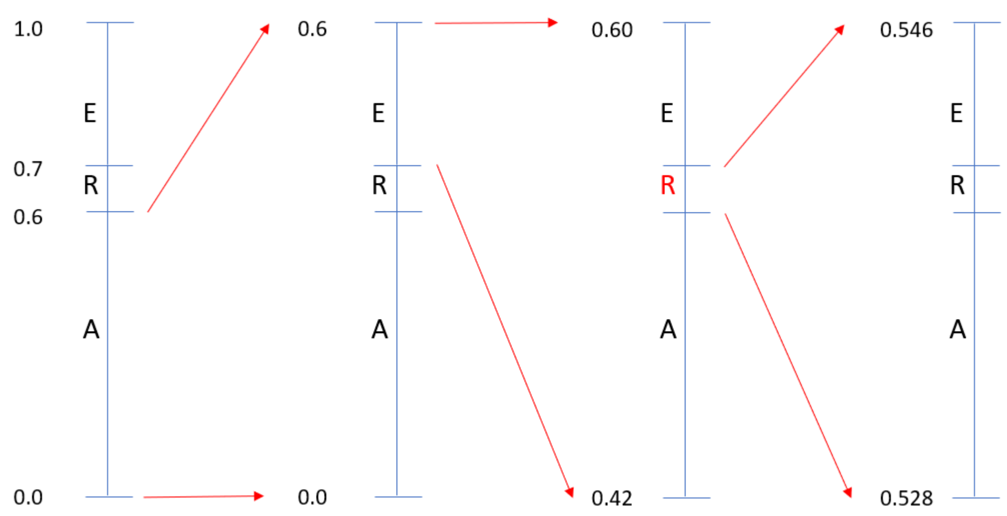


Codificação Aritmética décimal

- Mensagem: “AERA”

Low = 0.0				
High = 1.0				
Range = High - Low				
While (not end)				
s = next symbol	A	E	R	
High = Low + Range x High(s)	0.6	0.60	0.5460	
Low = Low + Range x Low(s)	0.0	0.42	0.5280	
Range = High - Low	0.6	0.18	0.0018	
end				
output code				

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

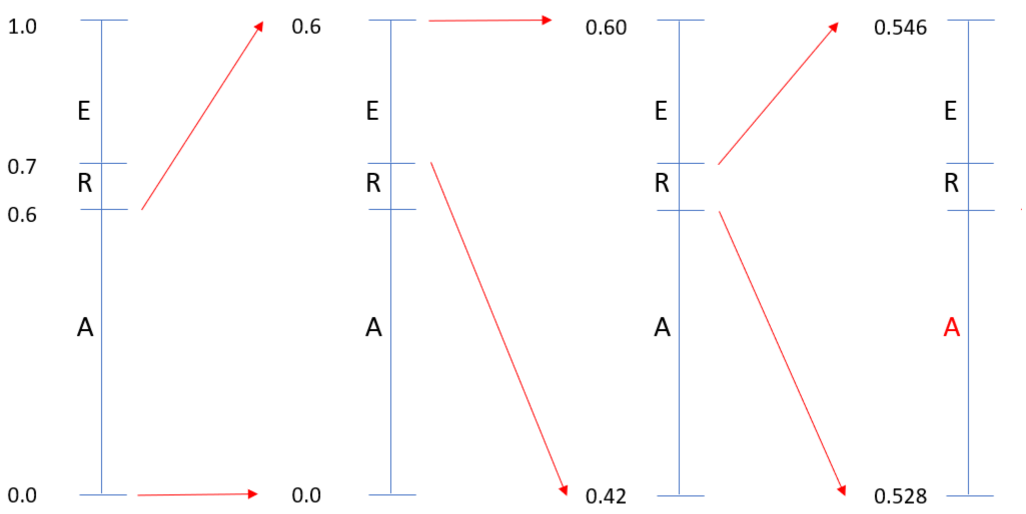


Codificação Aritmética décimal

- Mensagem: “AERA”

Low = 0.0				
High = 1.0				
Range = High - Low				
While (not end)				
s = next symbol	A	E	R	A
High = Low + Range x High(s)	0.6	0.60	0.5460	
Low = Low + Range x Low(s)	0.0	0.42	0.5280	
Range = High - Low	0.6	0.18	0.0018	
end				
output code				

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

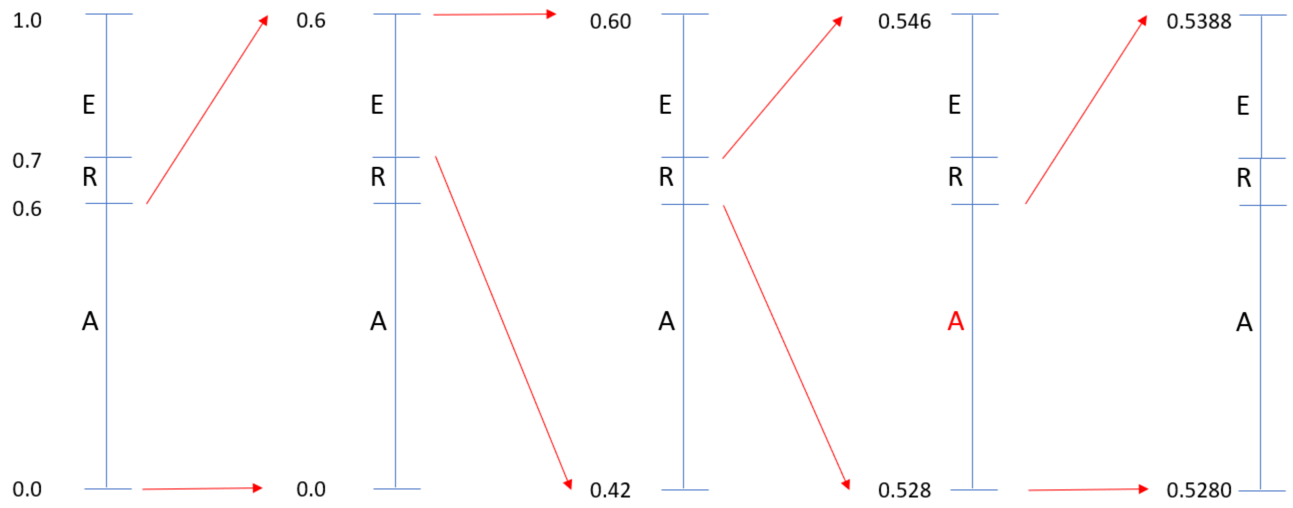


Codificação Aritmética décimal

- Mensagem: “AERA”

Low = 0.0				
High = 1.0				
Range = High - Low				
While (not end)				
s = next symbol	A	E	R	A
High = Low + Range x High(s)	0.6	0.60	0.5460	0.5388
Low = Low + Range x Low(s)	0.0	0.42	0.5280	0.5280
Range = High - Low	0.6	0.18	0.0018	0.0108
end				
output code				

Símbolo	prob.	Interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00



Codificação Aritmética décimal

- Valor entre 0.5280 e 0.5388

$$\text{número décimal} = \ldots + c_2 2^2 + c_1 2^1 + c_0 2^0 + c_{-1} 2^{-1} + c_{-2} 2^{-2} + \ldots$$

$$= \ldots + c_2 \times 4 + c_1 \times 2 + c_0 + \underbrace{c_{-1} \times 0.5 + c_{-2} \times 0.25 + \ldots}$$

Codificação Aritmética décimal

- Valor entre 0.5280 e 0.5388

$$\begin{aligned}\text{número décimal} &= \ldots + c_2 2^2 + c_1 2^1 + c_0 2^0 + c_{-1} 2^{-1} + c_{-2} 2^{-2} + \ldots \\ &= \ldots + c_2 \times 4 + c_1 \times 2 + c_0 + \underbrace{c_{-1} \times 0.5 + c_{-2} \times 0.25 + \ldots}_{}\end{aligned}$$

- Dado que o número está entre 0 e 1 só se envia a parte décimal

Codificação Aritmética décimal

- Valor entre 0.5280 e 0.5388

$$\text{número décimal} = \ldots + c_2 2^2 + c_1 2^1 + c_0 2^0 + c_{-1} 2^{-1} + c_{-2} 2^{-2} + \ldots$$

$$= \ldots + c_2 \times 4 + c_1 \times 2 + c_0 + \underbrace{c_{-1} \times 0.5 + c_{-2} \times 0.25 + \ldots}_{}$$

- Dado que o número está entre 0 e 1 só se envia a parte décimal

- 1 = 0.5

Codificação Aritmética décimal

- Valor entre 0.5280 e 0.5388

$$\text{número décimal} = \ldots + c_2 2^2 + c_1 2^1 + c_0 2^0 + c_{-1} 2^{-1} + c_{-2} 2^{-2} + \ldots$$

$$= \ldots + c_2 \times 4 + c_1 \times 2 + c_0 + \underbrace{c_{-1} \times 0.5 + c_{-2} \times 0.25 + \ldots}$$

- Dado que o número está entre 0 e 1 só se envia a parte décimal

- 1 = 0.5
- 1 1 = 0.5 + 0.25 = 0.75

Codificação Aritmética décimal

- Valor entre 0.5280 e 0.5388

$$\text{número décimal} = \ldots + c_2 2^2 + c_1 2^1 + c_0 2^0 + c_{-1} 2^{-1} + c_{-2} 2^{-2} + \ldots$$

$$= \ldots + c_2 \times 4 + c_1 \times 2 + c_0 + \underbrace{c_{-1} \times 0.5 + c_{-2} \times 0.25 + \ldots}_{}$$

- Dado que o número está entre 0 e 1 só se envia a parte décimal

- 1 = 0.5
- 1 1 = 0.5 + 0.25 = 0.75
- 1 0 1 = 0.5 + 0.125 = 0.625

Codificação Aritmética décimal

- Valor entre 0.5280 e 0.5388

$$\text{número décimal} = \ldots + c_2 2^2 + c_1 2^1 + c_0 2^0 + c_{-1} 2^{-1} + c_{-2} 2^{-2} + \ldots$$

$$= \ldots + c_2 \times 4 + c_1 \times 2 + c_0 + \underbrace{c_{-1} \times 0.5 + c_{-2} \times 0.25 + \ldots}$$

- Dado que o número está entre 0 e 1 só se envia a parte décimal

- 1 = 0.5
- 1 1 = 0.5 + 0.25 = 0.75
- 1 0 1 = 0.5 + 0.125 = 0.625
- 1 0 0 1 = 0.5 + 0.0625 = 0.5625

Codificação Aritmética décimal

- Valor entre 0.5280 e 0.5388

$$\text{número décimal} = \ldots + c_2 2^2 + c_1 2^1 + c_0 2^0 + c_{-1} 2^{-1} + c_{-2} 2^{-2} + \ldots$$

$$= \ldots + c_2 \times 4 + c_1 \times 2 + c_0 + \underbrace{c_{-1} \times 0.5 + c_{-2} \times 0.25 + \ldots}$$

- Dado que o número está entre 0 e 1 só se envia a parte décimal

- 1 = 0.5
- 1 1 = 0.5 + 0.25 = 0.75
- 1 0 1 = 0.5 + 0.125 = 0.625
- 1 0 0 1 = 0.5 + 0.0625 = 0.5625
- 1 0 0 0 1 = 0.5 + 0.03125 = 0.53125

Codificação Aritmética décimal

- Valor entre 0.5280 e 0.5388

$$\text{número décimal} = \ldots + c_2 2^2 + c_1 2^1 + c_0 2^0 + c_{-1} 2^{-1} + c_{-2} 2^{-2} + \ldots$$

$$= \ldots + c_2 \times 4 + c_1 \times 2 + c_0 + \underbrace{c_{-1} \times 0.5 + c_{-2} \times 0.25 + \ldots}_{}$$

- Dado que o número está entre 0 e 1 só se envia a parte décimal

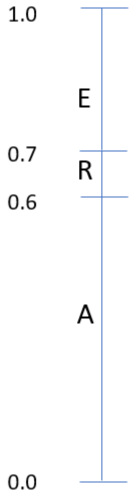
- 1 = 0.5
- 1 1 = 0.5 + 0.25 = 0.75
- 1 0 1 = 0.5 + 0.125 = 0.625
- 1 0 0 1 = 0.5 + 0.0625 = 0.5625
- 1 0 0 0 1 = 0.5 + 0.03125 = 0.53125

- São usados 5 bits para codificar a mensagem.

Descodificação Aritmética décimal

```
read input value
While (not end)
    find symbol s such that:   Low(s) < value < High(s)
    output symbol s
     High = High(s)
     Low = Low(s)
     Range  = High - Low
     value = ( value - Low ) / Range
end
```

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

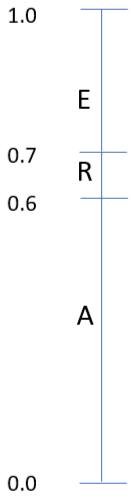


Descodificação Aritmética décimal

read input value	0.53125			
While (not end)				
find symbol s such that: Low(s) < value < High(s)				
output symbol s				
High = High(s)				
Low = Low(s)				
Range = High - Low				
value = (value - Low) / Range				
end				

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

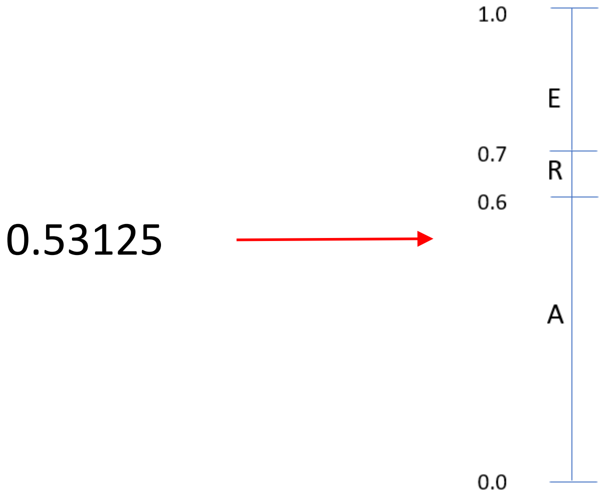
0.53125



Descodificação Aritmética décimal

read input value				
While (not end)				
find symbol s such that: Low(s) < value < High(s)				
output symbol s	A			
High = High(s)	0.6			
Low = Low(s)	0.0			
Range = High - Low	0.6			
value = (value - Low) / Range	~ 0.885			
end				

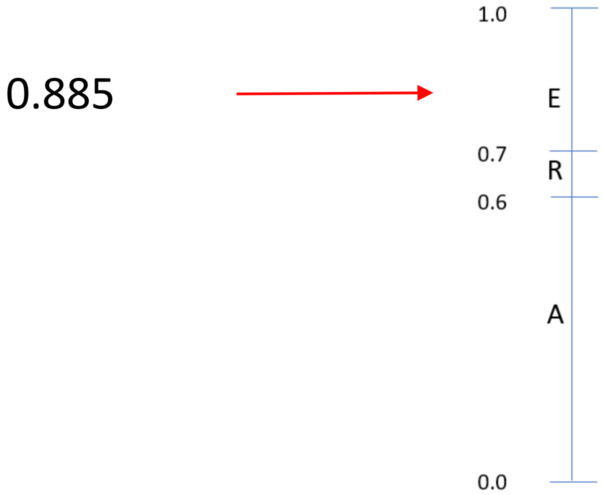
Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00



Descodificação Aritmética décimal

read input value				
While (not end)				
find symbol s such that: Low(s) < value < High(s)				
output symbol s	A	E		
High = High(s)	0.6	1.0		
Low = Low(s)	0.0	0.7		
Range = High - Low	0.6	0.3		
value = (value - Low) / Range	~ 0.885	~ 0.618		
end				

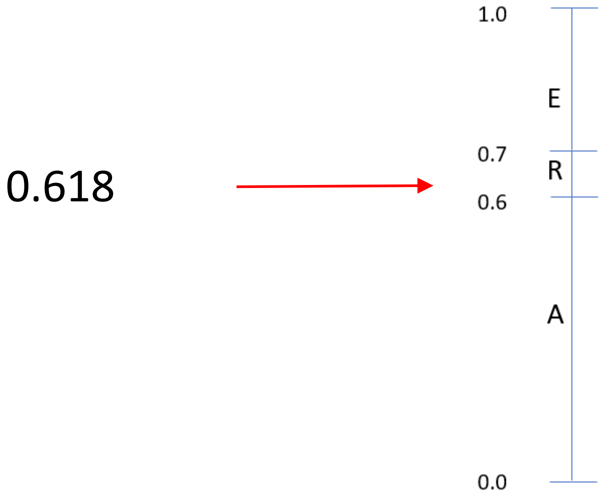
Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00



Descodificação Aritmética décimal

read input value				
While (not end)				
find symbol s such that: Low(s) < value < High(s)				
output symbol s	A	E	R	
High = High(s)	0.6	1.0	0.7	
Low = Low(s)	0.0	0.7	0.6	
Range = High - Low	0.6	0.3	0.1	
value = (value - Low) / Range	~ 0.885	~ 0.618	~ 0.180	
end				

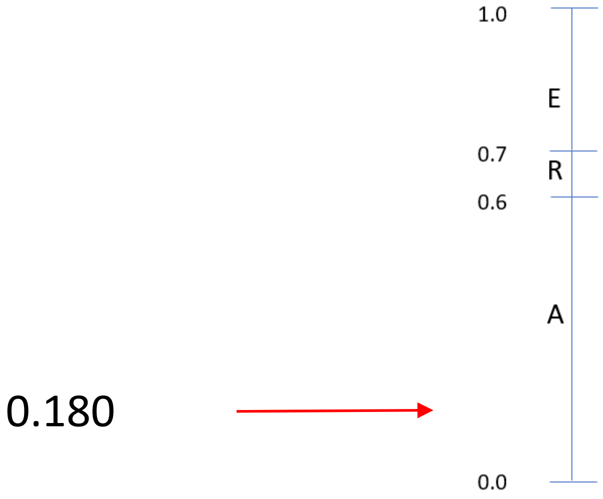
Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00



Descodificação Aritmética décimal

read input value				
While (not end)				
find symbol s such that: Low(s) < value < High(s)				
output symbol s	A	E	R	A
High = High(s)	0.6	1.0	0.7	0.6
Low = Low(s)	0.0	0.7	0.6	0.0
Range = High - Low	0.6	0.3	0.1	0.3
value = (value - Low) / Range	~ 0.885	~ 0.618	~ 0.180	~ 0.300
end				

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00



Codificação Aritmética

- A compressão é melhor que o código de Huffman;
- No pior caso, o código mais pequeno é no máximo

$$L \leq log_2 \frac{1}{range} = log_2 \frac{1}{\prod_i p_i}$$

  - Notar que o código de Huffman pode ultrapassar este limite.

- Desvantagens:
  - Precisa de saber à priori as probabilidades de cada símbolo;
  - Necessidade de trabalhar com números com muita precisão;
  - O tempo de compressão/descompressão pode ser elevado, devido à complexidade do cálculo.

- É usado na codificação JPEG.

Codificação Aritmética

- Há implementações práticas que escalam os intervalos para trabalhar com inteiros.

- Notar que o valor encontrado no exemplo “1 0 0 0 1 = 0.53125”

  o primeiro 1 informa que o valor está no intervalo [0, 0.5]

  o primeiro 0 informa que o valor está no intervalo [0.5, 0.75]

  etç…

Codificação Aritmética com Escalamento

- O algoritmo é o mesmo mas tem um loop tem duas condições antes de ler o próximo símbolo:

  - Condição E1:
    Se o intervalo [low high] pertence a [0, 0.5]
    - Escalar o intervalo [0, 0.5] para [0, 1.0]
    - Envia o bit 0

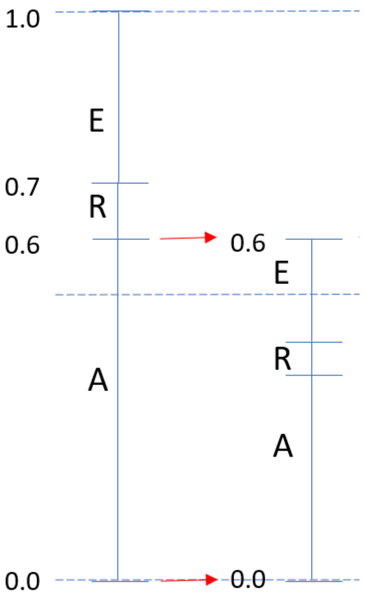
  - Condição E2:
    Se o intervalo [low high] pertence a [0.5, 1.0]
    - Escalar o intervalo [0.5, 1.0] para [0, 1.0]
    - Envia o bit 1

Codificação Aritmética com Escalamento

Símbolo	prob.	Interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

- Mensagem: “AERA”

s = next symbol	A								
High(s)	0.6								
Low(s)	0.0								
High = Low + Range x High(s)									
Low = Low + Range x Low(s)									
Range = High - Low									
output									

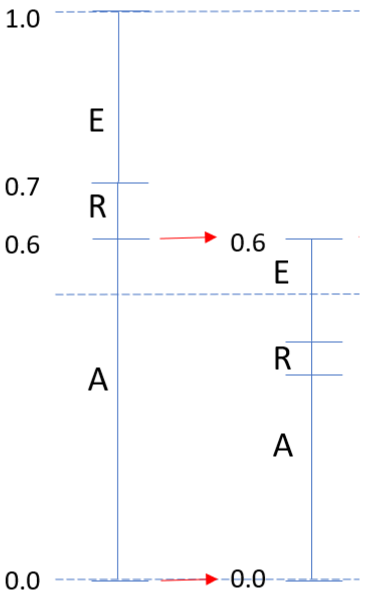


Codificação Aritmética com Escalamento

Símbolo	prob.	Interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

- Mensagem: “AERA”

s = next symbol	A								
High(s)	0.6								
Low(s)	0.0								
High = Low + Range x High(s)	0.6								
Low = Low + Range x Low(s)	0.0								
Range = High - Low	0.6								
output									

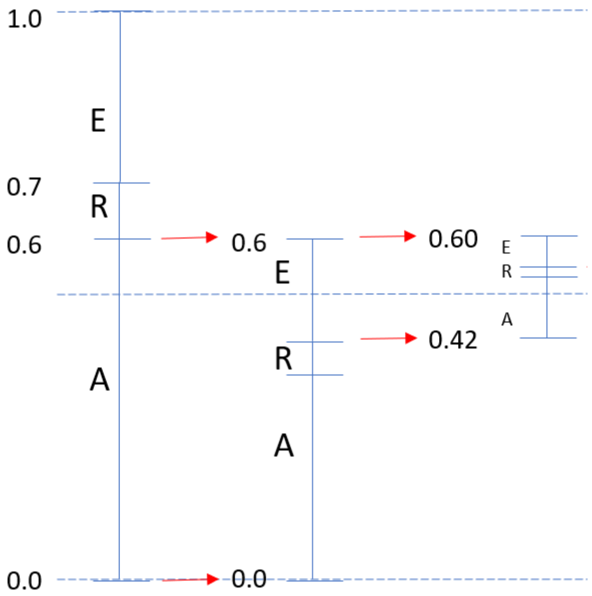


Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

Codificação Aritmética com Escalamento

- Mensagem: “AERA”

s = next symbol	A	E							
High(s)	0.6	1.0							
Low(s)	0.0	0.7							
High = Low + Range x High(s)	0.6	0.60							
Low = Low + Range x Low(s)	0.0	0.42							
Range = High - Low	0.6	0.18							
output									

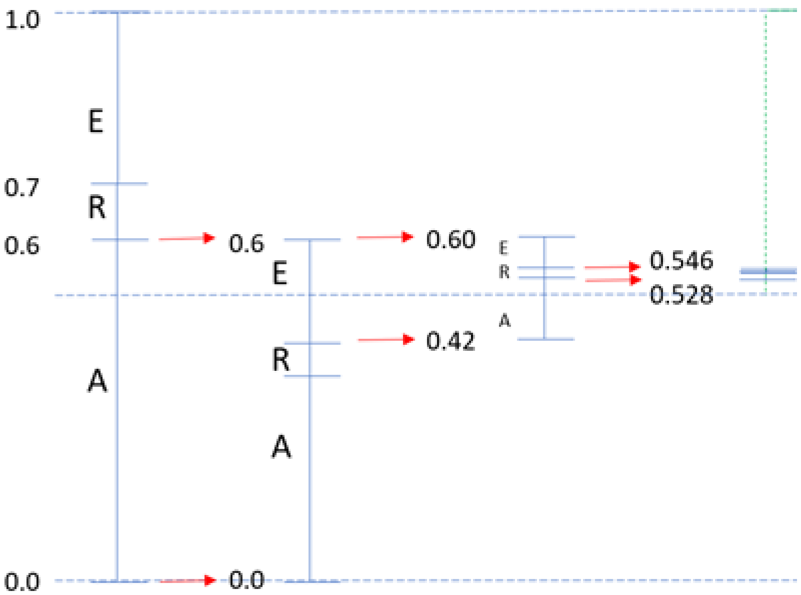


Codificação Aritmética com Escalamento

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

- Mensagem: “AERA”

s = next symbol	A	E	R						
High(s)	0.6	1.0	0.7						
Low(s)	0.0	0.7	0.6						
High = Low + Range x High(s)	0.6	0.60	0.546						
Low = Low + Range x Low(s)	0.0	0.42	0.528						
Range = High - Low	0.6	0.18	0.018						
output									

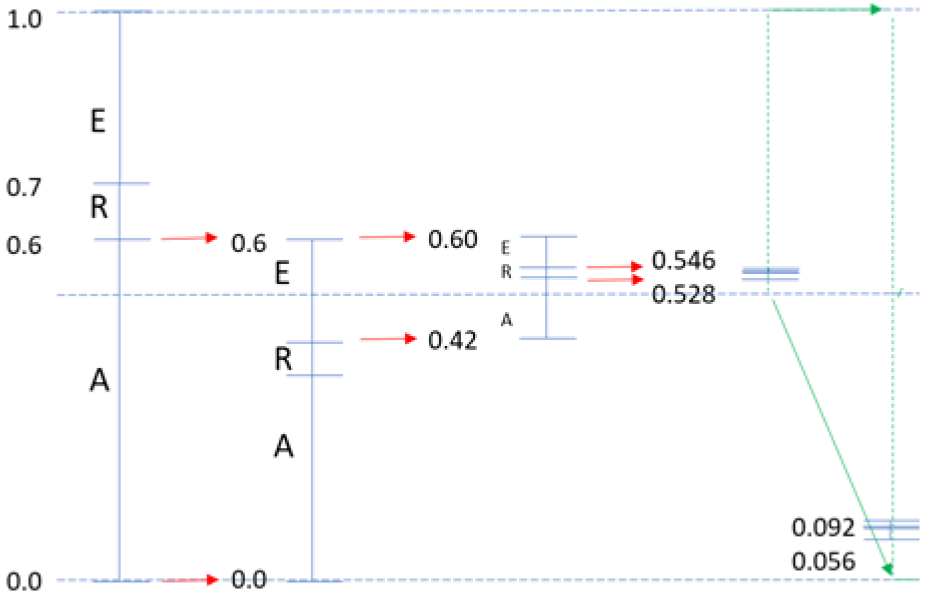


Codificação Aritmética com Escalamento

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

- Mensagem: “AERA”

s = next symbol	A	E	R						
High(s)	0.6	1.0	0.7						
Low(s)	0.0	0.7	0.6						
High = Low + Range x High(s)	0.6	0.60	0.546	0.092					
Low = Low + Range x Low(s)	0.0	0.42	0.528	0.056					
Range = High - Low	0.6	0.18	0.018	0.036					
output				1					

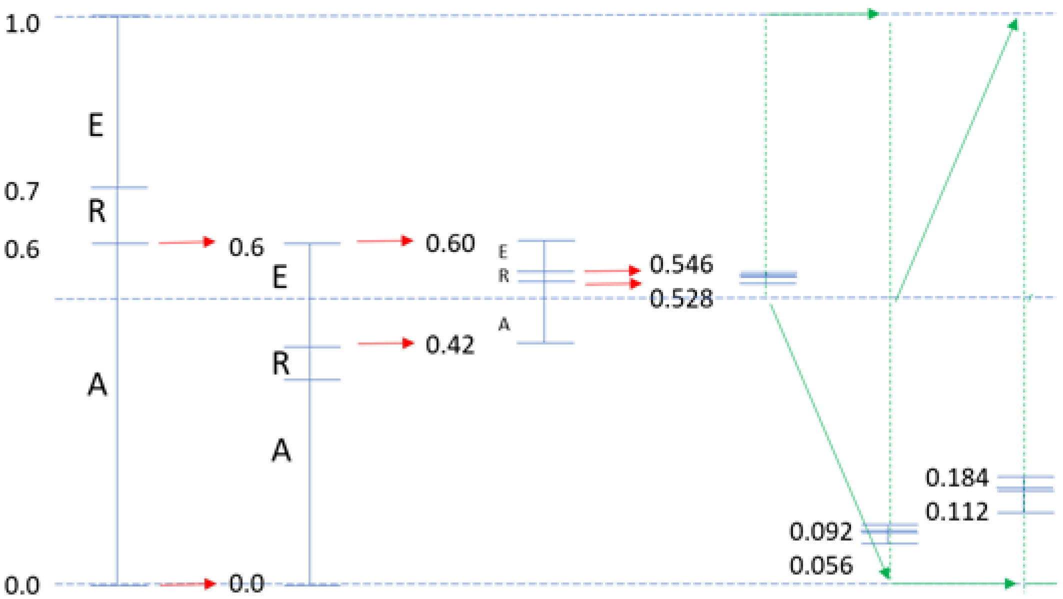


Codificação Aritmética com Escalamento

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

- Mensagem: “AERA”

s = next symbol	A	E	R						
High(s)	0.6	1.0	0.7						
Low(s)	0.0	0.7	0.6						
High = Low + Range x High(s)	0.6	0.60	0.546	0.092	0.184				
Low = Low + Range x Low(s)	0.0	0.42	0.528	0.056	0.112				
Range = High - Low	0.6	0.18	0.018	0.036	0.072				
output				1	0				

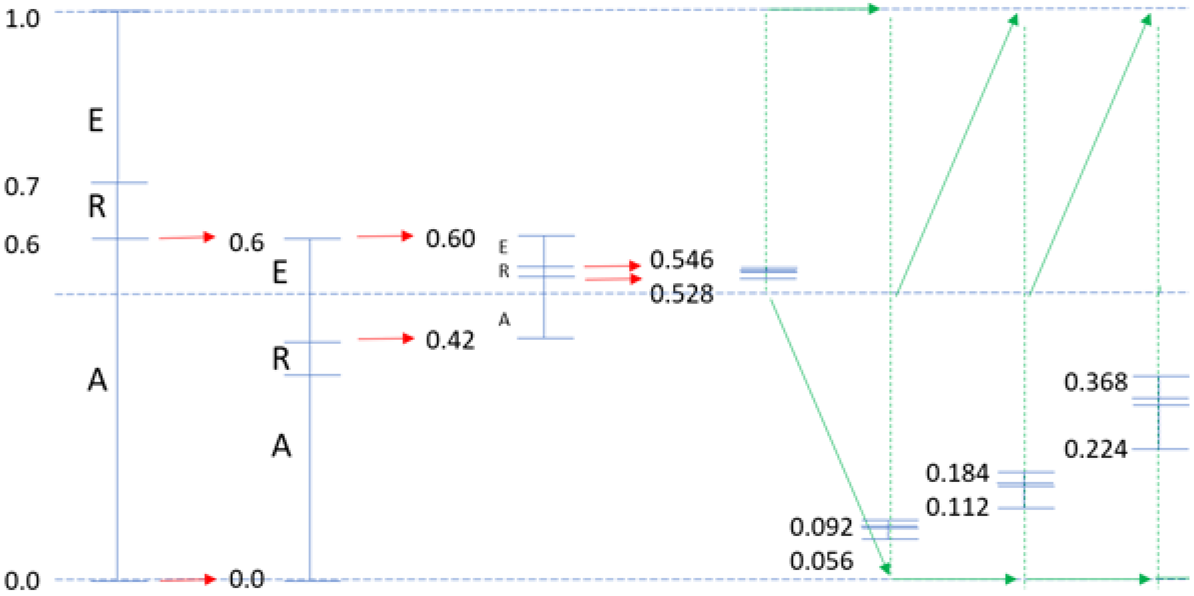


Codificação Aritmética com Escalamento

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

- Mensagem: “AERA”

s = next symbol	A	E	R						
High(s)	0.6	1.0	0.7						
Low(s)	0.0	0.7	0.6						
High = Low + Range x High(s)	0.6	0.60	0.546	0.092	0.184	0.368			
Low = Low + Range x Low(s)	0.0	0.42	0.528	0.056	0.112	0.224			
Range = High - Low	0.6	0.18	0.018	0.036	0.072	0.144			
output				1	0	0			

