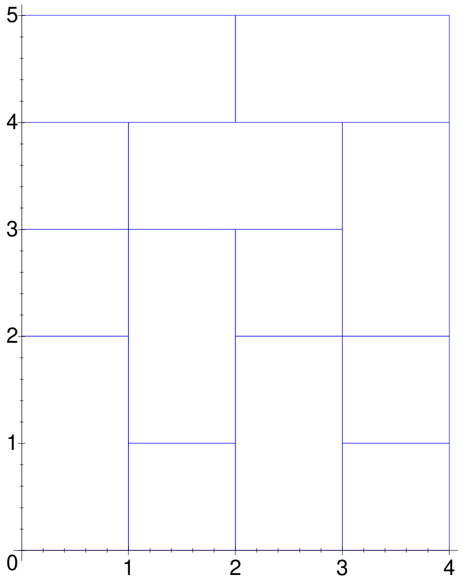
Monomer-Dimer Tilings

F. Cazals, December 1997.

A fundamental problem in lattice statistics is the monomer-dimer problem, in which the sites of a regular lattice are covered by non-overlapping monomers and dimers, that is squares and pairs of neighbor squares. An example of such a tiling for a *mxn* chessboard with $m = 4$ and $n = 5$ is depicted below. The relative number of monomers and dimers can be arbitrary or may be constrained to some density *p*, and the problem can be generalized to any fixed dimension *d*. This model was introduced long ago to investigate the properties of adsorbed diatomic molecules on a crystal surface [Rob35], and its three-dimensional version occurs in the theory of mixtures of molecules of different sizes [Gug52] as well as the cell cluster theory of the liquid state [CoAl55]. Practically, most of the thermodynamic properties of these physical systems can be derived from the number of ways a given lattice can be covered, so that a considerable attention has been devoted to this counting question. For any fixed dimension *d* and any monomer density *p*, a *provably good polynomial time approximation algorithm* is exposed in [KenAl95]. But exact counting results are still unknown even in dimension two.



The goal of this worksheet is to show that these questions are amenable to an automated computer algebra treatment which goes from the specifications of the coverings constructions in terms of Combstruct grammars, to the asymptotics using rational generating functions and the numeric-symbolic method exposed in [GoSa96]. In particular we shall be interested in enumerating the tilings for a vertical strip of constant width *m* in terms of multivariate rational generating functions, from which the average number of pieces or the expected proportions of the three types of pieces in a random tiling are easily derived.
This will also enable us to establish a *provably good* sequence of upper and lower bounds for the connectivity constant $\tau = \lim_{n \to \infty} g(n)^{\left(\frac{1}{n^2}\right)}$ where $g(n)$ counts the number of ways to tile an *nxn* cheesboard.

But before getting started, we need to load the Combstruct library, as well as the piece of code doing the asymptotics of rational fractions:

```
> with(combstruct): with(gfun): read `ratasympt.mpl`;read
  `./gfsolve.mpl`;
```

References
[CoAl55] E.G.D. Cohen et al., A cell-cluster theory for the liquid state II, Physica XXI, 1955.
[Fin97]S. Finch, Favorite Mathematical Constants, http://www.mathsoft.com/cgi-shl/constant.bat.
[GoSa96] X. Gourdon and B. Salvy, Effective Asymptotics of linear recurrences with rational coefficients, Discrete Mathematics, Vol. 153, 1996.
[Gug52] E.A. Guggenheim, Mixtures, Clarendon Press, 1952.
[Ken95] C. Kenyon et al., Approximating the number of Monomer-Dimer Coverings of a Lattice, Proc. of the 25th ACM STOC, 1993.
[Rob35]J.K. Robert, Some properties of adsorbed films of oxygen on tungsten, Proc. of the Royal Society of London, Vol. A 152, 1935.
[Sloa95] N.J.A. Sloane and S. Plouffe, The Encyclopedia of Integer Sequences, Academic Press, 1995.

A step-by-step example

We first observe that the number T_n counting the different tilings of a vertical slice of width 1 has a well known expression: since height n can be reached from height $n-1$ by adding a monomer and from height $n-2$ with a vertical dimer, we have $T_n = T_{n-1} + T_{n-2}$ with $T_0 = 1$, $T_1 = 1$, that is the Fibonacci recurrence. This can be checked directly with Combstruct:

```
> TGr:={T=Sequence(Union(monomer,dimer)),monomer=Z,dimer=Prod(Z
  ,Z)};
```

$$TGr := \{ T = \mathrm{Sequence}(\mathrm{Union}(monomer, dimer)), monomer = Z, dimer = \mathrm{Prod}(Z, Z) \}$$

And we can retrieve the corresponding rational Generating Function with gfsolve:

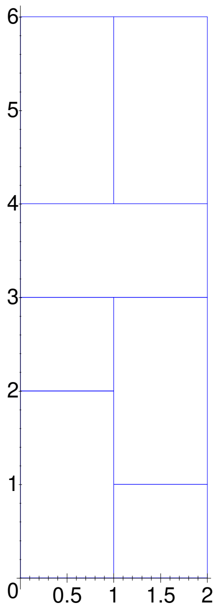
```
> gfsolve(TGr,unlabelled, z);
```

$$\{ \mathrm{Z}(z) = z, \mathrm{monomer}(z) = z, \mathrm{dimer}(z) = z^2, \mathrm{T}(z) = -\frac{1}{-1 + z + z^2} \}$$

More interesting is the case m=2 which we examine examine now.

Tiling a slice of width $m = 2$

An example covering of a 2x6 lattice is depicted below. If we draw a horizontal line at height 0, it turns out that we do not ‘cut’ any piece, which we encode by MM. At height 1, we cut the lefmost vertical dimer but just touch the monomer topmost side, which we encode by PM. At height 2 the leftmost P turned into an M since we now touch the dimer boundary, while on the right side we added a dimer and have a P. More generally, we shall *assign to each height of the construction containing a monomer or dimer boundary* a word of length m on the alphabet $\{ M, P \}$ as follows: the ith digit of the word is P if an horizontal line at this particular height splits a vertical domino located in the ith column, and M otherwise. To summarize our example we therefore have MM, PM, MP, MM, MM,MM at the heights 0,1,2,3,4,6. (BTW, M stands for Minus and P for Plus!)



