# CONSTRAINED PERMUTATIONS AND THE PRINCIPLE OF INCLUSION-EXCLUSION

*Philippe Flajolet, January 10, 1998*

This worksheet explores variations of some classical problems in combinatorial analysis, like the *derangement* problem (also called *rencontre* problem), or the *menage* problem. The descriptions of these problems as borrowed from [Comtet, 1974] are:

-- Derangement/Rencontre: If guests to a party leave their hats on hooks in the cloakroom, and grab a hat at good luck when leaving, what is the probability that nobody gets back his own hat? The problem is equivalent to estimating the number of permutations without fixed point, that is to say, without singleton cycles.

-- Menage: What is the number of possible ways one can arrange $n$ married couples (=menages) around a table such that men and women alternate but no woman seats next to her husband?

These problems are in fact permutation enumeration problems from classical combinatorial analysis. The constraints considered concern, for a permutation $\sigma = \sigma_1, ..., \sigma_n$, its "***succession gaps***", that is, differences between consecutive elements, $\sigma_{i+1} - \sigma_i$, The derangement problem corresponds to permutations such that $\sigma_i - i \neq 0$, the menage problem to $\sigma_i - i \neq 0, 1$. In the second case, the constraints on indices and values may also be taken to be modulo $n$, that is taken "cyclically".

The symbolic method in enumerative combinatorics as well as the Combstruct package that implements it are based on the concept of decomposability. Combinatorial objects defined by "constraints" are thus not normally accessible to this framework. However, as shown in this worksheet, the enumeration of various types of constrained *permutations* can be treated by a combination of Combstruct , Gfun, and the Maple system. One may either impose that all such gaps be ***forced*** to belong a finite set $\Omega$, or dually impose that all gaps have values that ***exclude*** elements of $\Omega$. Forcing gaps to belong to a finite set $\Omega$ leads to finite-state models, while the exclusion of gaps from $\Omega$ builds on the finite-state models models via an inclusion-exclusion argument.

The general idea of counting by inclusion-exclusion is to enumerate by generating functions (GF's) objects with a number of ***distinguished exceptions*** to a set of constraints. The principle is as follows: If $\mathrm{F}(z, u)$ is the bivariate generating function of such objects, where $z$ records size and $u$ records the number of distinguished exceptions of some type $\Omega$, then inclusion-exclusion [e.g., Comtet, 1974] provides $\mathrm{F}(z, -1)$ as the univariate generating function of $\Omega$-free configurations.

Here, we show cases where so-called "***permutation templates***" can be described by finite-state mechanisms, so that a multivariate generating function of templates is directly accessible to combstruct. (A template describes a class of permutations by specifying what happens at some places while others are free, which is rendered by a "dont-care" symbol.) A specialization of the multivariate GF that involves a sign-change (for inclusion-exclusion) and an integral transformation effected on an auxiliary variable (for "filling" the free positions in templates and transforming them into permutations) yields a counting GF for the original problem. Generating functions, recurrences, and numerical values can be obtained automatically in this framework.

This demonstration worksheet was originally inspired by works of Kostas Hatzis (Patras) relative to edge-disjoint paths in random graphs and of Bruno Codenotti (Pisa) relative to computing permanents of circulants matrices.

**References**.
[Comtet, 1974] : L. Comtet, *Advanced Combinatorics*, Reidel, 1974.
[EIS]: N. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, 1995.

This Maple worksheet is based on the current versions of the Maple packages Combstruct and Gfun (for version V.4) that can be found under *http://www-rocq.inria.fr/algo/*

```
> restart;
> with(combstruct);
```

[*allstructs*, *count*, *draw*, *finished*, *gfeqns*, *gfseries*, *gfsolve*, *iterstructs*, *mark*, *moment*, *nextstruct*, *prog_gfeqns*, *prog_gfseries*, *prog_gfsolve*, *variance*]

```
> with(gfun);
```

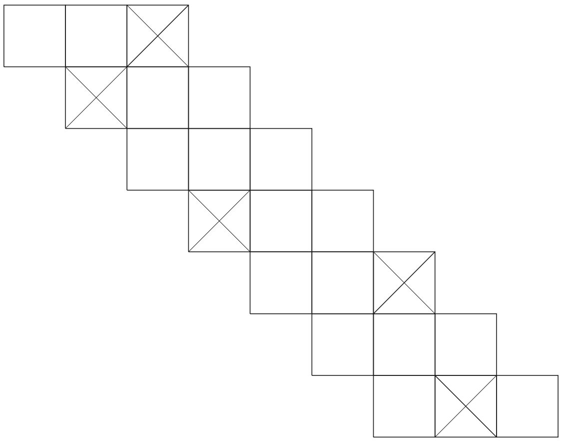
[*Laplace*, *algebraicsubs*, *algeqtodiffeq*, *algeqtoseries*, *algfuntoalgeq*, *borel*, *cauchyproduct*, *diffeq*diffeq*, *diffeq+diffeq*, *diffeqtohomdiffeq*, *diffeqtorec*, *guesseqn*, *guessgf*, *hadamardproduct*, *holexprtodiffeq*, *invborel*, *listtoalgeq*, *listtodiffeq*, *listtohypergeom*, *listtolist*, *listtoratpoly*, *listtorec*, *listtoseries*, *listtoseries/Laplace*, *listtoseries/egf*, *listtoseries/lgdegf*, *listtoseries/lgdogf*, *listtoseries/ogf*, *listtoseries/revegf*, *listtoseries/revogf*, *maxdegcoeff*, *maxdegeqn*, *maxordereqn*, *mindegcoeff*, *mindegeqn*, *minordereqn*, *optionsgf*, *poltodiffeq*, *poltorec*, *ratpolytocoeff*, *rec*rec*, *rec+rec*, *rectodiffeq*, *rectohomrec*, *rectoproc*, *seriestoalgeq*, *seriestodiffeq*, *seriestohypergeom*, *seriestolist*, *seriestoratpoly*, *seriestorec*, *seriestoseries*]

# 1. Permutations with forbidden "position gaps" and the principle of Inclusion-Exclusion

This section is devoted to the combinatorial basis on which this whole worksheet relies. We define the notion of **templates** that are simple combinatorial objects out of which constrained permutations can be built. An elementary integral transformation implements the inclusion-exclusion principle and leads to generating functions of constrained permutations.

The set $\Omega$ is a subset of $\{ 0 .. d \}$ with $d$ being its maximal element. We propose to count the number $P_n$ of permutations $\sigma = \sigma_1, \sigma_2 .. \sigma_n$ such that *none* of the position gaps $\sigma_k - k$ belongs to $\Omega$. By inclusion-exclusion, we first enumerate permutations with a certain number of distinguished exceptions.

For this purpose, one needs to determine
$F_{n,j}$ = the number of permutations $\sigma = \sigma_1, \sigma_2 .. \sigma_n$ such that $n-j$ of the position gaps $\sigma_k - k$ are distinguished and forced to belong to $\Omega$, while the remaining $j$ positions may be occupied by arbitrary elements, as long as the permutation property is satisfied.



**Templates**. The diagram above is the beginning of a permutation template (read from top to bottom) such that the sequence of position gaps is

$$2, 0, *, 0, 2, *, 1.$$

There, the don't care symbol "*" means that no constraint is imposed at the corresponding position. Thus, any permutation of 1,2,3,etc, that starts like

$$3, 2, *, 4, 7, *, 8, \text{etc}$$

will match the template. As they specify classes of permutations, the templates are such that there is at most one position occupied (represented by a cross) in each line and each column. Legal templates can thus be viewed as a language over a finite alphabet of dominos where each domino corresponds to an element of $\Omega$. Clearly, the constraints that each column is occupied by at most one cross can be tested with finite memory, i.e., they can be described by a finite state model.
The generating functions that are defined in this way, $\mathrm{G}(z, u)$, are such that

$G_{n,j}$ = the coefficient $[z^n u^j]$ in $G$ represents the number of templates (domino placements) of size $n$ that have $j$ don't-care symbols (hence $n-j$ distinguished occurrences of a position gap lying in $\Omega$).

Then, by filling the don't-care symbols in templates, one has

$$F_{n,j} = G_{n,j}\, j!.$$

Finally, by inclusion-exclusion, the number of permutations without any position gap in $\Omega$ is

$$P_n = \sum_{j=0}^{n} (-1)^{(n-j)}\, G_{n,j}\, j!.$$

This is also expressible as an integral transform

$$\sum_{n=0}^{\infty} p_n z^n = \int_0^{\infty} \mathrm{G}(-z, -t)\, \mathbf{e}^{(-t)}\, dt.$$

since $j! = \int_0^{\infty} \mathbf{e}^{(-t)}\, t^j\, dj.$

## Plotting routines archive

The diagram above was produced by the following simple routines:

```
> domino:=proc(x,y,l) local j;
```

```
seq([[x+j,y],[x+j+1,y],[x+j+1,y+1],[x+j,y+1],[x+j,y]],j=0..l-1)
; end;
```

*domino* := **proc**($x$, $y$, $l$)

**local** $j$;

$\mathrm{seq}([[x+j, y], [x+j+1, y], [x+j+1, y+1], [x+j, y+1], [x+j, y]], j = 0 .. l-1)$

**end**

```
> domino2:=proc(x,y,l,s)
domino(x,y,l),[[x+s,y],[x+s+1,y+1]],[[x+s+1,y],[x+s,y+1]]; end;
```

*domino2* := **proc**($x$, $y$, $l$, $s$)

$\mathrm{domino}(x, y, l), [[x+s, y], [x+s+1, y+1]], [[x+s+1, y], [x+s, y+1]]$

**end**

```
> plot([domino2(1,-1,3,2),domino2(2,-2,3,0),domino(3,-3,3),domino
2(4,-4,3,0),domino2(5,-5,3,2),domino(6,-6,3),domino2(7,-7,3,1)]
,scaling=constrained,axes=none,color=blue,labels=[`A
permutation template`,``],thickness=3):
```

# 2. Templates and permutations with Ω-exceptions (programme)

Templates are decomposable objects and thus they can be specified in the Combstruct language. As it turns out, they are described by a finite state model that translates all the allowed transitions between adjacent positions. This section provides the main routines that compute the specification of templates associated with a given set of position gaps.

We consider templates with some positions marked that are elements of $\Omega$ (exceptions) and "don't-care" positions.

Templates can be described by a finite state device. For instance, if $\Omega = \{ 0, 1 \}$, we have a linear version of the classical *menage problem*: If the choice $\omega = 1$ of $\omega$ in $\Omega$ has been made at stage $k-1$, that is $\sigma_{k-1} = k$ has been chosen, then the permutation property implies that one must have $\sigma_k \neq k$, that is to say, the choice $\omega = 0$ is forbidden (at stage $k$) in a template immediately after (at stage $k-1$) a choice $\omega = 1$. In other words, the language of templates is such that a domino of type "-*x*" cannot be followed by a domino of type "*x*-".