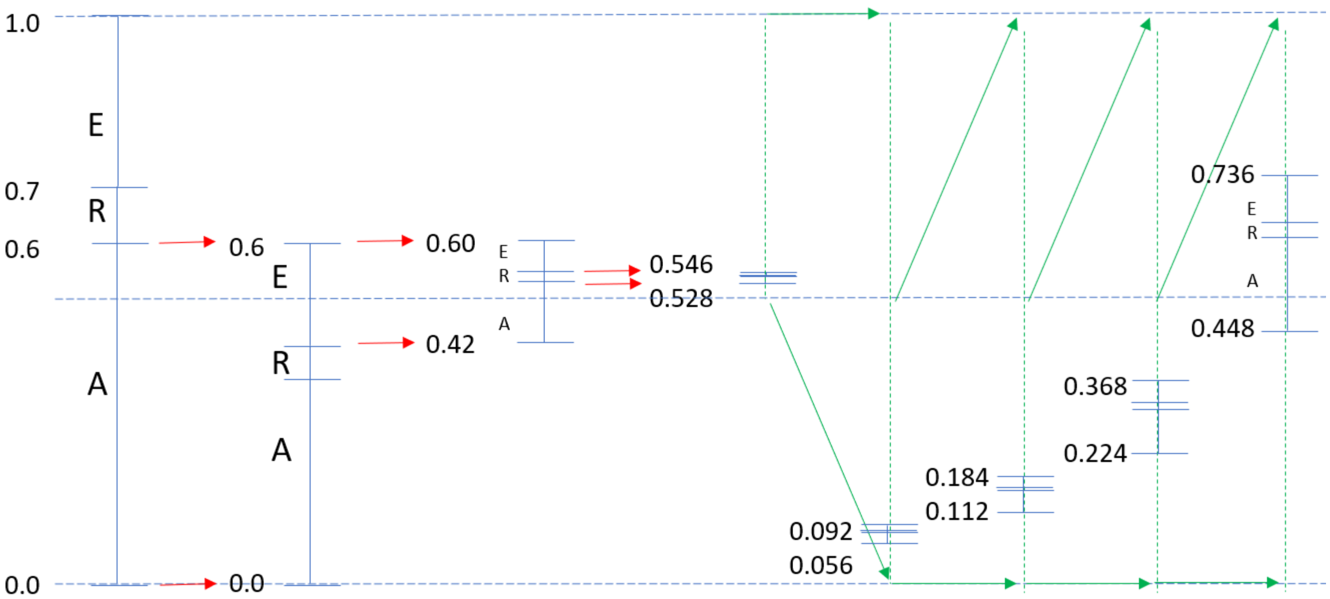


Codificação Aritmética com Escalamento

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

- Mensagem: “AERA”

s = next symbol	A	E	R					
High(s)	0.6	1.0	0.7					
Low(s)	0.0	0.7	0.6					
High = Low + Range x High(s)	0.6	0.60	0.546	0.092	0.184	0.368	0.736	
Low = Low + Range x Low(s)	0.0	0.42	0.528	0.056	0.112	0.224	0.448	
Range = High - Low	0.6	0.18	0.018	0.036	0.072	0.144	0.288	
output				1	0	0	0	

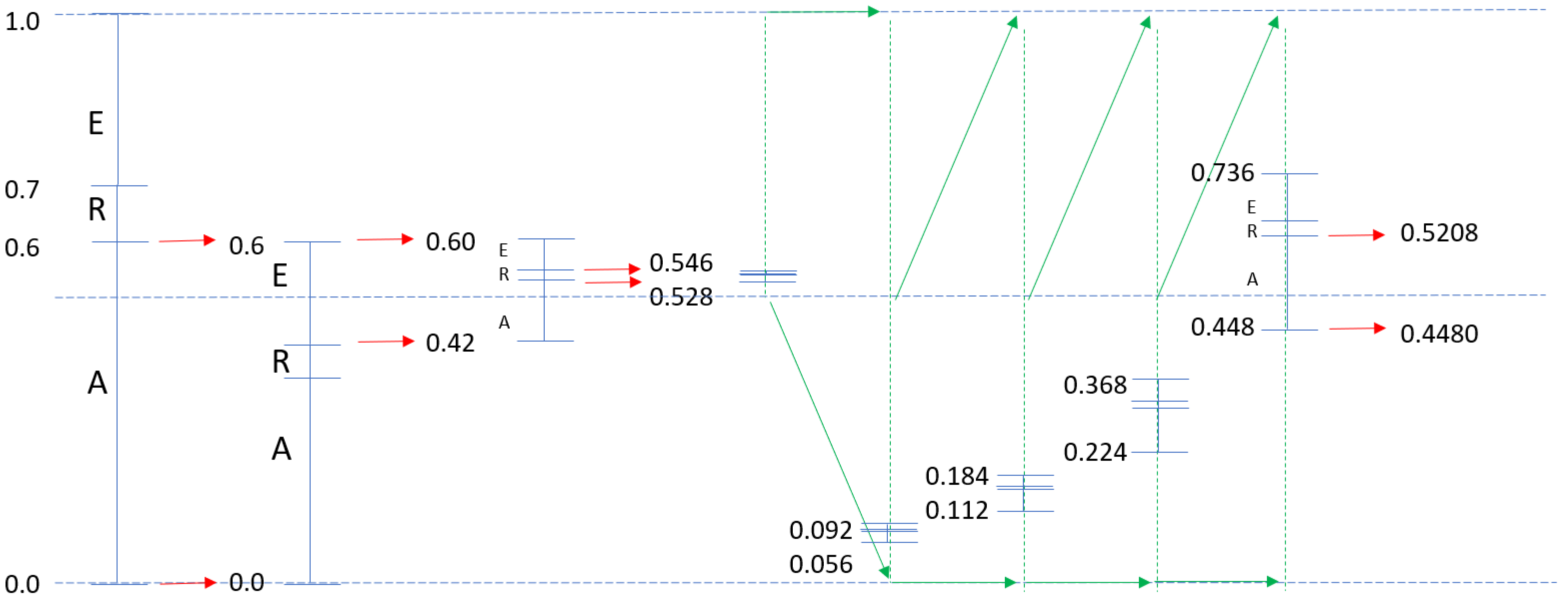


Codificação Aritmética com Escalamento

Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

- Mensagem: “AERA”

s = next symbol	A	E	R					A	
High(s)	0.6	1.0	0.7					0.6	
Low(s)	0.0	0.7	0.6					0.0	
High = Low + Range x High(s)	0.6	0.60	0.546	0.092	0.184	0.368	0.736	0.6208	
Low = Low + Range x Low(s)	0.0	0.42	0.528	0.056	0.112	0.224	0.448	0.4480	
Range = High - Low	0.6	0.18	0.018	0.036	0.072	0.144	0.288	0.1728	
output				1	0	0	0		

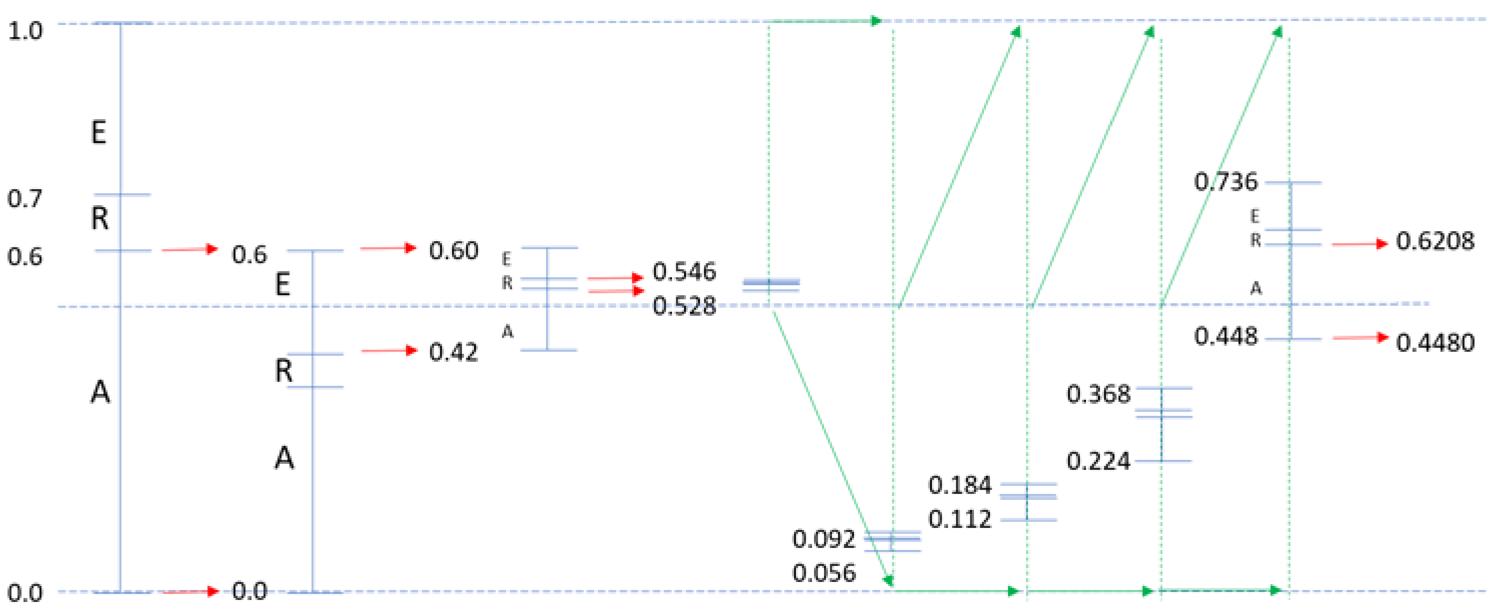


Símbolo	prob.	interval
A	0.60	0.00 - 0.60
R	0.10	0.60 - 0.70
E	0.30	0.70 - 1.00

Codificação Aritmética com Escalamento

- Mensagem: “AERA”

s = next symbol	A	E	R					A	
High(s)	0.6	1.0	0.7					0.6	
Low(s)	0.0	0.7	0.6					0.0	
High = Low + Range x High(s)	0.6	0.60	0.546	0.092	0.184	0.368	0.736	0.6208	
Low = Low + Range x Low(s)	0.0	0.42	0.528	0.056	0.112	0.224	0.448	0.4480	
Range = High - Low	0.6	0.18	0.018	0.036	0.072	0.144	0.288	0.1728	
output				1	0	0	0		1



Valor neste intervalo

Codificação Aritmética com Escalamento

- Mensagem: “AERA”
- Código transmitido: “10001”

  - Notar que os reescalamentos não são mais que um “shift” e é transmitido o bit de maior peso que é igual para o High e par o Low

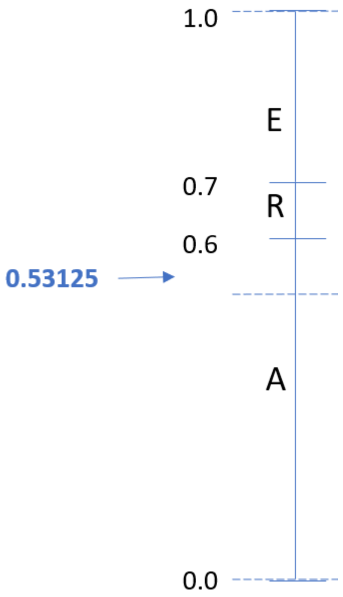
  - O último valor informa o receptor que a trama terminou, o valor escolhido é o 0.5

Símbolo	interval
A	0.00 - 0.60
R	0.60 - 0.70
E	0.70 - 1.00

Codificação Aritmética com Escalamento

- Descodificação
  - Reproduz o codificador

output symbol s									
High(s)									
Low(s)									
High = Low + Range x High(s)	1.0								
Low = Low + Range x Low(s)	0.0								
Range = High - Low	1.0								
value	0.53125								
Binary Value	10001								

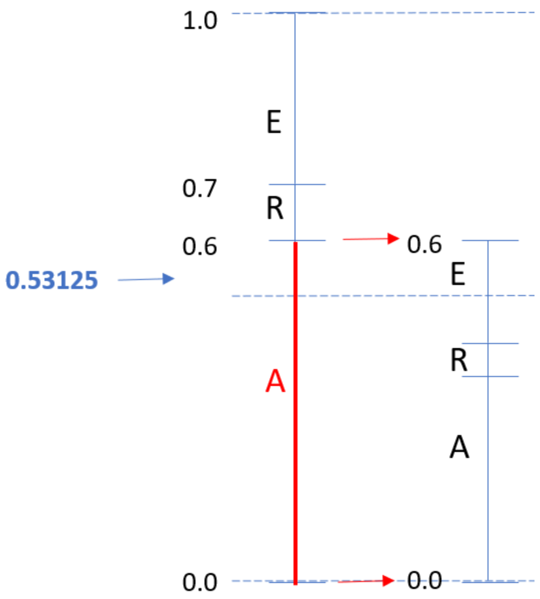


Codificação Aritmética com Escalamento

Símbolo	interval
A	0.000 - 0.360
R	0.360 - 0.420
E	0.420 - 1.000

- Descodificação
  - Reproduz o codificador

output symbol s		A							
High(s)		0.6							
Low(s)		0.0							
High = Low + Range x High(s)	1.0	0.6							
Low = Low + Range x Low(s)	0.0	0.0							
Range = High - Low	1.0	0.6							
value	0.53125	0.53125							
Binary Value	10001	10001							

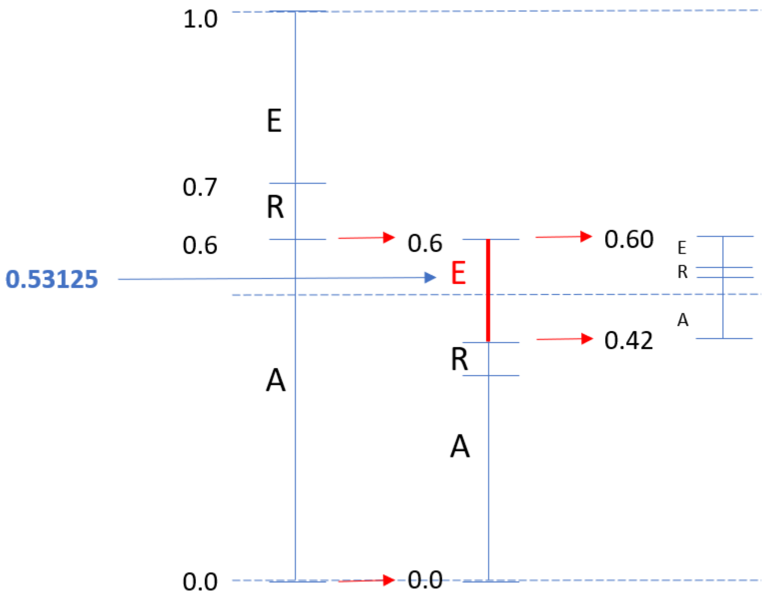


Codificação Aritmética com Escalamento

Símbolo	interval
A	0.420 - 0.528
R	0.528 - 0.546
E	0.546 – 0.600

- Descodificação
  - Reproduz o codificador

output symbol s		A	E						
High(s)		0.6	1.0						
Low(s)		0.0	0.7						
High = Low + Range x High(s)	1.0	0.6	0.60						
Low = Low + Range x Low(s)	0.0	0.0	0.42						
Range = High - Low	1.0	0.6	0.18						
value	0.53125	0.53125	0.53125						
Binary Value	10001	10001	10001						

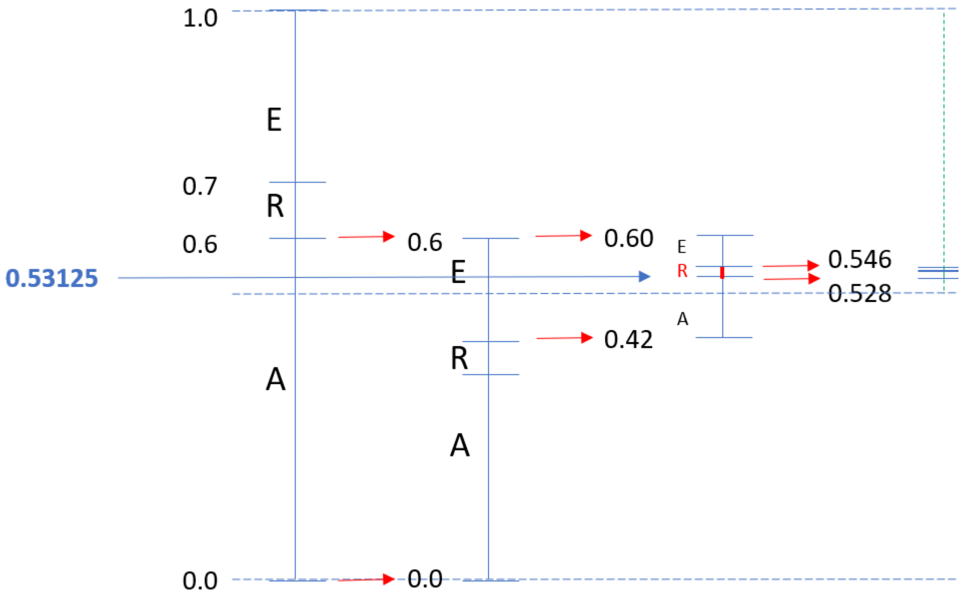


Símbolo	interval
A	0.5280 - 0.5388
R	0.5388 - 0.5406
E	0.5406 – 0.5460

Codificação Aritmética com Escalamento

- Descodificação
  - Reproduz o codificador

output symbol s		A	E	R					
High(s)		0.6	1.0	0.7					
Low(s)		0.0	0.7	0.6					
High = Low + Range x High(s)	1.0	0.6	0.60	0.546					
Low = Low + Range x Low(s)	0.0	0.0	0.42	0.528					
Range = High - Low	1.0	0.6	0.18	0.018					
value	0.53125	0.53125	0.53125	0.53125					
Binary Value	10001	10001	10001	10001					

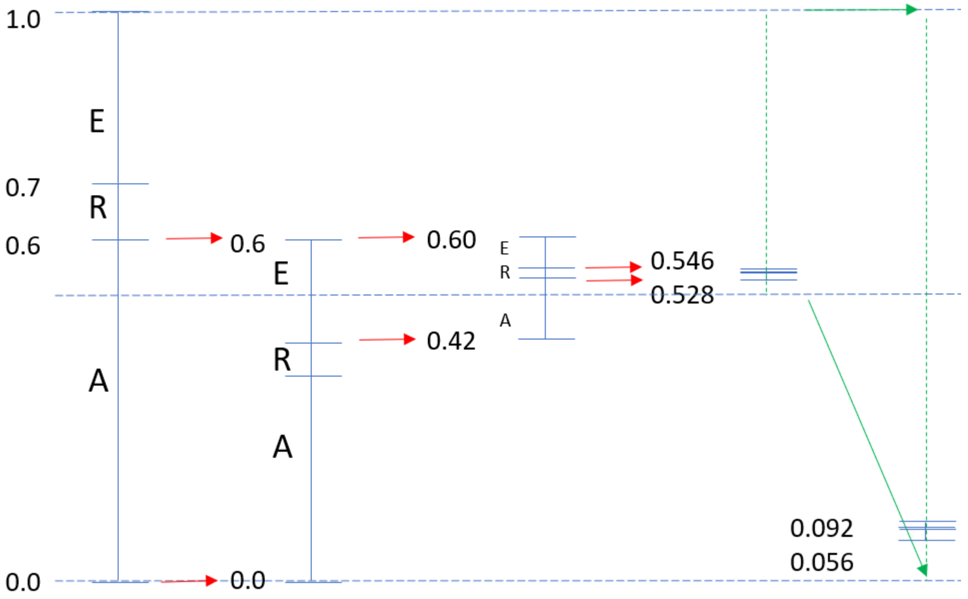


Codificação Aritmética com Escalamento

Símbolo	interval
A	0.0560 - 0.0776
R	0.0776 - 0.0812
E	0.0812 – 0.0920

- Descodificação
  - Reproduz o codificador

output symbol s		A	E	R					
High(s)		0.6	1.0	0.7					
Low(s)		0.0	0.7	0.6					
High = Low + Range x High(s)	1.0	0.6	0.60	0.546	0.092				
Low = Low + Range x Low(s)	0.0	0.0	0.42	0.528	0.056				
Range = High - Low	1.0	0.6	0.18	0.018	0.036				
value	0.53125	0.53125	0.53125	0.53125	0.0625				
Binary Value	10001	10001	10001	10001	0001				

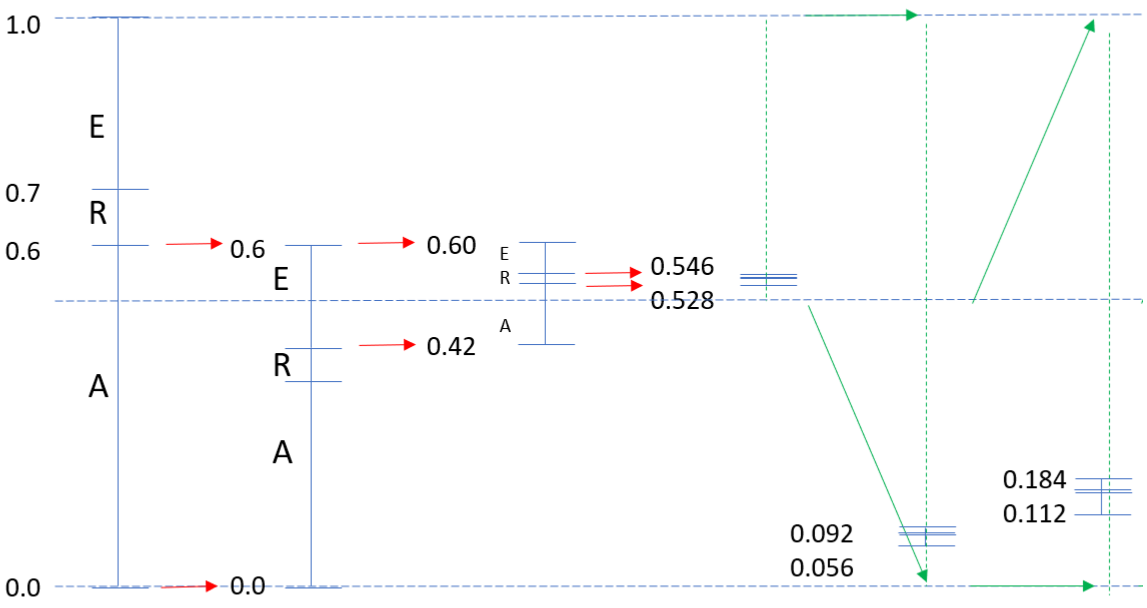


Símbolo	interval
A	0.1120 - 0.1552
R	0.1552 - 0.1624
E	0.1624 - 0.1840

Codificação Aritmética com Escalamento

- Descodificação
  - Reproduz o codificador

output symbol s		A	E	R					
High(s)		0.6	1.0	0.7					
Low(s)		0.0	0.7	0.6					
High = Low + Range x High(s)	1.0	0.6	0.60	0.546	0.092	0.184			
Low = Low + Range x Low(s)	0.0	0.0	0.42	0.528	0.056	0.112			
Range = High - Low	1.0	0.6	0.18	0.018	0.036	0.072			
value	0.53125	0.53125	0.53125	0.53125	0.0625	0.125			
Binary Value	10001	10001	10001	10001	0001	001			

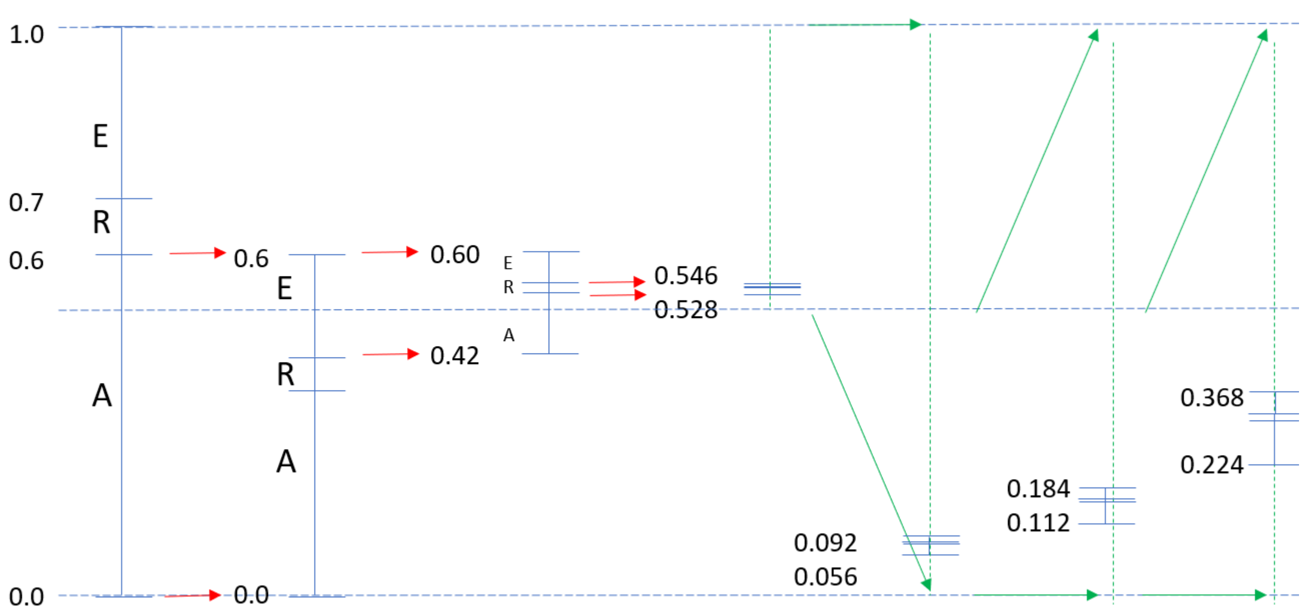


Codificação Aritmética com Escalamento

Símbolo	interval
A	0.2240 - 0.3104
R	0.3104 - 0.3248
E	0.3248 - 0.3680

- Descodificação
  - Reproduz o codificador

output symbol s		A	E	R					
High(s)		0.6	1.0	0.7					
Low(s)		0.0	0.7	0.6					
High = Low + Range x High(s)	1.0	0.6	0.60	0.546	0.092	0.184	0.368		
Low = Low + Range x Low(s)	0.0	0.0	0.42	0.528	0.056	0.112	0.224		
Range = High - Low	1.0	0.6	0.18	0.018	0.036	0.072	0.144		
value	0.53125	0.53125	0.53125	0.53125	0.0625	0.125	0.25		
Binary Value	10001	10001	10001	10001	0001	001	01		

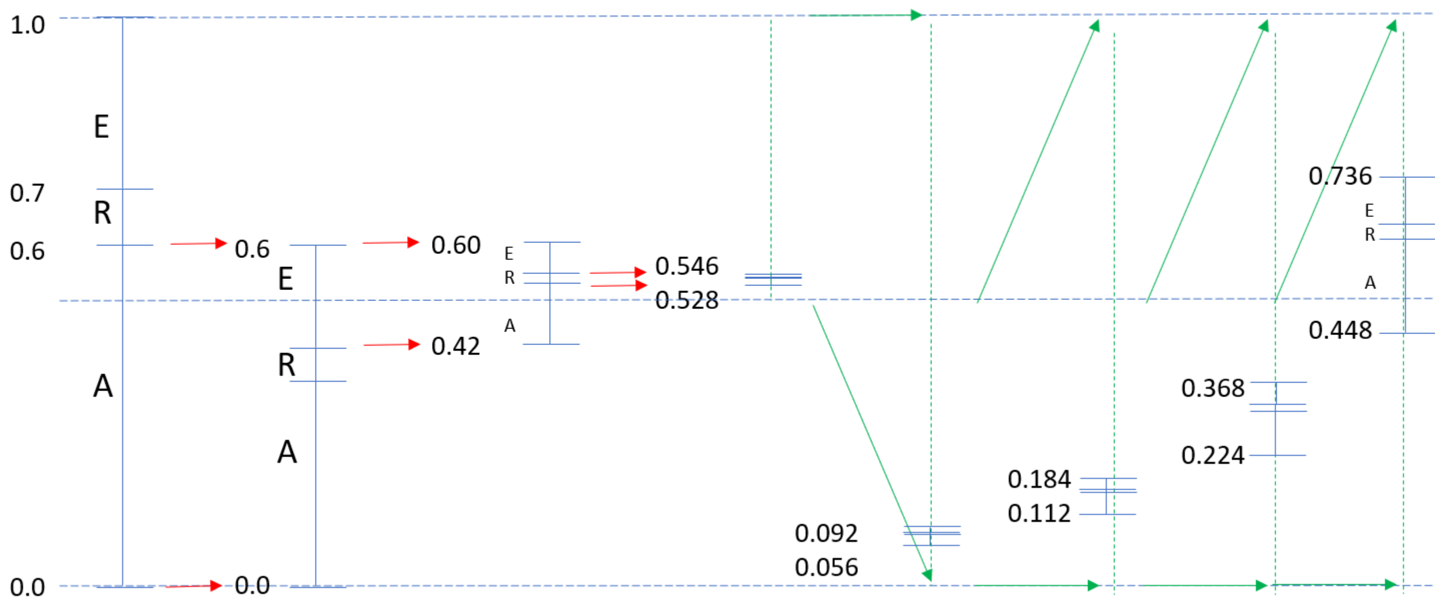


Codificação Aritmética com Escalamento

Símbolo	interval
A	0.4480 - 0.6208
R	0.6208 - 0.6496
E	0.6496 - 0.7360

- Descodificação
  - Reproduz o codificador

output symbol s		A	E	R					
High(s)		0.6	1.0	0.7					
Low(s)		0.0	0.7	0.6					
High = Low + Range x High(s)	1.0	0.6	0.60	0.546	0.092	0.184	0.368	0.736	
Low = Low + Range x Low(s)	0.0	0.0	0.42	0.528	0.056	0.112	0.224	0.448	
Range = High - Low	1.0	0.6	0.18	0.018	0.036	0.072	0.144	0.288	
value	0.53125	0.53125	0.53125	0.53125	0.0625	0.125	0.25	0.5	
Binary Value	10001	10001	10001	10001	0001	001	01	1	

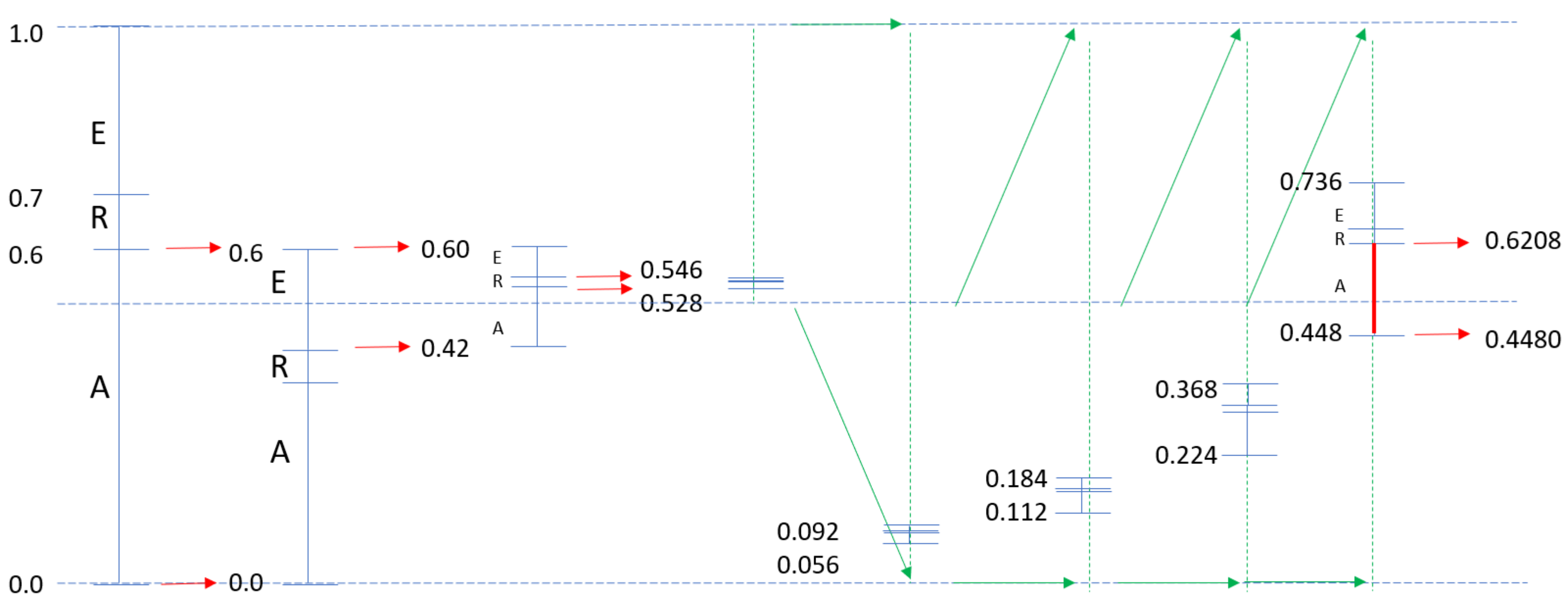


Símbolo	interval
A	0.44800 - 0.55168
R	0.55168 - 0.56896
E	0.56896 - 0.62080

Codificação Aritmética com Escalamento

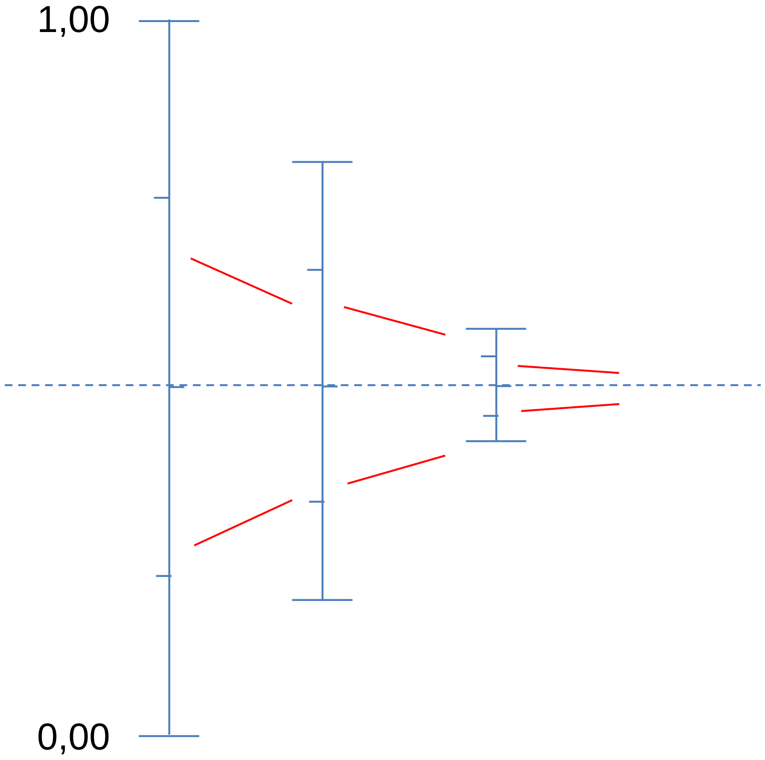
- Descodificação
  - Reproduz o codificador

output symbol s		A	E	R					A
High(s)		0.6	1.0	0.7					0.6
Low(s)		0.0	0.7	0.6					0.0
High = Low + Range x High(s)	1.0	0.6	0.60	0.546	0.092	0.184	0.368	0.736	0.6208
Low = Low + Range x Low(s)	0.0	0.0	0.42	0.528	0.056	0.112	0.224	0.448	0.4480
Range = High - Low	1.0	0.6	0.18	0.018	0.036	0.072	0.144	0.288	0.1728
value	0.53125	0.53125	0.53125	0.53125	0.0625	0.125	0.25	0.5	
Binary Value	10001	10001	10001	10001	0001	001	01	1	



