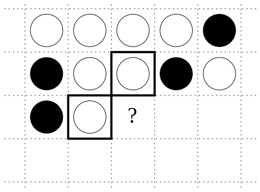
## *Example: Adaptively Compressing Black-and-White Images*

Suppose we have images (of some fixed dimensions) in which pixels are either black or white. We may expect large regions of white pixels and large regions of black pixels, and wish to use that knowledge to compress them.

But how large will the regions be? We probably don't know, so we use an adaptive scheme, with contexts defined by pixels above and to the left:




## *The Encode Program*

```
/* Initialize model. */

for (a = 0; a<2; a++) {
    for (l = 0; l<2; l++) {
        freq0[a][l] = 1;               /* Set frequencies of 0's */
        freq1[a][l] = 1;               /* and 1's to be equal.   */
    }
}

/* Encode image. */

for (i = 0; i<Height; i++) {
    for (j = 0; j<Width; j++) {
        a = i==0 ? 0 : image[i-1][j];  /* Find current context.  */
        l = j==0 ? 0 : image[i][j-1];
        encode_bit(image[i][j],        /* Encode pixel.          */
                   freq0[a][l],freq1[a][l]);
        if (image[i][j]) {             /* Update frequencies for */
            freq1[a][l] += 1;          /* this context.          */
        }
        else {
            freq0[a][l] += 1;
        }
        if (freq0[a][l]+freq1[a][l]>Freq_full) { /* Avoid huge   */
            freq0[a][l] = (freq0[a][l]+1) >> 1;  /* frequencies  */
            freq1[a][l] = (freq1[a][l]+1) >> 1;
        }
    }
}
```

## *The Decode Program*

```
/* Initialize model. */

for (a = 0; a<2; a++) {
    for (l = 0; l<2; l++) {
        freq0[a][l] = 1;               /* Set frequencies of 0's */
        freq1[a][l] = 1;               /* and 1's to be equal.   */
    }
}

/* Decode and write image. */

for (i = 0; i<Height; i++) {
    for (j = 0; j<Width; j++) {
        a = i==0 ? 0 : image[i-1][j];  /* Find current context.  */
        l = j==0 ? 0 : image[i][j-1];
        image[i][j] =                  /* Decode pixel.          */
          decode_bit(freq0[a][l],freq1[a][l]);
        printf("%c%c",image[i][j] ? '#' : '.',
                      j==Width-1 ? '\n' : ' ');
        if (image[i][j]) {             /* Update frequencies for */
            freq1[a][l] += 1;          /* this context.          */
        }
        else {
            freq0[a][l] += 1;
        }
        if (freq0[a][l]+freq1[a][l]>Freq_full) { /* Avoid huge   */
            freq0[a][l] = (freq0[a][l]+1) >> 1;  /* frequencies  */
            freq1[a][l] = (freq1[a][l]+1) >> 1;
        }
    }
}
```