More generally, a *state* is any subset $s$ of $\{ 0 .. d-1 \}$ whose meaning is that the values in $s$ are unavailable as current position gaps, due to previous "commitments". Only don't cares or values of $\omega$ in $\Omega$ that are compatible with $s$ can be then be taken, and there is a transition to a new state that records the necessary information about occupied positions. The finite state system thus comprises $2^{(d-1)}$ states, and the cardinality of the alphabet is equal to the cardinality of $\Omega$ plus one (for the don't care symbol).

This can be expressed by combstruct specifications. Let $a_i$ be a letter that represents a position gap equal to $i$. An $\Omega$ template of length $n$ is thus described by a word of length $n$ over the alphabet $A$ formed with the $a_\omega$ for all $\omega$ in $\Omega$ and a don't care symbol.
First build the alphabet:

```
> build_alphabet:=proc(Omega) local omega;
seq(a[omega]=Atom,omega=Omega),dontcare=Prod(Atom,marked),
marked=Epsilon end;
```

*build_alphabet* :=

**proc**($\Omega$) **local** $\omega$; $\mathrm{seq}(a[\omega] = Atom, \omega = \Omega)$, $dontcare = \mathrm{Prod}(Atom, marked)$, $marked = \mathrm{E}$ **end**

```
> build_alphabet({0,1,3});
```

$$a_0 = Atom, a_1 = Atom, a_3 = Atom, dontcare = \mathrm{Prod}(Atom, marked), marked = \mathrm{E}$$

Next, build the transitions. The auxiliary procedure *MinusOne* decreases all the elements of a set by 1.

```
> MinusOne:=proc(S) map(proc(x) x-1 end,S) minus {-1} end;
```

*MinusOne* := **proc**(*S*) map( **proc**(*x*) *x* – 1 **end**, *S* ) minus { -1 } **end**

The transitions (in a raw form) are thus given by the compatibility relations between each state and the symbols.

```
> build_transition:=proc(state,Omega) local x,t; if state={} then
  t:=Epsilon else t:=NULL
  fi;s[state]=Union(t,Prod(dontcare,s[MinusOne(state)]),seq(Prod(a[x
  ],s[MinusOne(state union {x})]),x=Omega minus state)) end;
```

*build_transition* := **proc**(*state*, Ω)

**local** *x*, *t*;

**if** *state* = { } **then** *t* := E **else** *t* := *NULL* **fi**;

*s*[ *state* ] = Union( *t*, Prod( *dontcare*, *s*[ MinusOne( *state* ) ] ),

seq( Prod( *a*[ *x* ], *s*[ MinusOne( *state* union { *x* } ) ] ), *x* = Ω minus *state* ) )

**end**

```
> build_transition({},{0,1},1);build_transition({0},{0,1},1);
```

$$s_{\{\ \}} = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(dontcare, s_{\{\ \}}), \mathrm{Prod}(a_0, s_{\{\ \}}), \mathrm{Prod}(a_1, s_{\{0\}}))$$

$$s_{\{0\}} = \mathrm{Union}(\mathrm{Prod}(dontcare, s_{\{\ \}}), \mathrm{Prod}(a_1, s_{\{0\}}))$$

The grammar is then obtained by collecting transitions and taking into account the don't care symbol. The initial state is the empty set { } and it is also the final state (since no domino can "protrude").

```
> build_grammar:=proc(Omega) [s[{}],{build_alphabet(Omega)} union
  map(build_transition,combstruct[allstructs](Subset({$0..max(op(Ome
  ga))-1})),Omega),unlabelled] end;
```

*build_grammar* := **proc**(Ω)

[ *s*[ { } ], { build_alphabet( Ω ) } union

map( *build_transition*, *combstruct*[ *allstructs* ]( Subset( { '$'( 0 .. max( op( Ω ) ) – 1 ) } ) ), Ω ),

*unlabelled* ]

**end**

Here is an example:

```
> build_grammar({0,1});
```

$$[s_{\{\ \}}, \{ s_{\{\ \}} = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(dontcare, s_{\{\ \}}), \mathrm{Prod}(a_0, s_{\{\ \}}), \mathrm{Prod}(a_1, s_{\{0\}})),$$
$$s_{\{0\}} = \mathrm{Union}(\mathrm{Prod}(dontcare, s_{\{\ \}}), \mathrm{Prod}(a_1, s_{\{0\}})), marked = \mathrm{E}, a_0 = Atom, a_1 = Atom,$$
$$dontcare = \mathrm{Prod}(Atom, marked) \}, unlabelled]$$

In general, the grammar comprises $2^{(d-1)}$ nonterminal, which corresponds to a finite-stae device with as many states. The template GF's are thus rational functions of degree $2^{(d-1)}$, where $d$ is the maximal element of Ω.

# 3. Special cases of permutations with forbidden "position gaps"

In this section, we enumerate various classes of permutations associated with a given finite set Ω of forbidden position gaps. The process is in three steps: (i) Generate the grammar by means of the procedures of Section 2; (ii) compute automatically a bivariate GF of templates by means of Combstruct[gfsolve]; (iii) transform the corresponding GF by means of the integral transform of Section 1 that implements the inclusion-exclusion principle.All cases can be treated automatically, and we detail here the derangement and menage problems. Several generalizations of the menage problem are also tabulated. The ordinary GF's turn out to have hypergeometric forms and satisfy simple (but combinatorially nonobvious) linear recurrences of the holonomic type that can once more be derived automatically.

## Derangements (or Rencontres)

A *derangement* [Comtet, 1974] is a permutation without fixed point, that is to say, the set $\Omega = \{0\}$. of position gaps is forbidden. A derangement is thus a permutation without singleton cycles.

```
> G0:=build_grammar({0},0);
```

$$G0 := [s_{\{\ \}}, \{s_{\{\ \}} = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(dontcare, s_{\{\ \}}), \mathrm{Prod}(a_0, s_{\{\ \}})), marked = \mathrm{E}, a_0 = Atom, dontcare = \mathrm{Prod}(Atom, marked)\}, unlabelled]$$

The language of templates is thus just the set of all words over the alphabet $\{a_0, *\}$ where * is shorthand for a don't-care symbol.

```
> draw(G0,size=10);
```

$\mathrm{Prod}(a_0, \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{Prod}(a_0, \mathrm{Prod}($
$\mathrm{Prod}(Atom, marked),$
$\mathrm{Prod}(a_0, \mathrm{Prod}(a_0, \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{Prod}(a_0, \mathrm{E})))))))))$

```
> seq(count(G0,size=n),n=0..10);
```

$$1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024$$

The bivariate GF $\mathrm{G}(z, u)$ is obtained by combstruct[gfsolve]:

```
> gfsolve(op(2,G0),op(3,G0),z,[[u,marked]]);
```

$$\left\{\mathrm{dontcare}(z, u) = z\,u, s_{\{\ \}}(z, u) = -\frac{1}{-1 + z\,u + z}, a_0(z, u) = z, \mathrm{marked}(z, u) = u\right\}$$

In particular,

```
> g0:=subs(",s[{}](z,u));
```

$$g0 := -\frac{1}{-1 + z\,u + z}$$

The counting sequence is obtained by the transformation corresponding to the inclusion-exclusion principle.

```
> series(subs([z=-z,u=-t],g0),z=0,11):map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$1 + z^2 + 2\,z^3 + 9\,z^4 + 44\,z^5 + 265\,z^6 + 1854\,z^7 + 14833\,z^8 + 133496\,z^9 + 1334961\,z^{10} + \mathrm{O}(z^{11})$$

The sequence 1, 0, 1, 2, 9, 44, 265, ..., is of course classical. It is **M1937** of [EIS] where it is described as "subfactorial or rencontre numbers".

The ordinary generating function (OGF) is also accessible in closed-form as

```
> P0:=int(exp(-t)*subs([z=-z,u=-t],g0),t=0..infinity);
```

$$P0 := -\frac{\mathrm{Ei}\left(1, -\frac{z+1}{z}\right)\mathbf{e}^{\left(-\frac{z+1}{z}\right)}}{z}$$

This involves the exponential integral Ei, where the exponential integral symbol means

$$\mathrm{Ei}(1, x) = \int_1^{\infty} \frac{\mathbf{e}^{(-x t)}}{t}\, dt$$

The OGF is purely divergent and P0 is to be interpreted as asymptotic to the right-hand side as $z$ tends to $0^-$.

```
> map(simplify,asympt(subs(z=-1/y,P0),y,10));
```

$$1 + \frac{1}{y^2} - \frac{2}{y^3} + \frac{9}{y^4} - \frac{44}{y^5} + \frac{265}{y^6} - \frac{1854}{y^7} + \frac{14833}{y^8} + \mathrm{O}\left(\frac{1}{y^9}\right)$$

**Verification with Combstruct**. Derangements are characterized by their cycle decomposition as labelled sets of cycles with length at least 1. They are thus specifiable in combstruct.

```
> der:=[S,{S=Set(Cycle(Z,card>1))},labelled];
```

$$der := [S, \{ S = \mathrm{Set}(\mathrm{Cycle}(Z, 1 < card)) \}, labelled]$$

```
> seq(count(der,size=n),n=0..10);
```

1, 0, 1, 2, 9, 44, 265, 1854, 14833, 133496, 1334961

```
> gfsolve(op(2,der),op(3,der),z);
```

$$\{ \mathrm{S}(z) = -\frac{1}{\mathbf{e}^z\,(-1+z)}, \mathrm{Z}(z) = z \}$$

The exponential generating function (EGF) is well-known

```
> der_egf:=simplify(subs(",S(z)));
```

$$der\_egf := -\frac{\mathbf{e}^{(-z)}}{-1+z}$$

```
> series(der_egf,z=0,13);
```

$$1 + \frac{1}{2}z^2 + \frac{1}{3}z^3 + \frac{3}{8}z^4 + \frac{11}{30}z^5 + \frac{53}{144}z^6 + \frac{103}{280}z^7 + \frac{2119}{5760}z^8 + \frac{16687}{45360}z^9 + \frac{16481}{44800}z^{10} + \frac{1468457}{3991680}z^{11} + \frac{16019531}{43545600}z^{12} + \mathrm{O}(z^{13})$$

We check that the coefficients in the EGF *der_egf* and in the OGF *P0* are the same. (The two GF's are related by the Laplace transform.)

**Asymptotics**. The number of derangements of size $n$ satisfies the well-known asymptotic formula:

$$\frac{p_n}{n!} = \mathbf{e}^{(-1)}\,(1 + \mathrm{o}(1)),$$

a fact which is obvious given the singularity of the EGF at 1:

```
> series(der_egf,z=1,3);
```

$$-\mathbf{e}^{(-1)}\,(-1+z)^{-1} + \mathbf{e}^{(-1)} - \frac{1}{2}\mathbf{e}^{(-1)}\,(-1+z) + \mathrm{O}((-1+z)^2)$$

## Menage numbers

The *menage numbers* are defined by the excluded set $\Omega = \{ 0, 1 \}$. In nineteenth century terminology, this is phrased as follows: In how many ways can one place couples with men and women alternating in such a way that no husband can seat next to his wife? The problem is usually posed in terms of a round table (=cyclic permutations); see [Comtet, 1974]. In this section, the linear version of the problem is treated.