Descodificação Aritmética décimal

- No caso do intervalo estar dentro de [0.25, 0.75]
- Condição E3:



- Faz-se o escalamento de [0.25 0.75] para [0, 1.0]

- Guarda o registo do escalamento

- Quando o intervalo estiver [0.5 1] Transmite-se o **1** seguido de **0**

(o número de vezes que está no registo)

- Se o intervalo estiver [0 0.5] Transmite-se o **0** seguido de **1**

(o número de vezes que está no registo)

Codificação Aritmética - inteira

- O procedimento anterior pode ser implementado com valores inteiros, permitindo que o processo de codificação e descodificação sejam mais rápidos.
- Para esta implementação escolhe-se um valor máximo que seja potência de 2 para optimizar os cálculos, assim:
  - o valor mínimo 0 é 000…0
  - o valor máximo 1 é 111…1
  - O valor 0,5 é 100…0
- As equações são muito semelhantes:

$$\begin{aligned} low &= low + \left\lfloor \frac{(high - low + 1) \times cumcount(s-1)}{totalcount} \right\rfloor \\ high &= low + \left\lfloor \frac{(high - low + 1) \times cumcount(s)}{totalcount} \right\rfloor - 1 \end{aligned}$$

- onde

$$cumcount(s_i) = \sum_{i=1}^{N\,simbolos} \texttt{N ocorrencias do simbolo } s_i$$

Codificação Aritmética - inteira

- A escolha do tamanho (nº de bits) para o totalcount, relaciona-se com o menor intervalo que temos de representar:

$$\frac{1}{4}2^m > totalcount$$

$$m = 2 + \lceil log_2(totalcount) \rceil$$

- Processo de codificação:

```
Set low and high
countE3 = 0
While not EOF
    Get symbol
    Update low and high
    While( E1 or E2 or E3)
        If E1,  Send 1, send countE3 times 0, scale value
        If E2,  Send 0, send countE3 times 1, scale value
        If E3,  countE3 = countE3+1, scale value
    end
end
Send MSB(Low), send countE3 times the complement of MSB(Low), send remaining bits of  Low
```

Codificação Aritmética - inteira

- Assumindo a tabela de occorrências:

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

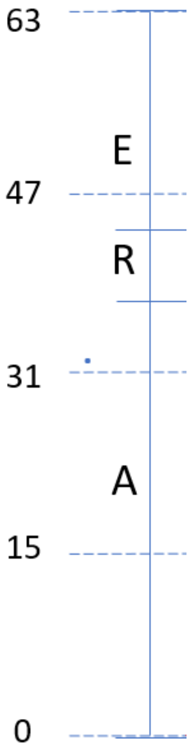
- A mensagem a codificar: ”AERA”
- Totalcount = 10
- $m = 2 + \lceil log_2(totalcount) \rceil$ = 6
- Low = 0 [000000]
- High = 63 [111111]

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

Codificação Aritmética - inteira

- Mensagem ”AERA”

s = read symbol		A										
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63											
	[111111]											
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0											
	[000000]											
Output												
CountE3	0											
conditions E1 or E2 or E3												

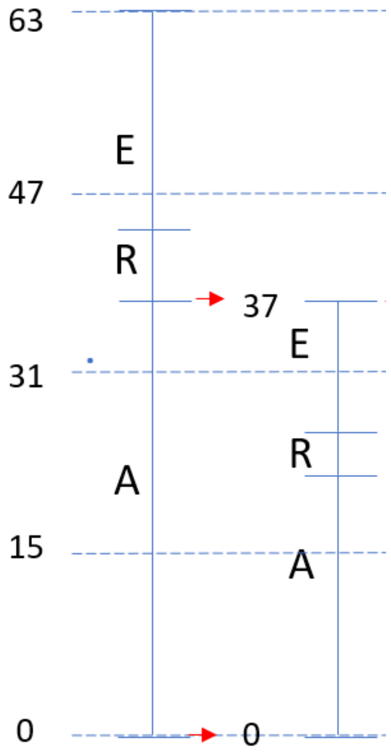


	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

Codificação Aritmética - inteira

- Mensagem ”AERA”

s = read symbol		A										
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37										
	[111111]	[100101]										
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0										
	[000000]	[000000]										
Output												
CountE3	0	0										
conditions E1 or E2 or E3												

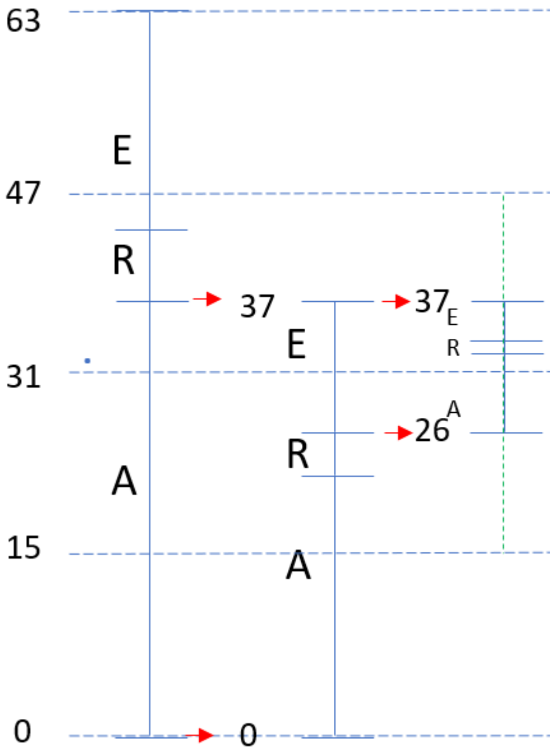


Codificação Aritmética - inteira

- Mensagem ”AERA”

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

s = read symbol		A	E									
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37									
	[111111]	[100101]	[100101]									
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26									
	[000000]	[000000]	[011010]									
Output												
CountE3	0	0	0									
conditions E1 or E2 or E3			E3									

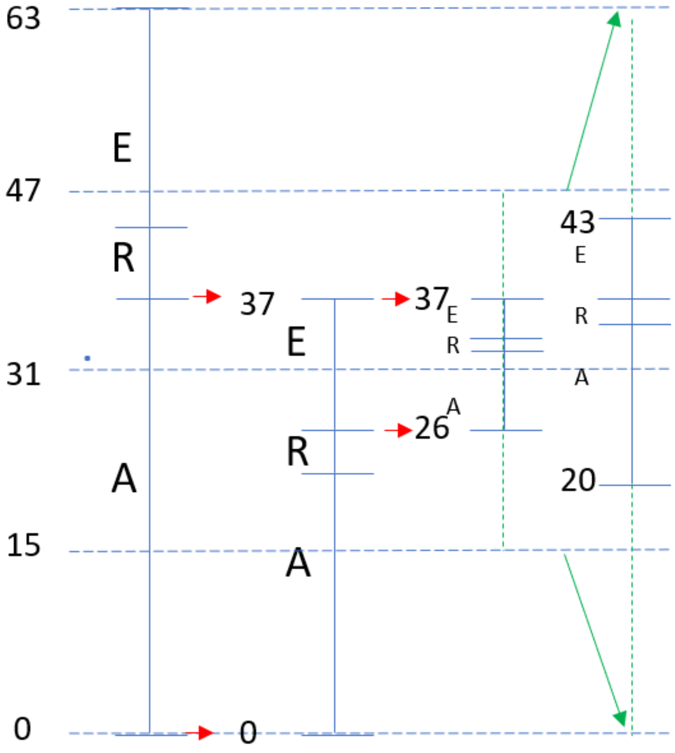


Codificação Aritmética - inteira

- Mensagem ”AERA”

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

s = read symbol		A	E									
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43								
	[111111]	[100101]	[100101]	[101011]								
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20								
	[000000]	[000000]	[011010]	[010100]								
Output												
CountE3	0	0	0	1								
conditions E1 or E2 or E3			E3	E3								

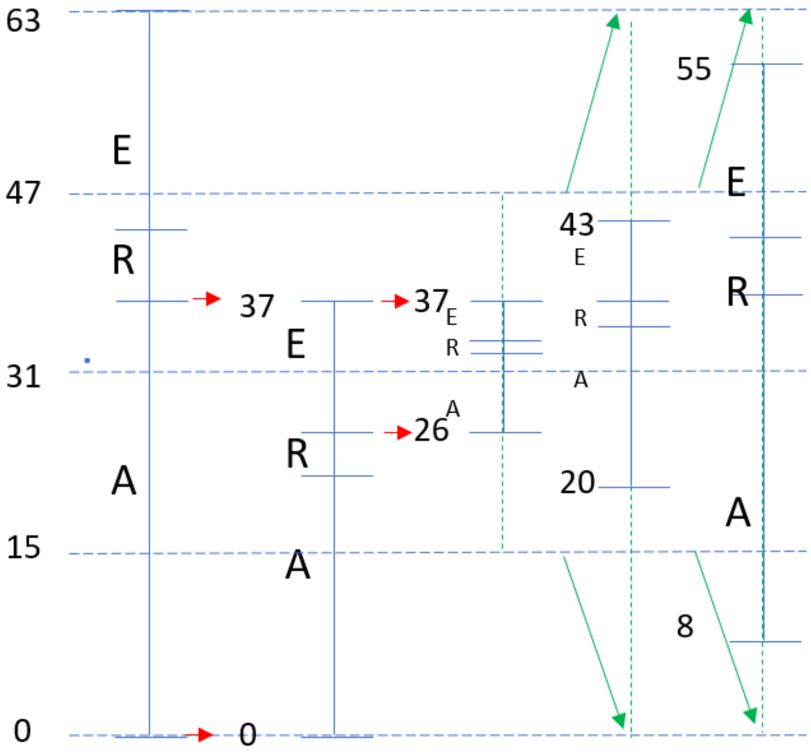


	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

Codificação Aritmética - inteira

- Mensagem ”AERA”

s = read symbol		A	E									
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55							
	[111111]	[100101]	[100101]	[101011]	[110111]							
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8							
	[000000]	[000000]	[011010]	[010100]	[000100]							
Output												
CountE3	0	0	0	1	2							
conditions E1 or E2 or E3			E3	E3								

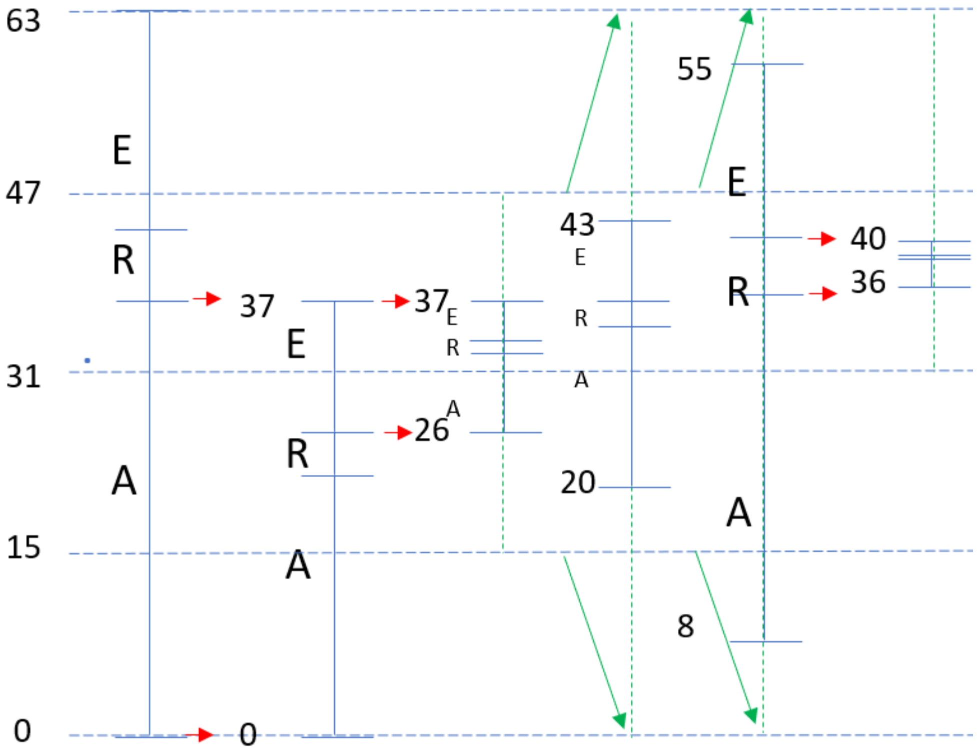


	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

Codificação Aritmética - inteira

- Mensagem ”AERA”

s = read symbol		A	E			R						
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55	40						
	[111111]	[100101]	[100101]	[101011]	[110111]	[101000]						
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8	36						
	[000000]	[000000]	[011010]	[010100]	[000100]	[100100]						
Output												
CountE3	0	0	0	1	2	2						
conditions E1 or E2 or E3			E3	E3		E2						

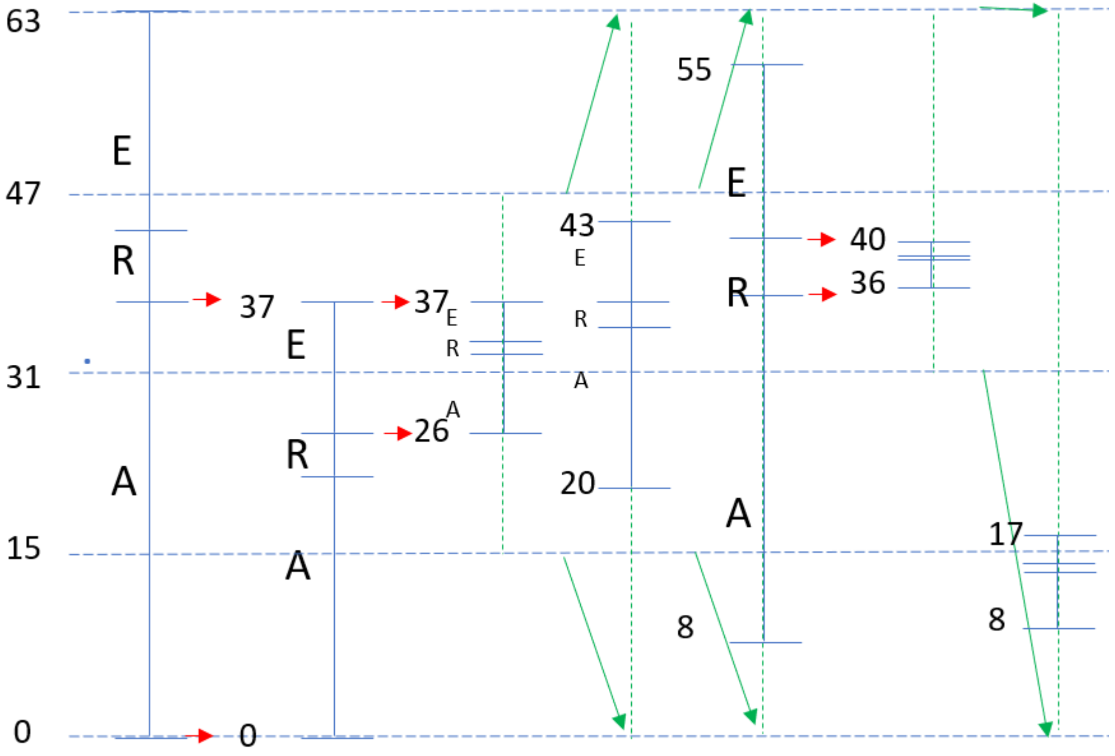


Codificação Aritmética - inteira

- Mensagem ”AERA”

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

s = read symbol		A	E			R						
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55	40	17					
	[111111]	[100101]	[100101]	[101011]	[110111]	[101000]	[010001]					
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8	36	8					
	[000000]	[000000]	[011010]	[010100]	[000100]	[100100]	[000100]					
Output							100					
CountE3	0	0	0	1	2	2	0					
conditions E1 or E2 or E3			E3	E3		E2	E1					

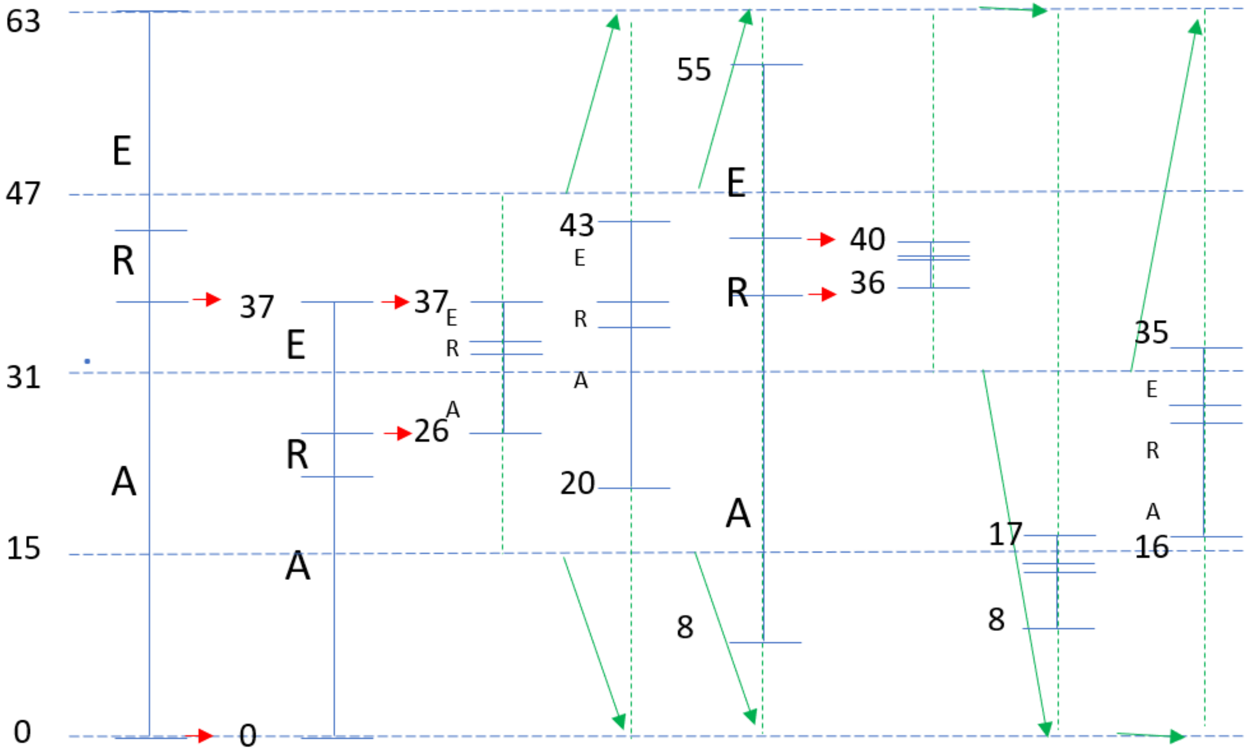


	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

Codificação Aritmética - inteira

- Mensagem ”AERA”

s = read symbol		A	E			R						
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55	40	17	35				
	[111111]	[100101]	[100101]	[101011]	[110111]	[101000]	[010001]	[100011]				
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8	36	8	16				
	[000000]	[000000]	[011010]	[010100]	[000100]	[100100]	[000100]	[010000]				
Output							100	0				
CountE3	0	0	0	1	2	2	0	0				
conditions E1 or E2 or E3			E3	E3		E2	E1	E3				

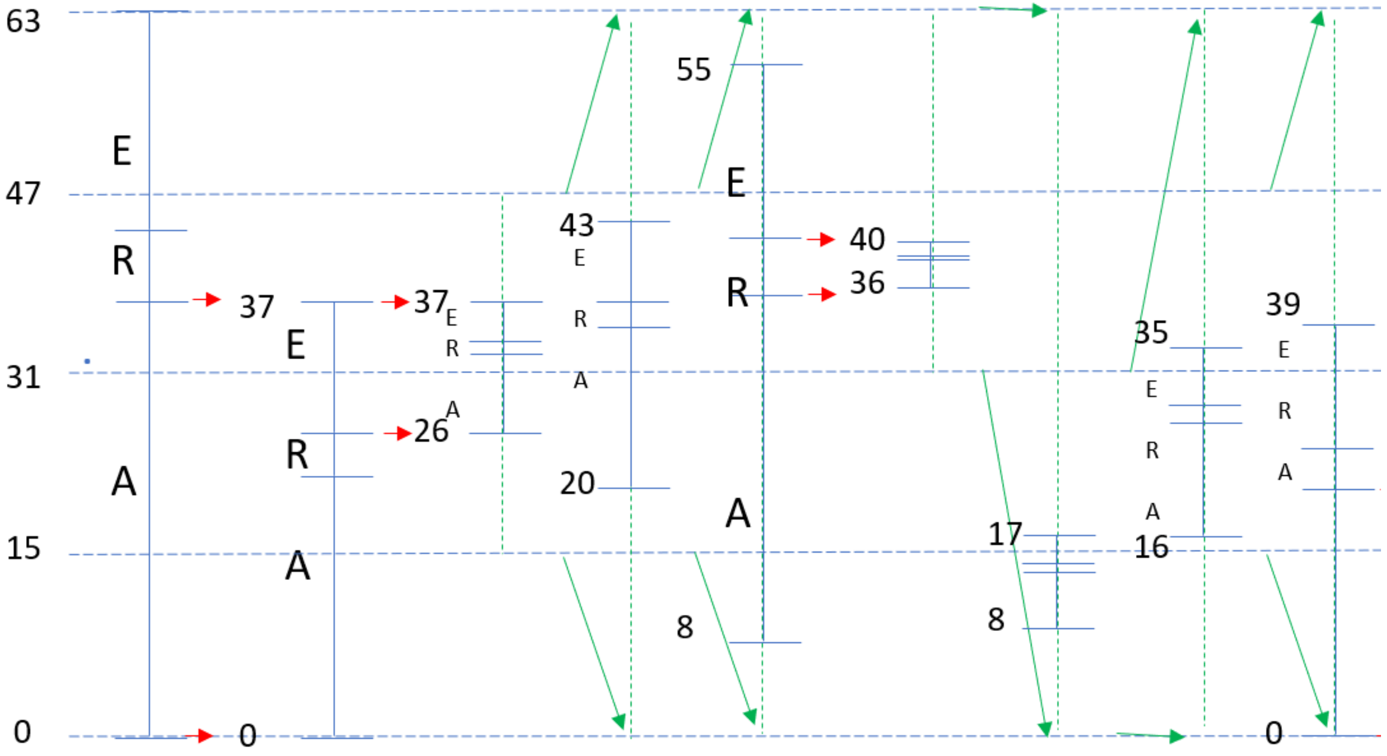


Codificação Aritmética - inteira

- Mensagem ”AERA”

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

s = read symbol		A	E			R						
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55	40	17	35	39			
	[111111]	[100101]	[100101]	[101011]	[110111]	[101000]	[010001]	[100011]	[100111]			
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8	36	8	16	0			
	[000000]	[000000]	[011010]	[010100]	[000100]	[100100]	[000100]	[010000]	[000000]			
Output							100	0				
CountE3	0	0	0	1	2	2	0	0	1			
conditions E1 or E2 or E3			E3	E3		E2	E1	E3				

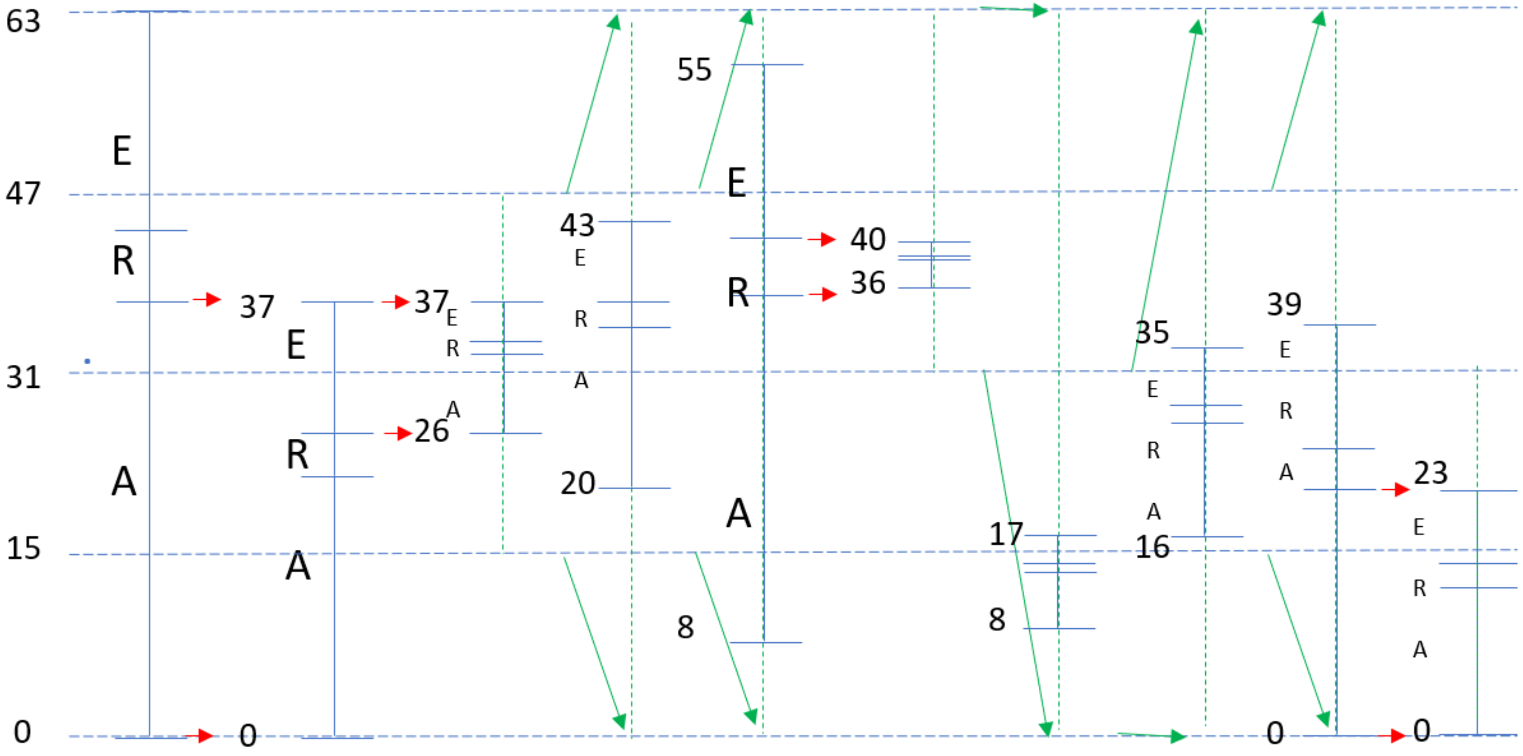


	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

Codificação Aritmética - inteira

- Mensagem ”AERA”

s = read symbol		A	E			R				A		
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55	40	17	35	39	23		
	[111111]	[100101]	[100101]	[101011]	[110111]	[101000]	[010001]	[100011]	[100111]	[010111]		
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8	36	8	16	0	0		
	[000000]	[000000]	[011010]	[010100]	[000100]	[100100]	[000100]	[010000]	[000000]	[000000]		
Output							100	0				
CountE3	0	0	0	1	2	2	0	0	1	1		
conditions E1 or E2 or E3			E3	E3		E2	E1	E3		E1		

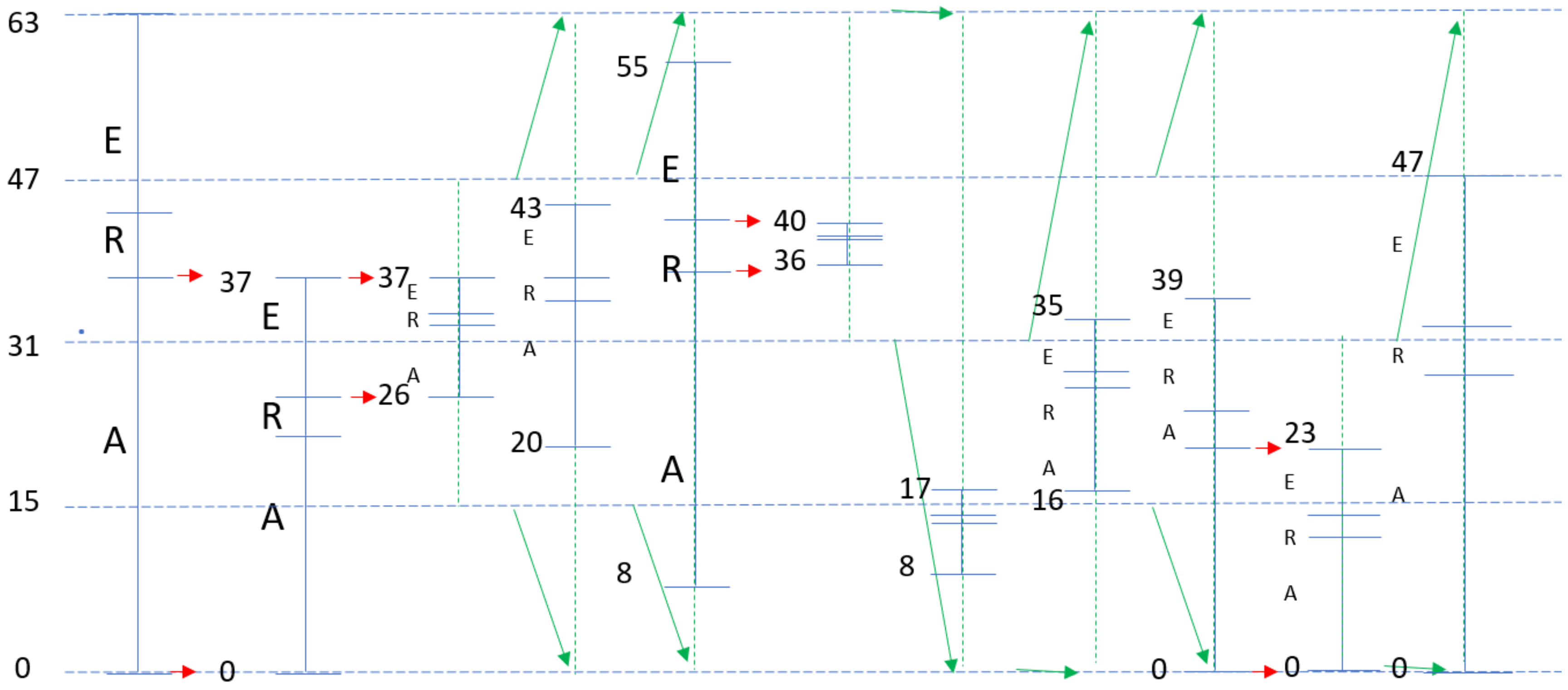


Codificação Aritmética - inteira

- Mensagem ”AERA”

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

s = read symbol		A	E			R				A		end
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55	40	17	35	39	23	47	
	[111111]	[100101]	[100101]	[101011]	[110111]	[101000]	[010001]	[100011]	[100111]	[010111]	[101111]	
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8	36	8	16	0	0	0	
	[000000]	[000000]	[011010]	[010100]	[000100]	[100100]	[000100]	[010000]	[000000]	[000000]	[000000]	
Output							100	0			01	000000
CountE3	0	0	0	1	2	2	0	0	1	1	0	
conditions E1 or E2 or E3			E3	E3		E2	E1	E3		E1		



Descodificação Aritmética - inteira

```
Set low and high
read m bits to tag
k = 0
While k < Nsimbolos
      k=k+1
      decode symbol
      Update low and high
      While( E1 or E2 or E3)
         If  E1,
             scale low and high (2x)
             scale tag and add next bit
         If E2,
            scale low and high (2(x-2^(m-1)))
            scale tag and add next bit
         If E3
           scale low and high (2(x-2^(m-2)))
           scale tag and add next bit
```

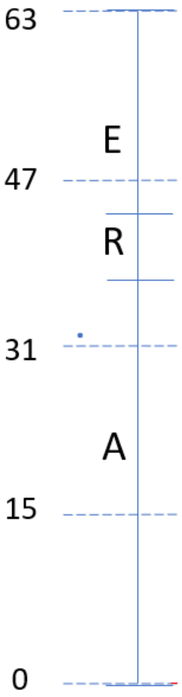
- Precisa da informação de quantos simbolos foram transmitidos e da tabela com ocorrências.