This encoding is not one-to-one since whenever we find two consecutive Ms, we do not know wether they are on top of two monomers or of a horizontal dimer. But it is sufficient to incrementally build all the possible configurations by recording the status of the *fringe*. If $m = 2$, the possible fringes are MM,MP,PM and each of them can be derived from a combination of the others and of monomers and dimers. For example, the configuration MM can be reached in 5 different ways by:

-stacking a horizontal dimer H, two monomers C,C, or two vertical dimers V,V on top of a MM configuration,

-adding a monomer C to the right column of a PM configuration or to the left one of a MP.

The remaining transitions follow similar rules. And in order to characterize the ordinate reached by the construction, we can mark the height reached by the bottommost piece whose elevation gain is 1 or 2 at each step of the construction. Putting everything together and associating the symbols H, V, C and S to the number of horizontal dimers, vertical dimers, monomers and the height yields the following Combstruct grammar:

```
> Gr2:={MM=Union(Epsilon,Prod(S, MM, H), Prod(S, MM, C,C),
                     Prod(S,PM, C),Prod(S,MP,
  C),Prod(S,S,MM,V,V)),
           PM=Union(Prod(S,MM, V, C), Prod(S,MP,V)),
           MP=Union(Prod(S,MM, C, V), Prod(S,PM,V)),
           H=Epsilon,V=Epsilon,C=Epsilon,S=Atom}:
```

The ordinary generating functions can be derived by Combstruct[gfsolve]:

```
> GF2Sys:=gfsolve(Gr2, unlabelled, z, [[h,H], [v,V],
  [c,C]]);
```

$$GF2Sys := \{\mathrm{PM}(z,h,v,c) = \frac{z\,v\,c}{\%1},\ \mathrm{MP}(z,h,v,c) = \frac{z\,v\,c}{\%1},\ \mathrm{MM}(z,h,v,c) = -\frac{z\,v-1}{\%1},$$

$$\mathrm{C}(z,h,v,c) = c,\ \mathrm{H}(z,h,v,c) = h,\ \mathrm{V}(z,h,v,c) = v,\ \mathrm{S}(z,h,v,c) = z\}$$

$$\%1 := -z\,v + 1 + h\,z^2\,v - z\,h - z^2\,c^2\,v - z\,c^2 + z^3\,v^3 - z^2\,v^2$$

Furthermore we can isolate the GF corresponding to the MM fringes; the coefficient of $z^n\,h^i\,v^j\,c^l$ in this GF counts the number of ways to tile a chessboard $2\,xn$ with respectively j, k and l horizontal and vertical dimers and monomers:

```
> GF2:=subs(GF2Sys,MM(z,h,v,c));
```

$$GF2 := -\frac{z\,v-1}{-z\,v+1+h\,z^2\,v-z\,h-z^2\,c^2\,v-z\,c^2+z^3\,v^3-z^2\,v^2}$$

The number of configurations up to a given height independently of the number and kind of pieces used can be retrieved by erasing the dimers and monomers markers followed by a Taylor expansion:

```
> GF2h:=subs([h=1,v=1,c=1],GF2);series(GF2h,z=0,11);
```

$$GF2h := -\frac{z-1}{-3\,z+1-z^2+z^3}$$

$$1+2\,z+7\,z^2+22\,z^3+71\,z^4+228\,z^5+733\,z^6+2356\,z^7+7573\,z^8+24342\,z^9+78243\,z^{10}+\mathrm{O}(z^{11})$$

This sequence does not appear in [Sloa95]. It can be checked that these values match those computed directly from the grammer by Combstruct[count]:

```
> seq(count([MM,Gr2], size=i), i=0..10);
```

$$1, 2, 7, 22, 71, 228, 733, 2356, 7573, 24342, 78243$$

Another way to compute the exact number of tilings for large values of n is through the recurrence equation satisfied by the Taylor coefficients and computed by gfun[diffeqtorec]:

```
> diffeqtorec(y(z)-GF2h,y(z),u(n));
```

$$\{\mathrm{u}(n)-\mathrm{u}(n+1)-3\,\mathrm{u}(n+2)+\mathrm{u}(n+3),\ \mathrm{u}(0)=1,\ \mathrm{u}(1)=2,\ \mathrm{u}(2)=7\}$$

```
> p2:=rectoproc(",u(n)):
> for i from 1 to 10 do i,p2(i) od;
```

1, 2

2, 7

3, 22

4, 71

5, 228

6, 733

7, 2356

8, 7573

9, 24342

10, 78243

For example:

```
> p2(1000);evalf(");
```

815888066415607016906952404112307595151512759687748484936134195851 69\
068227333558126428001382175144687730963257784525280829122573218066 04\
921460098220012515791506449413777539281203616366978286496772287518 85\
765204525356935063173344728895706423812409533412081112594313161139 94\
658744317182852871108762302870178745947878236817953193772066640148 32\
468621873354064032355390570879129423719987103789535781625389457675 35\
336756272709110771725017725079954729083470771691615025972534483749 53\
76786431924372193442043699516 85

$$.8158880664\ 10^{507}$$

But as we shall see now, asymptotic estimates can be derived much faster.