```
> G01:=build_grammar({0,1},1);
```

$G01 := [s_{\{\ \}}, \{marked = \mathrm{E}, a_0 = Atom, dontcare = \mathrm{Prod}(Atom, marked), a_1 = Atom,$
$s_{\{0\}} = \mathrm{Union}(\mathrm{Prod}(dontcare, s_{\{\ \}}), \mathrm{Prod}(a_1, s_{\{0\}})),$
$s_{\{\ \}} = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(dontcare, s_{\{\ \}}), \mathrm{Prod}(a_0, s_{\{\ \}}), \mathrm{Prod}(a_1, s_{\{0\}}))\}, unlabelled]$

```
> g01:=subs(gfsolve(op(2,G01),op(3,G01),z,[[u,marked]]),s[{}](z,u
  ));
```

$$g01 := -\frac{-1+z}{1-z\,u-2\,z+z^2}$$

The linear "menage numbers" immediately result by the general transformation.

```
> series(subs([z=-z,u=-t],g01),z=0,17):ser01:=map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$ser01 := 1+z^3+3\,z^4+16\,z^5+96\,z^6+675\,z^7+5413\,z^8+48800\,z^9+488592\,z^{10}+5379333\,z^{11} +64595975\,z^{12}+840192288\,z^{13}+11767626752\,z^{14}+176574062535\,z^{15}+2825965531593\,z^{16}+\mathrm{O}(z^{17})$$

This is sequence **M3020** of [*EIS*], defined there as "sums of menage numbers".

**Checking**. Combstruct is also useful for checking and debugging purposes. Consistency of templates can be checked for small values of $n$ by expansions and by *combstruct[allstructs]*.

```
> series(g01,z=0,10):map(expand,");
```

$$1+(u+1)\,z+(u^2+3\,u+1)\,z^2+(u^3+5\,u^2+6\,u+1)\,z^3+(7\,u^3+15\,u^2+10\,u+1+u^4)\,z^4 +(9\,u^4+28\,u^3+35\,u^2+15\,u+u^5+1)\,z^5+(84\,u^3+70\,u^2+21\,u+1+45\,u^4+11\,u^5+u^6)\,z^6 +(165\,u^4+210\,u^3+126\,u^2+28\,u+66\,u^5+1+13\,u^6+u^7)\,z^7+ (462\,u^3+210\,u^2+36\,u+1+495\,u^4+286\,u^5+91\,u^6+15\,u^7+u^8)\,z^8+ (1287\,u^4+924\,u^3+330\,u^2+45\,u+1001\,u^5+1+455\,u^6+120\,u^7+17\,u^8+u^9)\,z^9+\mathrm{O}(z^{10})$$

```
> allstructs(G01,size=2);allstructs(G01,size=3);
```

$[\mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{E})),$
$\mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{Prod}(a_0, \mathrm{E})), \mathrm{Prod}(a_1, \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{E})),$
$\mathrm{Prod}(a_0, \mathrm{Prod}(a_0, \mathrm{E})), \mathrm{Prod}(a_0, \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{E}))]$

$[\mathrm{Prod}(a_1, \mathrm{Prod}(a_1, \%1)), \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{Prod}(a_0, \%1)),$
$\mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{Prod}(a_0, \mathrm{Prod}(a_0, \mathrm{E}))), \mathrm{Prod}(a_1, \%2),$
$\mathrm{Prod}(a_0, \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \%1)), \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{Prod}(a_1, \%1)),$
$\mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \%1)), \mathrm{Prod}(a_0, \mathrm{Prod}(a_1, \%1)),$
$\mathrm{Prod}(a_0, \mathrm{Prod}(a_0, \%1)), \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \%2), \mathrm{Prod}(a_0, \%2),$
$\mathrm{Prod}(a_1, \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \%1)), \mathrm{Prod}(a_0, \mathrm{Prod}(a_0, \mathrm{Prod}(a_0, \mathrm{E})))]$
$\%1 := \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{E})$
$\%2 := \mathrm{Prod}(\mathrm{Prod}(Atom, marked), \mathrm{Prod}(a_0, \mathrm{E}))$

**Holonomic forms**. A recurrence can be heuristically guessed by *gfun*, proved by the hypergeometric method detailed here (this is however specific to the problem at hand), and also obtained automatically by *Mgfun*. Problems of this type a priori belong to the so-called *holonomic class*, introduced by Zeilberger, that is to say, sequences that satisfy linear recurrences with polynomial coefficients. Alternatively, the GF’s satisfy linear differential equations with polynomial coefficients.

As a first approach, the procedure gfun[listtodiffeq] provides a plausible differential equation that accounts for a given number sequence.

```
> l01:=[seq(coeff(ser01,z,i),i=0..16)];
```

$l01 := [1, 0, 0, 1, 3, 16, 96, 675, 5413, 48800, 488592, 5379333, 64595975, 840192288, 11767626752, 176574062535, 2825965531593]$

```
> ode01:=listtodiffeq(l01,Y(z));
```

$$ode01 := \left[ \{ 1 - z^2 + (-1 + z^2 + z^3)\, \mathrm{Y}(z) + (z^3 + z^2) \left( \frac{\partial}{\partial z} \mathrm{Y}(z) \right), \mathrm{Y}(0) = 1 \}, ogf \right]$$

This suggests a formal representation of the OGF of the basic menage problem.

```
> dsolve(op(1,op(1,ode01)),Y(z));
```

$$\mathrm{Y}(z) = \frac{\left( \left( \int \frac{(-1+z)\, \mathbf{e}^{\left(\frac{1+z^2}{z}\right)}}{(z+1)\, z} dz\, z + z\_C1 + \int \frac{(-1+z)\, \mathbf{e}^{\left(\frac{1+z^2}{z}\right)}}{(z+1)\, z} dz + \_C1 \right) \mathbf{e}^{\left(-\frac{1+z^2}{z}\right)} \right)}{z}$$

This guess can be validated by direct integration:

```
> P01:=int(subs([z=-z,u=-t],g01)*exp(-t),t=0..infinity);
```

$$P01 := -\frac{\mathrm{Ei}\left(1, -\frac{1+2\,z+z^2}{z}\right)(z+1)\, \mathbf{e}^{\left(-\frac{(z+1)^2}{z}\right)}}{z}$$

**Hypergeometrics**. Alternatively any expression with the exponential integral Ei can be expressed in terms of a divergent hypergeometrics (that is also the OGF of permutations). One has

```
> y*exp(y)*Ei(1,y)=asympt(y*exp(y)*Ei(1,y),y,10);
```

$$y\, \mathbf{e}^y\, \mathrm{Ei}(1, y) = 1 - \frac{1}{y} + \frac{2}{y^2} - \frac{6}{y^3} + \frac{24}{y^4} - \frac{120}{y^5} + \frac{720}{y^6} - \frac{5040}{y^7} + \frac{40320}{y^8} + \mathrm{O}\left(\frac{1}{y^9}\right)$$

that is also expressible as a ${}_2F_0$-hypergeometric:

```
> hypergeom([1,1],[],z)=series(hypergeom([1,1],[],z),z=0,12);
```

$$\mathrm{hypergeom}([1, 1], [\ ], z) = 1 + z + 2\,z^2 + 6\,z^3 + 24\,z^4 + 120\,z^5 + 720\,z^6 + 5040\,z^7 + 40320\,z^8 + 362880\,z^9 + 3628800\,z^{10} + 39916800\,z^{11} + \mathrm{O}(z^{12})$$

Thus the menage OGF has an hypergeometric form:

```
> hg_menage:=1/(1+z)*hypergeom([1,1],[],z/(1+z)^2);
```

$$hg\_menage := \frac{\mathrm{hypergeom}\left([1, 1], [\ ], \frac{z}{(z+1)^2}\right)}{z+1}$$

```
> series(hg_menage,z=0,10);
```

$$1 + z^3 + 3\,z^4 + 16\,z^5 + 96\,z^6 + 675\,z^7 + 5413\,z^8 + 48800\,z^9 + \mathrm{O}(z^{10})$$

This gives a single alternating sum for the coefficients that is the counterpart of the one of [Comtet, 1974] for the circular menage problem.

This process gives rise to a general procedure for conversion from exponential integral to hypergeometric form:

```
> convert_hypergeom:=proc(e) subs(Ei=proc(a,b) if a<>1 then
  ERROR(`unable to convert`) else
```

```
exp(-b)/b*hypergeom([1,1],[],-1/b) fi end,e); simplify("); end;
```

*convert_hypergeom* := **proc**($e$)
subs( Ei = ( **proc**($a$, $b$)
**if** $a \neq 1$ **then** ERROR( *'unable to convert'* )
**else** exp($-b$)*hypergeom( [ 1, 1 ], [ ], $-1/b$ ) / $b$
**fi**
**end** ), $e$ );
simplify( ″)
**end**

```
> convert_hypergeom(P01);
```

$$\frac{\mathrm{hypergeom}\left([1,1],[\ ],\dfrac{z}{1+2z+z^2}\right)}{z+1}$$

**Asymptotics**. Experimentally, we have

```
> seq(op(i,l01)/(i-1)!*1.0,i=1..16);
```

1.0, 0, 0, .1666666667, .1250000000, .1333333333, .1333333333, .1339285714, .1342509921, .1344797178, .1346428571, .1347636334, .1348554472, .1349268478, .1349834528, .1350290767

```
> exp(-2)=exp(-2.);
```

$$\mathbf{e}^{(-2)} = .1353352832$$

Now, two values are forbidden at each position. Empirically, the number of menage numbers is seen to satisfy

$$P_n = n!\,\mathbf{e}^{(-2)}\,(1+\mathrm{o}(1)).$$

This is to be compared to the asymptotics of derangements. The asymptotic estimate can be established directly from the alternating sum expression of coefficients.

### *A simplified menage problem*

The case $\Omega = \{ 1 \}$ is in fact a variant of the derangement problem.

```
> G1:=build_grammar({1},1):
> g1:=subs(gfsolve(op(2,G1),op(3,G1),z,[[u,marked]]),s[{}](z,u
  ));
```

$$g1 := \frac{-1+z}{-1+z\,u+z}$$

```
> series(subs([z=-z,u=-t],g1),z=0,11):map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$1+z+z^2+3\,z^3+11\,z^4+53\,z^5+309\,z^6+2119\,z^7+16687\,z^8+148329\,z^9+1468457\,z^{10} + \mathrm{O}(z^{11})$$

The sequence appears as **M2905** in [EIS] where the EGF is given:

```
> exp(-z)/(1-z)^2=series(exp(-z)/(1-z)^2,z=0,8);
```

$$\frac{\mathbf{e}^{(-z)}}{(1-z)^2} = 1+z+\frac{3}{2}z^2+\frac{11}{6}z^3+\frac{53}{24}z^4+\frac{103}{40}z^5+\frac{2119}{720}z^6+\frac{16687}{5040}z^7+\mathrm{O}(z^8)$$

## 2-Menage numbers

This is a generalization of the menage problem where $\Omega = \{ 0, 1, 2 \}$: In order to avoid family fights, couples are now at distance at least 3 (!).

