Math 192r, Problem Set #20: Solutions

1. *Let $f(\cdot)$ be some polynomial (in one variable), and define a sequence of rational functions $r_n = r_n(x, y)$ with the initial conditions $r_0 = x$, $r_1 = y$ and the recurrence $r_{n+2} = f(r_{n+1})/r_n$ $(n \geq 0)$. Here x and y are formal indeterminates, so you don't need to worry about ill-definedness arising from a vanishing denominator.*

   (a) *Find an f such that the sequence of polynomials r_n is periodic with period 5.*

   The polynomial $f(t) = 1 + t$ will do:

$$r_2 = (1 + r_1)/r_0 = (1 + y)/x,$$

$$r_3 = (1 + r_2)/r_1 = (1 + x + y)/xy,$$
$$r_4 = (1 + r_3)/r_2 = (1 + x)/y,$$
$$r_5 = (1 + r_4)/r_3 = x,$$
$$r_5 = (1 + r_5)/r_4 = y,$$

   etc.

   (b) *For which f does it seem to be the case that each of the rational functions $r_2, r_3, \ldots$ is a Laurent polynomial in x and y? Try to find necessary and sufficient conditions on f. (Proofs are not required for this part of the problem.)*

   I retracted this problem. One general pattern that appears is that if $f(t)$ is off the form $t^m + at^{m-1}$, then Laurentness seems to hold. However, I do not know necessary and sufficient conditions on $f()$ that predict exactly when the sequence $r_1, r_2, \ldots$ is an infinite sequence of Laurent polynomials.

   (c) *Find an f (of degree at least 3, and with at least two terms) for which the Laurentness property holds, such that the one-dimensional recurrence associated with f is a special case of a two-dimensional recurrence (analogous to frieze patterns or number walls) that also has the Laurentness property. (Proofs are not required for this part of the problem.)*

The one-dimensional recurrence associated with $f(t) = t^3 + 1$ is a special case of the two-dimensional recurrence

$$F(n,k) = (F(n-1,k-1)F(n-1,k)F(n-1,k+1)+1)/F(n-2,k),$$

which appears to satisfy the Laurentness property: if we put $F(0,k) = x_k$ and $F(1,k) = y_k$ for all integers k, then this recurrence gives us $F(2,k) = (y_{k-1}y_k y_{k+1}+1)/x_k$, $F(3,k) = (y_{k-2}y_{k-1}^2 y_k^3 y_{k+1}^2 y_{k+2} + y_{k-2}y_{k-1}^2 y_k^2 y_{k+1} + y_{k-2}y_{k-1}y_k^2 y_{k+1}y_{k+2} + y_{k-2}y_{k-1}y_k + y_{k-1}y_k^2 y_{k+1}^2 y_{k+2} + y_{k-1}y_k y_{k+1} + y_k y_{k+1} y_{k+2} + x_{k-1}x_k x_{k+1} + 1)/(x_{k-1}x_k x_{k+1} y_k)$, etc.

(d) *Find a two-variable polynomial $f(\cdot,\cdot)$ that genuinely involves both of its variables such that the sequence of rational functions r_n with the initial conditions $r_0 = x$, $r_1 = y$, $r_2 = z$ and the recurrence $r_{n+3} = f(r_{n+1}, r_{n+2})/r_n$ is not periodic, and such that each r_n is a Laurent polynomial in x, y and z. (Proofs are not required for this problem.)*

If we put $f(s,t) = st + 1$, then we get the recurrence $r_{n+3} = (r_{n+1}r_{n+2} + 1)/r_n$. Writing this as $r_n r_{n+3} - r_{n+1}r_{n+2} = 1$, we can see that (with a change of indexing) this is a special case of the recurrence we studied in the second problem of assignment 19.

2. *Given formal indeterminates $x_{i,j}$ and $y_{i,j}$, define $f(i,j,k)$ to be $x_{i,j}$ if $k = 0$ and $y_{i,j}$ if $k = 1$, and for $k > 1$ recursively define*

$$f(i,j,k) = \frac{f(i-1,j-1,k-1)f(i+1,j+1,k-1) + f(i-1,j+1,k-1)f(i+1,j-1,k-1)}{f(i,j,k-2)}.$$

(Note that this is Dodgson condensation with the minus-sign replaced by a plus-sign.)

(a) *Submit code that demonstrates that $f(i,j,k)$ is a Laurent polynomial in the x- and y-variables for $k = 2,3,4$, and that all coefficients in this Laurent polynomial equal $+1$. (To say that code "demonstrates" the truth of a proposition, I don't mean that it generates output which a human could look over in order to convince herself/himself that the proposition is true. I mean that the code evaluates a boolean expression that encodes the proposition in question, and the proposition evaluates to* `true`*.)*

We define $f(i,j,k)$ by

```
f := proc(i,j,k) option remember;
if k=0 then x(i,j); elif k=1 then y(i,j);
else simplify((f(i-1,j-1,k-1)*f(i+1,j+1,k-1)
+f(i-1,j+1,k-1)*f(i+1,j-1,k-1))/f(i,j,k-2));
fi; end;
```