Asymptotic estimates of the number of tilings

We have just seen that the number of configurations is encoded by the rational generating function GF2h(z). An elegant way to access its Taylor coefficients is therefore through a full partial fraction decomposition yielding linear denominators:

```
> fpf:=convert(GF2h,fullparfrac,z);
```

$$fpf := \sum_{\_\alpha = \%1} \frac{-\frac{8}{37}\_\alpha + \frac{7}{74}\_\alpha^2 - \frac{11}{74}}{z - \_\alpha}$$

$$\%1 := \mathrm{RootOf}(-3\,\_Z + 1 - \_Z^2 + \_Z^3)$$

The term in z^n comes from the contributions of the roots of $Z^3 - Z^2 - 3\,Z + 1 = 0$ in the expansion of

```
> el:=op(1,fpf);
```

$$el := \frac{-\frac{8}{37}\_\alpha + \frac{7}{74}\_\alpha^2 - \frac{11}{74}}{z - \_\alpha}$$

and since there are 3 singularities, the main asymptotic contribution comes from the one with smallest modulus:

```
> fsolve(-3*_Z+1+_Z^3-_Z^2,_Z);
```

$$-1.481194304, .3111078175, 2.170086487$$

```
> root1:=RootOf(-3*_Z+1+_Z^3-_Z^2,.3111078175);
  root2:=RootOf(-3*_Z+1+_Z^3-_Z^2,-1.481194304);
  root3:=RootOf(-3*_Z+1+_Z^3-_Z^2,2.170086487);
```

$$root1 := \mathrm{RootOf}(-3\,\_Z + 1 - \_Z^2 + \_Z^3, .3111078175)$$

$$root2 := \mathrm{RootOf}(-3\,\_Z + 1 - \_Z^2 + \_Z^3, -1.481194304)$$

$$root3 := \mathrm{RootOf}(-3\,\_Z + 1 - \_Z^2 + \_Z^3, 2.170086487)$$

On this example the dominant pole is clearly .31 so that the main contribution is encoded by:

```
> el1:=subs(_alpha=root1,el);
```

$$el1 := \left(-\frac{8}{37}\mathrm{RootOf}(-3\,\_Z + 1 - \_Z^2 + \_Z^3, .3111078175) + \frac{7}{74}\mathrm{RootOf}(-3\,\_Z + 1 - \_Z^2 + \_Z^3, .3111078175)^2 - \frac{11}{74}\right) \Big/ (z - \mathrm{RootOf}(-3\,\_Z + 1 - \_Z^2 + \_Z^3, .3111078175))$$

```
> evalf(");
```

$$-\frac{.2067595751}{z - .3111078175}$$

Extracting the term in z^n in the previous expression produces the estimate:

```
> es2:=n->.2067595751*(1/.3111078175)^(n+1);
```

$$es2 := n \to .2067595751\ 3.214319743^{(n+1)}$$

```
> seq(es2(i),i=1..10);
```

2.136209208, 6.866459431, 22.07099612, 70.94323855, 228.0342523, 732.9749992, 2356.016011, 7572.988780, 24342.00736, 78242.99481

To sum up, from the rational generating function we have:
-performed a full partial fraction decomposition,

-computed the singularities and sorted them by increasing moduli,
-extracted the contribution of the singularity with smallest modulus.

The key step consists in deciding which are the singularity (ies) with smallest modulus (i), and can be performed numerically using properties of polynomials with integer coefficients --see [GoSa96]. This is implemented by the **ratasympt** function --whose optional 4th argument corresponds to the number of singularity layers the user wants to take into account. In particular to retrieve the main contribution, one writes:

```
> layer1:=ratasympt(GF2h,z,n,1);nbCfs1:=evalf(layer1);
```

$$layer1 := -\left(-\frac{8}{37}\mathrm{RootOf}(-3\_Z+1-\_Z^2+\_Z^3, .311107817466) + \frac{7}{74}\mathrm{RootOf}(-3\_Z+1-\_Z^2+\_Z^3, .311107817466)^2 - \frac{11}{74}\right) \Big/ \mathrm{RootOf}(-3\_Z+1-\_Z^2+\_Z^3, .311107817466)^{(n+1)}$$

$$nbCfs1 := \frac{.2067595751}{.3111078175^{(n+1.)}}$$

And to take into account all the layers:

```
> layers:=ratasympt(GF2h,z,n);nbCfs:=evalf(layers);
```

$$layers := -\left(-\frac{8}{37}\mathrm{RootOf}(-3\_Z+1-\_Z^2+\_Z^3, .311107817466) + \frac{7}{74}\mathrm{RootOf}(-3\_Z+1-\_Z^2+\_Z^3, .311107817466)^2 - \frac{11}{74}\right) \Big/ \mathrm{RootOf}(-3\_Z+1-\_Z^2+\_Z^3, .311107817466)^{(n+1)} - \left(-\frac{8}{37}\mathrm{RootOf}(-3\_Z+1-\_Z^2+\_Z^3, \text{-}1.48119430409) + \frac{7}{74}\mathrm{RootOf}(-3\_Z+1-\_Z^2+\_Z^3, \text{-}1.48119430409)^2 - \frac{11}{74}\right) \Big/ \mathrm{RootOf}(-3\_Z+1-\_Z^2+\_Z^3, \text{-}1.48119430409)^{(n+1)} - \left(-\frac{8}{37}\mathrm{RootOf}(-3\_Z+1-\_Z^2+\_Z^3, 2.17008648663) + \frac{7}{74}\mathrm{RootOf}(-3\_Z+1-\_Z^2+\_Z^3, 2.17008648663)^2 - \frac{11}{74}\right) \Big/ \mathrm{RootOf}(-3\_Z+1-\_Z^2+\_Z^3, 2.17008648663)^{(n+1)}$$

$$nbCfs := \frac{.2067595751}{.3111078175^{(n+1.)}} - \frac{.3791441193}{(\text{-}1.481194304)^{(n+1.)}} + \frac{.1723845440}{2.170086487^{(n+1.)}}$$

We can check that the second approximation is more accurate:

```
> evalf(seq(subs(n=i, layer1), i=1..10));
```

2.136209208, 6.866459430, 22.07099611, 70.94323856, 228.0342523, 732.9749992, 2356.016012, 7572.988781, 24342.00735, 78242.99482

```
> evalf(seq(subs(n=i, layers), i=1..10));
```