```
> G012:=build_grammar({0,1,2},2);
```

$$G012 := [s_{\{\ \}}, \{marked = \mathrm{E}, a_0 = Atom, dontcare = \mathrm{Prod}(Atom, marked), a_2 = Atom,$$
$$s_{\{0\}} = \mathrm{Union}(\mathrm{Prod}(dontcare, s_{\{\ \}}), \mathrm{Prod}(a_1, s_{\{0\}}), \mathrm{Prod}(a_2, s_{\{1\}})),$$
$$s_{\{\ \}} = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(dontcare, s_{\{\ \}}), \mathrm{Prod}(a_0, s_{\{\ \}}), \mathrm{Prod}(a_1, s_{\{0\}}), \mathrm{Prod}(a_2, s_{\{1\}})),$$
$$s_{\{0,1\}} = \mathrm{Union}(\mathrm{Prod}(dontcare, s_{\{0\}}), \mathrm{Prod}(a_2, s_{\{0,1\}})),$$
$$s_{\{1\}} = \mathrm{Union}(\mathrm{Prod}(dontcare, s_{\{0\}}), \mathrm{Prod}(a_0, s_{\{0\}}), \mathrm{Prod}(a_2, s_{\{0,1\}})), a_1 = Atom\}, unlabelled]$$

```
> gfsolve(op(2,G012),op(3,G012),z,[[u,marked]]);
  g012:=subs(",s[{}](z,u));
```

$$\{a_0(z,u) = z, \mathrm{marked}(z,u) = u, a_2(z,u) = z, s_{\{0\}}(z,u) = \frac{z\,u}{\%1}, a_1(z,u) = z,$$
$$s_{\{0,1\}}(z,u) = -\frac{z^2 u^2}{\%1\,(-1+z)}, s_{\{\ \}}(z,u) = -\frac{1-2z-z^2u+z^3}{\%1\,(-1+z)}, s_{\{1\}}(z,u) = \frac{z^2 u\,(-u-1+z)}{\%1\,(-1+z)},$$
$$\mathrm{dontcare}(z,u) = z\,u\}$$
$$\%1 := z^3 - z^2 u - 2z - z u + 1$$
$$g012 := -\frac{1-2z-z^2u+z^3}{(z^3-z^2u-2z-zu+1)(-1+z)}$$

```
> series(subs([z=-z,u=-t],g012),z=0,13):P012_ser:=map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$P012\_ser := 1 + z^4 + 5z^5 + 33z^6 + 236z^7 + 1918z^8 + 17440z^9 + 175649z^{10} + 1942171z^{11} + 23396353z^{12} + \mathrm{O}(z^{13})$$

Only the beginning of this sequence (till 17440 included) appears in [EIS] as sequence **M3970.** The computation above makes it possible to extend the values of [EIS] to an arbitrary order.

**Holonomic forms**. Since the denominator of *g012* has degree 1 in *u*, the Laplace-like integral applied to it must be expressible in terms of the exponential integral $\mathrm{Ei}(1,z)$. This, in turn is equivalent to a divergent hypergeometric form.

```
> convert(subs([z=-z,u=-t],g012),parfrac,u);h012:=int("*exp(-t),t
  =0..infinity);
```

$$-\frac{1+2z+z^2t-z^3}{(-z^3+z^2t+2z-zt+1)(-1-z)}$$

$$h012 := \left(-\mathrm{Ei}\left(1, -\frac{z^3-2z-1}{z(-1+z)}\right)\mathbf{e}^{\left(\frac{1+2z}{z(-1+z)}\right)} - 2\,\mathrm{Ei}\left(1, -\frac{z^3-2z-1}{z(-1+z)}\right)z\,\mathbf{e}^{\left(\frac{1+2z}{z(-1+z)}\right)} + z^3\,\mathbf{e}^{\left(\frac{z^2}{-1+z}\right)} - z^2\,\mathbf{e}^{\left(\frac{z^2}{-1+z}\right)} + \mathrm{Ei}\left(1, -\frac{z^3-2z-1}{z(-1+z)}\right)z^3\,\mathbf{e}^{\left(\frac{1+2z}{z(-1+z)}\right)}\right)\mathbf{e}^{\left(-\frac{z^2}{-1+z}\right)} \Big/ \left(z(-1+z)^2(z+1)\right)$$

```
> hh012:=convert_hypergeom(h012);
```

$$hh012 := \frac{z - \mathrm{hypergeom}\left([1,1],[\ ],\frac{z(-1+z)}{z^3-2z-1}\right)}{(z+1)(-1+z)}$$

```
> series(",z=0,12);
```

$$1 + z^4 + 5z^5 + 33z^6 + 236z^7 + 1918z^8 + 17440z^9 + 175649z^{10} + 1942171z^{11} + \mathrm{O}(z^{12})$$

One can then get the holonomic forms by *gfun*.

```
> holexprtodiffeq(hh012,y(z)): subs(_C[0]=1,");
```

$$(-2 z^7 - z^8 - 4 z^4 + z^9 + 2 z^6 + 4 z^2 - 1 + z)\, y(z) + (-2 z^7 - 2 z^8 + z^9 + 5 z^6 - 4 z^4 + z^5 + z^2)\left(\frac{\partial}{\partial z} y(z)\right) - z^8 + 3 z^7 - 3 z^6 + 5 z^4 - 4 z^2 - z + 1$$

```
> diffeqtorec(",y(z),u(n));
```

$$\{ u(0) = 1, u(1) = 0, u(2) = 0, u(3) = 0, u(n) + (n+1)\, u(n+1) + (-4-n)\, u(n+2) + (-3n-9)\, u(n+3) + (2n+8)\, u(n+4) + (11+3n)\, u(n+5) + (-10-n)\, u(n+6) + (-n-7)\, u(n+7) + u(n+8), u(7) = 236, u(4) = 1, u(5) = 5, u(6) = 33 \}$$

This gives a fast procedure for computing the numbers.

```
> s012:=rectoproc(",u(n),remember);
```

*s012* := **proc**(*n*)

**option** *remember*;

−procname(*n* − 8) + 7*procname(*n* − 7) − 4*procname(*n* − 6) − 15*procname(*n* − 5) + 8*procname(*n* − 4) + 13*procname(*n* − 3) + 2*procname(*n* − 2) − procname(*n* − 1) − (procname(*n* − 7) − procname(*n* − 6) − 3*procname(*n* − 5) + 2*procname(*n* − 4) + 3*procname(*n* − 3) − procname(*n* − 2) − procname(*n* − 1))**n*

**end**

```
> seq(s012(j),j=0..30);
```

1, 0, 0, 0, 1, 5, 33, 236, 1918, 17440, 175649, 1942171, 23396353, 305055960, 4280721564, 64330087888, 1030831875953, 17545848553729, 316150872317105, 6012076099604308, 120330082937778554, 2528525819886170112, 55657855167451780993, 1280736404605380413303, 30750394025631567131329, 769039694784998460886896, 20001468420258808491667512, 540194118569060356429415712, 15129297070110498161942887009, 438850019810675038935803418941, 13168211505240199345047950457921

**Asymptotics**. This sequence obeys a general asymptotic pattern that was alluded to before. Verification to high orders is easy thanks to the holonomic recurrences.

```
> seq(1.0*s012(m)/m!,m=0..50);
```

1.0, 0, 0, 0, .04166666667, .04166666667, .04583333333, .04682539683, .04756944444, .04805996473, .04840415564, .04865547840, .04884399760, .04898907034, .04910306804, .04919427150, .04926837585, .04932940320, .04938025908, .04942308469, .04945948605, .04949068685, .04951763268, .04954106357, .04956156538, .04957960679, .04959556635, .04960975251, .04962241862, .04963377437, .04964399447, .04965322547, .04966159102, .04966919602, .04967612996, .04968246954, .04968828079, .04969362084, .04969853929, .04970307938, .04970727896, .04971117123, .04971478548, .04971814756, .04972128040, .04972420438, .04972693766, .04972949647, .04973189535, .04973414735, .04973626423

Such a numerical computation supports the assumption that asymptotically, the number of such permutations should grows like

$$p_n = n!\, \mathbf{e}^{(-3)}\, (1 + o(1)).$$

In effect, one has:

```
> exp(-3)=exp(-3.0);
```

$$\mathbf{e}^{(-3)} = .04978706837$$

(See the conclusion section for a proof of this observation.)

### *Variants of 2-menage numbers*

The case $\Omega = \{ 1, 2 \}$ is close to the classical menage problem.

```
> G12:=build_grammar({1,2},2):gfsolve(op(2,G12),op(3,G12),z,[[
  u,marked]]): g12:=subs(",s[{}](z,u));
```

$$g12 := -\frac{-1+2z-z^2+z^2u}{1-zu-2z+z^2}$$

```
> series(subs([z=-z,u=-t],g12),z=0,11):map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$1+z+z^2+z^3+5z^4+23z^5+131z^6+883z^7+6859z^8+60301z^9+591605z^{10}+O(z^{11})$$

## 3-Menage numbers

The basic version is defined by $\Omega = \{ 0, 1, 2, 3 \}$.

```
> G0123:=build_grammar({0,1,2,3},3):
> gfsolve(op(2,G0123),op(3,G0123),z,[[u,marked]]):
  g0123:=subs(",s[{}](z,u));
```

$$g0123 := -\frac{z^7-z^6u+z^6-z^5u+z^5-3z^4-3z^4u-z^3+z^3u+z^3u^2+2z^2u-z^2+3z-1}{-zu+4z^4u+4z^2+2z^4+1-4z+z^3u-4z^5-3z^5u+z^4u^2-z^7u+z^8}$$

```
> series(subs([z=-z,u=-t],g0123),z=0,11):map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$1+z^5+9z^6+71z^7+631z^8+5908z^9+60742z^{10}+O(z^{11})$$

This sequence is not to be found in [EIS].

### Variants of the 3-Menage problem

We just offer here a few random samples of sequences.

First, the case $\Omega = \{ 0, 3 \}$.

```
> G03:=build_grammar({0,3},3):
> gfsolve(op(2,G03),op(3,G03),z,[[u,marked]]):
  g03:=subs(",s[{}](z,u));
```

$$g03 := -\frac{z^7-z^6u-z^6-z^5u-2z^4-z^4u+2z^3+z^3u^2+2z^3u+z^2u+z-1}{1-zu-z^3u-2z^3+z^2+z^4u^2+4z^4u+4z^4-2z-2z^5-z^5u+z^6-z^7u-2z^7+z^8}$$

```
> series(subs([z=-z,u=-t],g03),z=0,11):map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$1+z^2+2z^3+6z^4+26z^5+142z^6+933z^7+7136z^8+62141z^9+605735z^{10}+O(z^{11})$$

Next, the case $\Omega = \{ 0, 1, 3 \}$.

