Enumerating alcohols and other classes of chemical moleculs, an example of Polya theory

Frederic Chyzak
(Version of January 13, 1997)

Alkanes are a simple class of chemical compounds. They are generically described by the chemical formula $C_n H_{2n+2}$. First examples for small n are methane ($n = 1$), ethane ($n = 2$), propane ($n = 3$), butane ($n = 4$), a.s.o. For a given n however, there exist several different *isomers*, i.e., different structures of bonds between atoms. In chemistry, there is much interest in knowing the number, or better yet the list, of such isomers. Alcohols are obtained from alkanes by replacing a hydrogen atom by an *OH* group. It follows that they are isomorphic to carbon chains with a distinguished node, or again to alkyl radicals $C_n H_{2n+1}$, which are alkanes with a missing hydrogen atom. If we disregard geometrical constraints (i.e., if we consider *structural* isomers only, and not *conformational* isomers), this leads to a pure graph-theoretical problem: how many rooted trees are there with n internal nodes, where each internal node has degree 4?
In this session, we thus consider *rooted* trees, so that we count and enumerate alkyls, with generic formula $C_n H_{2n+1}$. The combinatorics also corresponds to simple alcohols $C_n H_{2n+1} \mathrm{O} H$, organo-metalic compounds $C_n H_{2n+1} X$, and any other monosubstituted alkanes. We next treat the cases of disubstituted and trisubstituted alkanes. We develop the study of our models using the package Combstruct.

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

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

Enumerations of such classes of chemical compounds are part of Polya theory. We refer to the book by G. Polya and R. C. Read [*Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds*, (1987), Springer-Verlag] for more extensive results.

Monosubstituted alkanes, $C_n H_{2n+1} X$

In this section, we study monosubstituted alkanes, i.e., *rooted* trees, first without any constraint, next according to the *height*.

General alkyls

Definition

An alkyl radical can be viewed as a carbon atom linked to at most 3 alkyl radicals. Thus, we only take into account hydrogen atoms implicitly. There is no loss of information, since hydrogen atoms can always be recovered from the carbon skeleton. This yields the class equation $Alkyl = Carbon\,(\mathrm{E} + Alkyl + Alkyl^2 + Alkyl^3)$, which we map into the following grammar:

```
> gramm_Alkyl:=Alkyl=Prod(Carbon,Set(Alkyl,card<=3)),Carb
  on=Atom:
  specs_Alkyl:=[Alkyl,{gramm_Alkyl},unlabelled]:
```

Note that since the Set construct denotes multisets, i.e., sets with repetitions, a carbon atom of an alkyl is allowed to be bound to two copies of the same subtree (but the order of the subtrees does not matter).

Define the size of an alkyl as the number of carbon atoms it contains. We compute the number of alkyls of a given size using combstruct[count].

```
> seq(count(specs_Alkyl,size=i),i=0..50);
```

0, 1, 1, 2, 4, 8, 17, 39, 89, 211, 507, 1238, 3057, 7639, 19241, 48865, 124906, 321198, 830219, 2156010, 5622109, 14715813, 38649152, 101821927, 269010485, 712566567, 1891993344, 5034704828, 13425117806, 35866550869, 95991365288, 257332864506, 690928354105, 1857821351559, 5002305607153, 13486440075669, 36404382430278, 98380779170283, 266158552000477, 720807976831447, 1954002050661819, 5301950692017063, 14398991611139217, 39137768751465752, 106465954658531465, 289841389106439413, 789642117549095761, 2152814945971655556, 5873225808361331954, 16033495247557039074, 43797554941937577760

This series appears as the entry **M1146** ("quartic planted trees with *n* nodes") in the book by N. J. A. Sloane and S. Plouffe [*The Encyclopedia of Integer Sequences*, (1995), Academic Press].