Descodificação Aritmética - inteira

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

- Código ”100001000000”

tag											
decode tag = floor([(t+1-L)*total_counts - 1]/[H+1-L]											
output											
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63										
	[111111]										
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0										
	[000000]										
conditions E1 or E2 or E3											

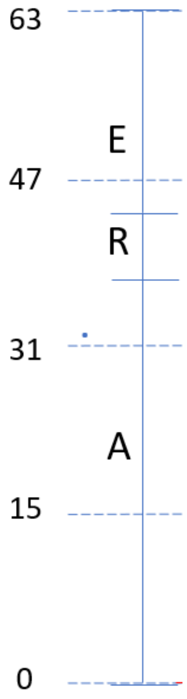


	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

Descodificação Aritmética - inteira

- Código ”100001000000”

tag		33									
		[100001]									
decode tag = floor([(t+1-L)*total_counts - 1]/[H+1-L]											
output											
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63										
	[111111]										
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0										
	[000000]										
conditions E1 or E2 or E3											

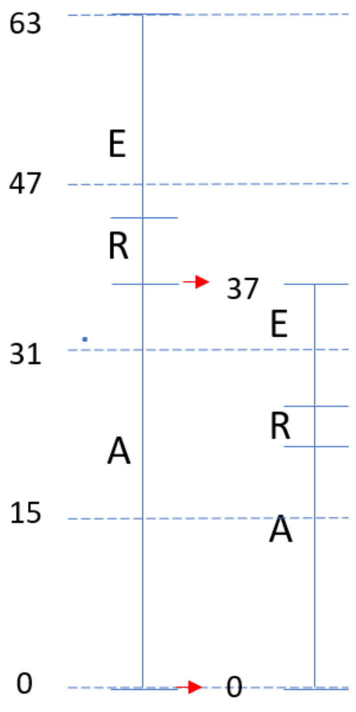


	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

Descodificação Aritmética - inteira

- Código ”100001000000”

tag		33									
		[100001]									
decode tag = floor([(t+1-L)*total_counts - 1]/[H+1-L]		5									
output		A									
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37									
	[111111]	[100101]									
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0									
	[000000]	[000000]									
conditions E1 or E2 or E3											

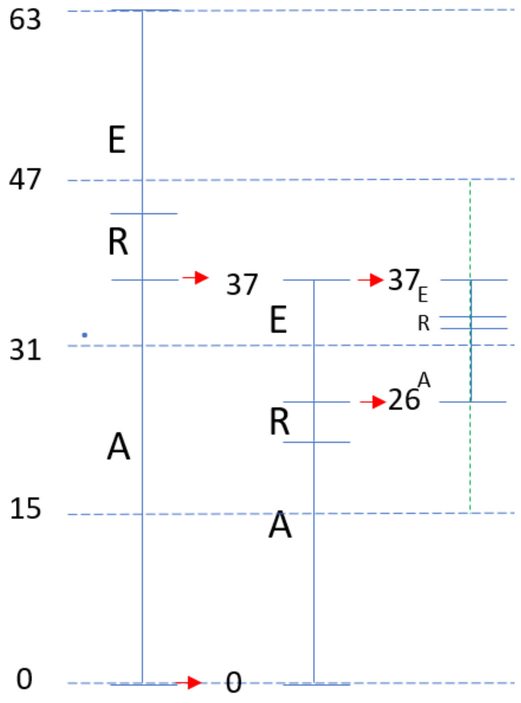


Descodificação Aritmética - inteira

- Código ”100001000000”

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

tag		33	33								
		[100001]	[100001]								
decode tag = floor([(t+1-L)*total_counts - 1]/[H+1-L]		5	8								
output		A	E								
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37								
	[111111]	[100101]	[100101]								
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26								
	[000000]	[000000]	[011010]								
conditions E1 or E2 or E3			E3								

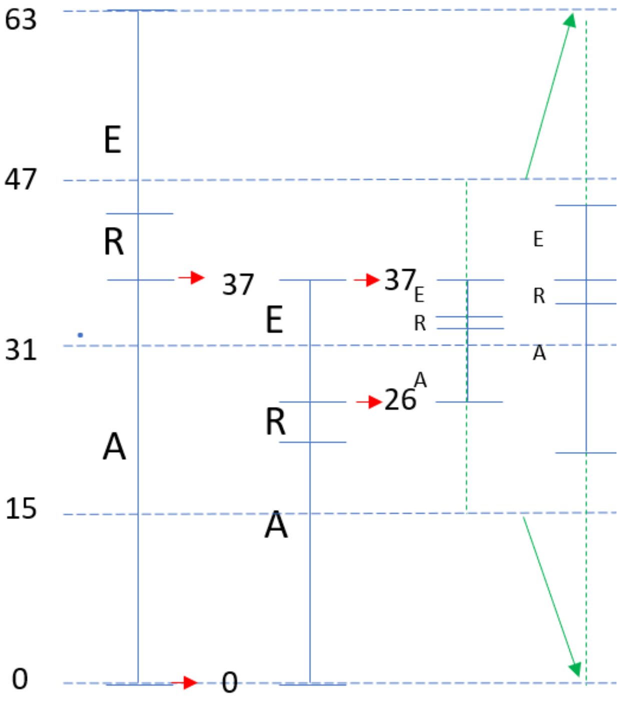


	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

Descodificação Aritmética - inteira

- Código ”100001000000”

tag		33	33	34							
		[100001]	[100001]	[100010]							
decode tag = floor([(t+1-L)*total_counts - 1]/[H+1-L]		5	8								
output		A	E								
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43							
	[111111]	[100101]	[100101]	[101011]							
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20							
	[000000]	[000000]	[011010]	[010100]							
conditions E1 or E2 or E3			E3	E3							

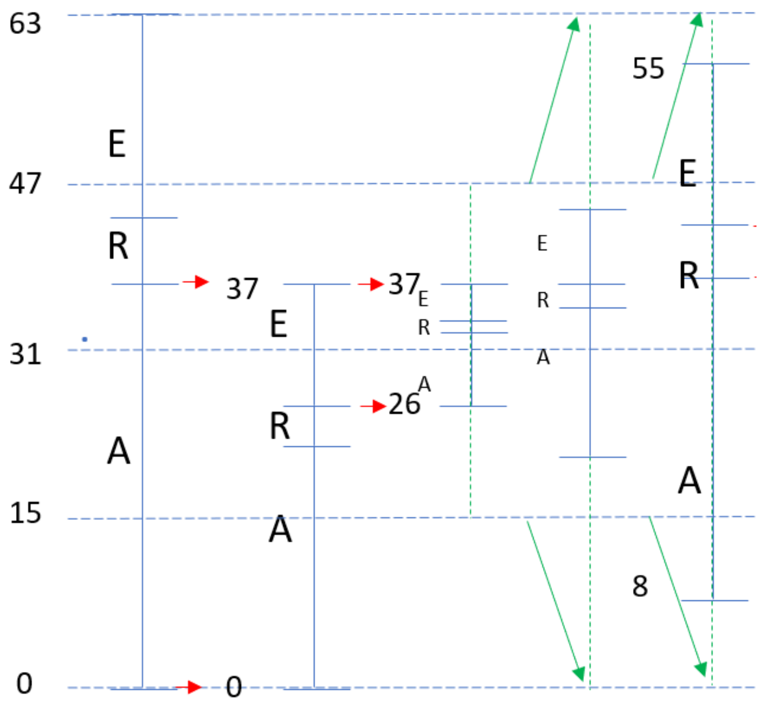


Descodificação Aritmética - inteira

- Código ”100001000000”

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

tag		33	33	34	36						
		[100001]	[100001]	[100010]	[100100]						
decode tag = floor([(t+1-L)*total_counts - 1]/[H+1-L]		5	8								
output		A	E								
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55						
	[111111]	[100101]	[100101]	[101011]	[110111]						
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8						
	[000000]	[000000]	[011010]	[010100]	[000100]						
conditions E1 or E2 or E3			E3	E3							

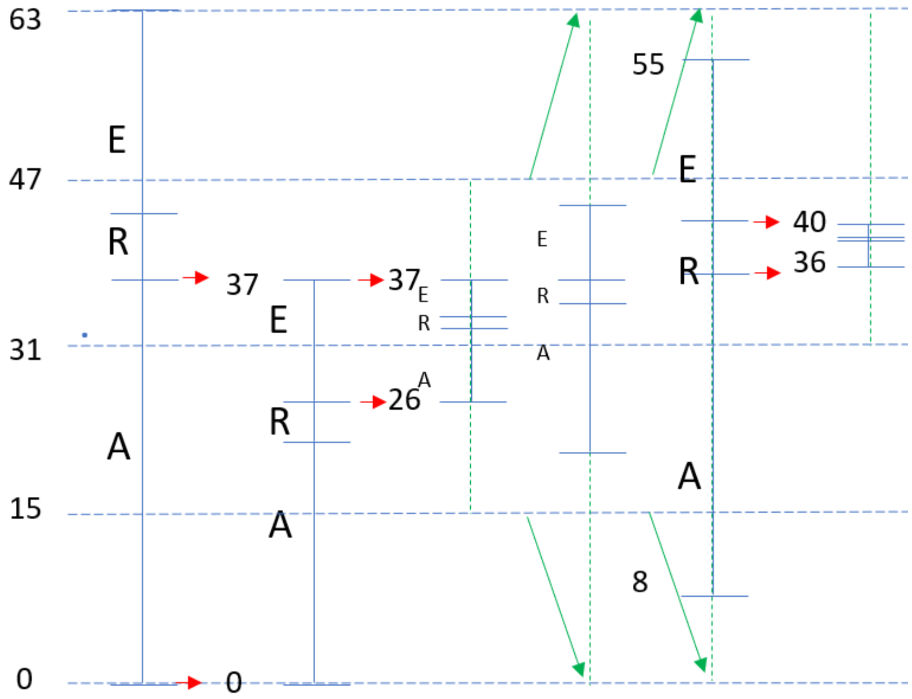


Descodificação Aritmética - inteira

- Código ”100001000000”

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

tag		33	33	34	36	36					
		[100001]	[100001]	[100010]	[100100]	[100100]					
decode tag = floor([(t+1-L)*total_counts - 1]/[H+1-L]		5	8			6					
output		A	E			R					
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55	40					
	[111111]	[100101]	[100101]	[101011]	[110111]	[101000]					
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8	36					
	[000000]	[000000]	[011010]	[010100]	[000100]	[100100]					
conditions E1 or E2 or E3			E3	E3		E2					

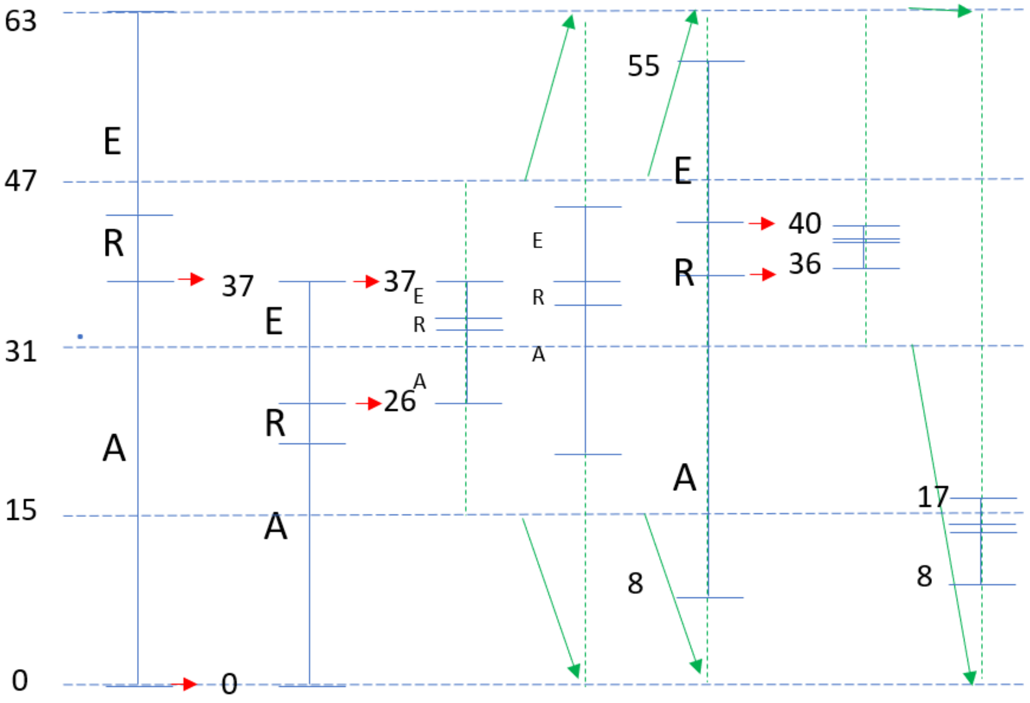


	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

Descodificação Aritmética - inteira

- Código ”1000010000000”

tag		33	33	34	36	36	8				
		[100001]	[100001]	[100010]	[100100]	[100100]	[001000]				
decode tag = floor([(t+1-L)*total_counts - 1]/[H+1-L]		5	8			6					
output		A	E			R					
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55	40	17				
	[111111]	[100101]	[100101]	[101011]	[110111]	[101000]	[010001]				
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8	36	8				
	[000000]	[000000]	[011010]	[010100]	[000100]	[100100]	[000100]				
conditions E1 or E2 or E3			E3	E3		E2	E1				

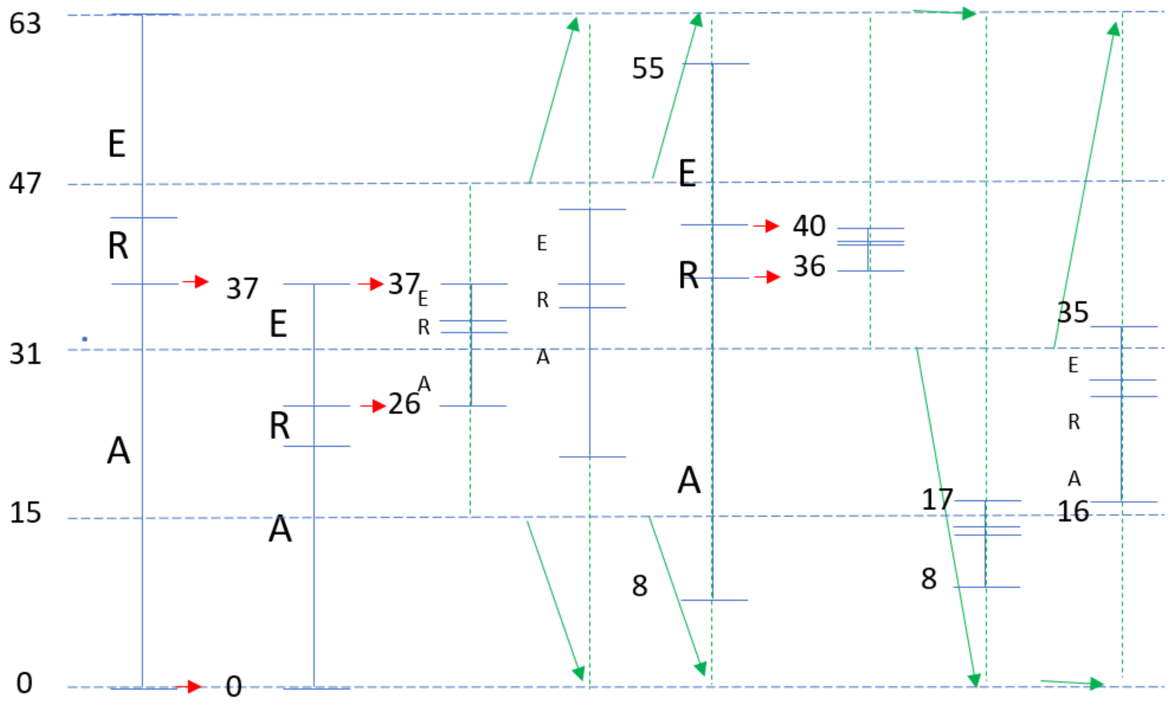


	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

Descodificação Aritmética - inteira

- Código ”100001000000”

tag		33	33	34	36	36	8	16				
		[100001]	[100001]	[100010]	[100100]	[100100]	[001000]	[010000]				
decode tag = floor([(t+1-L)*total_counts - 1]/[H+1-L]		5	8			6						
output		A	E			R						
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55	40	17	35				
	[111111]	[100101]	[100101]	[101011]	[110111]	[101000]	[010001]	[100011]				
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8	36	8	16				
	[000000]	[000000]	[011010]	[010100]	[000100]	[100100]	[000100]	[010000]				
conditions E1 or E2 or E3			E3	E3		E2	E1	E3				

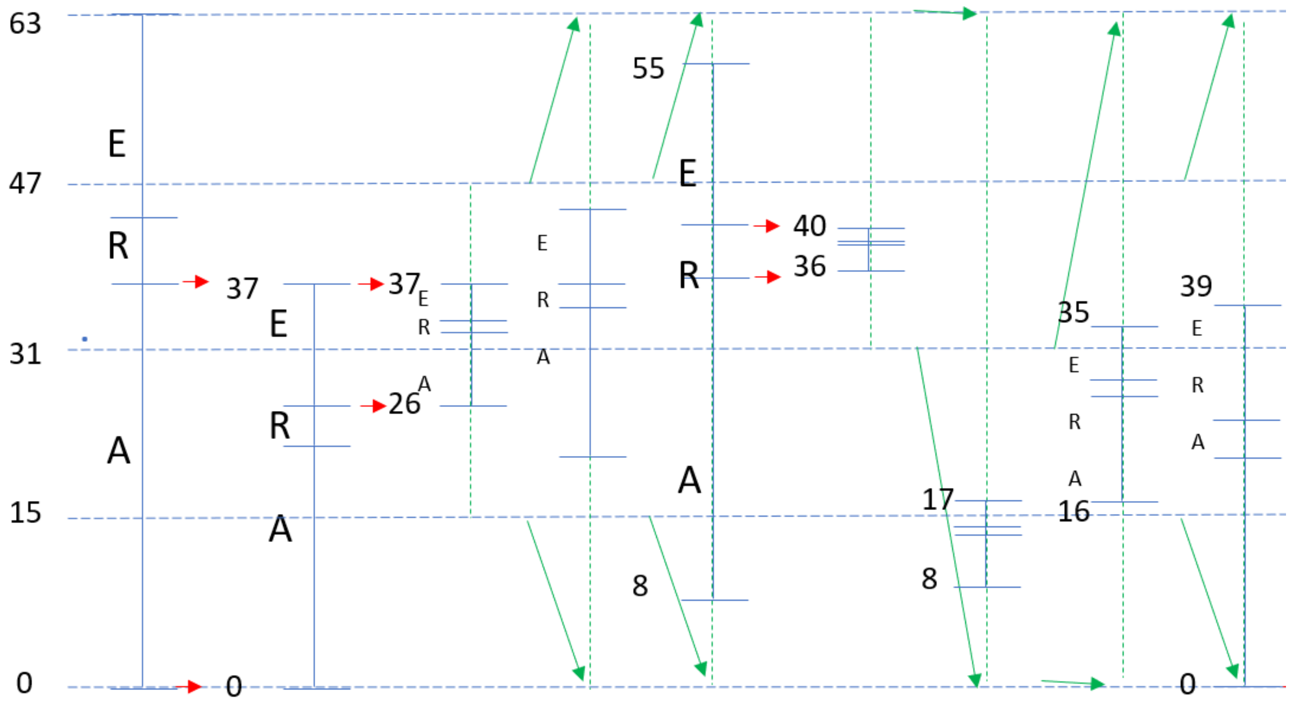


Descodificação Aritmética - inteira

- Código ”100001000000”

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

tag		33	33	34	36	36	8	16	0		
		[100001]	[100001]	[100010]	[100100]	[100100]	[001000]	[010000]	[000000]		
decode tag = floor([(t+1-L)*total_counts - 1]/[H+1-L]		5	8			6					
output		A	E			R					
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55	40	17	35	39		
	[111111]	[100101]	[100101]	[101011]	[110111]	[101000]	[010001]	[100011]	[100111]		
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8	36	8	16	0		
	[000000]	[000000]	[011010]	[010100]	[000100]	[100100]	[000100]	[010000]	[000000]		
conditions E1 or E2 or E3			E3	E3		E2	E1	E3			

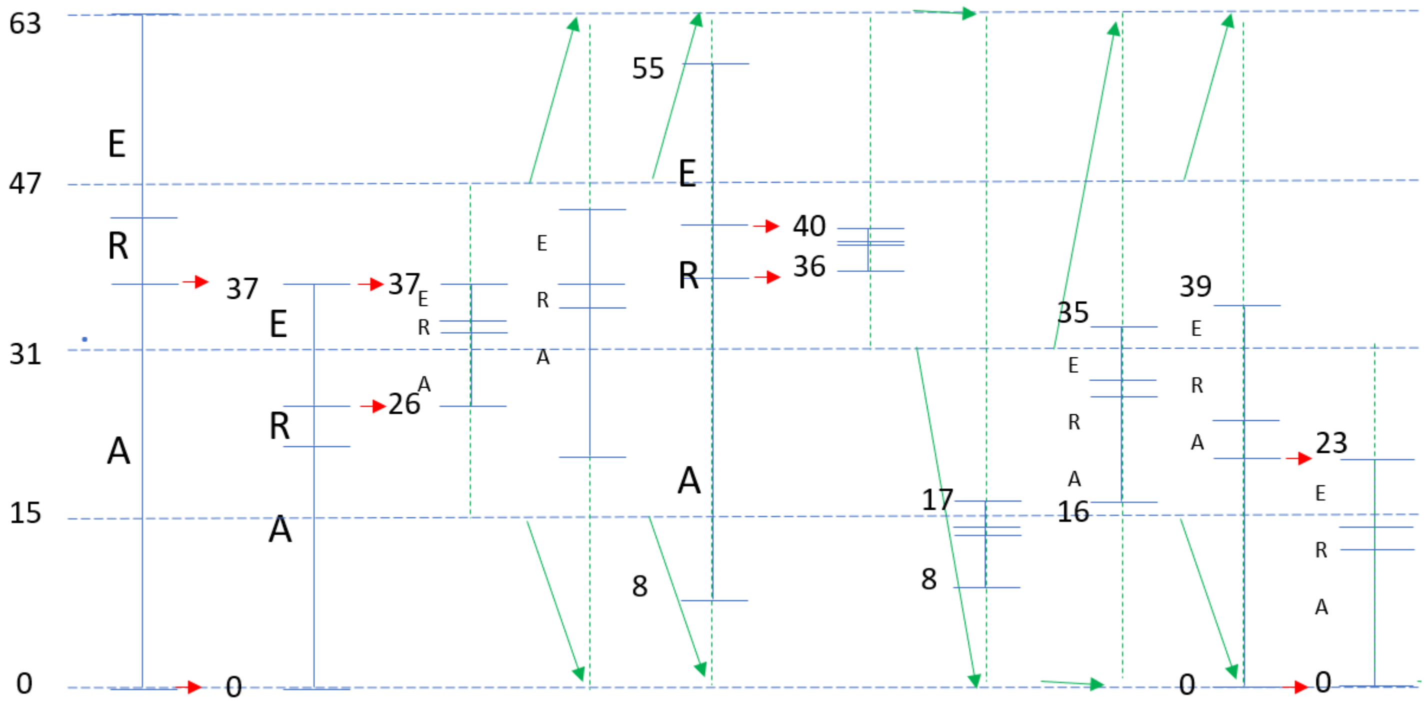


Descodificação Aritmética - inteira

- Código ”100001000000”

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

tag		33	33	34	36	36	8	16	0	0	
		[100001]	[100001]	[100010]	[100100]	[100100]	[001000]	[010000]	[000000]	[000000]	
decode tag = floor([(t+1-L)*total_counts - 1]/[H+1-L]		5	8			6				0	
output		A	E			R				A	
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55	40	17	35	39	23	
	[111111]	[100101]	[100101]	[101011]	[110111]	[101000]	[010001]	[100011]	[100111]	[010111]	
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8	36	8	16	0	0	
	[000000]	[000000]	[011010]	[010100]	[000100]	[100100]	[000100]	[010000]	[000000]	[000000]	
conditions E1 or E2 or E3			E3	E3		E2	E1	E3		E1	

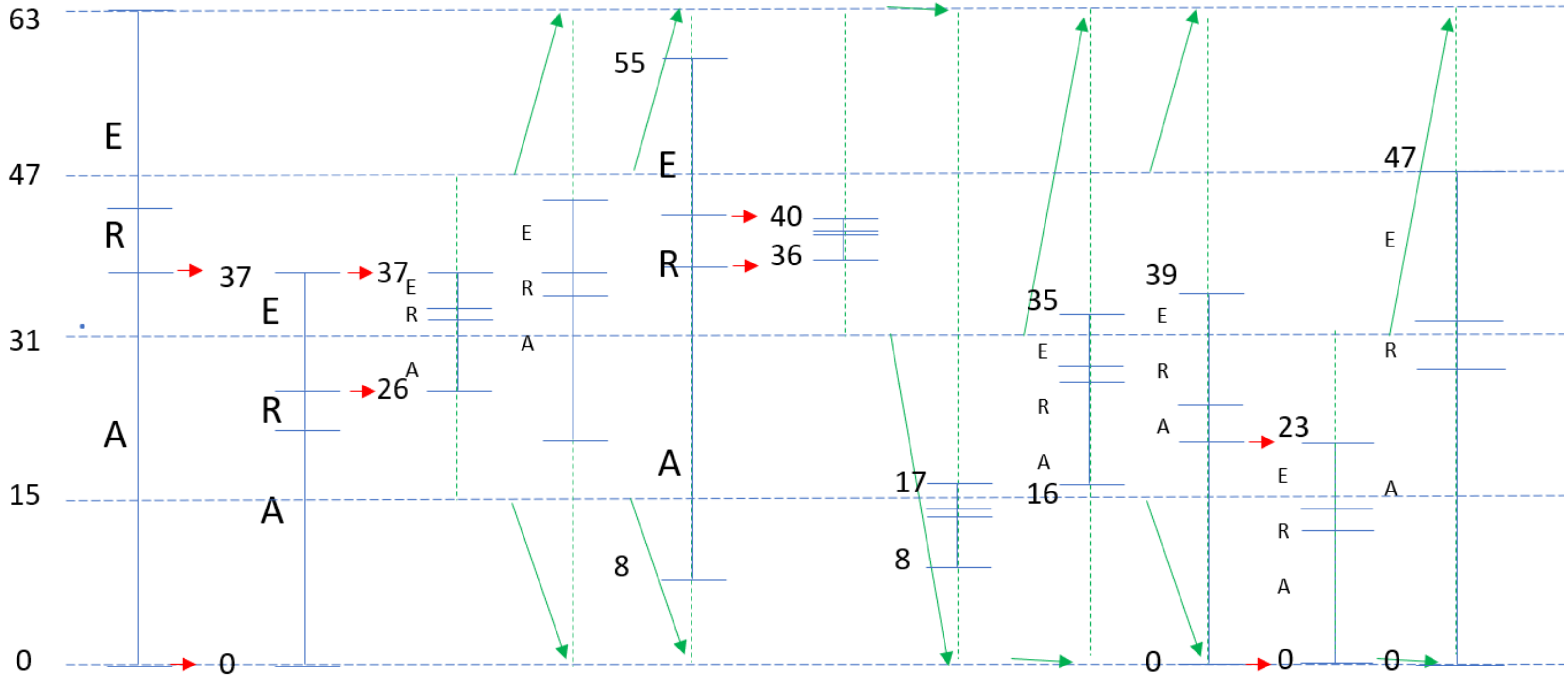


Descodificação Aritmética - inteira

- Código ”100001000000”

	Ocorrencias	cumcount
		0
a	6	6
r	1	7
e	3	10
total	10	

tag		33	33	34	36	36	8	16	0	0	0
		[100001]	[100001]	[100010]	[100100]	[100100]	[001000]	[010000]	[000000]	[000000]	[000000]
decode tag = floor([(t+1-L)*total_counts - 1]/[H+1-L]		5	8			6				0	
output		A	E			R				A	
H = L + floor((H+1-L)*cum_counts(s+1)/total_count) - 1	63	37	37	43	55	40	17	35	39	23	47
	[111111]	[100101]	[100101]	[101011]	[110111]	[101000]	[010001]	[100011]	[100111]	[010111]	[101111]
L = L + floor((H+1-L)*cum_counts(symbol)/total_count)	0	0	26	20	8	36	8	16	0	0	0
	[000000]	[000000]	[011010]	[010100]	[000100]	[100100]	[000100]	[010000]	[000000]	[000000]	[000000]
conditions E1 or E2 or E3			E3	E3		E2	E1	E3		E1	



Codificação Aritmética décimal

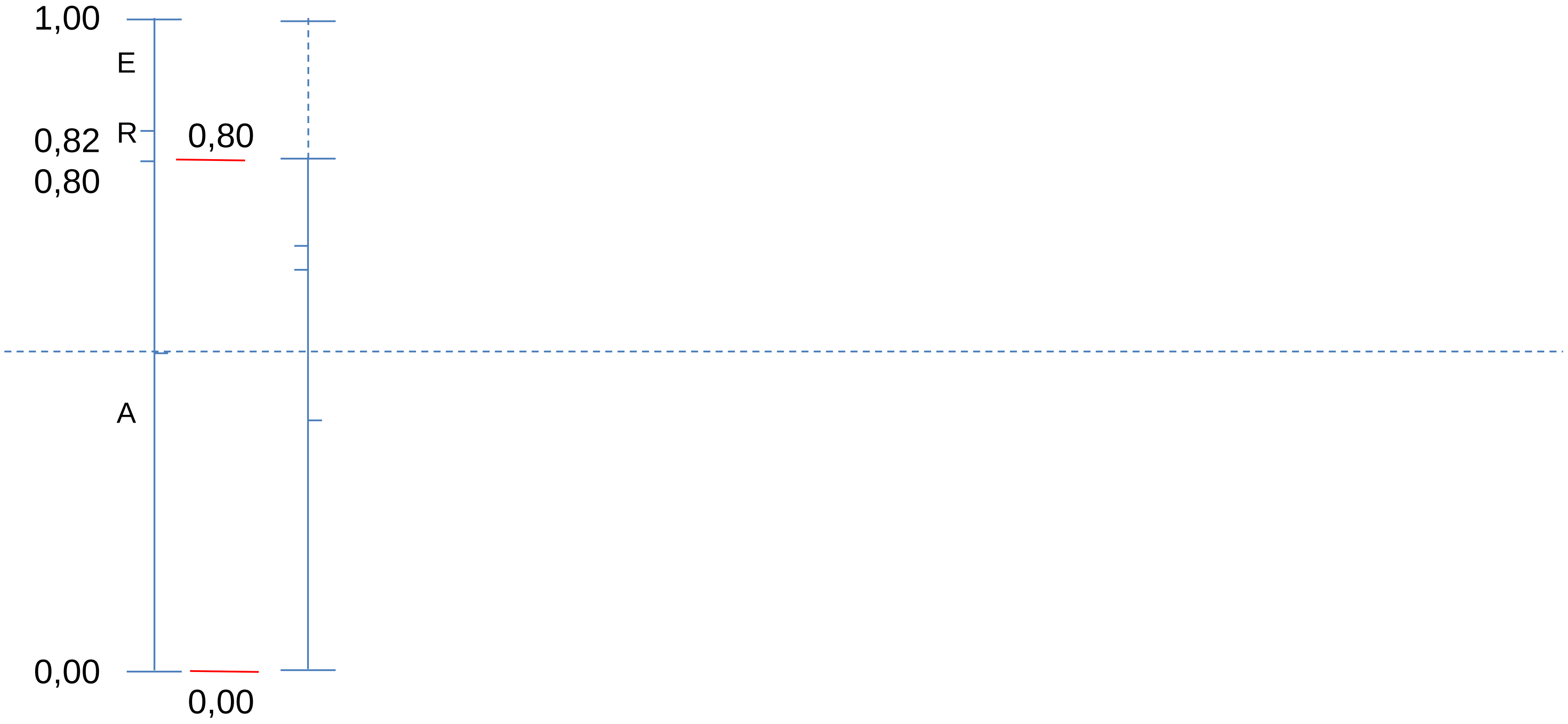
Símbolo	prob.	range
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00

O mesmo exemplo mas com probabilidades diferentes

		low	high	range
		0.0000	1.0000	1.0000
A	0.00-0.80	0.0000	0.8000	0.8000
E				
R				
A				