```
> G013:=build_grammar({0,1,3},3):gfsolve(op(2,"),op(3,"),z,[[u
  ,marked]]):
  g013:=subs(",s[{}](z,u));series(subs([z=-z,u=-t],g013),z=0,1
  1):map(proc(x) int(x*exp(-t),t=0..infinity) end,");
```

$$g013 := -\frac{-1+2z+2z^3u+z^3u^2-2z^4+z^2u-2z^4u+z^7-z^6u-z^5u}{2z^2-zu+1+z^4u^2+4z^4u-3z-2z^5u-z^7u-z^7+z^8-2z^5+2z^4}$$

$$1+z^3+2z^4+8z^5+42z^6+288z^7+2253z^8+20011z^9+198134z^{10}+O(z^{11})$$

And the case $\Omega = \{ 0, 2, 3 \}$.

```
> G023:=build_grammar({0,2,3},3):gfsolve(op(2,"),op(3,"),z,[[u
  ,marked]]):
  g023:=subs(",s[{}](z,u));series(subs([z=-z,u=-t],g023),z=0,1
  1):map(proc(x) int(x*exp(-t),t=0..infinity) end,");
```

$$g023 := -\frac{z^7 - z^6 u - z^5 u - 2 z^4 u - 2 z^4 + z^3 u^2 + z^3 u + 2 z^2 u + 2 z - 1}{2 z^2 - z u + 1 + z^4 u^2 + 4 z^4 u - 3 z - 2 z^5 u - z^7 u - z^7 + z^8 - 2 z^5 + 2 z^4}$$

$$1 + z^2 + z^3 + 2 z^4 + 10 z^5 + 50 z^6 + 328 z^7 + 2531 z^8 + 22216 z^9 + 217822 z^{10} + O(z^{11})$$

Finally, the case $\Omega = \{1, 3\}$.

```
> G13:=build_grammar({1,3},3):gfsolve(op(2,"),op(3,"),z,[[u,ma
  rked]]):
  g13:=subs(",s[{}](z,u));series(subs([z=-z,u=-t],g13),z=0,11)
  :map(proc(x) int(x*exp(-t),t=0..infinity) end,");
```

$$g13 := -\frac{-z^3 u^2 + z^4 u + 1 - z^2 u - 2 z + 2 z^3 - z^4}{-1 + 2 z + z u - z^3 u - 2 z^3 + z^4}$$

$$1 + z + z^2 + 3 z^3 + 7 z^4 + 31 z^5 + 161 z^6 + 1045 z^7 + 7903 z^8 + 68205 z^9 + 659809 z^{10} + O(z^{11})$$

## 4-Menage numbers

This and the next example serve to test the system on models of a fairly large size. The rational GF for the 4-menage proble is of degree 16, and the computations only take a few seconds.

```
> G01234:=build_grammar({0,1,2,3,4},4):gfsolve(op(2,G01234),op(3,
  G01234),z,[[u,marked]]):
> g01234:=subs(",s[{}](z,u));
```

$$g01234 := -(-1 + 4 z - 8 z^5 u^2 - 2 z^2 + z^3 u + 2 z^3 u^2 - 2 z^4 + 3 z^2 u - 5 z^4 u - 11 z^{10} + 5 z^9 u^2 + 2 z^{10} u + 3 z^4 u^2 + z^4 u^3 - 9 z^6 u^2 + z^{15} - z^{13} u - 3 z^{11} u^2 - 2 z^{11} u + z^{12} u - z^{14} u - 8 z^{11} - 2 z^{12} + 2 z^{14} + z^{10} u^3 - z^8 u^3 + z^9 u^3 - 4 z^3 - 5 z^5 + 12 z^6 + 12 z^7 - z^6 u^3 - 2 z^7 u^2 + 14 z^9 u - z^7 u^3 + 2 z^6 u - 22 z^5 u + 8 z^8 u + 12 z^7 u + 4 z^8) \Big/ ((z^{10} - z^9 u + 2 z^8 u - 2 z^8 - z^7 u^2 - 5 z^6 u - 4 z^6 + 2 z^5 u^2 + 6 z^5 u + 2 z^5 + 5 z^4 u + 4 z^4 - z^3 u^2 - 4 z^3 u - 2 z^3 + 4 z^2 - 4 z - z u + 1) (z^6 + z^5 - z^4 u - z^2 u - 2 z^2 - z + 1))$$

```
> series(subs([z=-z,u=-t],g01234),z=0,11):map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$1 + z^6 + 17 z^7 + 159 z^8 + 1753 z^9 + 19193 z^{10} + O(z^{11})$$

```
> convert(g01234,parfrac,u);
```

$$-\frac{z^4 + z^3 - 1}{z (z^4 - 1)} - \frac{1}{2} (z^{10} - z^9 u + 3 z^8 u - 2 z^8 + z^7 u - 4 z^6 - 7 z^6 u + 2 z^5 + 4 z^5 u + 3 z^4 u + 4 z^4 - 2 z^3 - 5 z^3 u + 4 z^2 + z^2 u - 4 z - z u + 1) \Big/ (z (z + 1) (-1 + z) (z^{10} - z^9 u + 2 z^8 u - 2 z^8 - z^7 u^2 - 5 z^6 u - 4 z^6 + 2 z^5 u^2 + 6 z^5 u + 2 z^5 + 5 z^4 u + 4 z^4 - z^3 u^2 - 4 z^3 u - 2 z^3 + 4 z^2 - 4 z - z u + 1)) - \frac{1}{2} \frac{z^6 + 3 z^5 + 4 z^4 + 4 z^3 + 2 z^2 - z - 1}{z (z^6 + z^5 - z^4 u - z^2 u - 2 z^2 - z + 1) (1 + z^2)}$$

## 5-Menage numbers

The computation time is now of the order of a minute (on a slow machine with about $10^8$ cycles per second). The denominator is now of degree 32 in $z$ and of degree 6 in $u$.

```
> G012345:=build_grammar({0,1,2,3,4,5},5):
  gfsolve(op(2,G012345),op(3,G012345),z,[[u,marked]]):
> g012345:=subs(",s[{}](z,u));
```

$$g012345 := -(24 z^{11} u^4 + 12 z^{12} u^4 + 19 z^8 u^2 + 384 z^{17} + 204 z^{16} + 31 z^{14} u^4 - 42 z^{18} u^3 + 95 z^{19} u^2 + 262 z^{18} - 218 z^{16} u^3 - 68 z^{15} u^3 - 318 z^{16} u + 44 z^{19} u + 11 z^{15} u^4 - 262 z^{13} - 1 + 17 z^{14} u^2 - 77 z^{13} u^2 + 5 z^{13} u^3 - 17 z^{10} u^4 + 173 z^{11} u^3 + 75 z^{12} u^3 + 18 z^{12} u^2 + 39 z^{14} u^3 + 5 z - 233 z^{16} u^2 - 152 z^{15} u^2 + 146 z^{19} + 105 z^{17} u^2 - 5 z^5 u^2 + 8 z^{13} u^4 + 124 z^{18} u^2 + 66 z^{18} u - 394 z^{15} u - 64 z^{17} u^3 + 48 z^{17} u - 55 z^{16} u^4 + 249 z^{10} u^2 - 4 z^2 + 3 z^3 u^2 + 6 z^5 u^3 - 2 z^4 - 25 z^{19} u^3 + 31 z^{23} u^2 + 22 z^{24} u^2 + 122 z^{21} u^2 - 38 z^{23} u + 35 z^{22} u^2 - 32 z^{24} u + 106 z^{20} u - 26 z^{22} u - z^{20} u^3 + 29 z^{21} u^3 - 7 z^{20} u^4 + 121 z^{20} u^2 + 82 z^{21} u - 13 z^{19} u^4 + 80 z^{20} - 40 z^{21} + 3 z^{23} u^4 - 2 z^7 u^4 - 8 z^8 u^4 + 4 z^2 u - 3 z^{18} u^5 - 7 z^{24} u^3 - 3 z^{26} u^3 + z^{25} u^4 - 45 z^{24} - 72 z^{23} - 112 z^{22} - 8 z^4 u - 32 z^{10} - z^{30} u + 2 z^{12} u^5 + 8 z^{14} u^5 - 2 z^{10} u^5 + z^{15} u^6 - 2 z^{20} u^5 + 66 z^9 u^2 + 274 z^{10} u + 8 z^9 + 5 z^4 u^2 + 2 z^4 u^3 - 92 z^6 u^2 + 6 z^{15} u^5 - 16 z^9 u^4 + 2 z^{13} u^5 + 2 z^{11} u^5 - 20 z^{17} u^4 - 6 z^{16} u^5 - 4 z^{15} - 332 z^{13} u + 437 z^{11} u^2 + 246 z^{11} u - 296 z^{12} u - 246 z^{14} u - 152 z^{11} - 370 z^{12} - 104 z^{14} + 7 z^{10} u^3 - 43 z^8 u^3 - 49 z^9 u^3 - 19 z^{18} u^4 - 2 z^{17} u^5 - 6 z^3 - 4 z^5 - 9 z^6 + 77 z^7 + z^5 u^4 - 14 z^6 u^3 - 63 z^7 u^2 + 158 z^9 u - 37 z^7 u^3 - 105 z^6 u - 31 z^5 u + 174 z^8 u + 108 z^7 u + 64 z^8 - z^9 u^5 - 2 u^5 z^{19} + z^{14} u^6 + 10 z^{27} + 3 u^2 z^{26} + 13 u^2 z^{25} - 7 u^3 z^{23} - 3 u^3 z^{22} - 3 u^3 z^{25} + 7 u^4 z^{21} + u^4 z^{24} + 2 u^4 z^{22} - 6 u z^{27} - 22 u z^{26} - 28 u z^{25} - 2 u z^{28} - u z^{29} + u^2 z^{27} - 4 z^{26} - 31 z^{25} + 6 z^{28} + 4 z^{29} + 3 z^{30} + z^{31}) \Big/ (-28 z^{11} u^4 + 16 z^{12} u^4 - 70 z^8 u^2 - 180 z^{17} - 208 z^{16} - 7 z^{14} u^4 - 24 z^{18} u^3 + 40 z^{19} u^2 + 122 z^{18} + 96 z^{16} u^3 + 95 z^{15} u^3 - 160 z^{16} u + 46 z^{19} u + 20 z^{15} u^4 - 108 z^{13} + 1 - 122 z^{14} u^2 + 64 z^{13} u^2 + 72 z^{13} u^3 - 7 z^{10} u^4 - 107 z^{11} u^3 + 160 z^{12} u^3 + 542 z^{12} u^2 - 40 z^{14} u^3 - 6 z + 30 z^{16} u^2 + 174 z^{15} u^2 - z u + 116 z^{19} - 386 z^{17} u^2 + 10 z^5 u^2 + 20 z^{13} u^4 - 10 z^{18} u^2 + 16 z^{18} u + 86 z^{15} u - 209 z^{17} u^3 - 410 z^{17} u + 54 z^{16} u^4 - 148 z^{10} u^2 + 9 z^2 + 3 z^3 u + z^5 u^3 - 4 z^4 - 24 z^{19} u^3 + 6 z^{23} u^2 + 10 z^{24} u^2 + 18 z^{21} u^2 + 8 z^{23} u + 100 z^{22} u^2 - 8 z^{24} u - 48 z^{20} u + 120 z^{22} u - 24 z^{20} u^3 - 31 z^{21} u^3 - 4 z^{20} u^4 - 10 z^{20} u^2 + 48 z^{21} u - 12 z^{19} u^4 + 66 z^{20} + 120 z^{21} + z^6 u^4 - 2 z^8 u^4 - 27 z^{24} - 40 z^{23} + 72 z^{22} + 4 z^4 u + 40 z^{10} - 2 z^9 u^2 - 56 z^{10} u + 56 z^9 + z^4 u^2 + 66 z^6 u^2 + 2 z^{15} u^5 + 2 z^{13} u^5 - 2 z^{11} u^5 - 52 z^{17} u^4 + 12 z^{16} u^5 - 100 z^{15} - 42 z^{13} u - 78 z^{11} u^2 + 112 z^{11} u + 600 z^{12} u - 168 z^{14} u + 120 z^{11} + 218 z^{12} - 158 z^{14} - 64 z^{10} u^3 - 24 z^8 u^3 - 5 z^9 u^3 - 3 z^{18} u^4 - 6 z^{17} u^5 + 2 z^3 + 2 z^5 + 5 z^6 - 86 z^7 + 16 z^6 u^3 - 112 z^7 u^2 + 72 z^9 u - 12 z^7 u^3 + 64 z^6 u + 19 z^5 u - 32 z^8 u - 237 z^7 u + 13 z^8 - 2 u^5 z^{19} - 2 u^5 z^{21} - 14 z^{27} + z^{16} u^6 + 10 u^2 z^{26} + 8 u^2 z^{25} + u^2 z^{28} + 3 u^3 z^{23} + 32 u^3 z^{22} - 3 u^3 z^{27} - 4 u^3 z^{25} - 12 u^4 z^{21} + 2 u^4 z^{24} + 5 u^4 z^{22} + u^4 z^{26} - 17 u z^{27} - 8 u z^{26} - 5 u z^{25} - 4 u z^{28} - u z^{29} - u z^{31} + 2 u^2 z^{27} - 27 z^{26} - 14 z^{25} + 4 z^{28} + 2 z^{29} + z^{30} + 2 z^{31} + z^{32})$$

```
> series(subs([z=-z,u=-t],g012345),z=0,13):map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$1 + z^7 + 33 z^8 + 371 z^9 + 5029 z^{10} + 64311 z^{11} + 859487 z^{12} + O(z^{13})$$