2.000000000, 6.999999995, 21.99999998, 70.99999995, 227.9999998, 732.9999991, 2355.999997, 7572.999991, 24341.99996, 78242.99989

```
> seq(p2(i),i=1..10);
```

2, 7, 22, 71, 228, 733, 2356, 7573, 24342, 78243

The proportion of monomers and dimers

We now address the computation of the average number of pieces in a random tiling. From the multivariate generating function GF2(z, h, v, c) we can merge the three types of pieces as follows:

```
> GF2;stij:=subs([h=t,v=t,c=t], GF2);
```

$$-\frac{z\,v-1}{-z\,v+1+h\,z^2\,v-z\,h-z^2\,c^2\,v-z\,c^2+z^3\,v^3-z^2\,v^2}$$

$$stij := -\frac{z\,t-1}{-2\,z\,t+1-z^2\,t^3-z\,t^2+z^3\,t^3}$$

The coefficient of $z^i\,t^j$ in *stij* counts the number of tilings at height i with exactly j pieces of any type. To get the total number of pieces we just have to compute the derivative with respect to t and substitute $t = 1$:

```
> sstij:=subs(t=1, diff(stij,t));
```

$$sstij := \frac{(z-1)\,(-4\,z-3\,z^2+3\,z^3)}{(-3\,z+1-z^2+z^3)^2}-\frac{z}{-3\,z+1-z^2+z^3}$$

For example, the total number of dimers and monomers used in all the configurations tilling the square 2 *x2* is 20:

```
> series(",z=0,5);
```

$$3\,z+20\,z^2+94\,z^3+402\,z^4+\mathrm{O}(z^5)$$

As before, we can compute an estimate of the total number of pieces in all the configurations at a given height:

```
> ratasympt(sstij,z,n,1);
```

$$\frac{\left(\frac{7}{74}\,\%1^2+\frac{5}{37}-\frac{13}{74}\,\%1\right)(n+1)}{\%1^{(n+2)}}-\frac{-\frac{376}{1369}\,\%1+\frac{551}{2738}\,\%1^2-\frac{1035}{2738}}{\%1^{(n+1)}}$$
$$-\frac{\frac{39}{74}\,\%1-\frac{12}{37}\,\%1^2+\frac{43}{74}}{\%1^{(n+1)}}$$

$$\%1 := \mathrm{RootOf}(-3\,\_Z+1-\_Z^2+\_Z^3,\ .311107817466)$$

```
> nBDPiecesN:=evalf(");
```

$$nBDPiecesN := .08963668765\,\frac{n+1.}{.3111078175^{(n+2.)}}-\frac{.2696705336}{.3111078175^{(n+1.)}}$$

So that the average number of pieces is asymptotically equivalent to:

```
> avNbD:=expand(nBDPiecesN/nbCfs1);
```

$$avNbD := 1.393507288\,n+.08923621157$$

And the average number of pieces per layer in a tiling of height n is therefore:

```
> asympt("/n,n);
```

$$1.393507288+\frac{.08923621157}{n}$$

The number of occurrences and the proportions of dimers and monomers can be computed in the same way by erasing the irrelevant indeterminates:

```
> pieceProportion:=proc(MGF, keptPiece)
   local forSubs, stij, sstij, nbp;

   forSubs:={h=1,v=1,c=1} minus {keptPiece=1};
   stij:=subs([op(forSubs)], MGF);
   sstij:=subs(keptPiece=1, diff(stij,keptPiece));
   nbp:=evalf(ratasympt(sstij,z,n,1));
   asympt(nbp/nBDPiecesN,n,2)
  end:
```

And we end up with:

```
> nbh:=pieceProportion(GF2,h);
  nbv:=pieceProportion(GF2,v);
  nbc:=pieceProportion(GF2,c);
```

$$nbh := .1483735155 + \mathrm{O}\left(\frac{1}{n}\right)$$

$$nbv := .2868539972 + \mathrm{O}\left(\frac{1}{n}\right)$$

$$nbc := .5647724884 + \mathrm{O}\left(\frac{1}{n}\right)$$

Plotting routines archive

The figures above were plotted with the following functions:

```
> dominoH:=proc(x,y) [[x,y], [x+2,y], [x+2,y+1], [x,y+1],
  [x,y]] end:
  dominoV:=proc(x,y) [[x,y], [x+1,y], [x+1,y+2], [x,y+2],
  [x,y]] end:
  dominoC:=proc(x,y) [[x,y], [x+1,y], [x+1,y+1], [x,y+1],
  [x,y]] end:
> plot([dominoV(0,0), dominoC(0,2),dominoC(0,3),
        dominoC(1,0),dominoV(1,1),dominoH(1,3),
        dominoV(2,0),dominoC(2,2),
        dominoC(3,0),dominoC(3,1),dominoV(3,2),

  dominoH(0,4),dominoH(2,4)],scaling=constrained,color=blue)
  ;
>
> plot([dominoV(0,0),
  dominoC(1,0),dominoV(1,1),dominoC(0,2),dominoH(0,3),domino
  V(0,4),dominoV(1,4)], scaling=constrained,color=blue);
```

Automatic counting in a slice of width *m*

Computing the generating functions

We now show how to automate the previous computations for any integer m. The first task consists in generating the $2^m - 1$ words on the binary alphabet $\{M, P\}$, and this is easily done with a Combstruct grammar as follows:

```
> allMPWords:=proc(m::integer)
   local i, MPGr, mps1, mps2, Pm;
```

```
  MPGr:={AllMP=Sequence(MP), MP=Union(M,P), M=Atom,
 P=Atom};
  mps1:=allstructs([AllMP, MPGr], size=m);
  mps2:=convert(map(proc(x) cat(op(x)) end, mps1), set);
  Pm:=cat(seq(P,i=1..m));
  [op(mps2 minus {Pm})]
 end:
```

