USING MAPLE TO ENUMERATE RESTRICTED BINARY RECTANGLES

Dr. Joel C. Fowler
Mathematics Department
Southern Polytechnic State University
1100 South Marietta Parkway
Marietta, GA 30060-2896
email: jfowler@spsu.edu

We generalize the counting of binary strings without certain substrings to the enumeration of binary rectangles without particular sub-rectangles. We examine how Maple can be used, at a student project level, to study the problem, generate examples, and motivate theoretical results.

```
> restart; with(LinearAlgebra): with(StringTools):
```

To illustrate the computational method we shall use, we first find a recurrence relation for the number of binary strings without a run of 3 consecutive 0's or 1's. Let b(n) = the number of length n binary strings without a run of 3 consecutive 0's or 1's .
We use a matrix approach for generating values.

Define the column vector $\mathbf{v}(n) = \begin{bmatrix} v0(n) \\ v1(n) \\ v2(n) \\ v3(n) \end{bmatrix}$, where v0(n) = the number of length n strings ending in 00 without a run of 3 zeros or ones; v1(n) = the number of length n strings ending in 01 without a run of 3 zeros or ones; v2(n) = the number of length n strings ending in 10 without a run of 3 zeros or ones; v3(n) = the number of length n strings ending in 11 without a run of 3 zeros or ones. We then have, by considering each case in turn:

v0(n+1) = v2(n), since such a string of length n+1 would have to end in 10
v1(n+1) = v0(n) + v2(n), since such a string of length n+1 would have to end in 001 or 101;
v2(n+1) = v1(n) + v3(n), since such a string of length n+1 would have to end in 010 or 110;
v3(n+1) = v1(n), since such a string of length n+1 would have to end in 011 .

Hence we have the matrix relationship $v(n+1) = \mathbb{B} * v(n)$, where $\mathbb{B} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$.

Note that $v(2) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ and thus $v(n) = \mathbb{B}\,\hat{}\,(n-2) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$.

Since $b(n) = v0(n) + v1(n) + v2(n) + v3(n) = [\,1\ 1\ 1\ 1\,]\, v(n)$ we then have, for $n > 1$:

$b(n) = [\,1\ 1\ 1\ 1\,]\, \mathbb{B}\,\hat{}\,(n-2) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

A quick route to a recurrence relation for b(n) is to note that the minimal polynomial for $\mathbb{B}$ yields a linear recurrence relation that the sequence must satisfy.

```
> B := <<0,1,0,0>|<0,0,1,1>|<1,1,0,0>|<0,0,1,0>>;
```

$$B := \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

```
> MinimalPolynomial(B,x);
```

$$-1 - 2x - x^2 + x^4$$

Hence we have the linear recurrence: $b(n) = b(n-2) + 2*b(n-3) + b(n-4)$, for $n > 5$; which, along with the initial conditions: [b(0) = 1, b(1) = 2], b(2) = 4, b(3) = 6, b(4) = 10, and b(5) = 16, determines the sequence.

Use of the minimal polynomial generates a linear recurrence, but not always one of lowest order. Using Maple, we can interate the powers of the matrix B to hunt for a lower order recurrence pattern.

```
> seq(<<1>|<1>|<1>|<1>>.B^i.<1,1,1,1>, i=0..10);
```

[4], [6], [10], [16], [26], [42], [68], [110], [178], [288], [466]

It appears that b(n) is determined by: $b(n) = b(n-1) + b(n-2)$, for $n > 3$; $b(2) = 4$, $b(3) = 6$.

It is easy to show that this recurrence relation holds in general using several basic facts about linear homogeneous recurrence relations. Suppose a sequence, {a(n)}, satisfies a linear, homogeneous recurrence relation, R, of order k. Let R' be a linear, homogeneous

recurrence relation of order p < k. It can be shown by induction that: If the first k terms of {a(n)} , after the initial conditions of R', satisfy R', then all of {a(n)} must satisfy R'. It is also easy to show that any recursively defined sequence has a unique recurrence relation of lowest order.