```
> denom(g012345):series(",u,8):
> map(factor,");
```

$$(-1+z)^2\,(z^5+z^4+z^3+z^2+z-1)^2\,(z^{10}+z^9+z^7+z^6-6\,z^5-5\,z^4-3\,z^3-2\,z^2-z+1)^2-z\,(-1+z)\,(z^5+z^4+z^3+z^2+z-1)\,(z^{24}+z^{22}+4\,z^{21}+17\,z^{20}+10\,z^{19}+4\,z^{18}+10\,z^{17}-z^{16}-90\,z^{15}-45\,z^{14}+46\,z^{13}-30\,z^{12}-28\,z^{11}+231\,z^{10}+160\,z^9+51\,z^8+62\,z^7+16\,z^6-110\,z^5-23\,z^4-2\,z^3+z^2+2\,z+1)\,u+z^4\,(18\,z^{17}-10\,z^{16}+100\,z^{18}-386\,z^{13}+1+10\,z+6\,z^{19}+66\,z^2-70\,z^4+10\,z^{20}+8\,z^{21}+z^{24}+2\,z^{23}+10\,z^{22}-122\,z^{10}+64\,z^9+40\,z^{15}+174\,z^{11}+30\,z^{12}-10\,z^{14}-112\,z^3-2\,z^5-148\,z^6-78\,z^7+542\,z^8)\,u^2-z^5\,(209\,z^{12}+4\,z^{20}-72\,z^8-160\,z^7+3\,z^{22}+24\,z^{13}+107\,z^6+40\,z^9+24\,z^3+31\,z^{16}-96\,z^{11}-1+64\,z^5-16\,z-3\,z^{18}-32\,z^{17}+12\,z^2-95\,z^{10}+5\,z^4+24\,z^{15}+24\,z^{14})\,u^3+z^6\,(1+5\,z^{16}-52\,z^{11}+54\,z^{10}-7\,z^4-7\,z^8-2\,z^2-4\,z^{14}-3\,z^{12}-28\,z^5+z^{20}+2\,z^{18}+20\,z^7+16\,z^6-12\,z^{13}-12\,z^{15}+20\,z^9)\,u^4-2\,z^{11}\,(z^8-z^4-z^2+1-6\,z^5+3\,z^6+z^{10})\,u^5+z^{16}\,u^6$$

Only partial patterns are apparent.

# 4. Cyclically constrained position gaps

The problems we consider in this section are of so-called "cyclic" or "toroidal type". In the menage formulation, one may for instace have a round table (the classical formulation of the menage problem is of this type). Cyclic templates are in fact easily derivable from the linear ones that were considered earlier. This gives directly the enumeration of permutations with forced position gaps (in the cyclic sense) and the generating functions are, like for templates, rational functions. The inclusion-exclusion argument still aplies in the form already encountered and this provides the enumeration of permutations with forbidden position gaps (in the cyclic sense). In this way, we solve automatically the classical menage problem and its generalizations while completing in passing some of the entries of the *Encyclopedia of Integer Sequences* [EIS].

## Forced position gaps

One first seeks to enumerate the permutations such that all position gaps are ***forced*** to lie in some fixed set Ω. For Ω a subset of the nonnegative integers, the problem only makes sense if one considers the constraints to be "*toroidal*". In this section, permutation are thus considered as values modulo $n$ and attached to positions 1 .. $n$ on a cycle. For instance [2,3,4,1] is such Sthat each position gap is exactly 1.

The system of transitions is the same as before. However, in the graph of transitions, one must restrict attention to paths such that the initial and the final states are the same: this corresponds to the fact that constraints from the end of the permutations are propagated back to the beginning. In this basic enumeration problem, no inclusion-exclusion argument is needed, and the problem is directly modelled by a finite-state device related to the earlier ones.

```
> build_transition:=proc(state,Omega,initial_state) local x,t; if
  state=initial_state then t:=Epsilon else t:=NULL
  fi;s[state]=Union(t,Prod(dontcare,s[MinusOne(state)]),seq(Prod(
  a[x],s[MinusOne(state union {x})]),x=Omega minus state)) end;
```

*build_transition* := **proc**(*state*, Ω, *initial_state*)

**local** *x*, *t*;

**if** *state* = *initial_state* **then** *t* := E **else** *t* := *NULL* **fi**;

*s*[*state*] = Union(*t*, Prod(*dontcare*, *s*[MinusOne(*state*)]),

$$\operatorname{seq}(\operatorname{Prod}(a[x], s[\operatorname{MinusOne}(\mathit{state} \text{ union } \{x\})]), x = \Omega \text{ minus } \mathit{state}))$$

**end**

```
> build_transition({1,2},{0,1,2,3},{});
```

$$s_{\{1,2\}} = \operatorname{Union}(\operatorname{Prod}(\mathit{dontcare}, s_{\{0,1\}}), \operatorname{Prod}(a_0, s_{\{0,1\}}), \operatorname{Prod}(a_3, s_{\{0,1,2\}}))$$

The following procedure builds the specification that corresponds to a loop that starts and ends with *initial_state* in the transition graph. It is a simple modification of the previous procedure *build_grammar*.

```
> build_loop:=proc(Omega,initial_state)
  [s[{}],{build_alphabet(Omega)} union
  map(build_transition,combstruct[allstructs](Subset({$0..max(op(
  Omega))-1})),Omega,initial_state),unlabelled] end;
```

*build_loop* := **proc**(Ω, *initial_state*)

$[s[\{\ \}], \{\operatorname{build\_alphabet}(\Omega)\} \text{ union } \operatorname{map}(\mathit{build\_transition},$

$\mathit{combstruct}[\mathit{allstructs}](\operatorname{Subset}(\{ `\$`(0 .. \max(\operatorname{op}(\Omega)) - 1)\})), \Omega, \mathit{initial\_state}),$

$\mathit{unlabelled}]$

**end**

```
> build_loop({0},{});build_loop({0,1},{0});
```

$[s_{\{\ \}}, \{s_{\{\ \}} = \operatorname{Union}(E, \operatorname{Prod}(\mathit{dontcare}, s_{\{\ \}}), \operatorname{Prod}(a_0, s_{\{\ \}})), \mathit{marked} = E, a_0 = \mathit{Atom},$

$\mathit{dontcare} = \operatorname{Prod}(\mathit{Atom}, \mathit{marked})\}, \mathit{unlabelled}]$

$[s_{\{\ \}}, \{\mathit{marked} = E, a_0 = \mathit{Atom}, \mathit{dontcare} = \operatorname{Prod}(\mathit{Atom}, \mathit{marked}),$

$s_{\{0\}} = \operatorname{Union}(E, \operatorname{Prod}(\mathit{dontcare}, s_{\{\ \}}), \operatorname{Prod}(a_1, s_{\{0\}})),$

$s_{\{\ \}} = \operatorname{Union}(\operatorname{Prod}(\mathit{dontcare}, s_{\{\ \}}), \operatorname{Prod}(a_0, s_{\{\ \}}), \operatorname{Prod}(a_1, s_{\{0\}})), a_1 = \mathit{Atom}\}, \mathit{unlabelled}]$

We could construct a composite grammar for all loops, it is however simpler to solve for the GF of each loop.

```
> gf_loops:=proc(Omega) local is;
  normal(add(subs(gfsolve(op(2,build_loop(Omega,is)),unlabelled,z
  ,[[u,marked]]),s[is](z,u)),is=combstruct[allstructs](Subset({$0
  ..max(op(Omega))-1})))) end;
```

*gf_loops* := **proc**(Ω)

**local** *is*;

normal( add(

$\operatorname{subs}(\operatorname{gfsolve}(\operatorname{op}(2, \operatorname{build\_loop}(\Omega, \mathit{is})), \mathit{unlabelled}, z, [[u, \mathit{marked}]]), s[\mathit{is}](z, u)),$

$\mathit{is} = \mathit{combstruct}[\mathit{allstructs}](\operatorname{Subset}(\{ `\$`(0 .. \max(\operatorname{op}(\Omega)) - 1)\}))))$

**end**

The bivariate GF counts configurations with possible exceptions (it is useful for the analysis of configurations with forbidden gaps).

In the case where only singletons are allowed, then there is only 1 possibility for each size *n*.

```
> l0:=gf_loops({0},0); subs(u=0,");series(",z=0,10);
```

$$\mathit{l0} := -\frac{1}{-1 + z\,u + z}$$

$$-\frac{1}{-1 + z}$$

$$1 + z + z^2 + z^3 + z^4 + z^5 + z^6 + z^7 + z^8 + z^9 + O(z^{10})$$

In the case $\Omega = \{ 0, 1 \}$, the first element (either 0 or 1) conditions all the rest.