Codificação Aritmética décimal

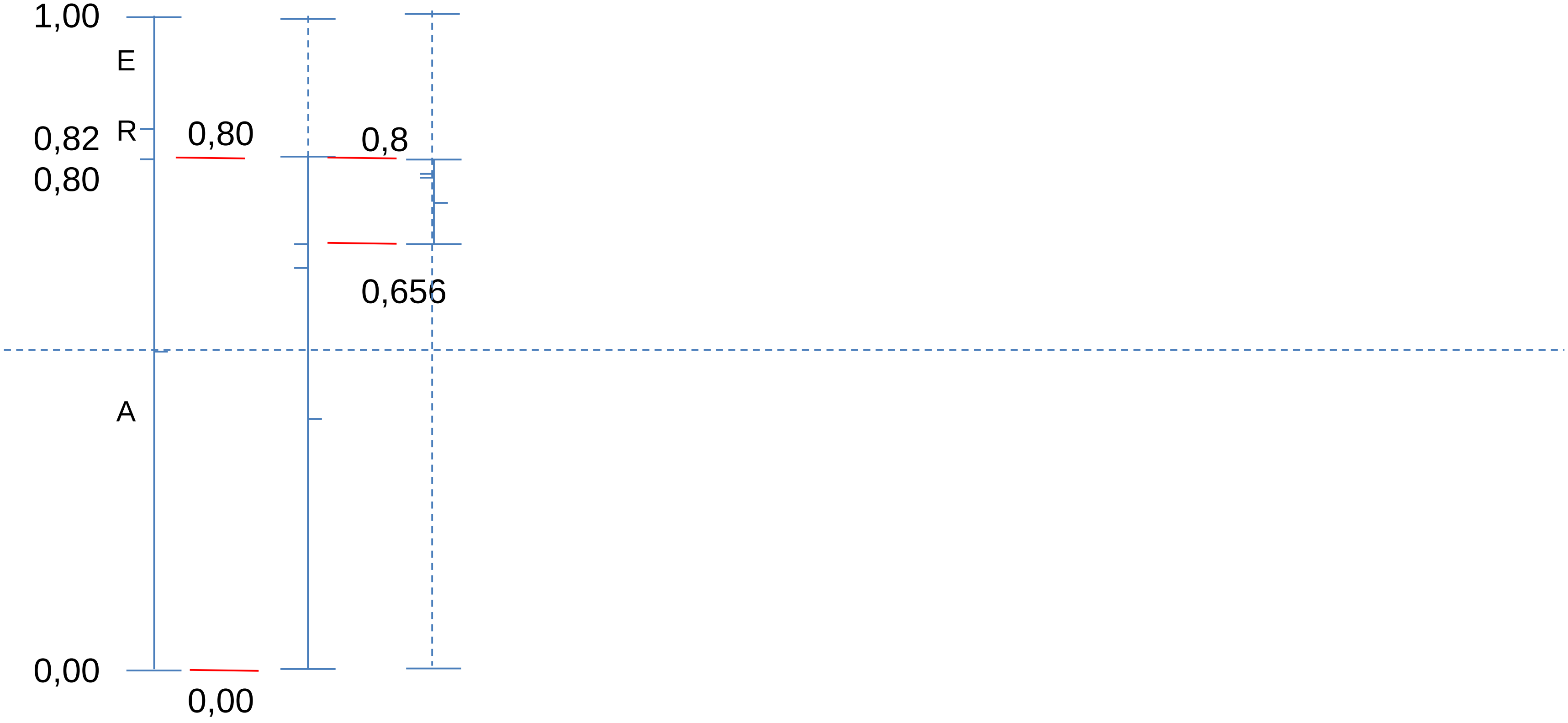
Símbolo	prob.	range
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00



Codificação Aritmética décimal

Símbolo	prob.	range
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00

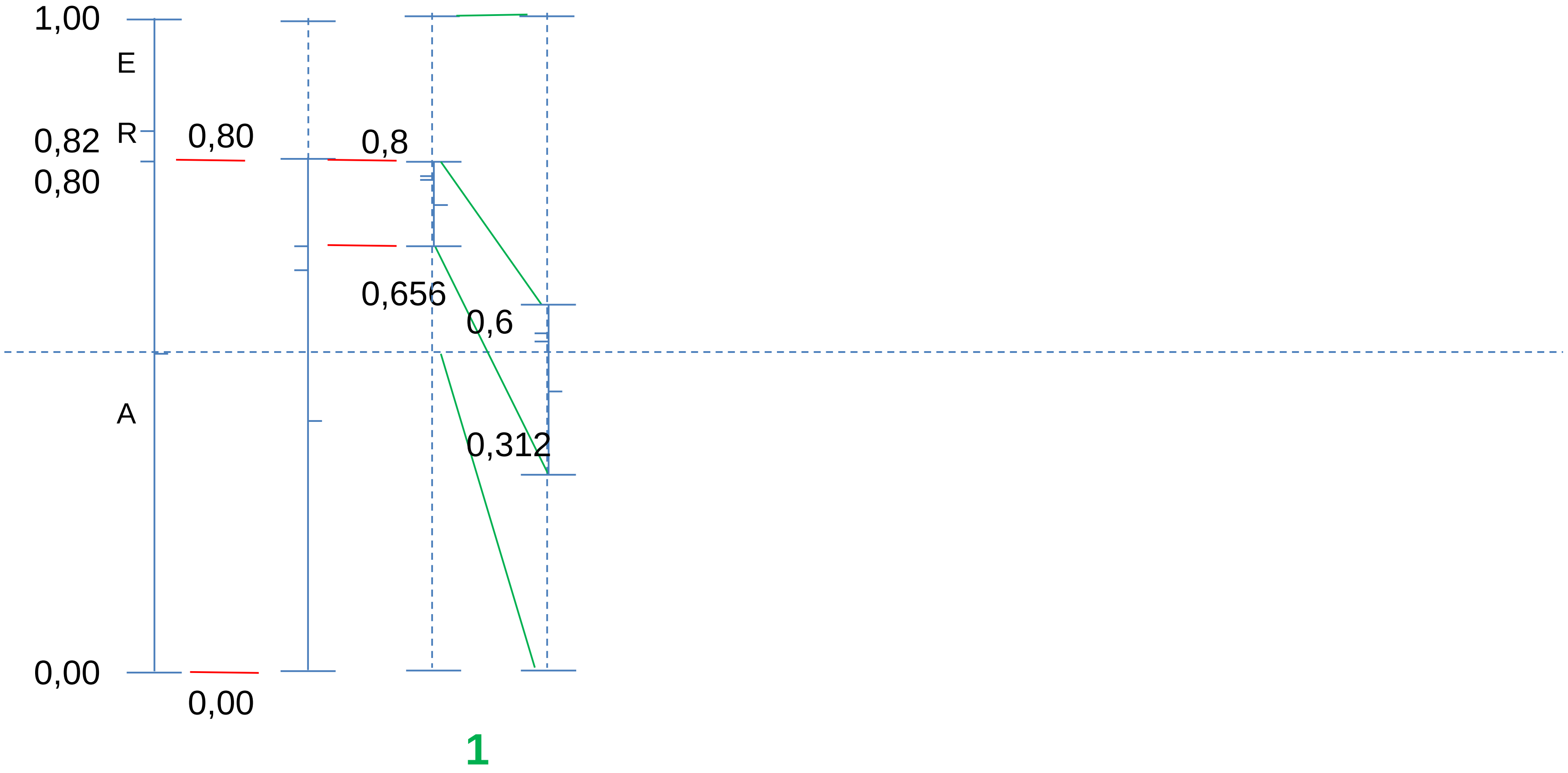
		low	high	range
		0.0000	1.0000	1.0000
A	0.00-0.80	0.0000	0.8000	0.8000
E	0.82-1.00	0.6560	0.8000	0.1440
R				
A				



Codificação Aritmética décimal

Símbolo	prob.	range
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00

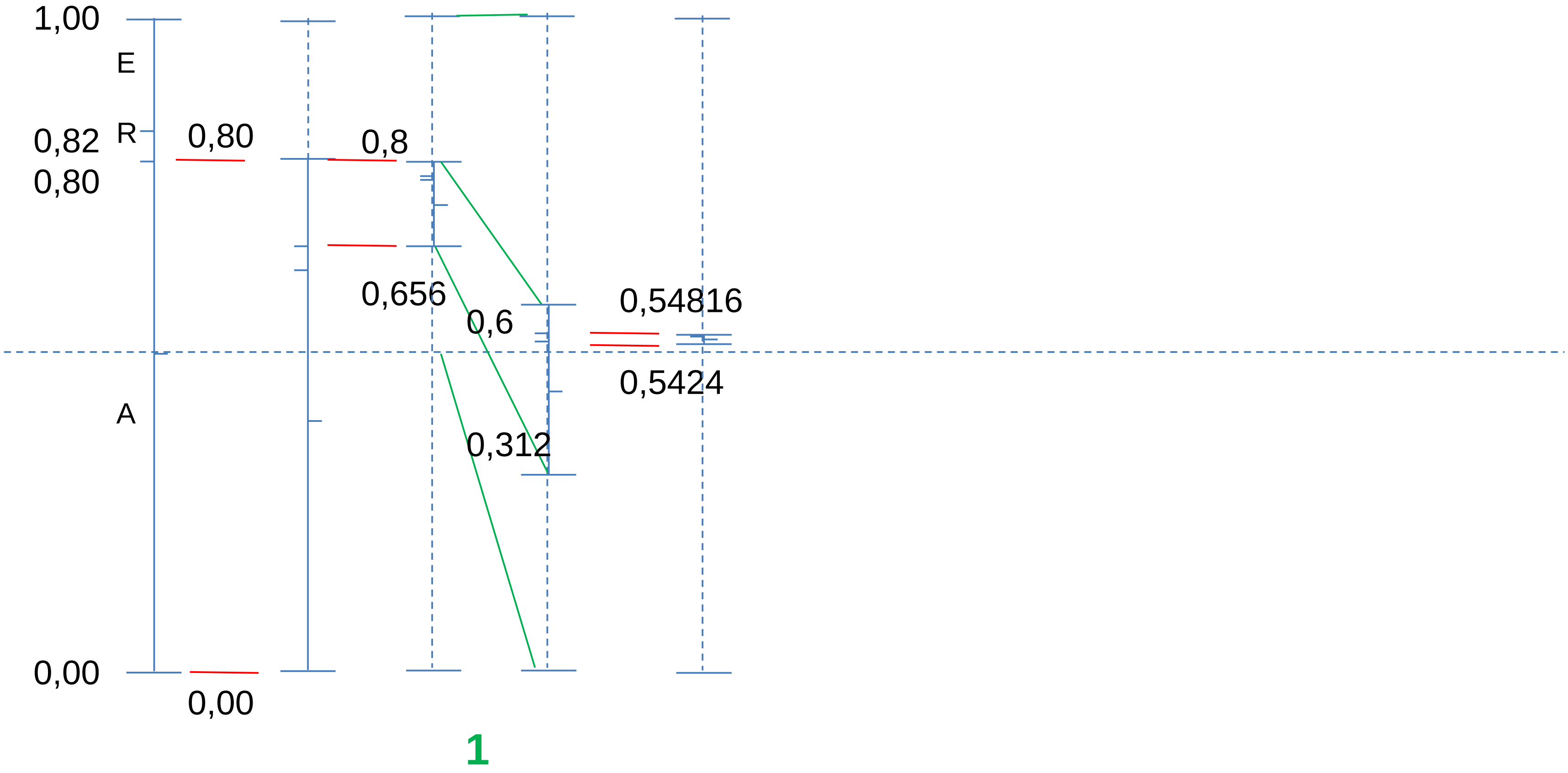
		low	high	range
		0.0000	1.0000	1.0000
A	0.00-0.80	0.0000	0.8000	0.8000
E	0.82-1.00	0.6560	0.8000	0.1440
		0.3120	0.6000	0.2880
R				
A				



Codificação Aritmética décimal

Símbolo	prob.	range
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00

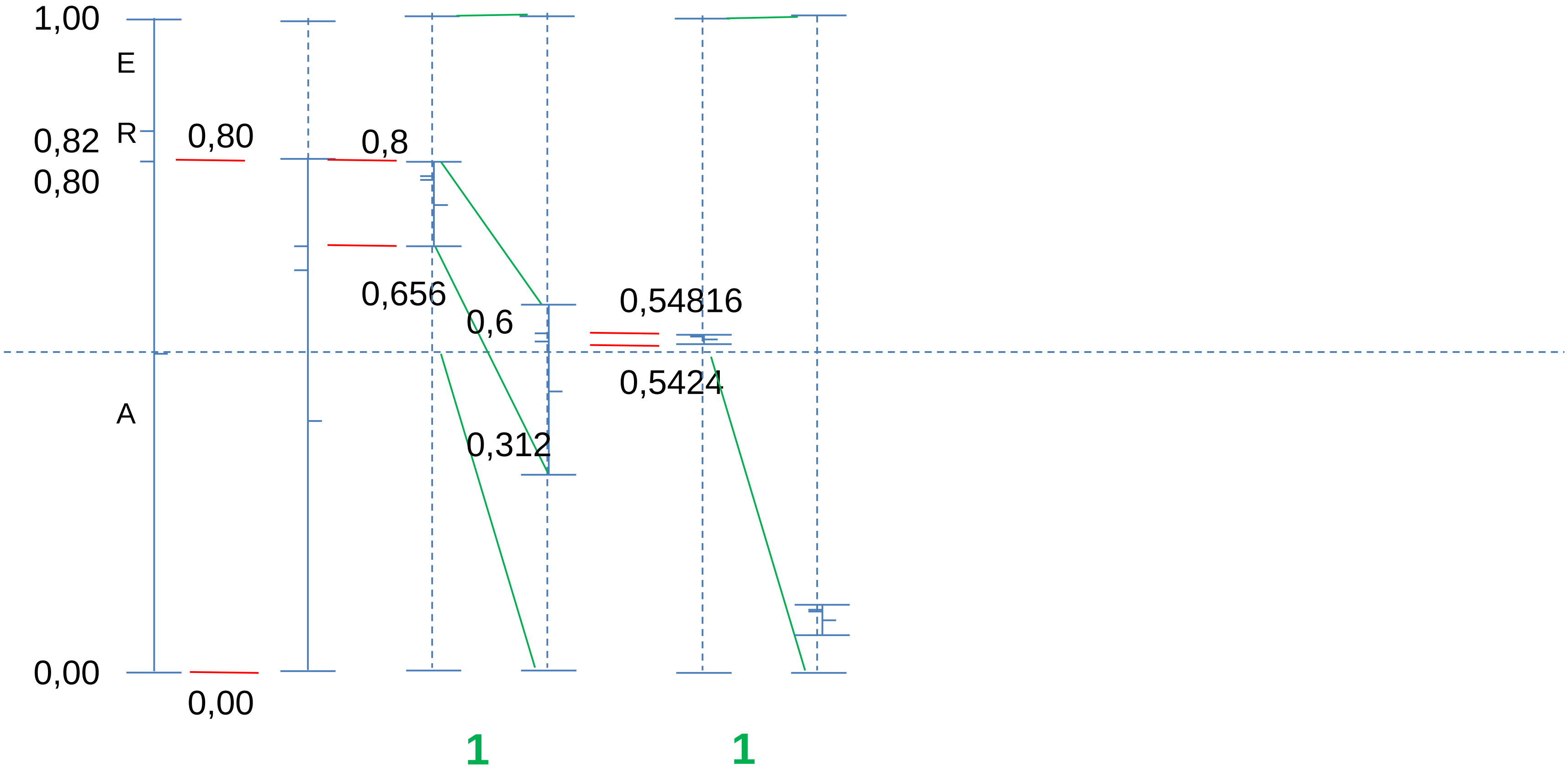
		low	high	range
		0.0000	1.0000	1.0000
A	0.00-0.80	0.0000	0.8000	0.8000
E	0.82-1.00	0.6560	0.8000	0.1440
		0.3120	0.6000	0.2880
R	0.8-0.82	0.5424	0.54816	
A				



Codificação Aritmética décimal

Símbolo	prob.	range
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00

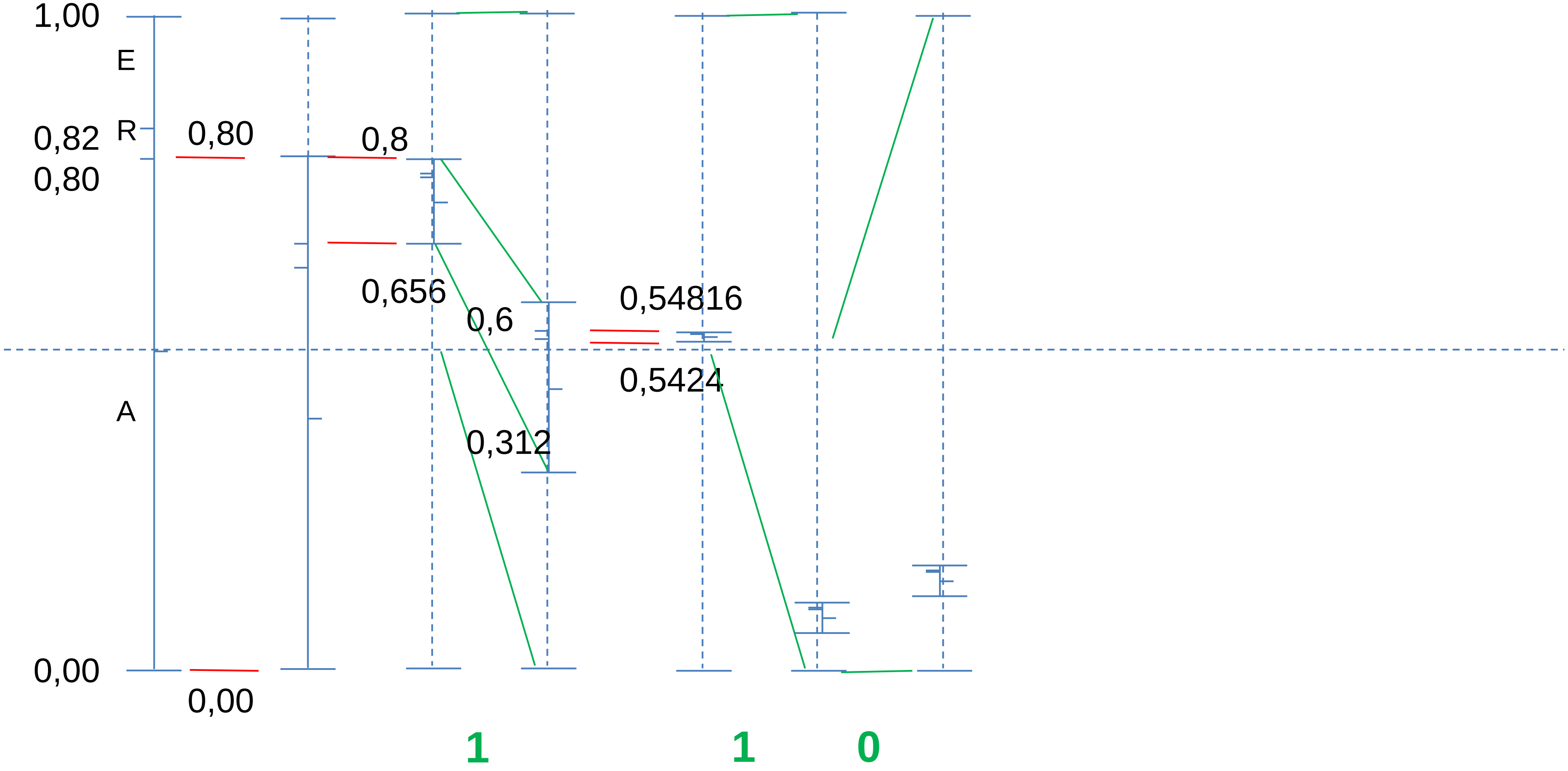
		low	high	range
		0.0000	1.0000	1.0000
A	0.00-0.80	0.0000	0.8000	0.8000
E	0.82-1.00	0.6560	0.8000	0.1440
		0.3120	0.6000	0.2880
R	0.8-0.82	0.5424	0.54816	
		0.0848	0.09632	
A				



Codificação Aritmética décimal

Símbolo	prob.	range
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00

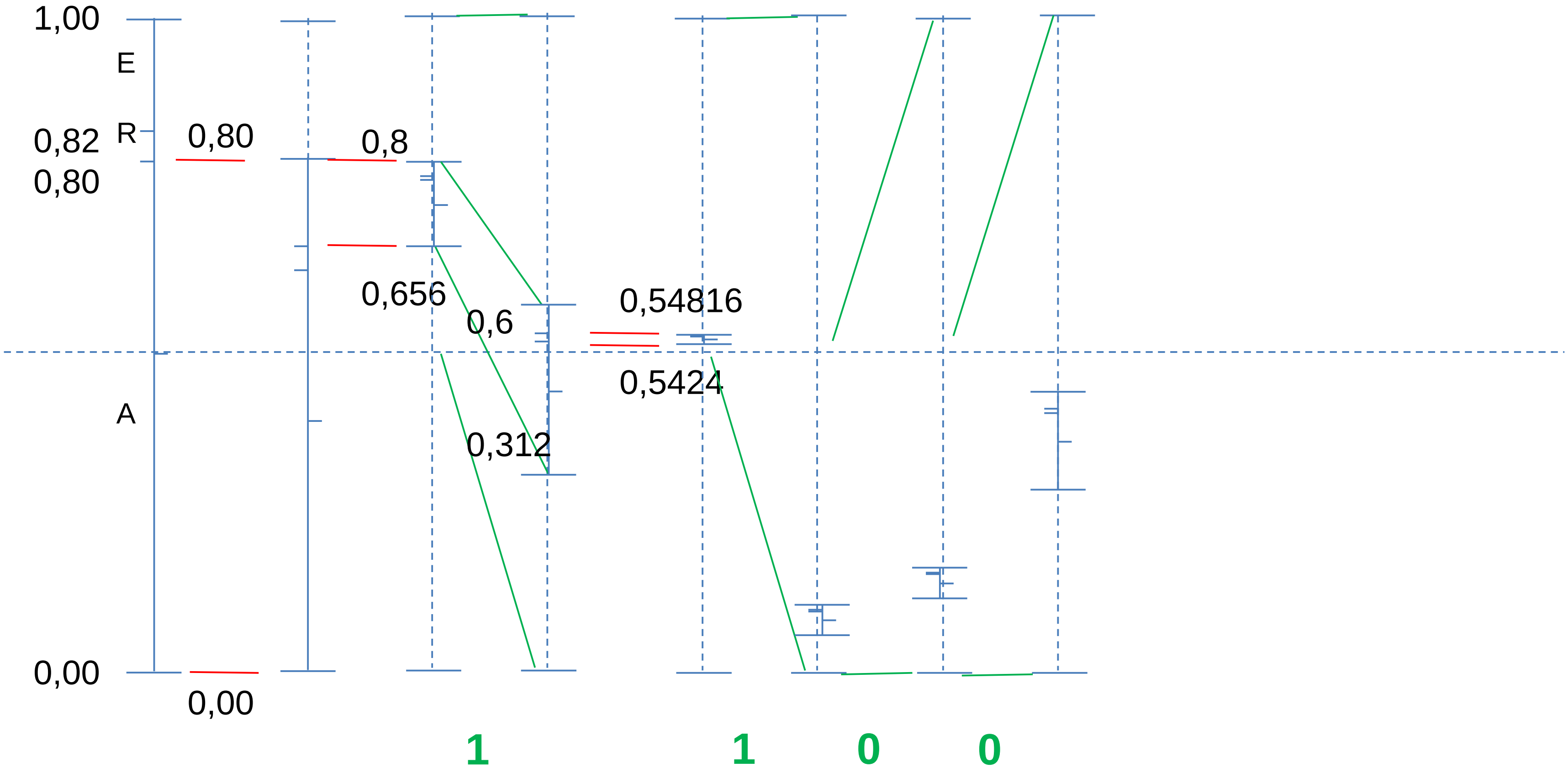
		low	high	range
		0.0000	1.0000	1.0000
A	0.00-0.80	0.0000	0.8000	0.8000
E	0.82-1.00	0.6560	0.8000	0.1440
		0.3120	0.6000	0.2880
R	0.8-0.82	0.5424	0.54816	
		0.0848	0.09632	
		0.1696	0.19264	
A				



Codificação Aritmética décimal

Símbolo	prob.	range
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00

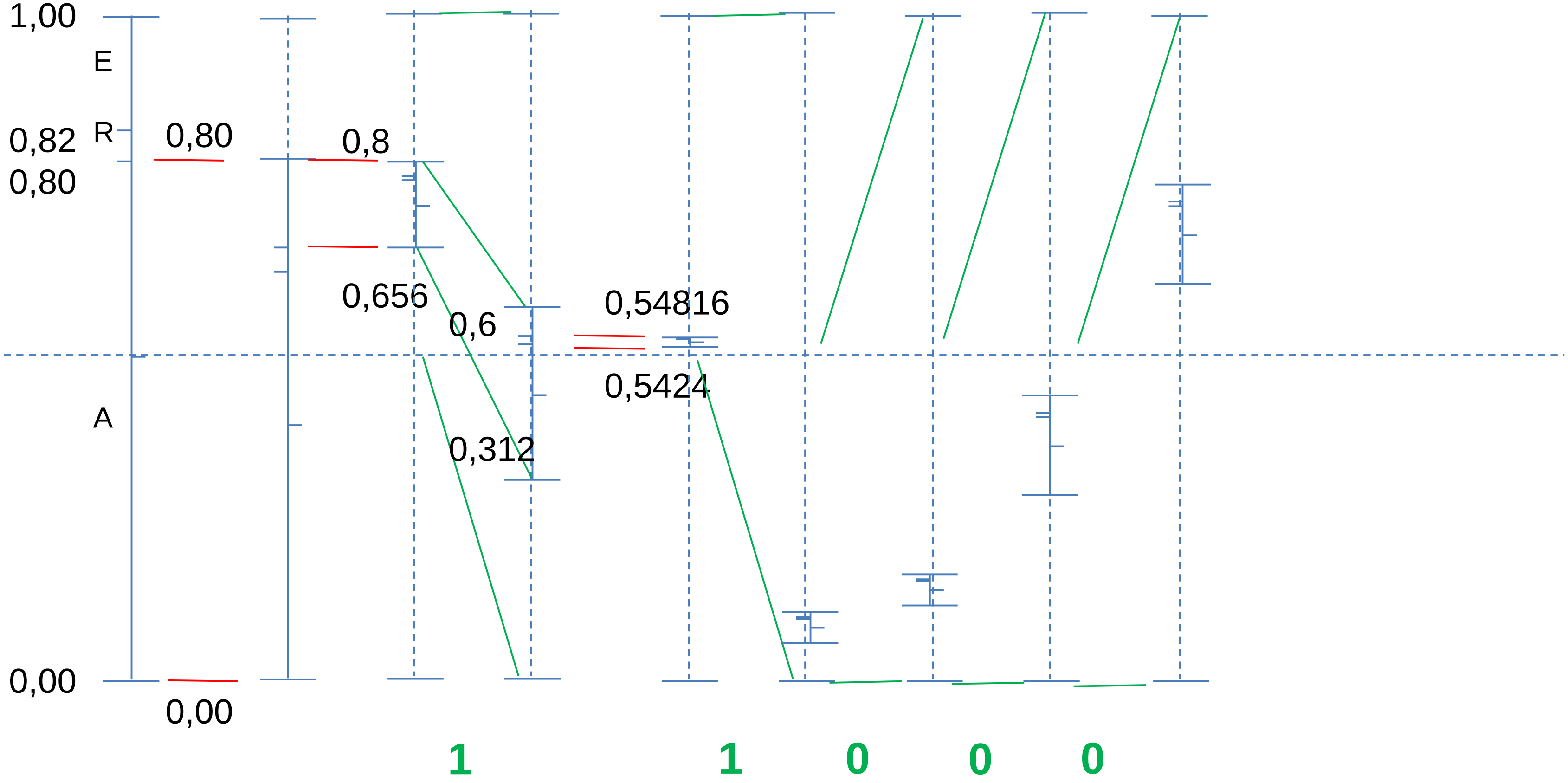
		low	high	range
		0.0000	1.0000	1.0000
A	0.00-0.80	0.0000	0.8000	0.8000
E	0.82-1.00	0.6560	0.8000	0.1440
		0.3120	0.6000	0.2880
R	0.8-0.82	0.5424	0.54816	
		0.0848	0.09632	
		0.1696	0.19264	
		0.3392	0.38528	
A				



Codificação Aritmética décimal

Símbolo	prob.	range
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00

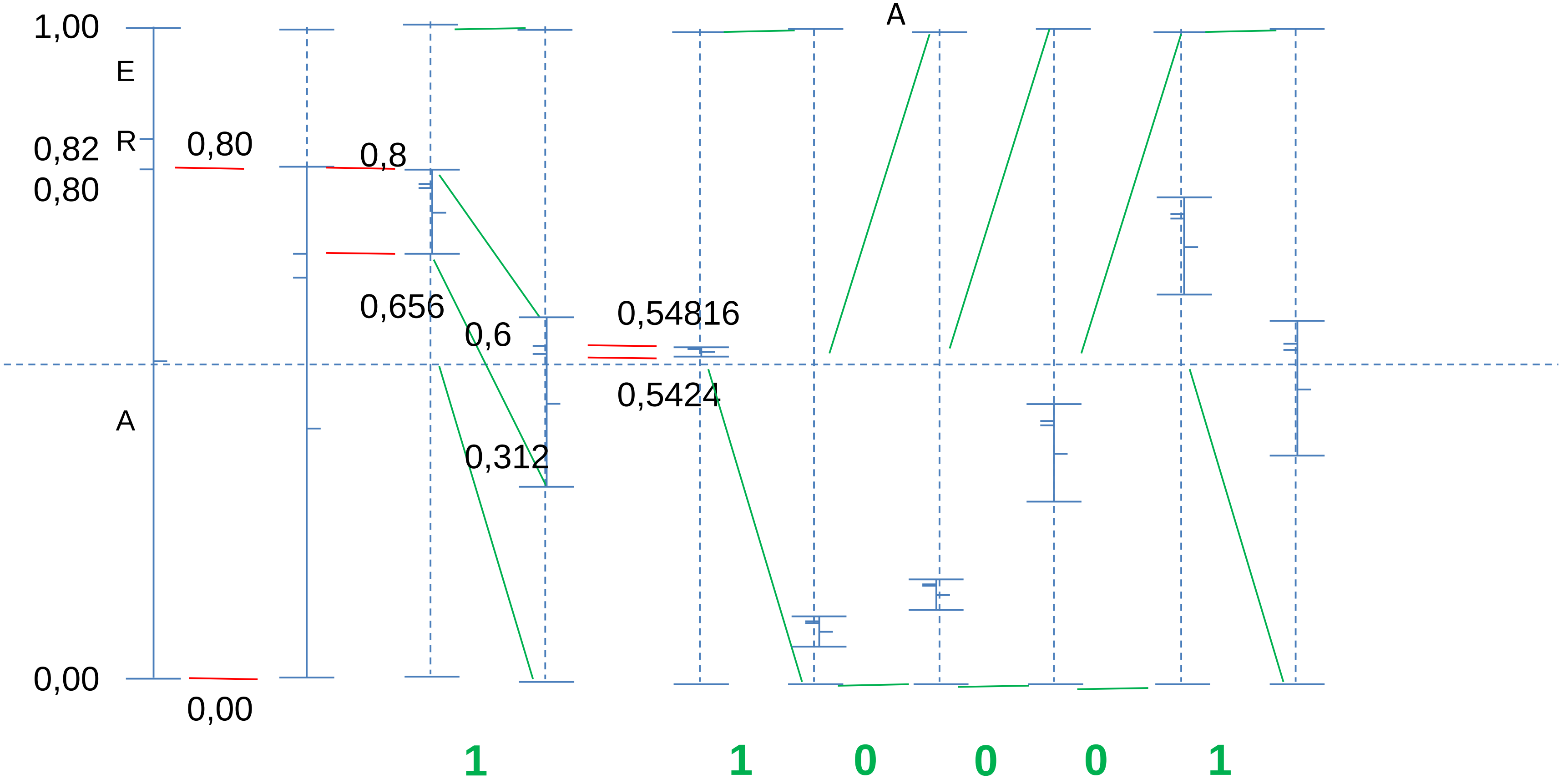
		low	high	range
		0.0000	1.0000	1.0000
A	0.00-0.80	0.0000	0.8000	0.8000
E	0.82-1.00	0.6560	0.8000	0.1440
		0.3120	0.6000	0.2880
R	0.8-0.82	0.5424	0.54816	
		0.0848	0.09632	
		0.1696	0.19264	
		0.3392	0.38528	
		0.6784	0.77056	
A				