We have already noted that, since b(n) is produced by the powers of the matrix, B, its minimal polynomial yields the 4th order linear, homogenous recurrence relation b(n) = b(n - 2) + 2*b(n - 3) + b(n - 4). Thus we need only note that the first 4 terms (past the first two) computed above for b(n) , satisfy b(n) = b(n-1) + b(n-2) to establish that the entire sequence satisfies this recurrence relation, which we have already seen by inspection. With the recurrence relation in hand, we can look for a more theoretical justification for it. In this case, arguments for the number of binary strings without a run of k consecutive zeros or ones are well known. A nice introduction the problem and the literature on its solution can be found in [M. Schilling, The Longest Run of Heads, The College Mathematics Journal, Vol. 21, No. 3 (1990) 196 - 207].

This matrix approach can be used to generate values and find the lowest order recurrence relation for strings with various pattern avoidances over any q-ary alphabet. And rather than use inspection to try to find the lowest order recurrence relation, we use Maple's regression routines to test the best fitting recurrence relation, of successive orders, until we find the lowest order that is a perfect fit. The following automates the entire algorithm for any set, S, of forbidden substrings (all with the same length) using a few basic routines and a Maple procedure.

```
> QaryRep := (n,q) -> Remove( IsPunctuation or IsSpace,
  convert(
  [seq(convert(n,base,q)[(nops(convert(n,base,q))+1)-i],
  i=1..(nops(convert(n,base,q))))],string));
```

QaryRep := $(n, q) \rightarrow$ *StringTools*:-*Remove*(*StringTools*:-*IsPunctuation* **or** *StringTools*:-*IsSpace*, convert([seq(convert(n, *base*, q)$_{\text{nops(convert}(n,\ base,\ q)) + 1 - i}$, $i = 1$ .. nops(convert(n, *base*, q)))], *string*))

```
> QaryRep(31,3);
```

"1011"

```
> FullQaryRep := (n,k,q) -> cat( Fill( "0", k - length(
  QaryRep(n,q))), QaryRep(n,q) );
```

FullQaryRep :=

$(n, k, q) \rightarrow$ cat(*StringTools*:-*Fill*("0", k – length(QaryRep(n, q))), QaryRep(n, q))

```
> FullQaryRep(100,6,5);
                                "000400"
> MinRecRelAnyBase := proc(S::set,q)   description "Minimal
  Recurrence Relation for enumerating q-ary strings over {0,
  1, 2, ..., q-1} without any element of S as a substring";

  slength := length(S[1]);

  # "Populate the matrix M"
  MatEnt := (i,j) -> `if`( Drop(
  FullQaryRep(j-1,slength-1,q),1) = Take(
  FullQaryRep(i-1,slength-1,q),slength-2) and not( cat(
  FullQaryRep(j-1,slength-1,q),
  FullQaryRep(i-1,slength-1,q)[slength-1]) in S ) , 1, 0): M
  := Matrix(q^(slength-1),q^(slength-1),MatEnt);

  # "Compute sequence terms from powers of M"
  Temp := (Matrix(1,q^(slength-1),1)) . M :  Seque(0) :=
  q^(slength-1) :
  for i from 1 to max(q^slength,12) do
      Seque(i) := ( Temp . Matrix(q^(slength-1),1,1))[1,1]:
      Temp := Temp . M :
  end do:

  # "Find recurrence relation fit, by first regressing
  against a short vector of terms, and then checking against
  the full list of terms."
  CheckFit := 1 :
  for SeqDepth from 1 while CheckFit<>0 do
     NormResult := 1 :
          for TryOrder from 1 while NormResult<>0 do
              MatEnt2 := (i,j) -> Seque(TryOrder+i-j-1) :
  SolveMat := Matrix(SeqDepth*slength,TryOrder, MatEnt2) :
              VectEnt := (i,j) -> Seque(TryOrder+i-1) :
  SolveVect := Matrix(SeqDepth*slength,1, VectEnt):
              Sol := LeastSquares(SolveMat,SolveVect):
              NormResult := Norm(SolveMat.Sol - SolveVect,2)
```

```
:
        end do:

   FullMat := Matrix(q^(slength-1),TryOrder-1, MatEnt2) :
FullVect := Matrix(q^(slength-1),1, VectEnt):
   CheckFit := Norm( FullMat.Sol - FullVect) :
end do:

op([S,a(n) = (Transpose(<seq( a(n-i), i=1..(TryOrder-1))>)
.Sol)[1], [seq(Seque(k),k=0..10)]]);

end proc:
```