For example if $m = 3$:

```
> allMPWords(3);
```

$$[PMM, MMP, MMM, PPM, MPP, MPM, PMP]$$

More interesting is the generation of the transitions between these words. Let *pattern* be one of them and suppose we want to figure out all the fringes *pattern* can be derived from. Suppose for example the *i*th letter of *pattern* is a *P*; this means that the *i*th letter of the fringe *pattern* was derived from was *M* and that a vertical dimer was put on top of this *M*. Similar rules applies if the *i*th digit is a *M*. And since the letter of a given fringe are independent --except for two consecutive *M*s that may come from an horizontal dimer, it suffices to recursively examine the digits from left to right as follows:

```
> #--pattern is the fringe to be built, e.g. MMPMM
 recComesFrom:=proc(pattern::string, idx::integer,
 prefix::string, mul::list, result::table)
  local prodRes, m, Mm, Pm;

  if (idx>length(pattern)) then #--stores the result into
 an indexed table
   prodRes:=Prod(S,prefix, op(mul));
   if not assigned(result[pattern]) then
 result[pattern]:={prodRes}
   else
 result[pattern]:=result[pattern] union {prodRes}
   fi
  else
   #--we examine the idx^{th} letter of the target
   if substring(pattern,idx)=P then
    recComesFrom(pattern, idx+1, cat(prefix,M), [op(mul),
 V], result)
   else #target=M
    recComesFrom(pattern, idx+1, cat(prefix,P), mul,
 result);
    recComesFrom(pattern, idx+1, cat(prefix,M), [op(mul),
 C], result);

    #--we may have MM=Prod(MM,H)
    if (length(pattern)>idx) and
 (substring(pattern,idx+1)=M) then
     recComesFrom(pattern, idx+2, cat(prefix,M,M),
 [op(mul), H], result)
    fi
   fi
  fi;

  #--some extra work for M^m
  m:=length(pattern);
```

```
    Mm:=cat(seq(M,i=1..m));
    if pattern=Mm then
     Pm:=cat(seq(P,i=1..m));
     result[Mm]:=result[Mm] minus {Prod(S,Pm)}
                            union
  {Epsilon,Prod(S,S,Mm,seq(V,i=1..m))}
    fi
  end:
```

Here is the table for $m = 3$:

```
> table3:=table():for i in allMPWords(3) do recComesFrom(i,
  1, ``, [], table3) od:print(table3);
```

table([

$MPM = \{\mathrm{Prod}(S, PMP, V), \mathrm{Prod}(S, MMM, C, V, C), \mathrm{Prod}(S, MMP, C, V), \mathrm{Prod}(S, PMM, V, C)\}$

$MMM = \{\mathrm{E}, \mathrm{Prod}(S, S, MMM, V, V, V), \mathrm{Prod}(S, PPM, C), \mathrm{Prod}(S, PMM, C, C), \mathrm{Prod}(S, PMP, C), \mathrm{Prod}(S, MPM, C, C), \mathrm{Prod}(S, MPP, C), \mathrm{Prod}(S, PMM, H), \mathrm{Prod}(S, MMP, C, C), \mathrm{Prod}(S, MMM, C, C, C), \mathrm{Prod}(S, MMM, C, H), \mathrm{Prod}(S, MMP, H), \mathrm{Prod}(S, MMM, H, C)\}$

$PMP = \{\mathrm{Prod}(S, MMM, V, C, V), \mathrm{Prod}(S, MPM, V, V)\}$

$PPM = \{\mathrm{Prod}(S, MMM, V, V, C), \mathrm{Prod}(S, MMP, V, V)\}$

$PMM = \{\mathrm{Prod}(S, MPP, V), \mathrm{Prod}(S, MPM, V, C), \mathrm{Prod}(S, MMM, V, C, C), \mathrm{Prod}(S, MMM, V, H), \mathrm{Prod}(S, MMP, V, C)\}$

$MPP = \{\mathrm{Prod}(S, MMM, C, V, V), \mathrm{Prod}(S, PMM, V, V)\}$

$MMP = \{\mathrm{Prod}(S, PMM, C, V), \mathrm{Prod}(S, MPM, C, V), \mathrm{Prod}(S, MMM, C, C, V), \mathrm{Prod}(S, MMM, H, V), \mathrm{Prod}(S, PPM, V)\}$

])

The tables entries are merged as follows:

```
> setGrammarFromTable:=proc(aTable)
    local aList, transitions, x;
    aList:=op(op(aTable));#--[a={Prod(...), Prod(...)}, ...]
    transitions:=seq(op(1,x)=Union(op(op(2,x))), x=aList);
    {transitions} union
  {H=Epsilon,V=Epsilon,C=Epsilon,S=Atom}
  end:
```

This yields the grammar:

```
> Gr3:=setGrammarFromTable(table3);
```

$Gr3 := \{MMP = \mathrm{Union}(\mathrm{Prod}(S, PMM, C, V), \mathrm{Prod}(S, MPM, C, V), \mathrm{Prod}(S, MMM, C, C, V), \mathrm{Prod}(S, MMM, H, V), \mathrm{Prod}(S, PPM, V)), PMM = \mathrm{Union}(\mathrm{Prod}(S, MPP, V), \mathrm{Prod}(S, MPM, V, C), \mathrm{Prod}(S, MMM, V, C, C), \mathrm{Prod}(S, MMM, V, H), \mathrm{Prod}(S, MMP, V, C)),$

$MPP = \mathrm{Union}(\mathrm{Prod}(S, MMM, C, V, V), \mathrm{Prod}(S, PMM, V, V)),$

$PMP = \mathrm{Union}(\mathrm{Prod}(S, MMM, V, C, V), \mathrm{Prod}(S, MPM, V, V)),$

$PPM = \mathrm{Union}(\mathrm{Prod}(S, MMM, V, V, C), \mathrm{Prod}(S, MMP, V, V)), MPM = \mathrm{Union}(\mathrm{Prod}(S, PMP, V), \mathrm{Prod}(S, MMM, C, V, C), \mathrm{Prod}(S, MMP, C, V), \mathrm{Prod}(S, PMM, V, C)), MMM = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(S, S, MMM, V, V, V),$