Codificação Aritmética décimal

Símbolo	prob.	range
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00

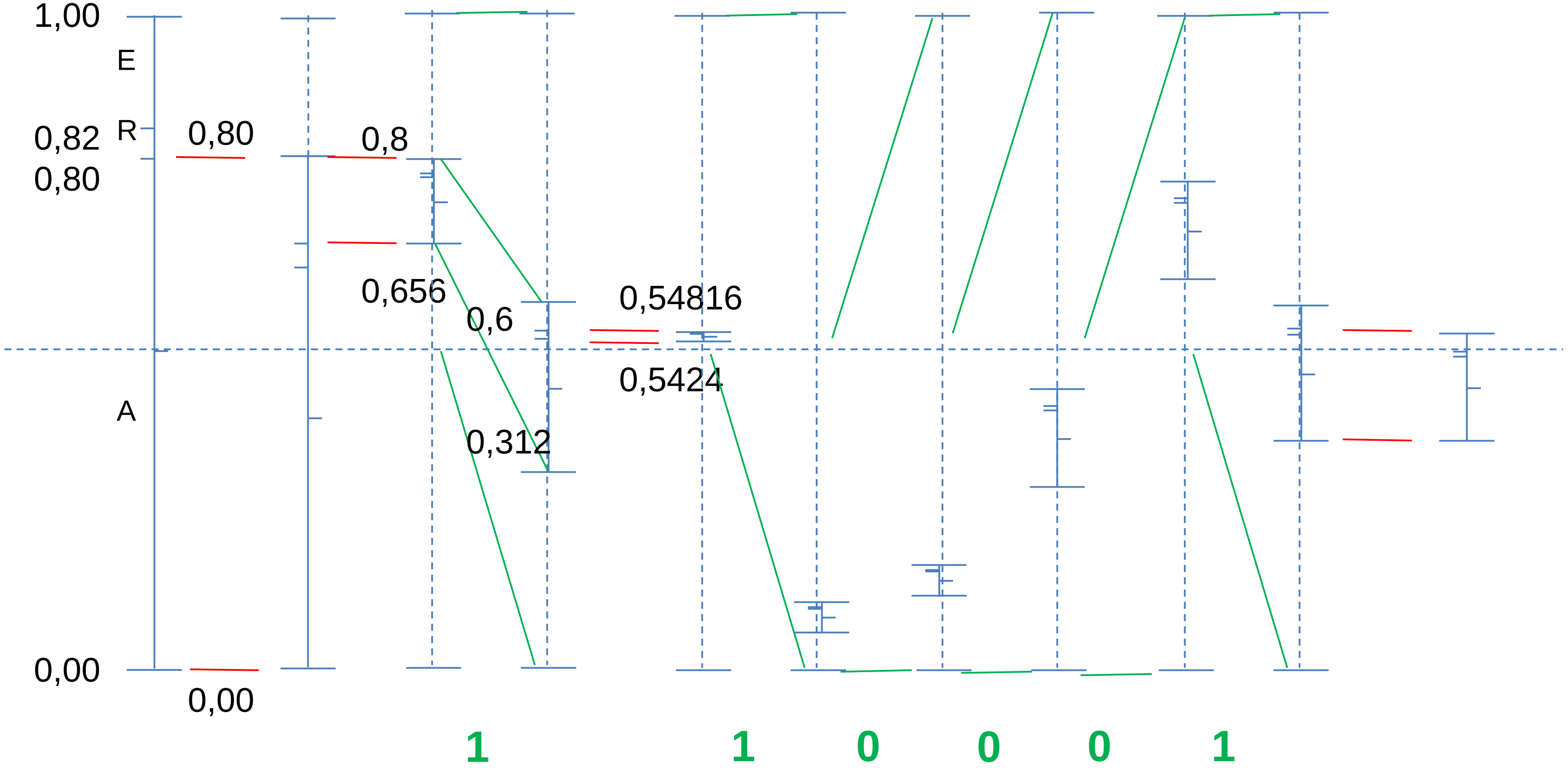
		low	high	range
		0.0000	1.0000	1.0000
A	0.00-0.80	0.0000	0.8000	0.8000
E	0.82-1.00	0.6560	0.8000	0.1440
		0.3120	0.6000	0.2880
R	0.8-0.82	0.5424	0.54816	
		0.0848	0.09632	
		0.1696	0.19264	
		0.3392	0.38528	
		0.6784	0.77056	
		0.3568	0.54112	0.18422
A				



Codificação Aritmética décimal

Símbolo	prob.	range
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00

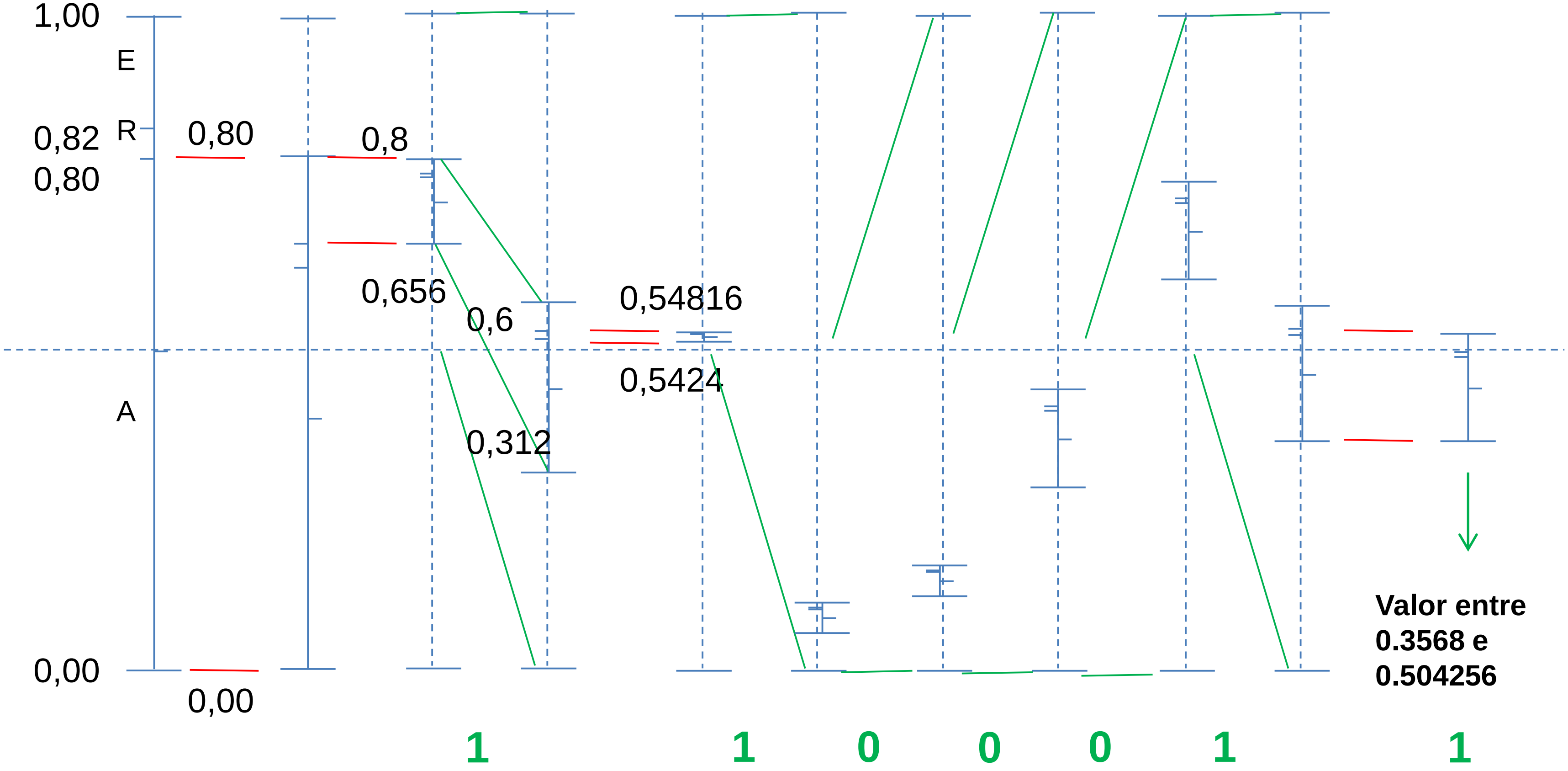
		low	high	range
		0.0000	1.0000	1.0000
A	0.00-0.80	0.0000	0.8000	0.8000
E	0.82-1.00	0.6560	0.8000	0.1440
		0.3120	0.6000	0.2880
R	0.8-0.82	0.5424	0.54816	
		0.0848	0.09632	
		0.1696	0.19264	
		0.3392	0.38528	
		0.6784	0.77056	
		0.3568	0.54112	0.18422
A	0.00-0.80	0.3568	0.504256	



Codificação Aritmética décimal

Símbolo	prob.	range
A	0.80	0.00 - 0.80
R	0.02	0.80 - 0.82
E	0.18	0.82 - 1.00

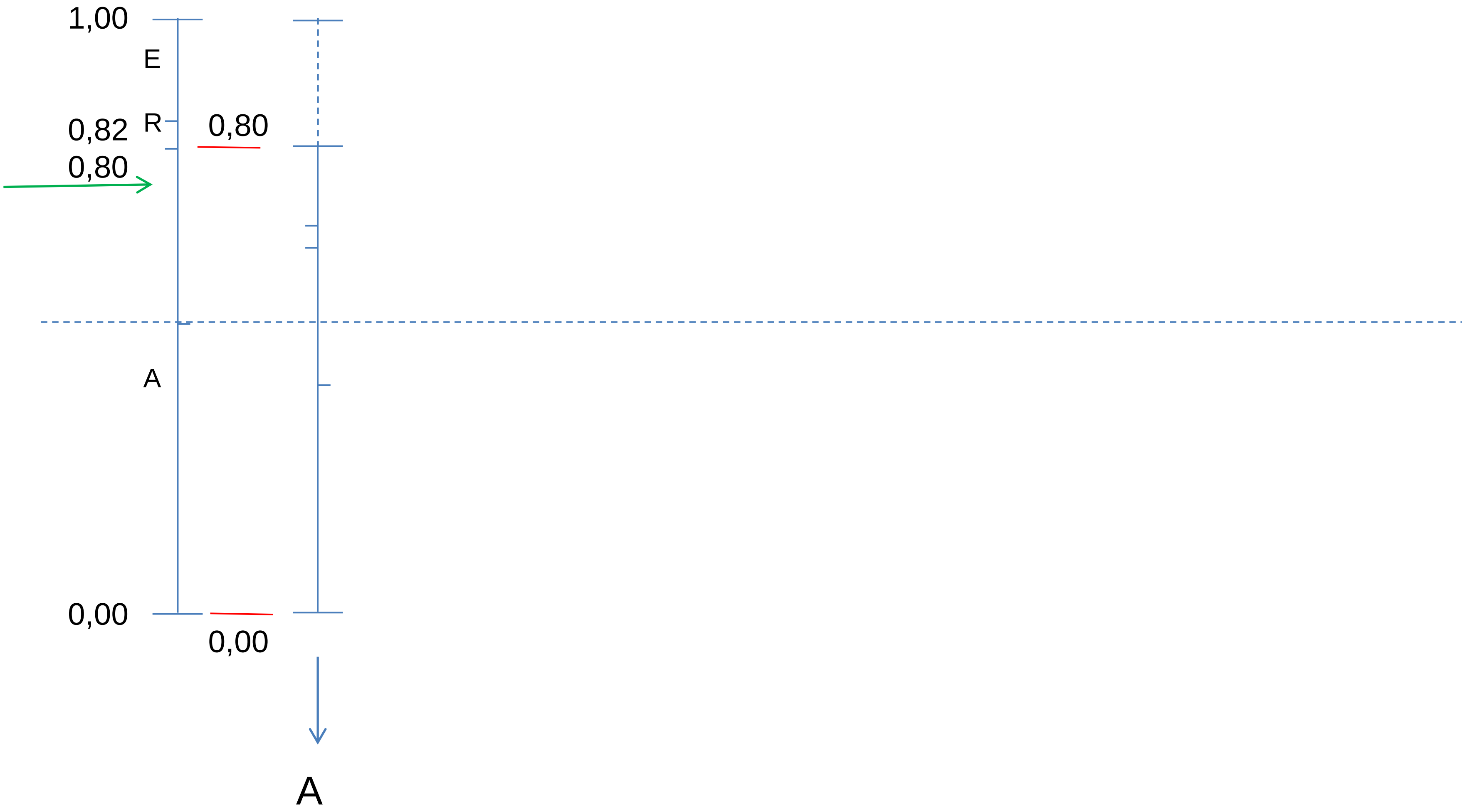
		low	high	range
		0.0000	1.0000	1.0000
A	0.00-0.80	0.0000	0.8000	0.8000
E	0.82-1.00	0.6560	0.8000	0.1440
		0.3120	0.6000	0.2880
R	0.8-0.82	0.5424	0.54816	
		0.0848	0.09632	
		0.1696	0.19264	
		0.3392	0.38528	
		0.6784	0.77056	
		0.3568	0.54112	0.18422
A	0.00-0.80	0.3568	0.504256	