Investigating and explaining various results for some small examples is useful. Each of the following recurrence relations can be explained with a simple purely theoretical argument.

```
> MinRecRelAnyBase({"01","11"}, 2 );
```

$$\{\text{"01"}, \text{"11"}\}, a(n) = a(n-1), [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]$$

```
> MinRecRelAnyBase({"10"}, 2 );
```

$$\{\text{"10"}\}, a(n) = 2\,a(n-1) - a(n-2), [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]$$

```
> MinRecRelAnyBase({"000", "111", "011","110","101"}, 2 );
```

$$\{\text{"000"}, \text{"011"}, \text{"101"}, \text{"110"}, \text{"111"}\}, a(n) = a(n-1), [4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]$$

```
> MinRecRelAnyBase({"00", "01", "10", "20", "21"},3);
```

$$\{\text{"00"}, \text{"01"}, \text{"10"}, \text{"20"}, \text{"21"}\}, a(n) = 2\,a(n-1) - a(n-2),$$
$$[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]$$

```
> MinRecRelAnyBase({ "10", "20", "21"},3);
```

$$\{\text{"10"}, \text{"20"}, \text{"21"}\}, a(n) = 3\,a(n-1) - 3\,a(n-2) + a(n-3),$$
$$[3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78]$$

```
> MinRecRelAnyBase({"200", "020", "002", "011", "101",
  "110", "122", "212", "221"},3);
```

$$\{\text{"002"}, \text{"011"}, \text{"020"}, \text{"101"}, \text{"110"}, \text{"122"}, \text{"200"}, \text{"212"}, \text{"221"}\}, a(n) = 2\,a(n-1),$$
$$[9, 18, 36, 72, 144, 288, 576, 1152, 2304, 4608, 9216]$$

We can use the generator to investigate strings without runs of repeated characters.

```
> MinRecRelAnyBase({"000"},4); MinRecRelAnyBase({"000",
  "111"},4); MinRecRelAnyBase({"000", "111", "222"},4);
  MinRecRelAnyBase({"000", "111", "222", "333"},4);
```

{"000"}, a(n) = 3 a($n-1$) + 3 a($n-2$) + 3 a($n-3$),

[16, 63, 249, 984, 3888, 15363, 60705, 239868, 947808, 3745143, 14798457]

{"000", "111"}, a(n) = 3 a($n-1$) + 3 a($n-2$) + 2 a($n-3$),

[16, 62, 242, 944, 3682, 14362, 56020, 218510, 852314, 3324512, 12967498]

{"000", "111", "222"}, a(n) = 3 a($n-1$) + 3 a($n-2$) + a($n-3$),

[16, 61, 235, 904, 3478, 13381, 51481, 198064, 762016, 2931721, 11279275]

{"000", "111", "222", "333"}, a(n) = 3 a($n-1$) + 3 a($n-2$),

[16, 60, 228, 864, 3276, 12420, 47088, 178524, 676836, 2566080, 9728748]

```
> MinRecRelAnyBase({"0000"},3);MinRecRelAnyBase({"0000",
  "1111"},3);MinRecRelAnyBase({"0000", "1111", "2222"},3);
```

{"0000"}, a(n) = 2 a($n-1$) + 2 a($n-2$) + 2 a($n-3$) + 2 a($n-4$),

[27, 80, 238, 708, 2106, 6264, 18632, 55420, 164844, 490320, 1458432]

{"0000", "1111"}, a(n) = 2 a($n-1$) + 2 a($n-2$) + 2 a($n-3$) + a($n-4$),

[27, 79, 233, 687, 2025, 5969, 17595, 51865, 152883, 450655, 1328401]

{"0000", "1111", "2222"}, a(n) = 2 a($n-1$) + 2 a($n-2$) + 2 a($n-3$),

[27, 78, 228, 666, 1944, 5676, 16572, 48384, 141264, 412440, 1204176]

So, if S is a set of s characters from a q-ary alphabet, then the recurrence relation for q-ary strings with no run of k identical characters from S, appears to be:

a(n) = (q - 1) a(n-1) + (q - 1) a(n-2) + . . . + (q - 1) a(n-k+1) + (q - s) a(n-k) .