Maple's way of simplifying a rational function is to write it as a ratio of two polynomials; this ratio can be written as a Laurent polynomial if and only if the denominator has one term in it. To determine this, we can use the `whattype` operator. If the denominator is a sum of two or more terms, its type will be `+`; otherwise its type will be `*` (if the denominator is a product of two or more factors) or `^` (if the denominator is a single factor raised to a power) or `function` (if the denominator is a single factor of the form $f(i, j, k)$ for some i, j, k). So a test of Laurentness would be

```
IsLaurent := proc(e) if type(denom(e),'+') then false;
else true fi; end;
```

and the commands

```
IsLaurent(f(i,j,2));
IsLaurent(f(i,j,3));
IsLaurent(f(i,j,4));
```

certify that $f(i, j, k)$ is a Laurent polynomial when $2 \leq k \leq 4$, by returning the value `true`.

As for the claim that all coefficients equal 1, we have to be a bit careful. In a simplified Maple polynomial (created with the Maple command `simplify`), each term could be a constant, a single variable, a single variable raised to some power, or a product of such things. If the product has two or more factors, and one of them is a number, it will appear as the first factor. So, the only way a monomial that occurs in $f(i, j, k)$ could have a coefficient different from 1 is if it is a fraction or an integer not equal to 1, or if it is a product in which the first factor is a fraction or an integer not equal to 1. We write:

```
IsMonic := proc(e)
if type(e,'*') then IsMonic(op(1,e));
```

```
elif type(e,fraction) then false;
elif type(e,integer) and e<>1 then false;
else true; fi; end;
```

(This code would need more cases to be useful in a broader setting, but for the purpose at hand it's adequate.) To handle all the coefficients of a polynomial, we write:

```
IsAllMonic := proc(e) local k,bit;
bit := true;
for k from 1 to nops(e) do
if not IsMonic(op(k,e)) then bit := false; break;
fi; od; bit; end;
```

(Note the use of the `break` command, to cause the execution of the loop to terminate once a non-monic term has been found.) The commands

```
IsAllMonic(numer(f(i,j,2)));
IsAllMonic(numer(f(i,j,3)));
IsAllMonic(numer(f(i,j,4)));
```

certify that all the coefficients of the Laurent polynomial $f(i,j,2)$, $f(i,j,3)$, $f(i,j,4)$ equal 1, by returning the value `true`.

(b) *Give a conjectural pairing between the terms of the Laurent polynomial $f(i,j,k)$ and domino tilings of the Aztec diamond of order $k-1$, and verify it for $k \leq 3$.*

Each domino tiling determines a Laurent monomial in the following way: Rotate the Aztec diamond by 45 degrees, so that its corners, instead of being north, west, south, and east, are northeast, northwest, southwest, and southeast. Associate the vertices inside the Aztec diamond with the variables $x_{i,j}$, $y_{i,j}$ in the fashion illustrated below for the case $n = 3$. Then for any tiling T, the exponent assigned to a variable v (in the Laurent monomial that represents T) is 1 minus the number of dominos in the tiling T that lie inside a 2-by-2 square centered at v. (Compare this with the rule we saw when studying domino tilings of a 2-by-$2n$

rectangle in assignment 17.)

$$\begin{array}{ccccc}
y_{-2,2} & & y_{0,2} & & y_{2,2} \\
& x_{-1,1} & & x_{1,1} & \\
y_{-2,0} & & y_{0,0} & & y_{2,0} \\
& x_{-1,-1} & & x_{1,-1} & \\
y_{-2,-2} & & y_{0,-2} & & y_{2,-2}
\end{array}$$

(c) *For $n \leq 6$, count how many terms there are in the Laurent polynomial obtained from $f(i,j,k)$ by replacing all the x-variables by 1. Repeat, this time instead replacing all the y-variables by 1.*

If after defining `f` we define `x:=1`, the command

```
seq(nops(numer(f(i,j,k))),k=2..6);
```

yields the numbers 2, 7, 42, 429, 7436. (Actually, that's only true in Maple 6; in Maple 5, the case $k = 6$ causes an object-too-large error.)

When doing this sort of thing (changing the meanings of symbols used in a program), we can run afoul of the fact that the `remember` option may cause the program to remember a value from a previous run that is no longer applicable. In cases like this, it's a good idea to redefine `f` whenever one of the symbols it depends on (`x` or `y` in this case) gets redefined; even though the new definition will agree with the old on a literal level, the fact that the definition is being re-assigned will cause Maple to relinquish what it thinks it "knows" about values of the function `f`.

To get Maple to help us with the second part of the problem, we can type `unassign('x'); y:=1` and then re-enter the definition of `f`. When we enter the command `seq(nops(numer(f(i,j,k))),k=2..6);` we get the numbers 1, 1, 2, 7, 42, 429.

It's not a coincidence that we see the same numbers for both questions. Indeed, the nth term of the sequence 1, 2, 7, 42, 429, ... is the number of alternating-sign matrices (ASMs) of order n, and in a very concrete way, every domino tiling of an Aztec diamond of order n results from superimposing an ASM of order n with a (compatible) ASM of order $n+1$.