```
> l01:=gf_loops({0,1},1);subs(u=0,");series(",z=0,10);
```

$$l01 := -\frac{-2 + z\,u + 2\,z}{1 - z\,u - 2\,z + z^2}$$

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

$$2 + 2\,z + 2\,z^2 + 2\,z^3 + 2\,z^4 + 2\,z^5 + 2\,z^6 + 2\,z^7 + 2\,z^8 + 2\,z^9 + \mathrm{O}(z^{10})$$

The case $\Omega = \{ 0, 1, 2 \}$ is the first one that is nontrivial.

```
> l012:=gf_loops({0,1,2},2);subs(u=0,");series(",z=0,10);
```

$$l012 := -\frac{4 - 9\,z - 3\,z\,u + 4\,z^2 + z^3 + z^3\,u}{(z^3 - z^2\,u - 2\,z - z\,u + 1)\,(-1 + z)}$$

$$-\frac{4 - 9\,z + 4\,z^2 + z^3}{(z^3 - 2\,z + 1)\,(-1 + z)}$$

$$4 + 3\,z + 5\,z^2 + 6\,z^3 + 9\,z^4 + 13\,z^5 + 20\,z^6 + 31\,z^7 + 49\,z^8 + 78\,z^9 + \mathrm{O}(z^{10})$$

The sequence 3, 5, 6, 9, 13,etc, is sequence **M2396** of [*EIS*].

```
> l0123:=gf_loops({0,1,2,3},3);subs(u=0,");series(",z=0,12);
```

$$l0123 := -\frac{-8 + 28\,z + 7\,z\,u - 24\,z^2 - 5\,z^3\,u - 8\,z^4 + 9\,z^5\,u + 12\,z^5 - 16\,z^4\,u - 4\,z^4\,u^2 + z^7\,u}{-z\,u + 4\,z^4\,u + 4\,z^2 + 2\,z^4 + 1 - 4\,z + z^3\,u - 4\,z^5 - 3\,z^5\,u + z^4\,u^2 - z^7\,u + z^8}$$

$$-\frac{-8 + 28\,z - 24\,z^2 - 8\,z^4 + 12\,z^5}{1 - 4\,z + 4\,z^2 + 2\,z^4 - 4\,z^5 + z^8}$$

$$8 + 4\,z + 8\,z^2 + 16\,z^3 + 24\,z^4 + 44\,z^5 + 80\,z^6 + 144\,z^7 + 264\,z^8 + 484\,z^9 + 888\,z^{10} + 1632\,z^{11} + \mathrm{O}(z^{12})$$

**Generating functions**. These are clearly rational. Let $\mathrm{f}(z) = \frac{\mathrm{N}(z)}{\mathrm{D}(z)}$ be the GF that arises in each case. Then, a general result concerning generating functions of loops in graphs [Biggs, *Algebraic Graph Theory*, 1993] implies that $\mathrm{f}(z) - m$, where $m = 2^{(d-1)}$ is the dimension of the state space, is a logarithmic derivative,

$$\mathrm{f}(z) - m = \frac{z\left(\frac{\partial}{\partial z}\mathrm{D}(z)\right)}{\mathrm{D}(z)}$$

For instance:

```
> L0123:=subs(u=0,l0123);
```

$$L0123 := -\frac{-8 + 28\,z - 24\,z^2 - 8\,z^4 + 12\,z^5}{1 - 4\,z + 4\,z^2 + 2\,z^4 - 4\,z^5 + z^8}$$

```
> int((L0123-8)/z,z);
```

$$-2\ln(z^4 + 1 - 2\,z)$$

This is also true of the bivariate GF:

```
> int((l0123-8)/z,z);
```

$$-\ln(-z\,u + 4\,z^4\,u + 4\,z^2 + 2\,z^4 + 1 - 4\,z + z^3\,u - 4\,z^5 - 3\,z^5\,u + z^4\,u^2 - z^7\,u + z^8)$$

**Recurrences**. The GF's for forced position gaps are always *rational*. Linear recurrences (with constant coefficients) are then available by gfun.

```
> series(L0123,z=0,12);
```

$$8+4z+8z^2+16z^3+24z^4+44z^5+80z^6+144z^7+264z^8+484z^9+888z^{10}+1632z^{11}+O(z^{12})$$

```
> holexprtodiffeq(L0123,y(z));
```

$$(z^4+1-2z)\,\mathrm{y}(z)+12z-8$$

```
> recl0123:=diffeqtorec(",y(z),u(n));
```

$$recl0123 := \{-\mathrm{u}(n)-\mathrm{u}(n+1)-\mathrm{u}(n+2)+\mathrm{u}(n+3)+4,\ \mathrm{u}(0)=8,\ \mathrm{u}(1)=4,\ \mathrm{u}(2)=8\}$$

High index values can be obtained quickly by the binary powering method that is implemented in gfun[rectoproc]. The recurrence needs to be made homogeneous, first.

```
> op(1,recl0123)-subs(n=n+1,op(1,recl0123));
```

$$-\mathrm{u}(n)+2\,\mathrm{u}(n+3)-\mathrm{u}(n+4)$$

```
> sl0123:=rectoproc({",u(0)=8,u(1)=4,u(2)=8,u(3)=16},u(n));
```

*sl0123* := **proc**(*n*)

**local** *a*, *res*, *i*, *l*, *u0*, *u1*, *u2*, *u3*, *u4*;

**if** $n \le 256$ **then**

*u0* := 8;

*u1* := 4;

*u2* := 8;

*u3* := 16;

**for** *i* **from** 4 **to** $n-1$ **do** *u4* := –*u0* + 2**u3*; *u0* := *u1*; *u1* := *u2*; *u2* := *u3*; *u3* := *u4*

**od**;

–*u0* + 2**u3*

**else**

*a* := array( 1 .. 4, 1 .. 4, [ [ 2, 0, 0, -1 ], [ 1, 0, 0, 0 ], [ 0, 1, 0, 0 ], [ 0, 0, 1, 0 ] ] );

*res* := array( 1 .. 4, [ 16, 8, 4, 8 ] );

*l* := convert( $n-3$, *base*, 2 );

**if** *l*[ 1 ] = 1 **then** *res* := array( 1 .. 4, [ 2 = 16, 3 = 8, 4 = 4, 1 = 24 ] ) **fi**;

**if** *l* = [ 1 ] **then** *res*[ 1 ]

**else**

**for** *i* **in** subsop( 1 = ( ), nops( *l* ) = ( ), *l* ) **do**

*a* := array( 1 .. 4, 1 .. 4, [ ( 4, 3 ) = *a*[ 4, 1 ]**a*[ 1, 3 ] + *a*[ 4, 2 ]**a*[ 2, 3 ] + *a*[ 4, 3 ]**a*[ 3, 3 ] + *a*[ 4, 4 ]**a*[ 4, 3 ], ( 1, 2 ) = *a*[ 1, 1 ]**a*[ 1, 2 ] + *a*[ 1, 2 ]**a*[ 2, 2 ] + *a*[ 1, 3 ]**a*[ 3, 2 ] + *a*[ 1, 4 ]**a*[ 4, 2 ], ( 4, 4 ) = *a*[ 1, 4 ]**a*[ 4, 1 ] + *a*[ 2, 4 ]**a*[ 4, 2 ] + *a*[ 3, 4 ]**a*[ 4, 3 ] + *a*[ 4, 4 ]^2, ( 2, 3 ) = *a*[ 2, 1 ]**a*[ 1, 3 ] + *a*[ 2, 2 ]**a*[ 2, 3 ] + *a*[ 2, 3 ]**a*[ 3, 3 ] + *a*[ 2, 4 ]**a*[ 4, 3 ], ( 3, 4 ) = *a*[ 3, 1 ]**a*[ 1, 4 ] + *a*[ 3, 2 ]**a*[ 2, 4 ] + *a*[ 3, 3 ]**a*[ 3, 4 ] + *a*[ 3, 4 ]**a*[ 4, 4 ], ( 2, 1 ) = *a*[ 2, 1 ]**a*[ 1, 1 ] + *a*[ 2, 2 ]**a*[ 2, 1 ] + *a*[ 2, 3 ]**a*[ 3, 1 ] + *a*[ 2, 4 ]**a*[ 4, 1 ], ( 2, 4 ) = *a*[ 2, 1 ]**a*[ 1, 4 ] + *a*[ 2, 2 ]**a*[ 2, 4 ] + *a*[ 2, 3 ]**a*[ 3, 4 ] + *a*[ 2, 4 ]**a*[ 4, 4 ], ( 3, 2 ) = *a*[ 3, 1 ]**a*[ 1, 2 ] + *a*[ 3, 2 ]**a*[ 2, 2 ]

```
                  + a[3, 3]*a[3, 2] + a[3, 4]*a[4, 2], (2, 2) =
                  a[1, 2]*a[2, 1] + a[2, 2]^2 + a[2, 3]*a[3, 2] + a[2, 4]*a[4, 2],
                  (4, 1) = a[4, 1]*a[1, 1] + a[4, 2]*a[2, 1] + a[4, 3]*a[3, 1]
                  + a[4, 4]*a[4, 1], (1, 4) = a[1, 1]*a[1, 4] + a[1, 2]*a[2, 4]
                  + a[1, 3]*a[3, 4] + a[1, 4]*a[4, 4], (1, 3) = a[1, 1]*a[1, 3]
                  + a[1, 2]*a[2, 3] + a[1, 3]*a[3, 3] + a[1, 4]*a[4, 3], (4, 2) =
                  a[4, 1]*a[1, 2] + a[4, 2]*a[2, 2] + a[4, 3]*a[3, 2]
                  + a[4, 4]*a[4, 2], (1, 1) =
                  a[1, 1]^2 + a[1, 2]*a[2, 1] + a[1, 3]*a[3, 1] + a[1, 4]*a[4, 1],
                  (3, 1) = a[3, 1]*a[1, 1] + a[3, 2]*a[2, 1] + a[3, 3]*a[3, 1]
                  + a[3, 4]*a[4, 1], (3, 3) =
                  a[1, 3]*a[3, 1] + a[2, 3]*a[3, 2] + a[3, 3]^2 + a[3, 4]*a[4, 3]]);
              if i = 1 then res := array(1 .. 4, [2 =
                  a[2, 1]*res[1] + a[2, 2]*res[2] + a[2, 3]*res[3] + a[2, 4]*res[4],
                  3 =
                  a[3, 1]*res[1] + a[3, 2]*res[2] + a[3, 3]*res[3] + a[3, 4]*res[4],
                  4 =
                  a[4, 1]*res[1] + a[4, 2]*res[2] + a[4, 3]*res[3] + a[4, 4]*res[4],
                  1 =
                  a[1, 1]*res[1] + a[1, 2]*res[2] + a[1, 3]*res[3] + a[1, 4]*res[4]])
              fi
          od;
          (a[1, 1]^2 + a[1, 2]*a[2, 1] + a[1, 3]*a[3, 1] + a[1, 4]*a[4, 1])*res[1] +
              (a[1, 1]*a[1, 2] + a[1, 2]*a[2, 2] + a[1, 3]*a[3, 2] + a[1, 4]*a[4, 2])*
              res[2] +
              (a[1, 1]*a[1, 3] + a[1, 2]*a[2, 3] + a[1, 3]*a[3, 3] + a[1, 4]*a[4, 3])*
              res[3] +
              (a[1, 1]*a[1, 4] + a[1, 2]*a[2, 4] + a[1, 3]*a[3, 4] + a[1, 4]*a[4, 4])*
              res[4]
      fi
   fi
end
```