$\mathrm{Prod}(S, PPM, C), \mathrm{Prod}(S, PMM, C, C), \mathrm{Prod}(S, PMP, C), \mathrm{Prod}(S, MPM, C, C),$
$\mathrm{Prod}(S, MPP, C), \mathrm{Prod}(S, PMM, H), \mathrm{Prod}(S, MMP, C, C), \mathrm{Prod}(S, MMM, C, C, C),$
$\mathrm{Prod}(S, MMM, C, H), \mathrm{Prod}(S, MMP, H), \mathrm{Prod}(S, MMM, H, C)), H = \mathrm{E}, V = \mathrm{E}, C = \mathrm{E},$
$S = Atom\}$

This is solved as usual:

```
> MM3GFSys:=gfsolve(Gr3, unlabelled, z, [[h,H], [v,V],
  [c,C]]);
  MM3GF:=subs(MM3GFSys,MMM(z,h,v,c));
```

$$MM3GFSys := \{\mathrm{PMM}(z, h, v, c) = \frac{z\,v\,(-z\,c\,v^2 - c^2 - h + z^3\,v^5\,c + z^2\,v^3\,h - z\,c^3\,v)}{\%1},$$

$$\mathrm{C}(z, h, v, c) = c, \mathrm{H}(z, h, v, c) = h, \mathrm{S}(z, h, v, c) = z, \mathrm{V}(z, h, v, c) = v,$$

$$\mathrm{PMP}(z, h, v, c) = -\frac{z\,c\,v^2\,(1 - z^2\,v^3 + z^3\,v^4\,c - z^2\,c^2\,v^2 + 2\,z^2\,v^2\,h)}{\%1},$$

$$\mathrm{MPP}(z, h, v, c) = -\frac{z\,v^2\,(c - z^2\,v^3\,c + z^3\,v^4\,c^2 - z^2\,c^3\,v^2 + z\,h\,v - z^3\,v^4\,h)}{\%1},$$

$$\mathrm{PPM}(z, h, v, c) = -\frac{z\,v^2\,(c - z^2\,v^3\,c + z^3\,v^4\,c^2 - z^2\,c^3\,v^2 + z\,h\,v - z^3\,v^4\,h)}{\%1},$$

$$\mathrm{MMM}(z, h, v, c) = -\frac{1 - 2\,z^2\,v^3 - z\,c\,v + z^4\,v^6 + z^3\,v^4\,c - 2\,z^2\,c^2\,v^2}{\%1},$$

$$\mathrm{MPM}(z, h, v, c) = \frac{z\,v\,c\,(-c - z\,v^2 + z^3\,v^5 - z\,c^2\,v - 2\,z\,h\,v)}{\%1},$$

$$\mathrm{MMP}(z, h, v, c) = \frac{z\,v\,(-z\,c\,v^2 - c^2 - h + z^3\,v^5\,c + z^2\,v^3\,h - z\,c^3\,v)}{\%1}\}$$

$$\%1 := -1 + 3\,z^2\,v^3 + 5\,z^2\,c^2\,v^2 - 3\,z^4\,v^6 + z\,c\,v - 2\,z^3\,v^4\,c - 2\,z^4\,c^4\,v^4 + 2\,z\,c\,h + z^3\,c^3\,v^3$$
$$+ 2\,z^4\,v^4\,h\,c^2 + z^5\,v^6\,c^3 - 5\,c^2\,z^4\,v^5 - 2\,z^5\,v^6\,c\,h + 2\,c^4\,z^2\,v + z^5\,c\,v^7 - 2\,z^4\,v^4\,h^2$$
$$+ 2\,c^2\,z^2\,h\,v + z^6\,v^9 + z\,c^3 + 2\,z^2\,h^2\,v + z^3\,c^5\,v^2$$

$$MM3GF := -(1 - 2\,z^2\,v^3 - z\,c\,v + z^4\,v^6 + z^3\,v^4\,c - 2\,z^2\,c^2\,v^2) \Big/ (-1 + 3\,z^2\,v^3$$
$$+ 5\,z^2\,c^2\,v^2 - 3\,z^4\,v^6 + z\,c\,v - 2\,z^3\,v^4\,c - 2\,z^4\,c^4\,v^4 + 2\,z\,c\,h + z^3\,c^3\,v^3 + 2\,z^4\,v^4\,h\,c^2$$
$$+ z^5\,v^6\,c^3 - 5\,c^2\,z^4\,v^5 - 2\,z^5\,v^6\,c\,h + 2\,c^4\,z^2\,v + z^5\,c\,v^7 - 2\,z^4\,v^4\,h^2 + 2\,c^2\,z^2\,h\,v + z^6\,v^9$$
$$+ z\,c^3 + 2\,z^2\,h^2\,v + z^3\,c^5\,v^2)$$

Putting everything together, we end up with a procedure which takes *m* as entry and returns the grammar:

```
> getGrammar:=proc(m::integer)
   local i, MPTable;

  MPTable:=table();
  for i in allMPWords(m) do recComesFrom(i,1,``,[],MPTable)
  od;
  setGrammarFromTable(MPTable)
  end:

  getMmGFun:=proc(m::integer)
   local i, MPTable,Grm,MMmGFSys;
```

```
Grm:=getGrammar(m);
MMmGFSys:=gfsolve(Grm, unlabelled, z, [[h,H], [v,V],
[c,C]]);
subs(MMmGFSys,cat(seq(M,i=1..m))(z,h,v,c));
end:
```

The computation to be carried out being quite heavy for 4-variate generating functions, we can alleviate it be keeping only the markers for the total number of pieces and the height:

```
> getMmGFunZ:=proc(m::integer)
   local i, MPTable,Grm,GrmM,MMmGFSys;

  MPTable:=table();
  for i in allMPWords(m) do recComesFrom(i,1,``,[],MPTable)
  od;
  Grm:=setGrammarFromTable(MPTable);
  MMmGFSys:=gfsolve(Grm, unlabelled, z);
  subs(MMmGFSys,cat(seq(M,i=1..m))(z));
  end:
```