This can be proved using a somewhat more involved argument than those presented earlier.

Now we are in a position to consider binary rectangles free of several two dimensional patterns. First consider the number of 2 x n binary rectangles that have no two ones side by side, horizontally or vertically. We can consider a 2 x n rectangle to be a length n string, over the alphabet:

```
> 0 = <0,0>,1=<1,0>,2=<0,1>;
```

$$0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, 1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, 2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Thus a 2 x n rectangle without two ones side by side corresponds to a string over the alphabet { 0, 1, 2 } that does not contain "11" or "22". So we have:

```
> MinRecRelAnyBase({"11", "22"},3);
```

{"11", "22"}, a(n) = 2 a($n-1$) + a($n-2$),

[3, 7, 17, 41, 99, 239, 577, 1393, 3363, 8119, 19601]

This is consistent with our observation that the recurrence relation for q-ary strings with no run of k identical characters from a set of s characters is:
a(n) = (q - 1) a(n-1) + (q - 1) a(n-2) + . . . + (q - 1) a(n-k+1) + (q - s) a(n-k) , with q = 3, k = 2, and s = 2.

Now we consider the number of 2 x n binary rectangles that have no two ones adjacent, horizontally, vertically, or diagonally. These may be identified with length n strings over the same alphabet. that do not contain "11", "12", "21", or "22". Hence we have:

```
> MinRecRelAnyBase({"11", "12", "21", "22"},3);
```

$$\{"11", "12", "21", "22"\},\ a(n) = a(n-1) + 2\,a(n-2),$$

$$[3, 5, 11, 21, 43, 85, 171, 341, 683, 1365, 2731]$$

After a moments thought we can find the following purely theoretical argument for this recurrence relation:
We may divide strings of length n over { 0, 1, 2 } into cases based on the first character.
If the 1st character is 0, the string must have the form: **"0 [any length n-1 string w/o 11, 12, 21, or 22]"** , hence a(n-1) ;
If the 1st character is 1, the string must have the form: **"1 0 [any length n-2 string w/o 11, 12, 21, or 22]"**, hence a(n-2) ;
If the 1st character is 2, the string must have the form: **"2 0 [any length n-2 string w/o 11, 12, 21, or 22]"**, hence a(n-2) ;
Summing these three cases yields: a(n) = a(n-1) + 2 a(n-2).

Now consider the number of 2 x n binary rectangles that have no 2 x 2 block of ones. These may be identified with length n strings over the larger alphabet

```
> 0 = <0,0>,1=<1,0>,2=<0,1>,3=<1,1>;
```

$$0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ 1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ 2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ 3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

that do not contain "33". Thus we have:

```
> MinRecRelAnyBase({"33"},4);
```

$$\{"33"\},\ a(n) = 3\,a(n-1) + 3\,a(n-2),$$

$$[4, 15, 57, 216, 819, 3105, 11772, 44631, 169209, 641520, 2432187]$$

This is an immediate consequence of the observation that the recurrence relation for q-ary strings with no run of k identical characters from a set of s characters is:
a(n) = (q - 1) a(n-1) + (q - 1) a(n-2) + . . . + (q - 1) a(n-k+1) + (q - s) a(n-k) , with q = 4, k = 2, and s = 1.

Carrying this idea further, consider the number of 2 x n binary rectangles that have no 2 x 2 block of ones or zeros. These may be identified with length n strings over the four character alphabet that do not contain "00" or "33". Thus we have:

```
> MinRecRelAnyBase({"00", "33"},4);
```

{"00", "33"}, $a(n) = 3\,a(n-1) + 2\,a(n-2)$,

[4, 14, 50, 178, 634, 2258, 8042, 28642, 102010, 363314, 1293962]

Again, however, this is already an immediate consequence of the observation that the recurrence relation for q-ary strings with no run of k identical characters from a set of s characters is:
a(n) = (q - 1) a(n-1) + (q - 1) a(n-2) + . . . + (q - 1) a(n-k+1) + (q - s) a(n-k) , this time with q = 4, k = 2, and s = 2.