Valor entre 0.3568 e 0.504256

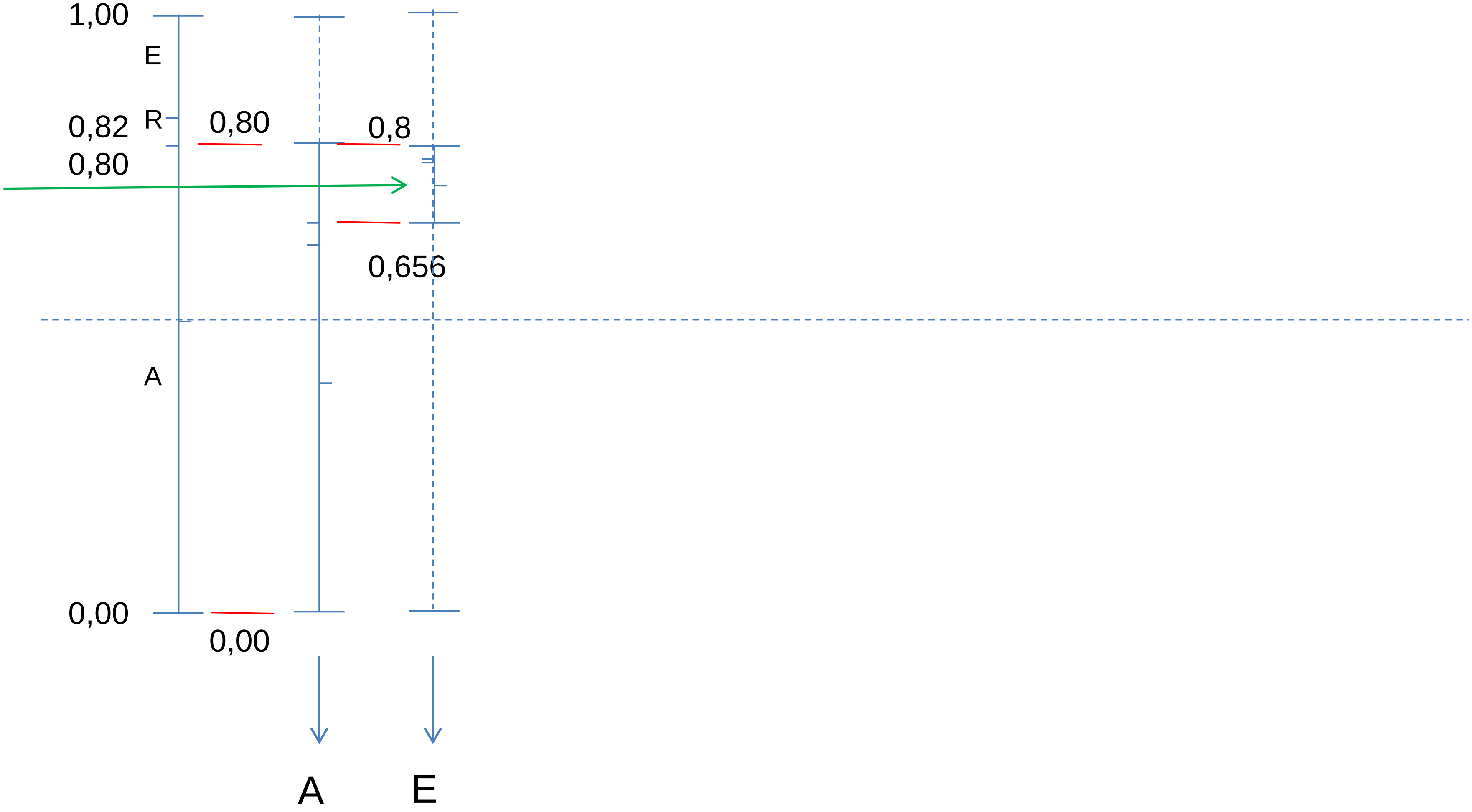
Descodificação Aritmética décimal

- Code: 1 1 0 0 0 1 1 = 0,7734375



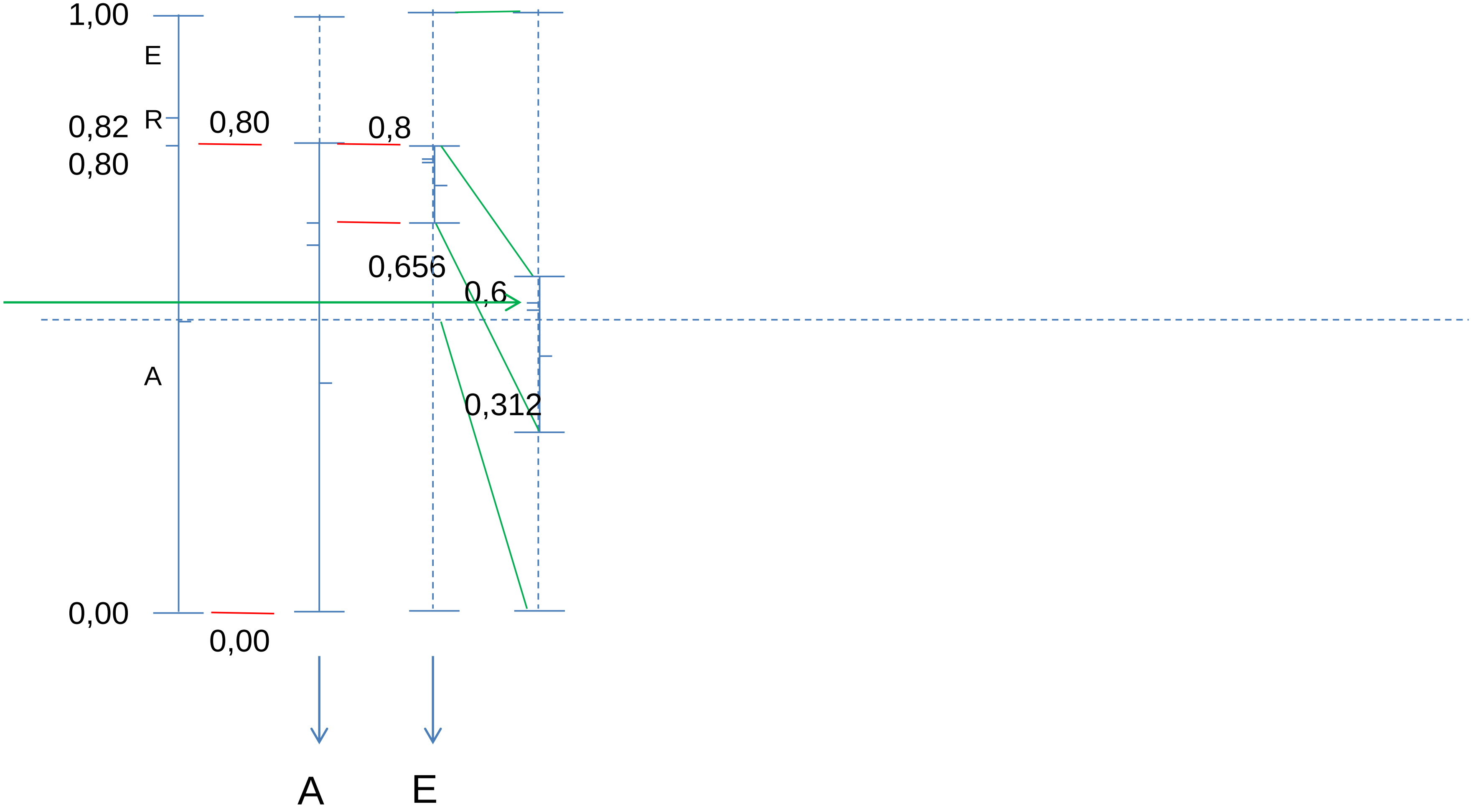
Descodificação Aritmética décimal

- Code: 1 1 0 0 0 1 1 = 0,7734375



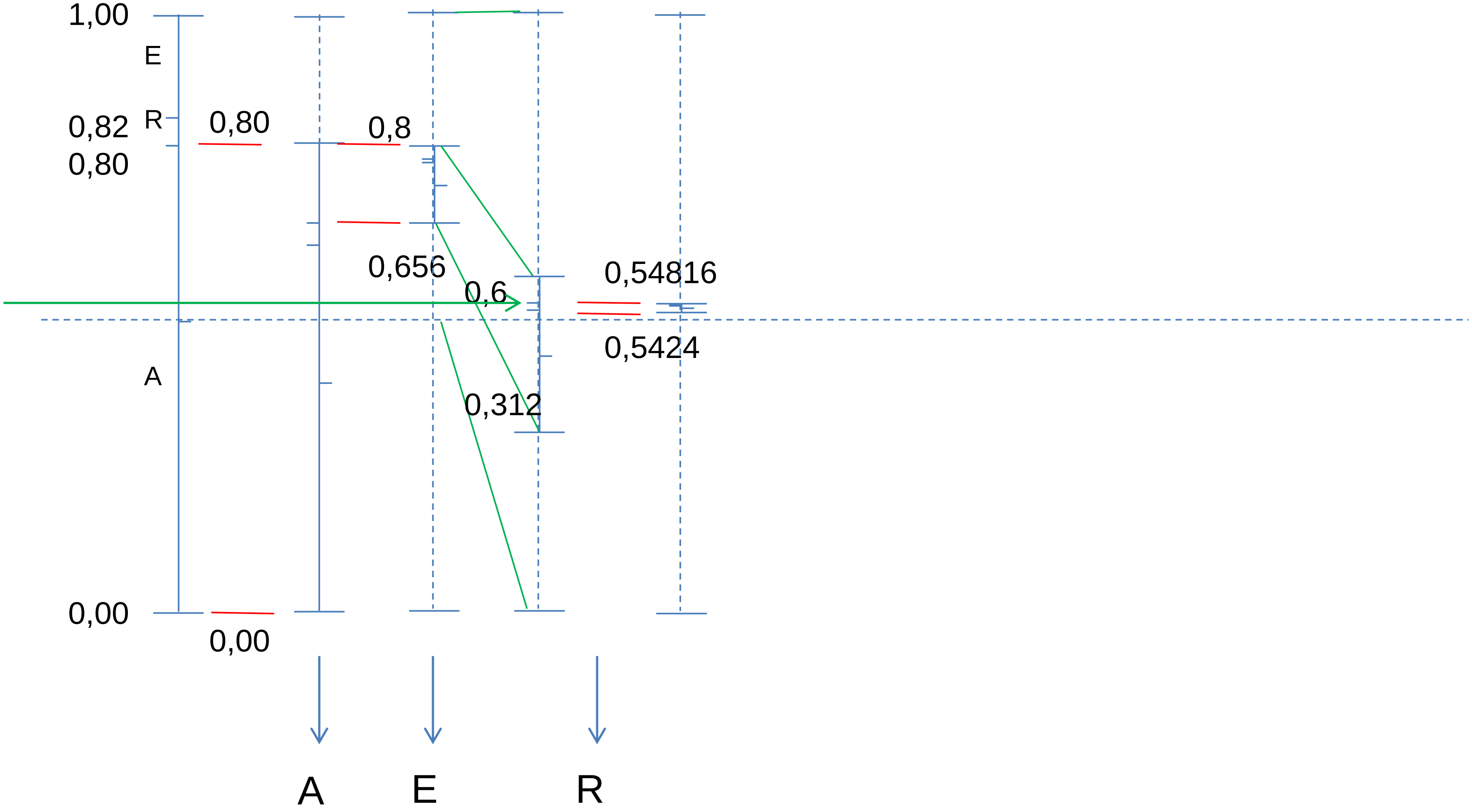
Descodificação Aritmética décimal

- Code: 1 0 0 0 1 1 = 0,546875



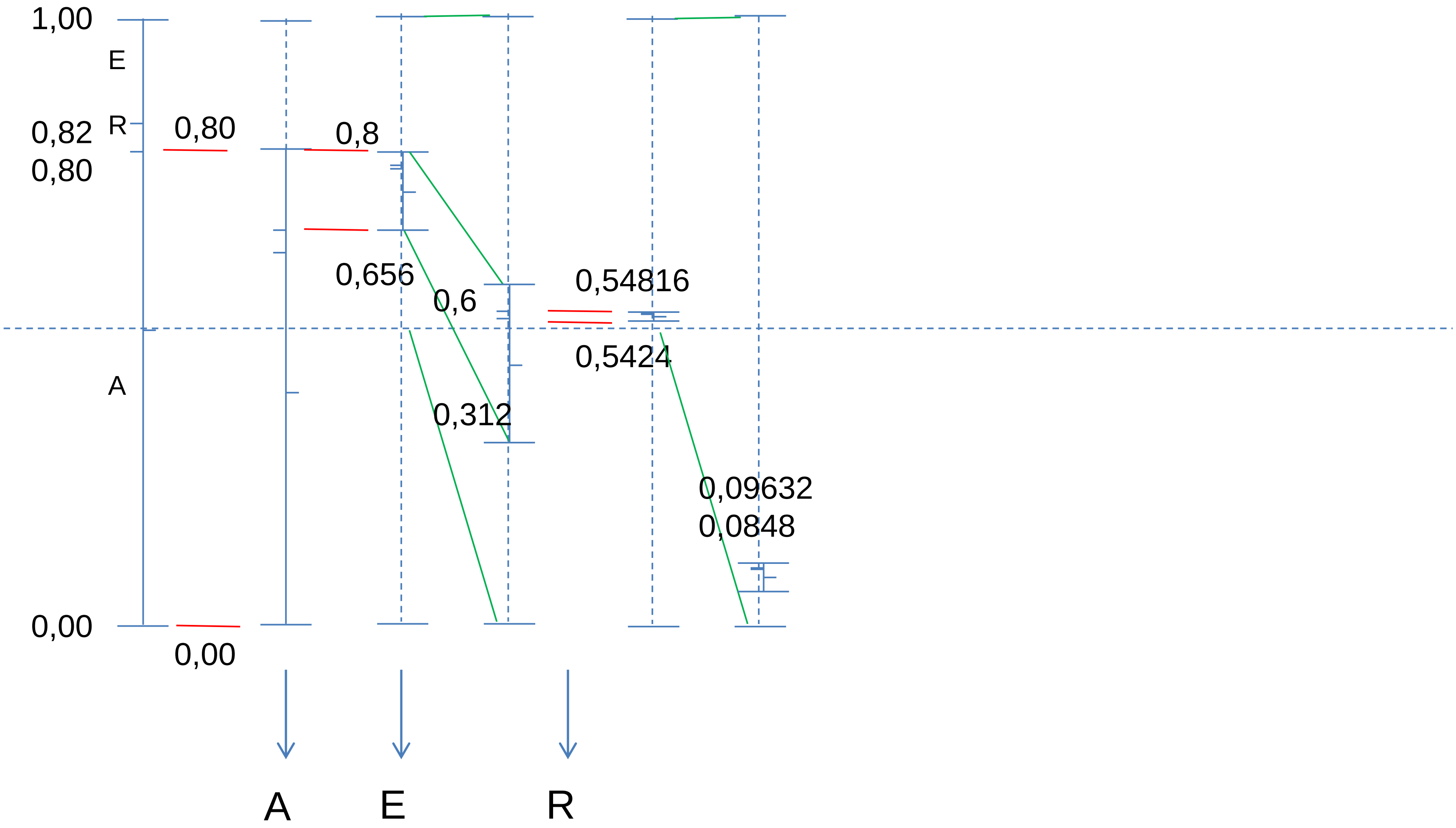
Descodificação Aritmética décimal

- Code: 1 0 0 0 1 1 = 0,546875



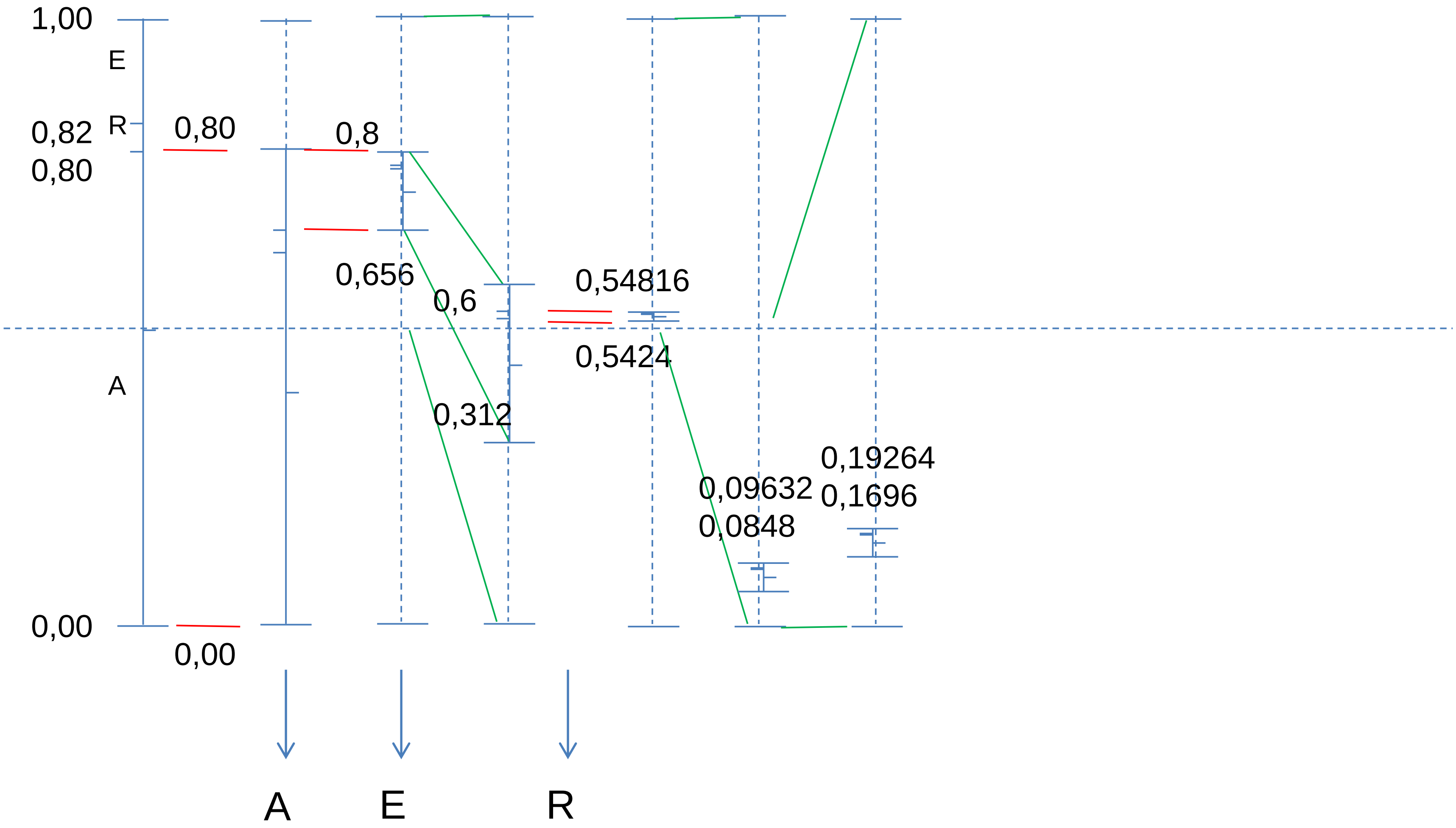
Descodificação Aritmética décimal

- Code: 0 0 0 1 1 = 0,09375



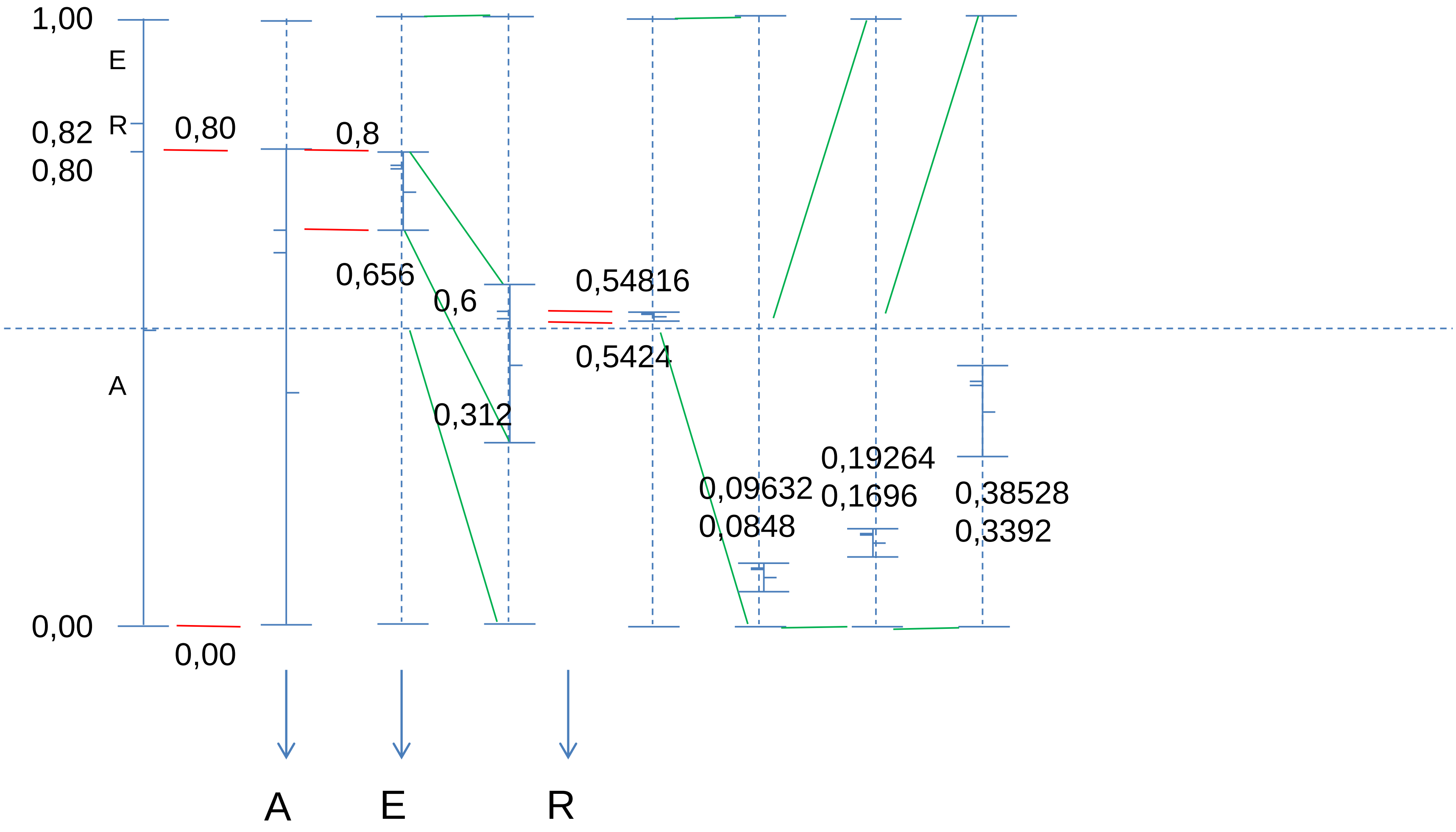
Descodificação Aritmética décimal

- Code: 0 0 1 1 = 0,1875



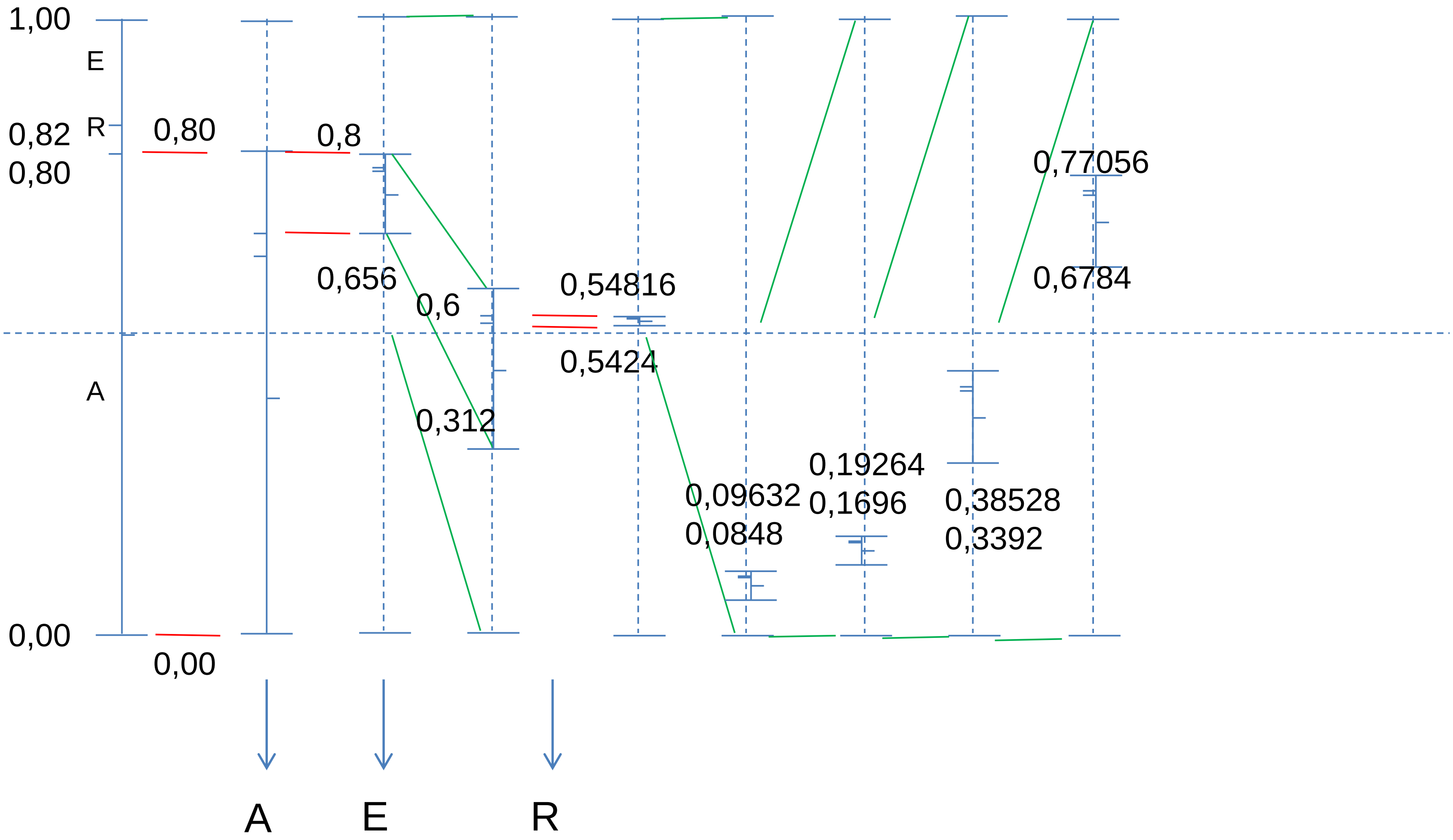
Descodificação Aritmética décimal

- Code: 0 1 1 = 0,375



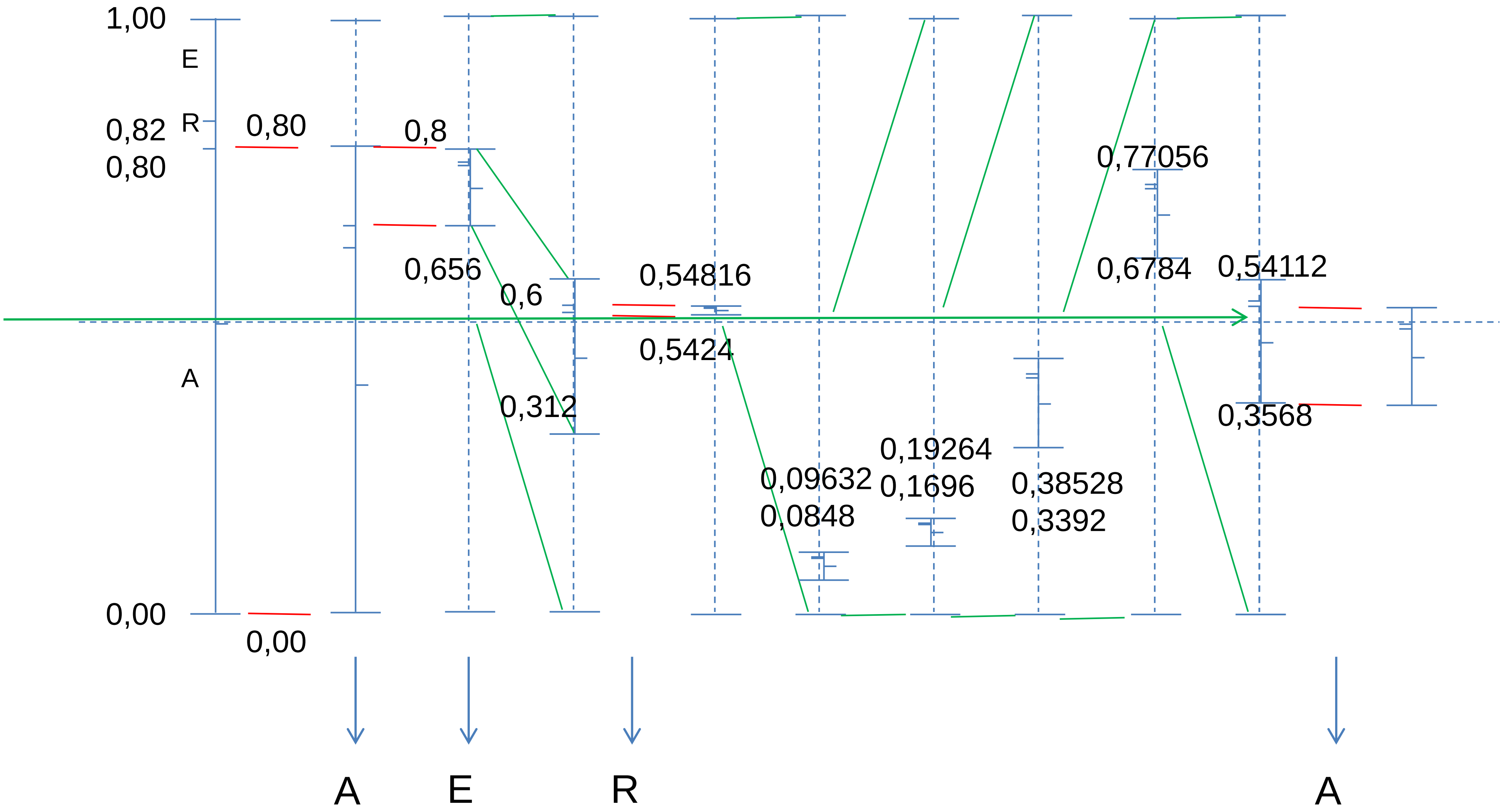
Descodificação Aritmética décimal

- Code: 1 1 = 0,75



Descodificação Aritmética décimal

- Code: ~~[illegible]~~ 1 = 0,5



Codificação Aritmética - inteira

- Assumindo a tabela de occorrências:
- A sequência a codificar: ”aera…”

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

- $m = 2 + \lceil log_2(totalcount) \rceil$ = 8
- Low = 0 [00000000]
- High = 255 [11111111]

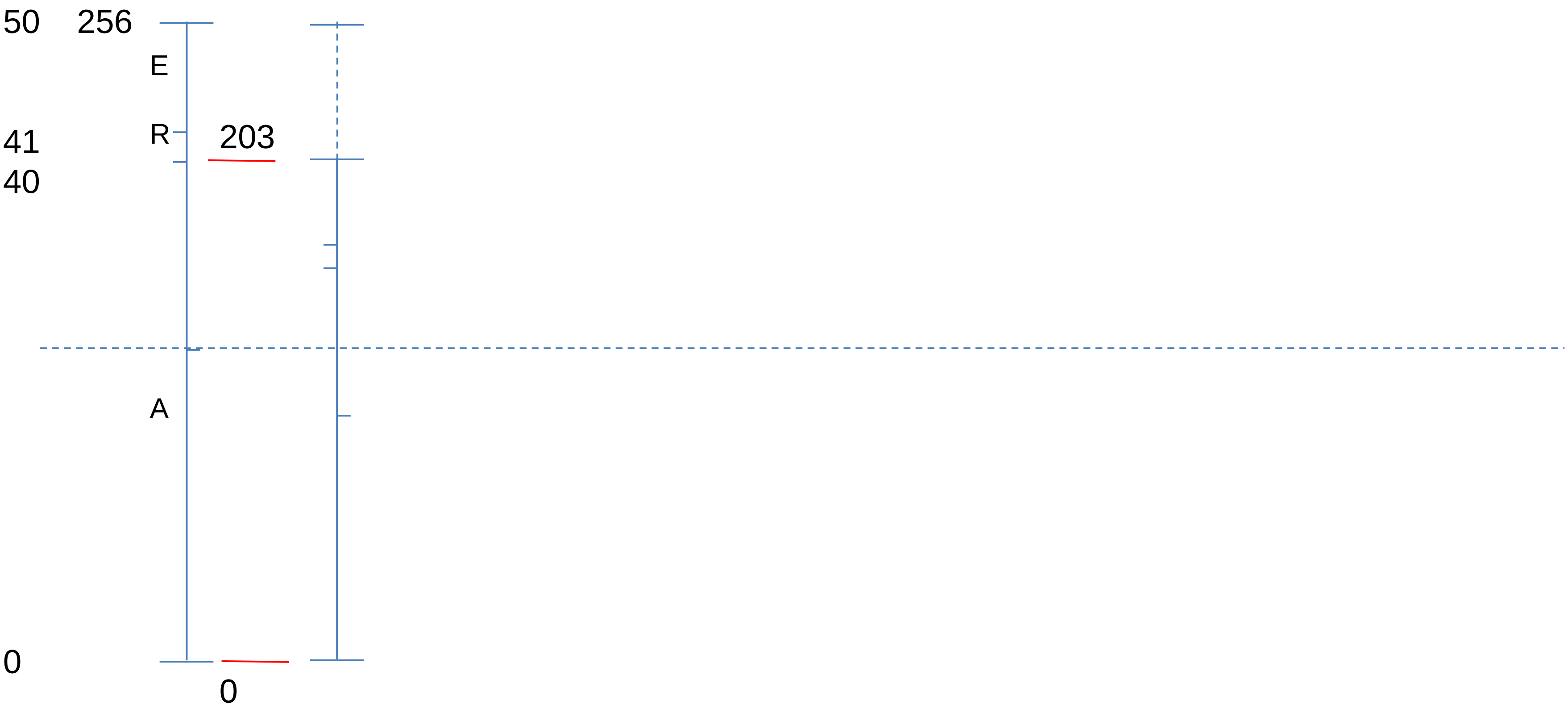
Codificação Aritmética décimal

- Codificar “a”

$$low \quad = \quad 0 + \left\lfloor \frac{256 \times 0}{50} \right\rfloor = 0 \ (00000000)$$

$$high \quad = \quad 0 + \left\lfloor \frac{256 \times 40}{50} \right\rfloor - 1 = 203 \ (11001011)$$

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	



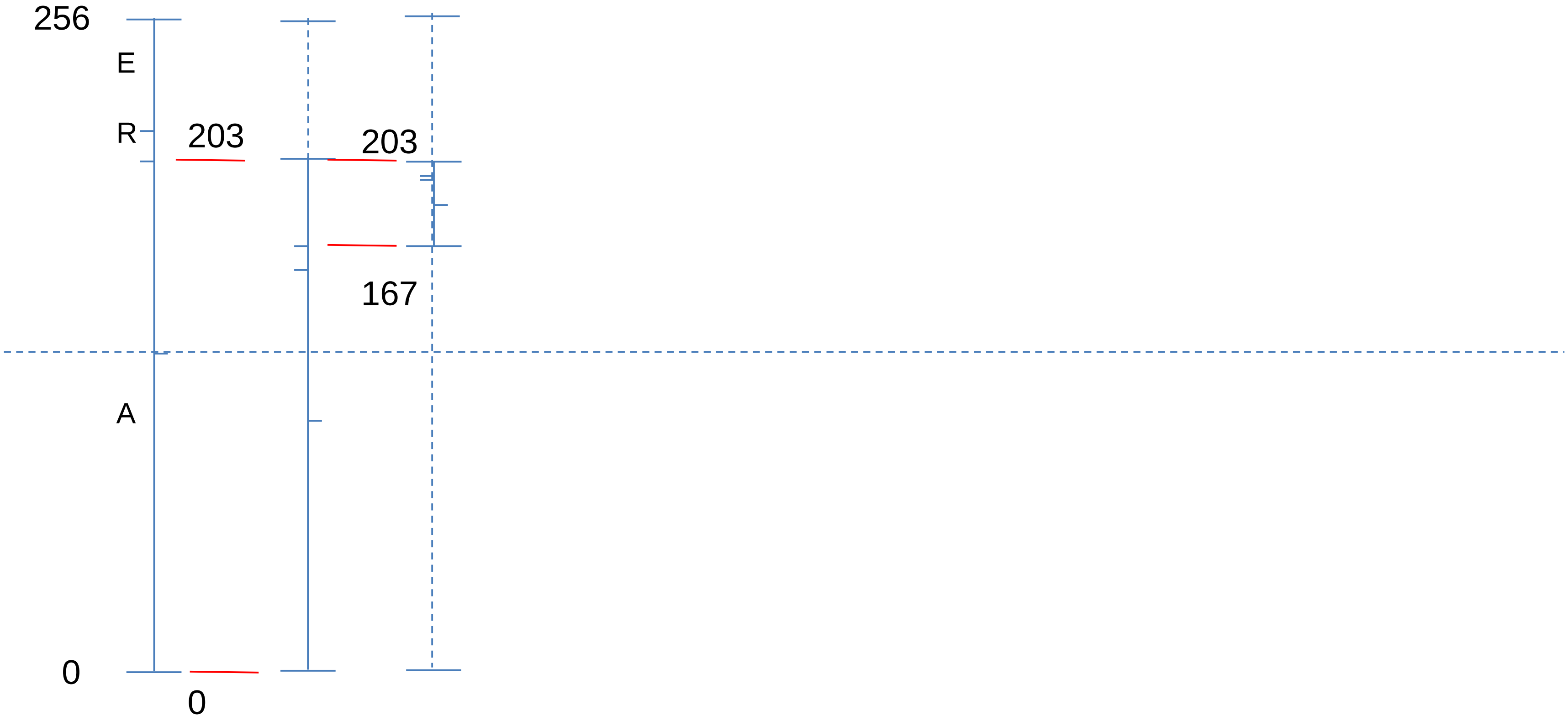
Codificação Aritmética décimal

- Codificar “e”

$$low \quad = \quad 0 + \left\lfloor \frac{204 \times 41}{50} \right\rfloor = 167 \ (10100111)$$

$$high \quad = \quad 0 + \left\lfloor \frac{204 \times 50}{50} \right\rfloor - 1 = 203 \ (11001011)$$

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

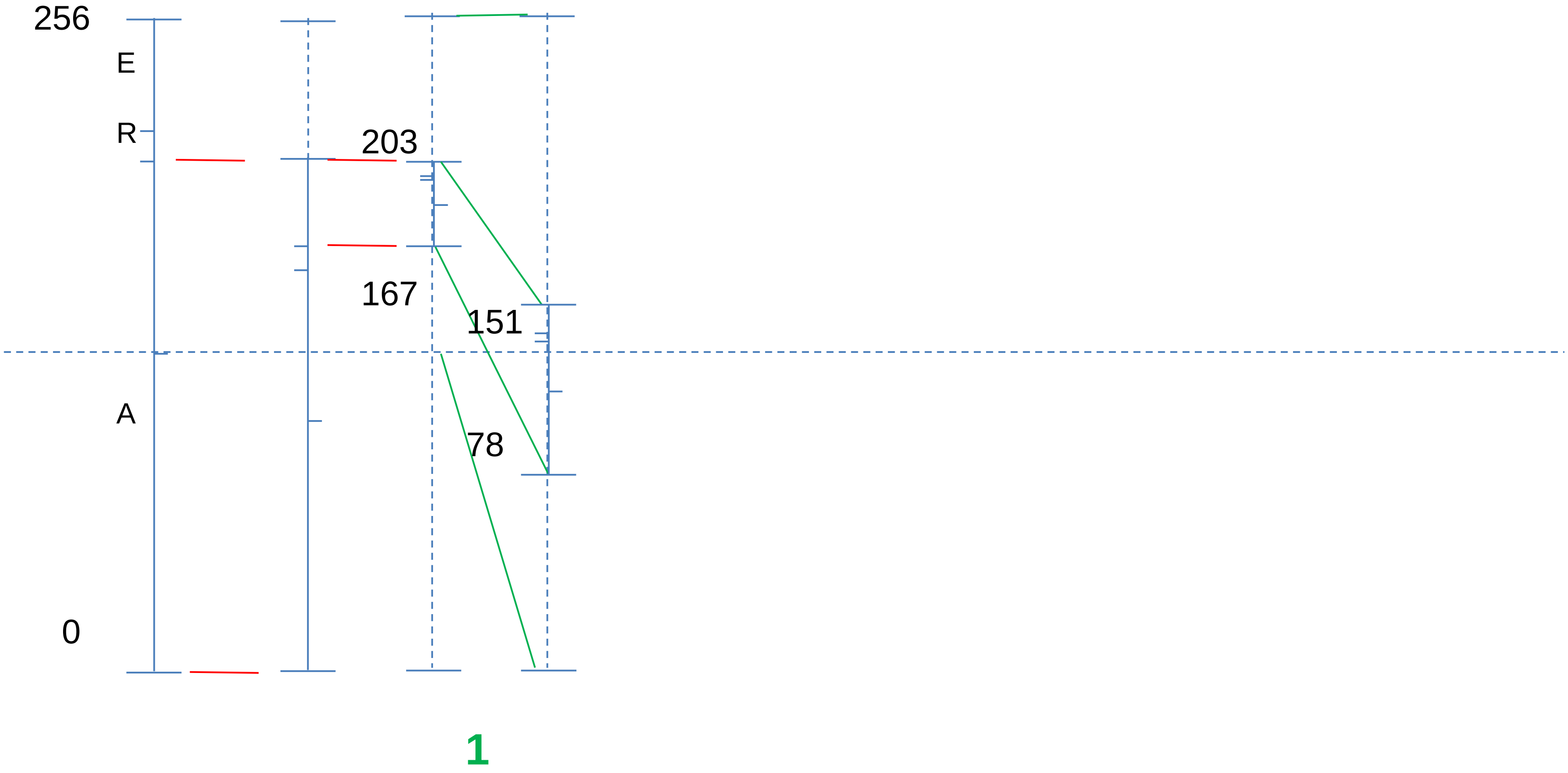


Codificação Aritmética décimal

- Condição E2 - escalar

$$
\begin{aligned}
low &= 2 \times (low - 128) = 78 \ (01001110) \\
high &= 2 \times (high - 128) + 1 = 151 \ (10010111)
\end{aligned}
$$

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

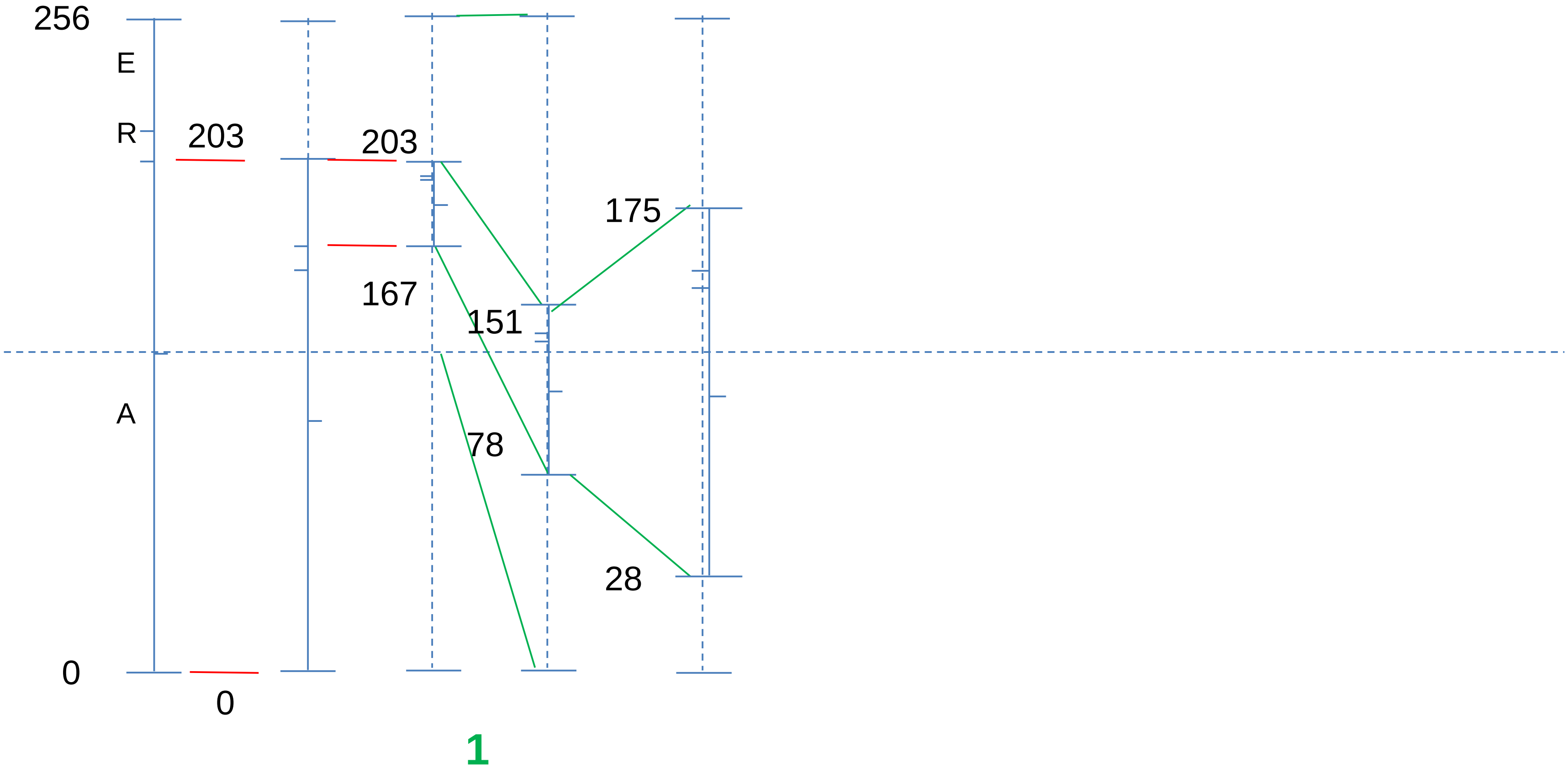


Codificação Aritmética décimal

- Condição E3 - escalar

$$
\begin{aligned}
low &= 2 \times (low - 64) = 28 \ (00011100) \\
high &= 2 \times (high - 64) + 1 = 175 \ (10101111) \\
count\ E3 &= count\ E3 + 1
\end{aligned}
$$

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	



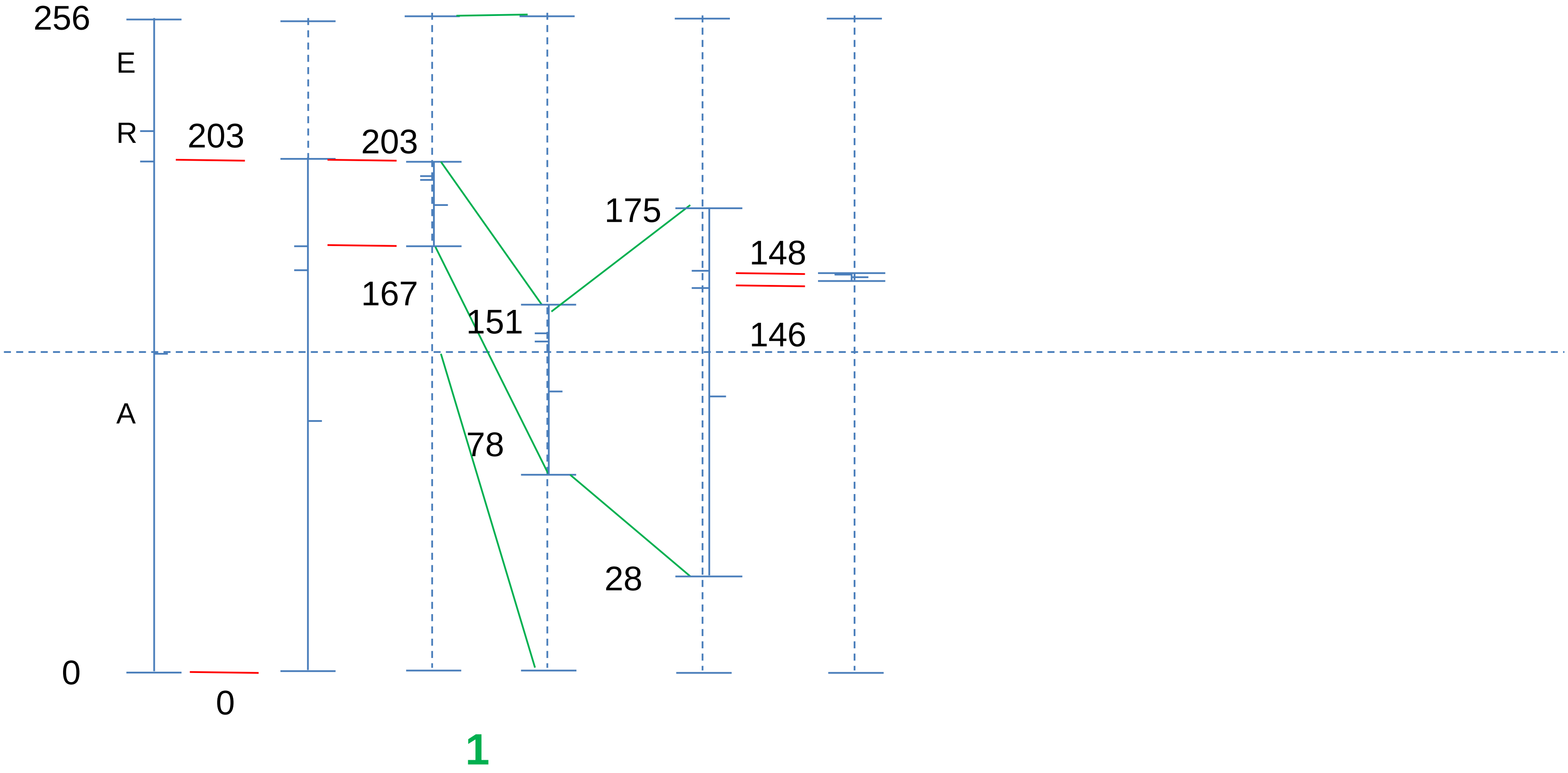
Codificação Aritmética décimal

- Codificar “r”

$$low = 28 + \left\lfloor \frac{148 \times 40}{50} \right\rfloor = 146\ (10010010)$$

$$high = 28 + \left\lfloor \frac{148 \times 41}{50} \right\rfloor - 1 = 148\ (10010100)$$

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

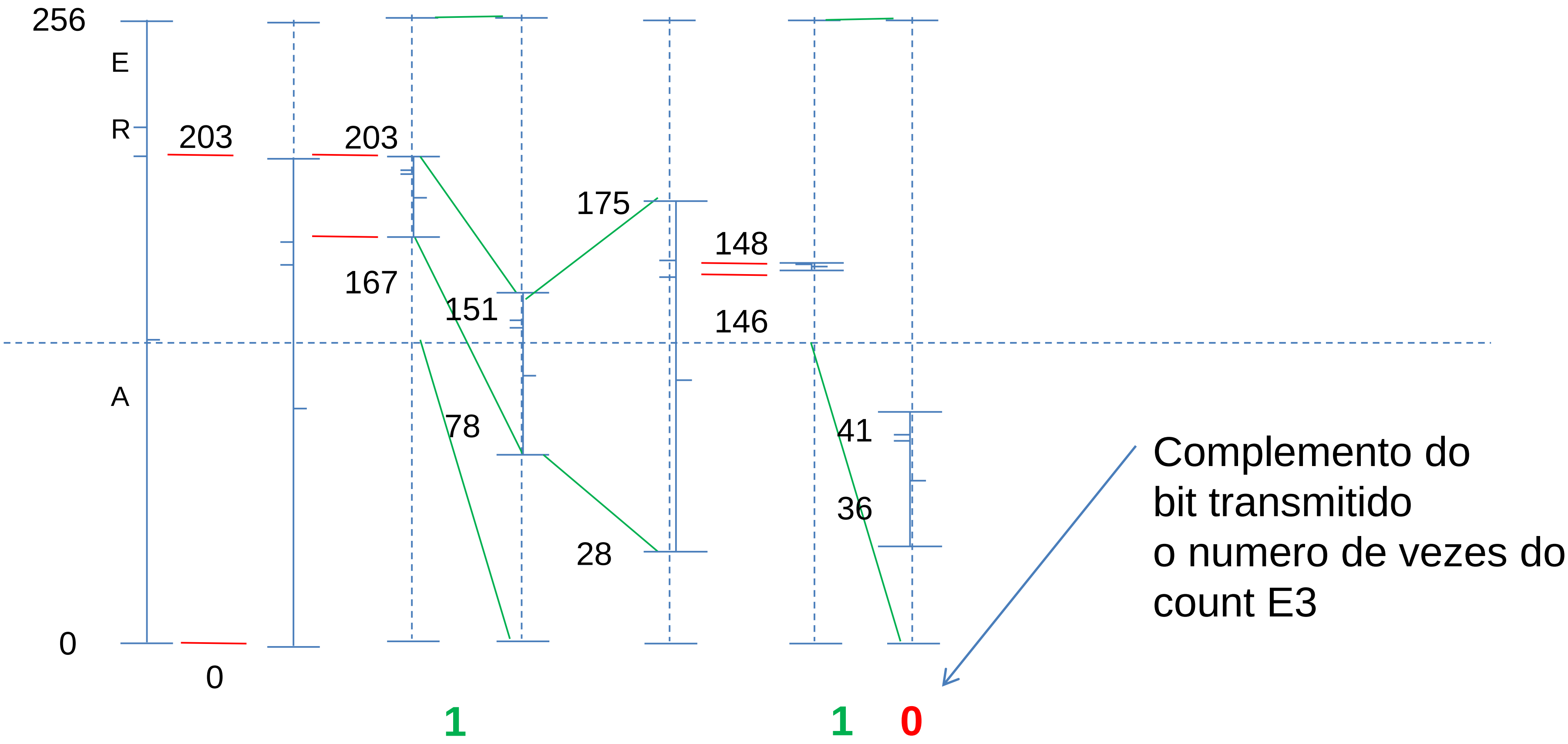


Codificação Aritmética décimal

- Condição E2 -escalar

$$low = 2 \times (low - 128) = 36 \ (00100100)$$
$$high = 2 \times (high - 128) + 1 = 41 \ (00101001)$$