Asymptotics

We can now compute the generating functions for small values of *m*:

```
> gf:='gf':
> for i from 1 to 5 do
  i,time(assign(gf[i],getMmGFunZ(i))),gf[i]  od;
```

$$1, .553, -\frac{1}{-1+z^2+z}$$

$$2, .554, -\frac{z-1}{-3\,z+1-z^2+z^3}$$

$$3, 3.593, -\frac{z^4+z^3-4\,z^2-z+1}{14\,z^2-1+4\,z+z^6-10\,z^4}$$

$$4, 16.026, -\frac{1-4\,z-15\,z^2+20\,z^3+z^7-11\,z^5-2\,z^6+10\,z^4}{z^9-z^8-23\,z^7+29\,z^6+91\,z^5-111\,z^4-41\,z^3+41\,z^2+9\,z-1}$$

$$5, 87.973, -(z^{18}+2\,z^{17}-45\,z^{16}-68\,z^{15}+654\,z^{14}+870\,z^{13}-3820\,z^{12}-4700\,z^{11} +9255\,z^{10}+9448\,z^9-11175\,z^8-7532\,z^7+6956\,z^6+1994\,z^5-1794\,z^4-88\,z^3 +113\,z^2+6\,z-1)\big/(z^{20}+2\,z^{19}-65\,z^{18}-140\,z^{17}+1281\,z^{16}+2538\,z^{15}-10366\,z^{14} -17604\,z^{13}+38553\,z^{12}+50158\,z^{11}-73623\,z^{10}-60482\,z^9+74665\,z^8+26564\,z^7 -35106\,z^6-898\,z^5+4757\,z^4+16\,z^3-229\,z^2-14\,z+1)$$

For bigger ones, the grammar size, that is 2^m-1, inherently yields a linear system $(2^m-1)\,x\,(2^m-1)$ with large coefficients whose resolution is very much time consuming. So that for $6 \le m$, a better alternative to running getMmGFunZ(m) is to retrieve the result in the archive below!

From these generating functions we can easily isolate the main contribution to the asymptotic equivalent with the ratasympt procedure:

```
> asGf:='asGf':
> for i from 1 to 6 do
  assign(asGf[i],ratasympt(gf[i],z,n,1)),evalf(asGf[i]) od;
```

$$\frac{.4472135955}{.6180339887^{(n+1.)}}$$

$$\frac{.2067595751}{.3111078175^{(n+1.)}}$$

$$\frac{.08874224656}{.1609769304^{(n+1.)}}$$

$$\frac{.03918944864}{.08292494619^{(n+1.)}}$$

$$\frac{.01715071699}{.04274350262^{(n+1.)}}$$

$$\frac{.007526396801}{.02203003755^{(n+1.)}}$$

It should be observed that these estimates correspond to huge expressions. For $m = 5$ for example:

```
> asGf[5];
```

$$-\Bigg(-\frac{46317991073918998121720552258680932532925588451820209990220301070039628171861928513419898232438907}{120850558058521484597383085076949653932773755377203083325329465551645583169804908047083249525312263}\%1$$

$$-\frac{931682897453087119538745510601441296451495658283945878361720206516086771199590611777946503991100285}{483402232234085938389532340307798615731095021508812333301317862206582332679219632188332998101249052}\%1^2-$$

$$\frac{18551439752819447155580817439282493644479761993895686131890023230427662209706394813020560553874835535}{80567038705680989731588723384633102621849170251468722216886310367763722113203272031388833016874842}\%1^6$$

$$+\frac{893625422988249197578261957226517109328967447650231154892782977894339378843092025911344101892784484}{120850558058521484597383085076949653932773755377203083325329465551645583169804908047083249525312263}\%1^3+$$

$$\frac{193708319498731493699752304705704543508418408085778657723321334213274386775701014464481094994730}{120850558058521484597383085076949653932773755377203083325329465551645583169804908047083249525312263}\%1^{19}$$

$$+\frac{1015218932016225640584188228154294259559112356559259461445927386026112478036725036266150682073879}{483402232234085938389532340307798615731095021508812333301317862206582332679219632188332998101249052}+$$

$$\frac{45198704571767327788052408899616217704135687357133828403238716165902982279604246916216536113310615581}{241701116117042969194766170153899307865547510754406166650658931103291166339609816094166499050624526}\%1^7-$$

$$\frac{42524520911700248083179297060315463788783027535516497322439903868502683474731802906790552164870930703}{241701116117042969194766170153899307865547510754406166650658931103291166339609816094166499050624526}\%1^9-$$

$$\frac{4473865675824098179536616314686946157253491215658322423677314250149194642370336855256377458653477759}{161134077411361979463177446769266205243698340502937444433772620735527444226406544062777666033749684}\%1^8$$

$$+\frac{322437564223543182921356891684857807904205410733103254197208591605614332870095717600579631367919545}{120850558058521484597383085076949653932773755377203083325329465551645583169804908047083249525312263}\%1^{14}-$$

$$\frac{4166627910892933720155491722292102473257871813089828506672166641148356873484045432836446527714439541}{241701116117042969194766170153899307865547510754406166650658931103291166339609816094166499050624526}\%1^{12}+$$

$$\frac{19940370933626274413560498309767975811352589867780736680105715067599511878285191110954491829498256139}{241701116117042969194766170153899307865547510754406166650658931103291166339609816094166499050624526}\%1^{11}+$$

$$\frac{20153184668984336671119741674009644173726439177573019439972135337302183328956929619131419214990499367}{48340223222340859383895323403077986157310950215088123333013178622065823326792196321883329981012490 52}$$

$$\%1^{10} -$$

$$\frac{33874666588245387831489189870160725924410034890694975563695383957605578201280553735343752482807823 1}{17264365436931640656769012153849950561824822196743297617904209364520797595686415435297607075044609} \%1^{13}$$