```
> sl0123(1000);
```

8922231345392004165904999737124438107493716088304945490594273343717650420\
   61538517012888490692235068242530557693062246508874517804919747179957545665\
   79370036542235073051049831184236881812088091583521208448630999657070043494\
   4638412930546113516282190295223135713441544

The complexity of this computation is thus of O(log(n)) arithmetic operations for the *n*th term:

```
> for m in [1000,2000,4000,8000,16000,32000] do
  st:=time():sl0123(m):print(time()-st); od:
```

.061

$$.082$$

$$.119$$

$$.182$$

$$.319$$

$$1.242$$

The computation of the value of index 32000 takes typically 1 second of computer time though the number has 8470 digits:

```
> sl0123(32000)*1.0;
```

$$.1211215700\ 10^{8470}$$

(Such fast algorithms are useful for instance when analyzing patterns in strings, like in DNA sequences.)

## Forbidden position gaps

We are dealing here with menage problems in the classical sense of a circular table. The same integral transformation as in the noncyclic/nontoroidal case works for obtaining the GF of permutations with excluded patterns, *i.e.*, no position gap in $\Omega$.

The case $\Omega = \{ 0 \}$ gives back the derangement numbers (again, this is **M1937** of [EIS]).

```
> series(subs([z=-z,u=-t],l0),z=0,13):map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$1 + z^2 + 2\,z^3 + 9\,z^4 + 44\,z^5 + 265\,z^6 + 1854\,z^7 + 14833\,z^8 + 133496\,z^9 + 1334961\,z^{10} + 14684570\,z^{11} + 176214841\,z^{12} + \mathrm{O}(z^{13})$$

Next comes the case $\Omega = \{ 0, 1 \}$.

```
> series(subs([z=-z,u=-t],l01),z=0,13):map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$2 - z + z^3 + 2\,z^4 + 13\,z^5 + 80\,z^6 + 579\,z^7 + 4738\,z^8 + 43387\,z^9 + 439792\,z^{10} + 4890741\,z^{11} + 59216642\,z^{12} + \mathrm{O}(z^{13})$$

This, apart from the first two values (see the next section for possible adjustments) is exactly the sequence of *Menage Numbers* that is listed as **M2062** in [EIS].

```
> int(x*exp(-t)*subs([z=-z,u=-t],l01-2),t=0..infinity);
```

$$-\frac{x\left(\mathbf{e}^{(-2)}\,\mathrm{Ei}\left(1, -\frac{1+2\,z+z^2}{z}\right) + z\,\mathbf{e}^{\left(\frac{1+z^2}{z}\right)} - \mathbf{e}^{(-2)}\,\mathrm{Ei}\left(1, -\frac{1+2\,z+z^2}{z}\right)z^2\right)\mathbf{e}^{\left(-\frac{1+z^2}{z}\right)}}{z}$$

```
> convert_hypergeom(");
```

$$-\frac{\left(z + z\,\mathrm{hypergeom}\left([1, 1], [\ ], \frac{z}{1+2\,z+z^2}\right) - \mathrm{hypergeom}\left([1, 1], [\ ], \frac{z}{1+2\,z+z^2}\right) + 1\right)x}{z+1}$$

This gives the ordinary GF for the menage problem. This GF is closely related to that of the linear menage problem.

The generalized menage numbers that correspond to $\Omega = \{ 0, 1, 2 \}$ appear as **M2121** in [EIS] under the name "discordant permutations of length $n$".

```
> series(subs([z=-z,u=-t],l012),z=0,13):map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$4-2z+z^2+z^4+2z^5+20z^6+144z^7+1265z^8+12072z^9+126565z^{10}+1445100z^{11}+17875140z^{12}+O(z^{13})$$

The next sequence called "hit polynomials" is given as **M2154** in [EIS], where it stops at the value 369321.

```
> series(subs([z=-z,u=-t],l0123),z=0,13):map(proc(x)
  int(x*exp(-t),t=0..infinity) end,");
```

$$8-3z+2z^2-z^3+z^5+2z^6+31z^7+264z^8+2783z^9+30818z^{10}+369321z^{11}+4745952z^{12}+O(z^{13})$$

It is possible to complete the table to arbitrary orders, obtaining exact forms of the GF, recurrences, etc. For instance:

```
> int(subs([z=-z,u=-t],l0123)*exp(-t),t=0..infinity):
> hl0123:=convert_hypergeom(");
```

$$\begin{aligned}hl0123 := 4z^2(&-4\sqrt{\%1}-12\sqrt{\%1}\,z+7z^4\%3+4z\%4+4\sqrt{\%1}\,z^3-8z^6\%4-4z\%3\\
&-7z^4\%4-z^{12}\%4-4z^9\%4+z^{12}\%3+8z^5\%3+4z^9\%3-10z^3\%4+8z^6\%3\\
&-2\%4\sqrt{\%1}-8z^5\%4-2z^{11}\%3+10z^3\%3+2z^{11}\%4+z^8\%3-2\%3\sqrt{\%1}+\%4-\%3\\
&-4z^2\sqrt{\%1}-2z^6\%4\sqrt{\%1}-z^8\%4+2z^7\%4\sqrt{\%1}+2z^7\%3\sqrt{\%1}-4z^7\sqrt{\%1}\\
&+4z^6\sqrt{\%1}-4z^5\sqrt{\%1}-12z^4\sqrt{\%1}-4z\%3\sqrt{\%1}+2z^5\%3\sqrt{\%1}-4z\%4\sqrt{\%1}\\
&+2z^5\%4\sqrt{\%1}-2z^6\%3\sqrt{\%1})\Big/\left(\%2\,(-z^5+z^4-4z^3+z-1+\sqrt{\%1})\sqrt{\%1}\right)\end{aligned}$$

$$\%1 := (z^6-2z^5+3z^4+4z^3-5z^2-2z+1)(1+z^2)^2$$

$$\%2 := -z^2+z^6+3z^4+4z^3+1+\sqrt{\%1}\,z+\sqrt{\%1}$$

$$\%3 := \mathrm{hypergeom}\left([1,1],[\ ],2\frac{z^3}{\%2}\right)$$

$$\%4 := \mathrm{hypergeom}\left([1,1],[\ ],2\frac{z^3}{-z^2+z^6+3z^4+4z^3+1-\sqrt{\%1}\,z-\sqrt{\%1}}\right)$$

```
> series(hl0123,z=0,33);
```

$$\begin{aligned}&8-3z+2z^2-z^3+z^5+2z^6+31z^7+264z^8+2783z^9+30818z^{10}+369321z^{11}+4745952z^{12}+65275999z^{13}+957874226z^{14}\\
&+14951584189z^{15}+247524019720z^{16}+4334022049377z^{17}+80052395326514z^{18}+1555999253409203z^{19}\\
&+31755107852542144z^{20}+679008663143893773z^{21}+15182701602959054546z^{22}+354364531995856105099z^{23}\\
&+8618865446674052425224z^{24}+218107566239993684272531z^{25}+5734302907005373513322674z^{26}\\
&+156419366922989437119483053z^{27}+4421296327690290833867429280z^{28}+129342793572697058001361979515z^{29}\\
&+3911872744020239697091063448610z^{30}+O(z^{31})\end{aligned}$$

# Conclusions

The combinatorial packages may be used to facilitate the enumeration of permutations with contraints on "positions". In essence, a finite-state model is solved that leads to a bivariate generating function $R(z,t)$ and the GF's of interest appear to be integral transforms of $R(z,t)$. In particular these GF are all holonomic. Thus, the corresponding counting sequences can be determined in $O(n)$ arithmetic operations.

If partial fraction decompositions are effected on $R(z, t)$, then the GF's appear to be systematically expressible as compositions of the divergent hypergeometric series hypergeom( [ 1, 1 ], [ ], $z$ ) with algebraic functions. The simplest of these forms have been made explicit above in some cases, like derangements, menage, and successions. The same method also allows for counting Ω-exceptions in *all* permutations by means of bivariate generating functions. A typical problem in this range is the statistics of the number of "conflicts" in a random assignment of seats in the menage problem.

The fast algorithms make it possible to perform numerical observations and conjecture various asymptotic patterns. One of these, which is confirmed in the particular cases treated above is [see, I. Vardi, *Computational Recreations with Mathematica*, Addison-Wesley, 1991; p. 123]:

**Property**. *The fraction of permutations with gaps that do not belong to a finite set Ω is asymptotic to*

$$e^{(-c)},$$

*where c = card( Ω ). More generally, the probability that a random permutation of size n has k exceptions of type Ω (with k fixed) is given by a Poisson law of parameter c,*

$$Pr(\#\ exceptions = k) = \frac{\mathbf{e}^{(-c)}\, c^k\, (1 + o(1))}{k!}.$$

A weaker form of this property (for position gaps where Ω is an initial segment of the integers) is established by probabilistic arguments in [Barbour, Holst, Svanson, *Poisson Approximation*, Oxford, 1992].

The approach via hypergeometrics points to a unified derivation via generating functions and divergent series that encompasses more general constraints (like succession gaps and the case where Ω is not an initial segment of the integers). The principle of the generating function method is that

$$\mathit{coeff}_{z^n}\ \mathrm{hypergeom}([1, 1], [\ ], z + d\, z^2 + O(z^3)) = n!\, e^d\, (1 + o(1)),$$

provided the argument of the hypergeometric is a function that is analytic at the origin. For instance, in the case of the menage problem, one has

```
> P01=convert_hypergeom(P01);
```

$$-\frac{\mathrm{Ei}\left(1, -\frac{1 + 2z + z^2}{z}\right)(z+1)\,\mathbf{e}^{\left(-\frac{(z+1)^2}{z}\right)}}{z} = \frac{\mathrm{hypergeom}\left([1, 1], [\ ], \frac{z}{1 + 2z + z^2}\right)}{z+1}$$

```
> series(1/(1+2*z+z^2),z=0,6);
```

$$1 - 2z + 3z^2 - 4z^3 + 5z^4 - 6z^5 + O(z^6)$$

so that the asymptotic proportion of menage permutations is *provedly* equal to $e^{(-2)}$. The method works for any of the examples discussed in this worksheet.

A companion worksheet shows that a similar approach applies to "succession gaps", by which we means constraints on the values of successive elements in a permutation. The corresponding solutions serve in the analysis of multiconnectivity in random interconnection graphs.