Codificação e decodificação aritmética com inteiros

Algoritmo de Boden/Clasen/Kneis (2007)

Codificação

Inicializar mHigh, mLow, total, high_count, low_count ...

Expansões **E1**, **E2** e **E3**

mStep = (mHigh – mLow + 1) / total;
mHigh = mLow + mStep * high_count – 1;
mLow = mLow + mStep * low_count;

```
while( ( mHigh < g_Half ) || ( mLow >= g_Half ) ) {
    if( mHigh < g_Half ) // E1
    {
        SetBit( 0 );
        mLow = mLow * 2;
        mHigh = mHigh * 2 + 1;

        // E3
        for(; mScale > 0; mScale-- )
        SetBit( 1 );
    }
    else if(mLow >= g_Half ) // E2
    {
        SetBit( 1 );
        mLow = 2 * ( mLow - g_Half );
        mHigh = 2 * ( mHigh - g_Half ) + 1;

        // E3
        for(; mScale > 0; mScale-- )
        SetBit( 0 );
    }
}

while( ( g_FirstQuarter <= mLow ) && ( mHigh < g_ThirdQuarter )) {
    mScale++;
    mLow = 2 * ( mLow - g_FirstQuarter );
    mHigh = 2 * ( mHigh - g_FirstQuarter ) + 1;
}
```

Decodificação

Inicializar mBuffer, mHigh, mLow, total, high_count, low_count ...

mStep = (mHigh – mLow + 1) / total;
value = (mBuffer – mLow) / mStep;

```
// update upper bound
mHigh = mLow + mStep * high_count - 1; // interval open at the top => -1

// update lower bound
mLow = mLow + mStep * low_count;

// e1/e2 scaling
while( ( mHigh < g_Half ) || ( mLow >= g_Half ) )
  {
    if( mHigh < g_Half )
    {
      mLow = mLow * 2;
      mHigh = mHigh * 2 + 1;
      mBuffer = 2 * mBuffer + GetBit();
    }
    else if( mLow >= g_Half )
    {
      mLow = 2 * ( mLow - g_Half );
      mHigh = 2 * ( mHigh - g_Half ) + 1;
      mBuffer = 2 * ( mBuffer - g_Half ) + GetBit();
    }


     mScale = 0;
   }

 // e3 scaling
 while( ( g_FirstQuarter <= mLow ) && ( mHigh < g_ThirdQuarter ) )
 {
   mScale++;
   mLow = 2 * ( mLow - g_FirstQuarter );
   mHigh = 2 * ( mHigh - g_FirstQuarter ) + 1;
   mBuffer = 2 * ( mBuffer - g_FirstQuarter ) + GetBit();
 }
```

Observação:

O trecho de codificação bem como de decodificação estão dentro de laços de repetição que permanecem ativos enquanto houver informação para ser codificada/decodificada.

Contagem empregada no exemplo:

symbol	frequency	low_count	high_count
a	2	0	2
b	1	2	3
c	3	3	6
d	1	6	7
e	1	7	8