Next we consider the number of 2 x n binary rectangles that have no 2 x 2 "checkerboard" block. These may be identified with length n strings over the four character alphabet that do not contain "12" or "21". Thus we have:

```
> MinRecRelAnyBase({"12", "21"},4);
```

{"12", "21"}, $a(n) = 3\,a(n-1) + 2\,a(n-2)$,

[4, 14, 50, 178, 634, 2258, 8042, 28642, 102010, 363314, 1293962]

It is interesting to note that this is exactly the same sequence as the previous example. However, it is not immediately obvious that avoiding "12" and "21" results from barring two repeated characters. We draw the connection as follows. We can transform any length n string over { 0, 1, 2, 3 } into another string over the same alphabet by applying the transposition permutation (0)(1, 2)(3) to the characters in the odd positions in the string. This mapping is a bijection and the original string will lack 12 and 21 if and only if its image lacks 11 and 22. Hence the number of strings without 12 or 21 is the same as the number without 11 or 22, which is the same as the number that lack 00 or 22, as in the previous example.

Unfortunately, the regularity of these recurrence relations and simple theoretical explanations does not continue into larger forbidden patterns or rectangles.

Consider the number of 2 x n binary rectangles that have no three ones in a row. These may be identified with length n strings over the alphabet {0, 1, 2, 3}, which we have used before, that do not contain "111", "113", "131", "133", "222", "223", "232", "233", "333", "331", "311", "332", "322", "313", or "323". Thus we have:

```
> MinRecRelAnyBase({"111", "113", "131", "133", "222",
  "223", "233", "232", "333", "331", "311", "332", "322",
  "313", "323"},4);
```

that do not contain "33", "37", "66", "67", "73", "76", or "77". Thus we have:

```
> MinRecRelAnyBase({"33", "37", "66", "67", "73", "76",
  "77"},8);
```

{"33", "37", "66", "67", "73", "76", "77"}, a(n) = 6 a($n-1$) + 10 a($n-2$) – 5 a($n-3$), [
8, 57, 417, 3032, 22077, 160697, 1169792, 8515337, 61986457, 451223152, 3284626797
]

3 x n rectangles with no 2 x 2 block of ones or zeros may be identified with length n strings over the same alphabet that do not contain "00", "01", "04", "10", "11", "40", "44", "33", "37", "66", "67", "73", "76", or "77". Thus we have:

```
> MinRecRelAnyBase({"00", "01", "04", "10", "11", "40",
  "44", "33", "37", "66", "67", "73", "76", "77"},8);
```

{"00", "01", "04", "10", "11", "33", "37", "40", "44", "66", "67", "73", "76", "77"},
a(n) = 6 a($n-1$) + 3 a($n-2$) – 2 a($n-3$),
[8, 50, 322, 2066, 13262, 85126, 546410, 3507314, 22512862, 144506294, 927561722]

3 x n rectangles with no 2 x 2 checkerboard may be identified with length n strings over the same eight character alphabet that do not contain "12", "16", "21", "24", "25", "34", "35", "42", "43", "52", "53", "56", "61", or "65". Thus we have:

```
> MinRecRelAnyBase({"12", "16", "21", "24", "25", "34",
  "35", "42", "43", "52", "53", "56", "61", "65"},8);
```

{"12", "16", "21", "24", "25", "34", "35", "42", "43", "52", "53", "56", "61", "65"},
a(n) = 6 a($n-1$) + 3 a($n-2$) – 2 a($n-3$),
[8, 50, 322, 2066, 13262, 85126, 546410, 3507314, 22512862, 144506294, 927561722]

Note that this is the same as the number of 3 x n rectangles free of a 2 x 2 block of zeros or ones. This connection can be proved by swapping occurrences of zero and one in checkerboarded positions of a 3 x n rectangle.

Unfortunately, while the Maple and the algorithm can be used to find recurrence relations associated with the number of binary rectangles that avoid particular patterns, the results do not, in general appear to admit simple explanations.