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	



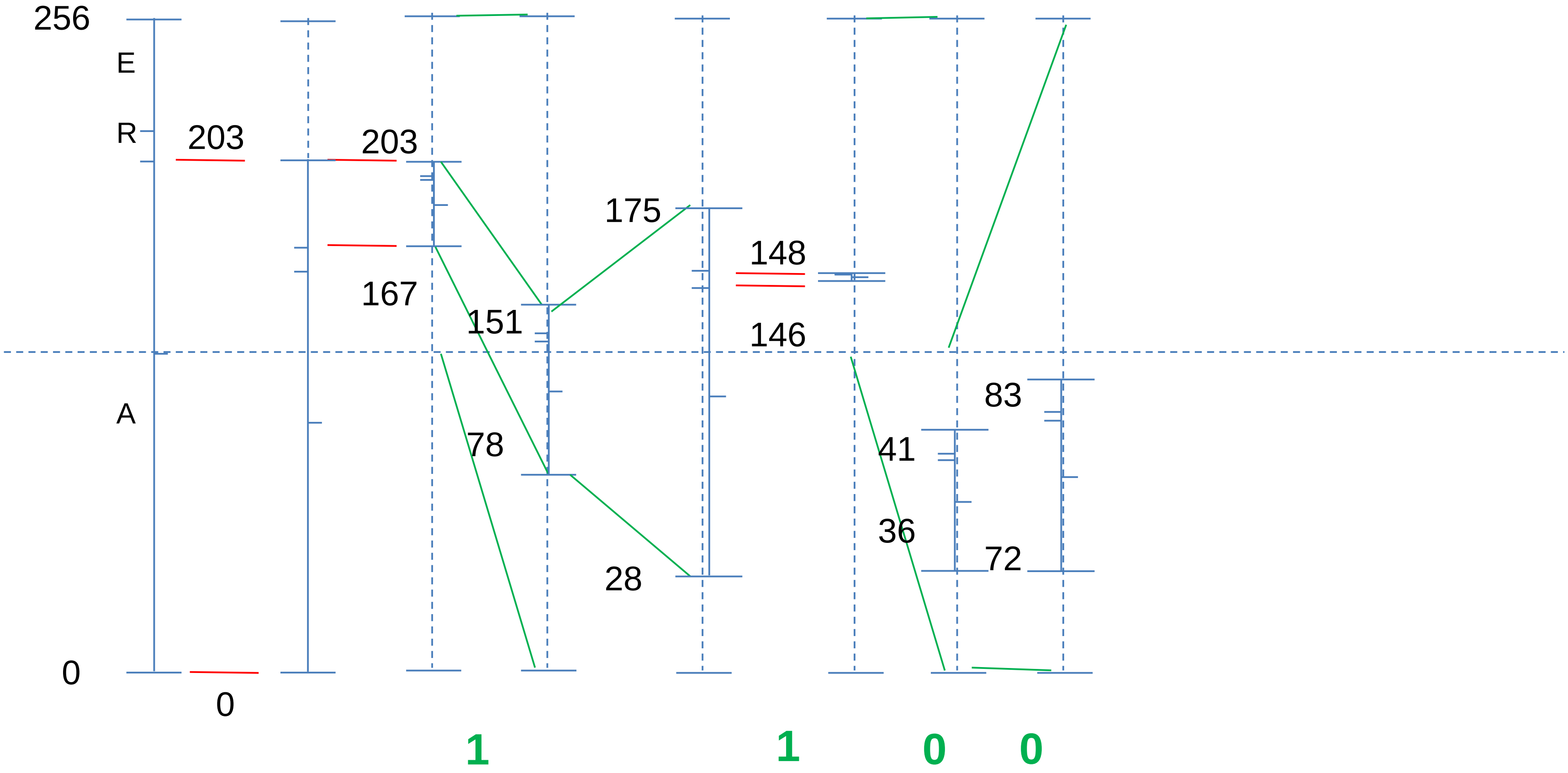
Complemento do bit transmitido o numero de vezes do count E3

Codificação Aritmética décimal

- Condição E1 -escalar

$$
\begin{aligned}
low &= 2 \times (low - 128) = 72\ (01001000) \\
high &= 2 \times (high - 128) + 1 = 83\ (01010011)
\end{aligned}
$$

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

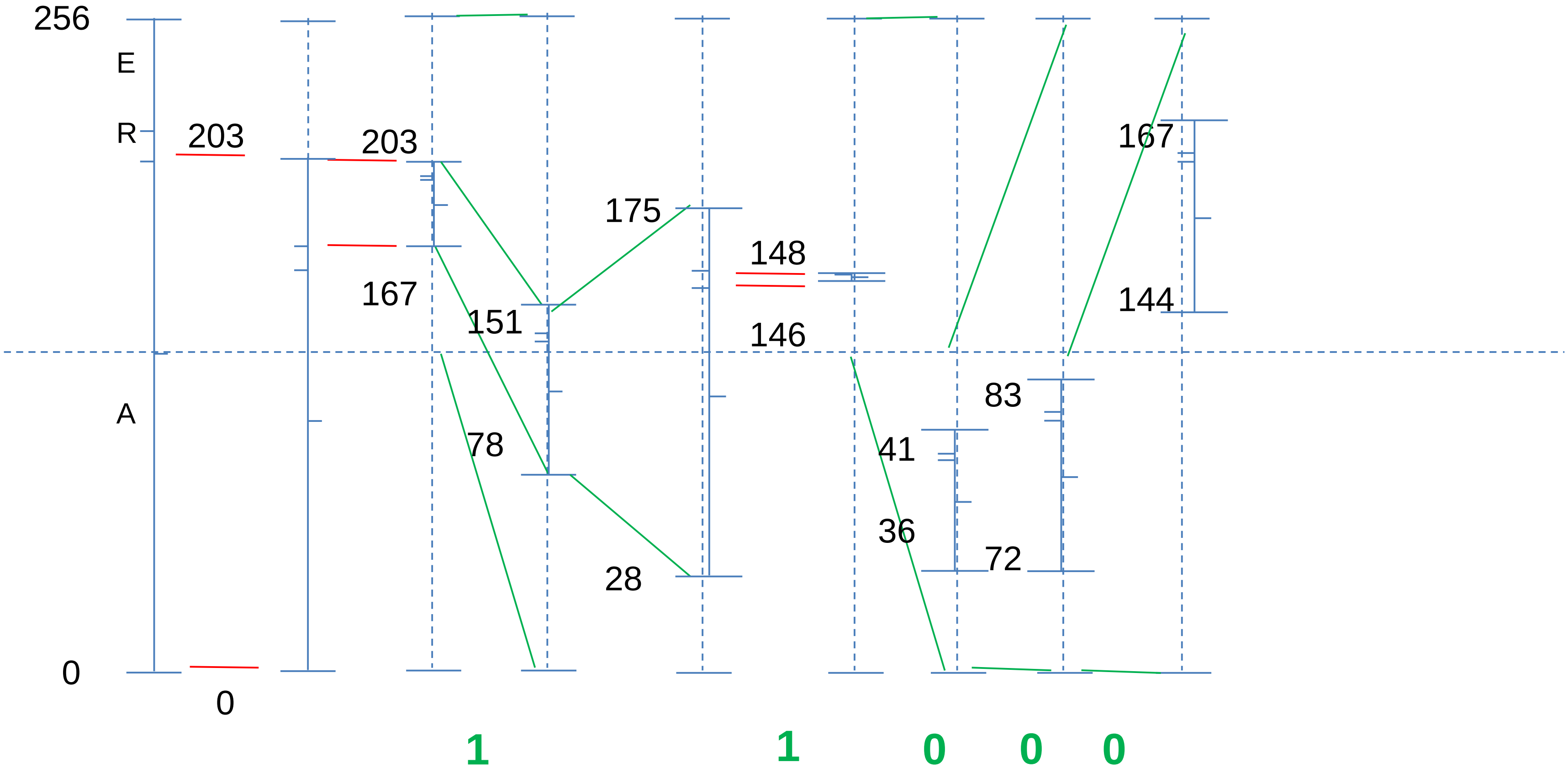


Codificação Aritmética décimal

- Condição E1 -escalar

$$\begin{aligned} low &= 2 \times (low - 128) = 144\ (10010000) \\ high &= 2 \times (high - 128) + 1 = 167\ (10100111) \end{aligned}$$

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

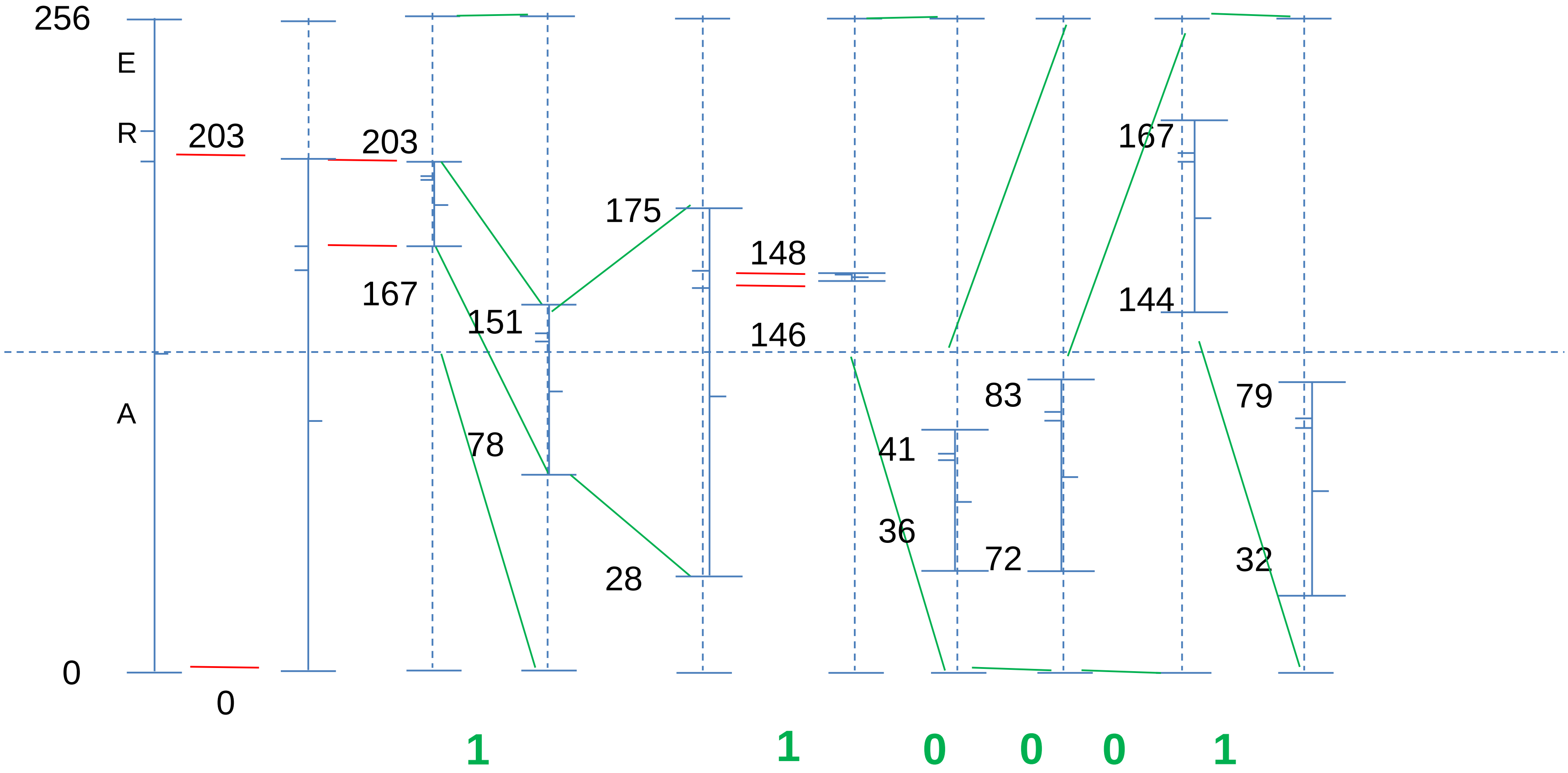


Codificação Aritmética décimal

- Condição E2 -escalar

$$low = 2 \times (low - 128) = 32\ (00100000)$$
$$high = 2 \times (high - 128) + 1 = 79\ (01001111)$$

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

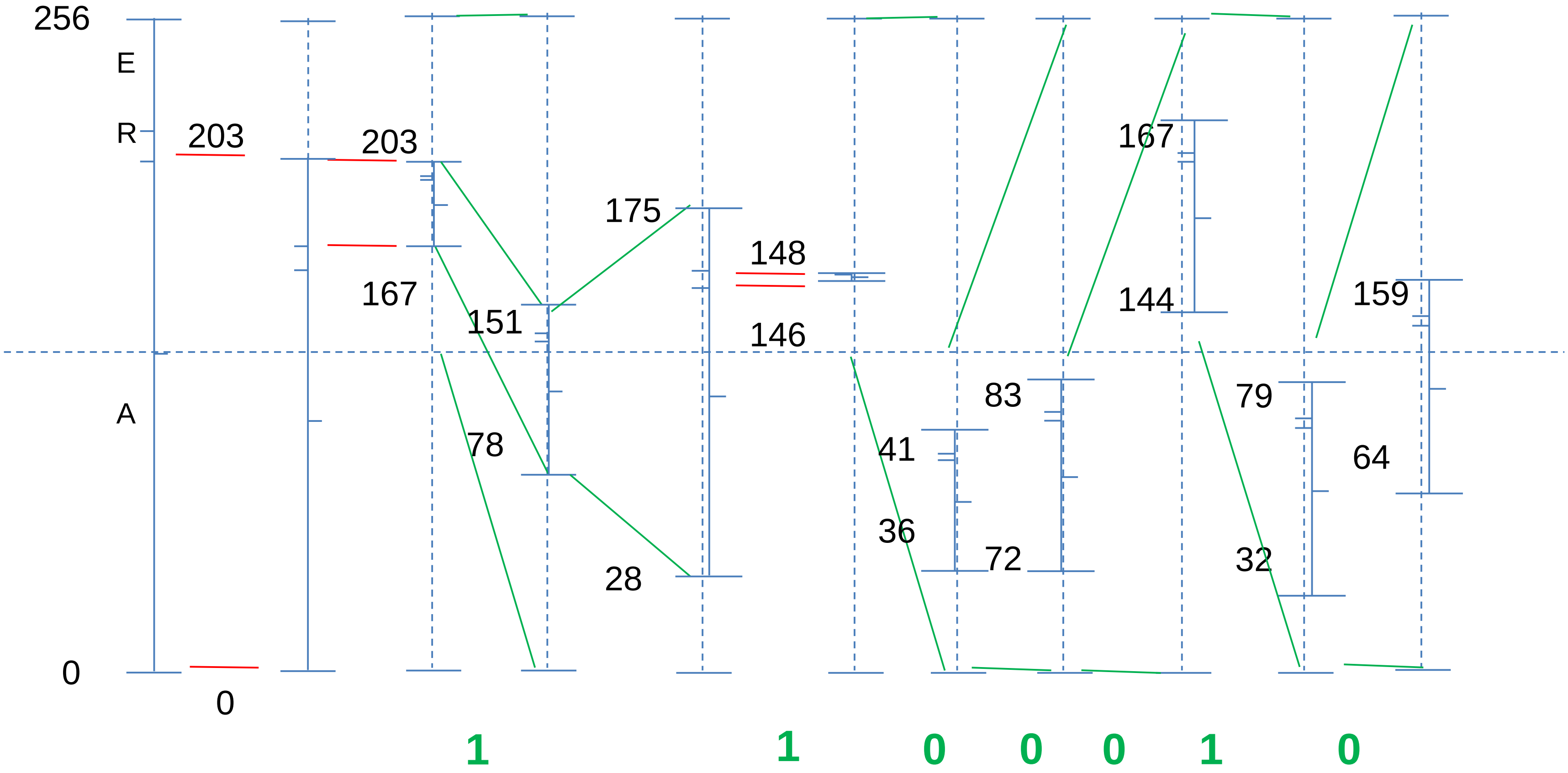


Codificação Aritmética décimal

- Condição E1 -escalar

$$\begin{aligned} low &= 2 \times (low - 128) = 64 \ (01000000) \\ high &= 2 \times (high - 128) + 1 = 159 \ (10011111) \end{aligned}$$

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

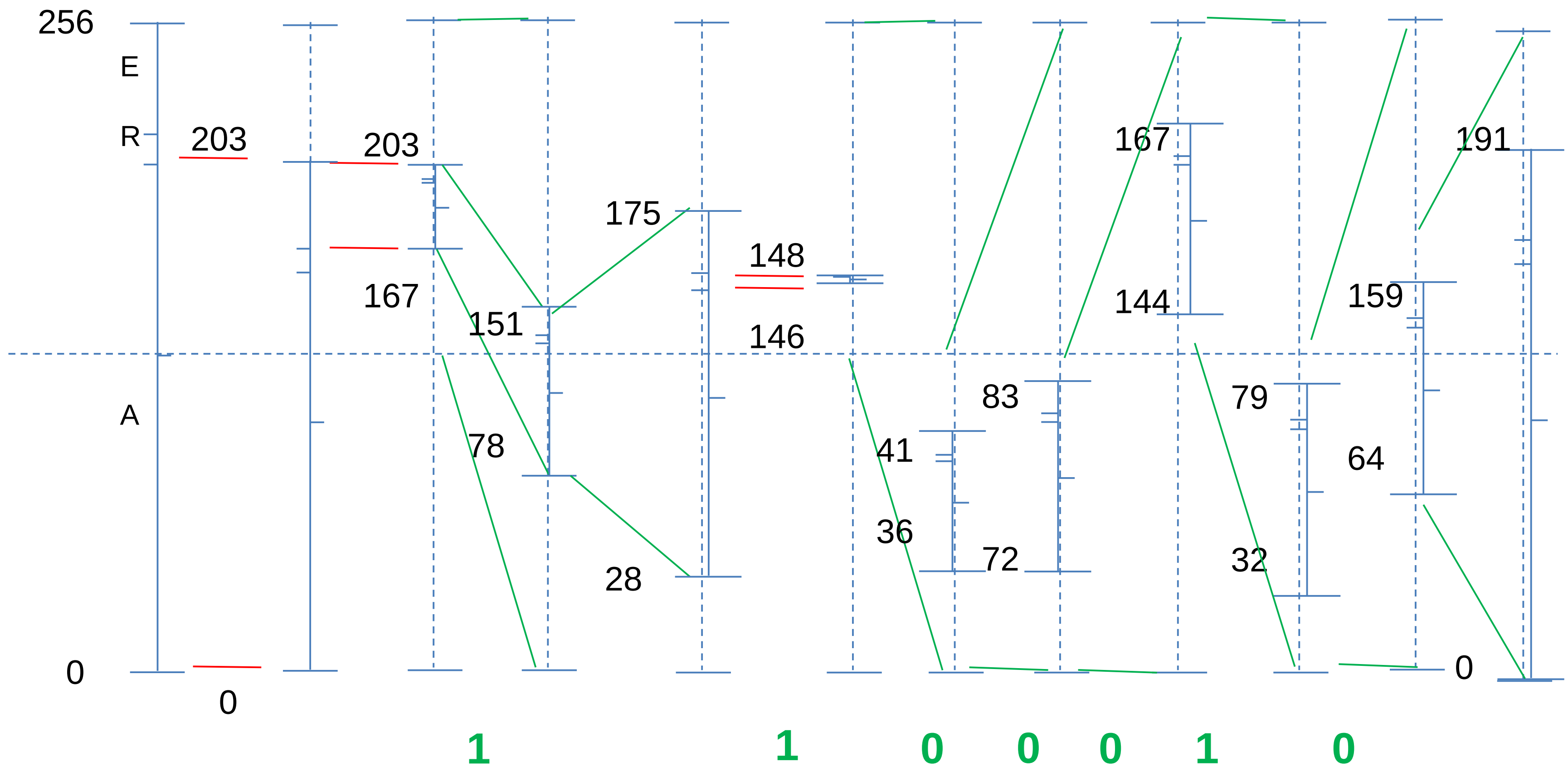


Codificação Aritmética décimal

- Condição E3 -escalar

$$\begin{aligned} low &= 2\times(low-64) = 0\ (00000000) \\ high &= 2\times(high-64)+1 = 191\ (10111111) \\ count\ E3 &= count\ E3 + 1 \end{aligned}$$

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

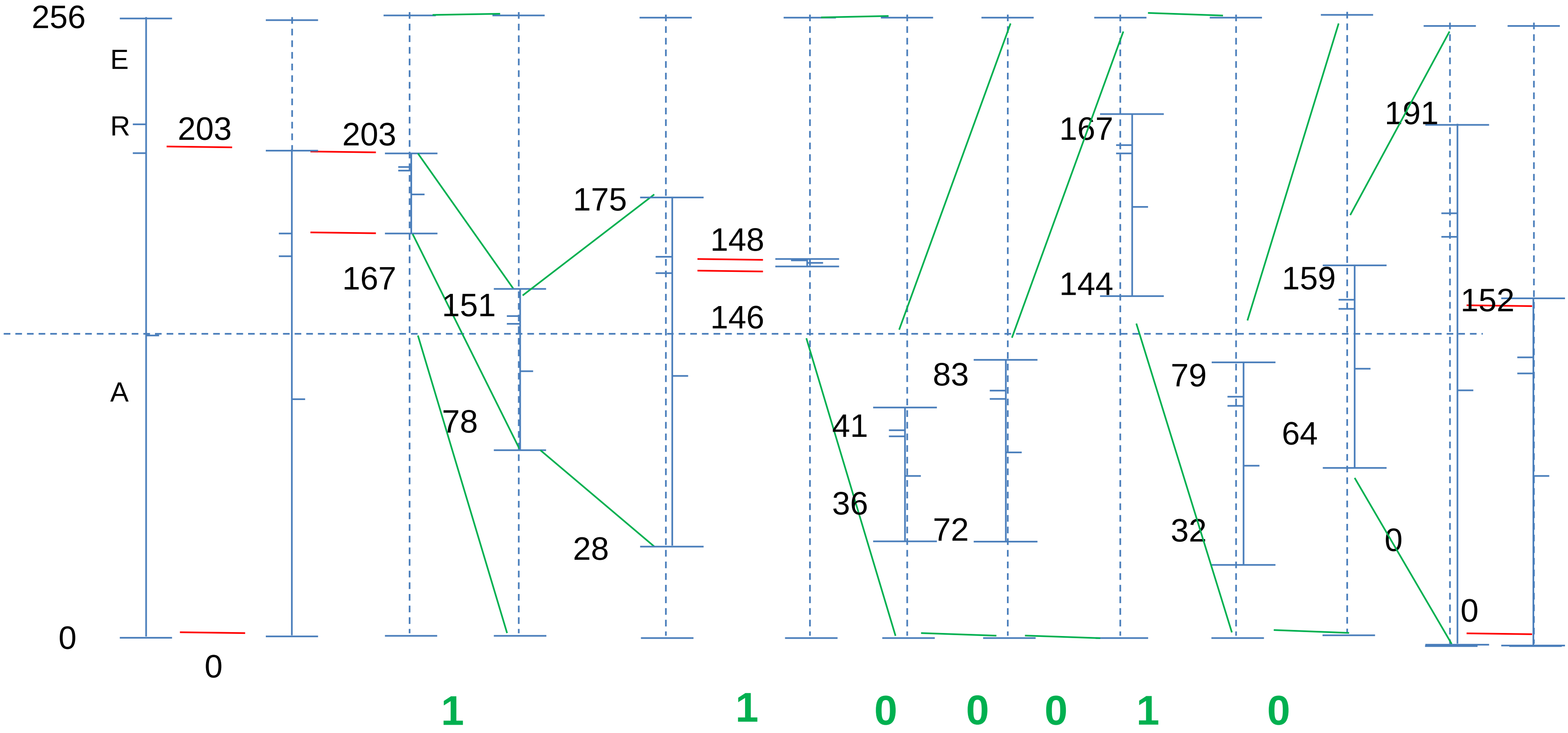


Codificação Aritmética décimal

- Codifica “a”

$$
\begin{aligned}
low &= 0 + \left\lfloor \frac{192 \times 0}{50} \right\rfloor = 0 \ (00000000) \\
high &= 0 + \left\lfloor \frac{192 \times 40}{50} \right\rfloor - 1 = 152 \ (10011000)
\end{aligned}
$$

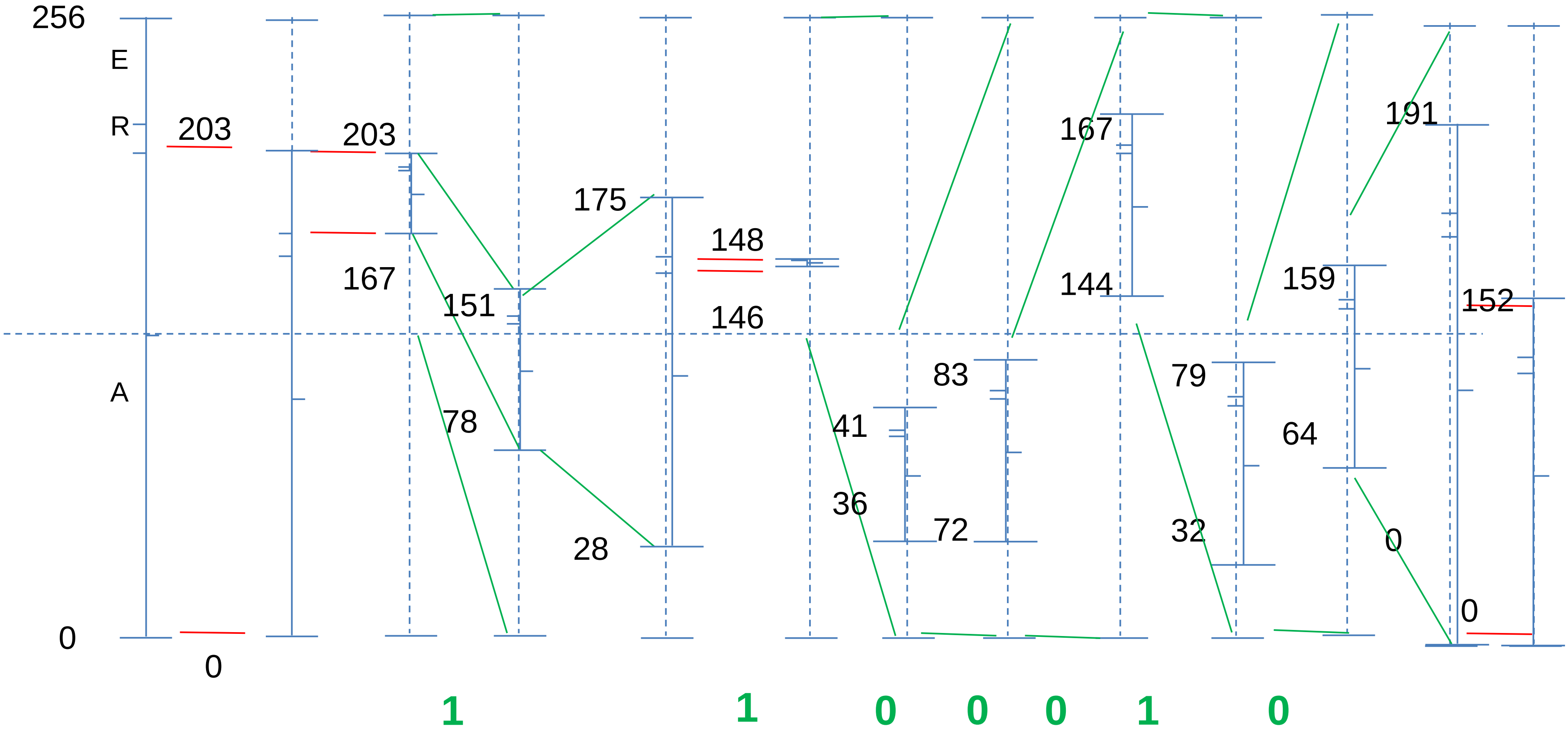
	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	



Codificação Aritmética décimal

- Terminou a mensagem
- Envia o low (00000000)
- Como o countE3 = 1
- O codigo fica 1 1 0 0 0 1 0 010000000

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	



Descodificação Aritmética - inteira

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- Lê m bits

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- Tag = (1 1 0 0 0 1 0 0) = 196
- low = 0
- high = 255

$$\left\lfloor \frac{(tag - low + 1) \times totalcount}{(high - low + 1)} \right\rfloor = \left\lfloor \frac{197 \times 50}{255 - 0 + 1} \right\rfloor = 38$$ = “a”

- Update low = 0 and high = 203
- Nenhuma condição é verificada

Descodificação Aritmética - inteira

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- low = 0
- high = 203
- Tag = (1 1 0 0 0 1 0 0) = 196
- $\left\lfloor \frac{(tag - low + 1) \times totalcount}{(high - low + 1)} \right\rfloor = \left\lfloor \frac{197 \times 50}{203 - 0 + 1} \right\rfloor = 48$ = “e”
- Update low = 167 and high = 203
- Condição E2

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

Descodificação Aritmética - inteira

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- low = 0
- high = 203
- Tag = (1 1 0 0 0 1 0 0) = 196

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

- $\left\lfloor \frac{(tag - low + 1) \times totalcount}{(high - low + 1)} \right\rfloor = \left\lfloor \frac{197 \times 50}{203 - 0 + 1} \right\rfloor = 48$ = “e”

- Update low = 167 and high = 203
- Condição E2: scale low = 78
  scale high = 151
  read next bit
  scale tag and add bit= 137 (10001001)

- Condição E3:

Descodificação Aritmética - inteira

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- low = 78
- high = 151
- Tag = 137 (10001001)
- Condição E3: scale low = 28
  scale high = 175
  read next bit
  scale tag = 146 (10010010)
- Nenhuma condição é verificada… decode simbol

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

Descodificação Aritmética - inteira

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- low = 28
- high = 175
- Tag = 146 (10010010)
- Condição E3: scale low = 28
  scale high = 175
  read next bit
  scale tag and add bit= 146 (10010010)
- Nenhuma condição é verificada… decode simbol

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

Descodificação Aritmética - inteira

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- low = 28
- high = 175
- Tag = 146 (10010010)

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

- $\left\lfloor \frac{(tag - low + 1) \times totalcount}{(high - low + 1)} \right\rfloor = \left\lfloor \frac{(146 - 28 + 1) \times 50}{175 - 28 + 1} \right\rfloor = 40$ = “r”

- Update low = 146
- Update high = 148
- Condição E2:

Descodificação Aritmética - inteira

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- low = 28
- high = 175
- Tag = 146 (10010010)

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

- $\left\lfloor \frac{(tag - low + 1) \times totalcount}{(high - low + 1)} \right\rfloor = \left\lfloor \frac{(146 - 28 + 1) \times 50}{175 - 28 + 1} \right\rfloor = 40$ = "r"

- Update low = 146
- Update high = 148
- Condição E2: scale low = 36
  scale high = 41
  read next bit
  scale tag and add bit= (00100100) = 36
- Condição E1:

Descodificação Aritmética - inteira

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- low = 36
- high = 41
- Tag = 36 (00100100)

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

- Condição E1: scale low = 72
  scale high = 83
  read next bit
  scale tag and add bit= (01001000) = 72
- Condição E1:

Descodificação Aritmética - inteira

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- low = 72
- high = 83
- Tag = 72 (01001000)

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

- Condição E1: scale low = 144
  scale high = 167
  read next bit
  scale tag and add bit= (10010000) = 144
- Condição E2:

Descodificação Aritmética - inteira

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- low = 144
- high = 167
- Tag = 144 (10010000)

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

- Condição E2: scale low = 32
  scale high = 79
  read next bit
  scale tag and add bit= (00100000) = 32
- Condição E1:

Descodificação Aritmética - inteira

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- low = 32
- high = 79
- Tag = 32 (00100000)

	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

- Condição E1: scale low = 64

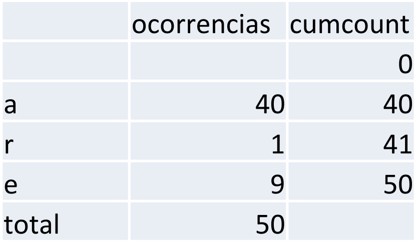
  scale high = 159

  read next bit

  scale tag and add bit= (01000000) = 64
- Nenhuma condição é verificada... decode simbol

Descodificação Aritmética - inteira

- 1 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
- low = 64
- high = 159
- Tag = 64 (01000000)



	ocorrencias	cumcount
		0
a	40	40
r	1	41
e	9	50
total	50	

- $\left\lfloor \frac{(tag - low + 1) \times totalcount}{(high - low + 1)} \right\rfloor = \left\lfloor \frac{(64 - 64 + 1) \times 50}{159 - 64 + 1} \right\rfloor = 0$ = “a”

- Foi o último simbolo!
- Mensagem “aera”