$$-\frac{1288028014849486898036418934280394418072486780117136617884693085595712752119730323334239514502536 2}{12085055805852148459738308507694965393277375537720308332532946555164558316980490804708324952531226 3}$$

$$\%1^{17} +$$

$$\frac{33615034572440983735963408796471759220661060005919004950809176641167351762167943798529741064017 3}{16113407741136197946317744676926620524369834050293744443377262073552744422640654406277766603374968 4} \%1^{18}$$

$$-\frac{73542184434291576707402737270482395945485642284538658810290438193203430528995852901720911441830269}{48340223222340859383895323403077986157310950215088123333013178622065823326792196321883329981012490 52}$$

$$\%1^{16} +$$

$$\frac{26823539392022455637189727971979359972559991939014086791153701228099514773710896216018845972215830 9}{12085055805852148459738308507694965393277375537720308332532946555164558316980490804708324952531226 3} \%1^{15}$$

$$-\frac{33627365933280014450579049512662300273806922290216853006037610453061926185417329445486158161725208 02}{40283519352840494865794361692316551310924585125734361108443155183881861056601636015694416508437421}$$

$$\%1^{5} +$$

$$\frac{31538432554647277828972647114485168871438054469763254749489413461823191558831847188065789861282993 76}{12085055805852148459738308507694965393277375537720308332532946555164558316980490804708324952531226 3} \%1^{4}$$

$$\Big) \Big/ \%1^{(n+1)}$$

$$\%1 := \mathrm{RootOf}(\_Z^{20} + 2\,\_Z^{19} - 65\,\_Z^{18} - 140\,\_Z^{17} + 1281\,\_Z^{16} + 2538\,\_Z^{15} - 10366\,\_Z^{14} - 17604\,\_Z^{13} + 38553\,\_Z^{12}$$
$$+ 50158\,\_Z^{11} - 73623\,\_Z^{10} - 60482\,\_Z^{9} + 74665\,\_Z^{8} + 26564\,\_Z^{7} - 35106\,\_Z^{6} - 898\,\_Z^{5} + 4757\,\_Z^{4} + 16\,\_Z^{3} - 229\,\_Z^{2}$$
$$- 14\,\_Z + 1, .0427435026222)$$

As observed in [Fin97], if $g(n)$ denotes the number of tilings of a *nxn* chessboard, an interesting value for the physical applications is $\tau = \lim_{n \to \infty} g(n)^{\left(\frac{1}{n^2}\right)}$.
No exact expression for this limit is known, although the approximation 1.940215531 is generally agreed on. The first terms of the sequence can be computed from the previous approximations and are consistent with 1.94:

```
> nn:='nn':
> for i from 1 to 6 do
  assign(nn[i],coeff(series(gf[i],z=0,i+1),z,i)),evalf((nn[i
  ])^(1/(i*i))) od;
```

$$1.$$
$$1.626576562$$
$$1.718906945$$
$$1.778412706$$
$$1.811142170$$
$$1.833198802$$

But more interesting is the following observation. Suppose for example n is a multiple of 6. To tile a *nxn* chessboard we can put side by side $\frac{n}{6}$ slices of width 6. In this case

$\tau = \alpha^{\left(\frac{1}{6}\right)}$ with α the singularity of smallest modulus of the denominator of gf_6. If n is not a multiple of 6, it suffices to complete with at most 5 vertical stripes of width 1, but this does not change the limit. The interest in using as many slices of maximal width is to minimize the number of joints where the overlaps are not taken into account. The sequence $\{\alpha_i^{\left(\frac{1}{i}\right)}, i = 1 .. 6\}$ therefore provides lower bounds for the constant τ. An upper bound can be obtained in the same way by having slices of width 6 overlap on a position, and the corresponding sequence is $\{\alpha_i^{\left(\frac{1}{i-1}\right)}, i = 2 .. 6\}$.

```
> for i from 2 to 6 do
  i,(1/op(1,denom(evalf(asGf[i]))))^(1./i),(1/op(1,denom(eva
  lf(asGf[i]))))^(1./(i-1)) od;
```

2, 1.792852404, 3.214319743

3, 1.838281935, 2.492402505

4, 1.863497010, 2.293180643

5, 1.878563927, 2.199289866

6, 1.888704987, 2.144850135

At last a trick we can use to try to guess the value of τ is Romberg's convergence acceleration. Let u_n be a sequence known to converge to l. If the rate of convergence is of the form $u_n = l + \frac{a_1}{n} + O\left(\frac{1}{n^2}\right)$, then $2\,u_{2\,n} - u_n$ is $l + O\left(\frac{1}{n^2}\right)$. On our example, although the upper bound does not make sense due to too erroneous initial values, after a single step the lower bound gets close to the commonly accepted value:

```
> u[2]:=1.792852404:u[4]:=1.863497010:
  v[2]:=3.214319743:v[4]:=2.293180643:
  2*u[4]-u[2],2*v[4]-v[2];


  u[3]:=1.838281935:u[6]:=1.888704987:
  v[3]:=2.492402505:v[6]:=2.144850135:
  2*u[6]-u[3],2*v[6]-v[3];
```

1.934141616, 1.372041543

1.939128039, 1.797297765

Generating functions archive

Conclusion

We showed that various parameters related to dimer-monomer tilings such as the average number of pieces or the relative numbers of horizontal dimers and monomers in a random tiling of height n in a strip of width m can be computed very easily using Combstruct and ratasympt. More precisely Combstruct is used to define the grammars the tilings are derived from, and ratasympt is used to perform asymptotic expansions on rational fractions with rational coeficients.

About the number $g(n)$ of different tilings of a *nxn* chessboard, altough the method presented here is limited due to the exponential growth of the grammar describing these tilings, the very first terms computed provide *provably good upper and lower bounds* for the connectivity constant $g(n)^{\left(\frac{1}{n^2}\right)}$. More precisely:

Theorem. The connectvity constant for two dimensional monomer-dimer tilings satisfies

$1.888 \leq \tau$ and $\tau \leq 2.144$