Here is an example of an alkyl with 6 carbon atoms, obtained by the command combstruct[draw].

```
> alk:=draw(specs_Alkyl,size=6);
```

$alk := \mathrm{Prod}(Carbon, \mathrm{Set}(\mathrm{Prod}(Carbon, \mathrm{Set}(\mathrm{Prod}(Carbon, \mathrm{Set}(\mathrm{Prod}(Carbon, \mathrm{E}))), \mathrm{Prod}(Carbon, \mathrm{Set}(\mathrm{Prod}(Carbon, \mathrm{E})))))))$

The following procedure rewrites an alkyl into a more readable way.

```
> nice:=proc(alk)
  eval(subs({Epsilon=NULL,Carbon=C,Prod=proc() global H;
  [args] end,Set=proc() args end},alk)) end:
> nice(alk);
```

$$[C, [C, [C, [C]], [C, [C]]]]$$

The following procedure computes the size of a given alkyl.

```
> size:=proc(alk) option remember;
  1+convert(map(size,op(2,alk)),`+`) end:
  size(Prod(Carbon,Epsilon)):=1:
```

The following procedure computes the height of a given alkyl.

```
> height:=proc(alk) option remember;
  1+max(op(map(height,op(2,alk)))) end:
  height(Prod(Carbon,Epsilon)):=1:
```

Here is an alkyl with 50 carbon atoms, its nice representation and height.

```
> alk:=draw(specs_Alkyl,size=50):
> nice(alk);
```

$[C, [C, [C, [C], [C, [C, [C, [C, [C], [C, [C, [C], [C, [C], [C]]]], [C, [C, [C, [C]]], [C, [C, [C]]], [C, [C, [C], [C], [C, [C], [C], [C, [C]]]], [C, [C], [C, [C]]]]]],$

$[C, [C, [C, [C, [C]], [C, [C], [C]]]]], [C, [C, [C, [C]]], [C, [C, [C]]]]]]]$
$]$

```
> height(alk);
```

$$13$$

Empirical study

Drawing

By drawing several random structures, we can study probabilistic properties of alkyls. For instance, the following is a probabilistic estimate of their height on average:

```
> for i to 10 do
  ho[i]:=height(draw(specs_Alkyl,size=50)) od;
```

$$ho_1 := 18$$

$$ho_2 := 15$$

$$ho_3 := 19$$

$$ho_4 := 13$$

$$ho_5 := 13$$

$$ho_6 := 15$$

$$ho_7 := 13$$

$$ho_8 := 16$$

$$ho_9 := 14$$

$$ho_{10} := 17$$

```
> add(ho[i],i=1..10)/10.;
```

$$15.30000000$$

In the same way, we get a probabilistic estimate of their standard deviation:

```
> sqrt(add((ho[i]-")^2,i=1..10)/10);
```

$$2.051828453$$

Exhaustive enumeration

The command combstruct[draw] permits us to draw *one* structure at random. We can also generate *all* alkyls of a given size, using combstruct[allstructs], so as to compute the mean of a particular parameter exactly, or to count all those with a particular property. For instance, the height of trees cannot be represented in the class of combinatorial structures when using Combstruct. For instance, by computing all alkyls of size 5, we get the distribution of height for these alkyls (in their nice representation).

```
> allstructs(specs_Alkyl,size=5):
> map(nice,");
```

$[[C, [C, [C], [C, [C]]]], [C, [C], [C], [C, [C]]], [C, [C], [C, [C], [C]]],$
$[C, [C, [C, [C, [C]]]]], [C, [C, [C], [C], [C]]], [C, [C, [C, [C], [C]]]],$
$[C, [C], [C, [C, [C]]]], [C, [C, [C]], [C, [C]]]]]$

```
> sort(map(height,""));
```

$$[3, 3, 3, 3, 4, 4, 4, 5]$$

Here we count 4 alkyls of size 5 and height 3, 3 alkyls of size 5 and height 4, and 1 alkyl of size 5 and height 5.

By the same method, we get the exact mean and standard deviation of the height for small sizes.

```
> esd:=proc(n) local i,as,mean;
       as:=map(height,allstructs(specs_Alkyl,size=n));
       mean:=evalf(convert(as,`+`)/nops(as));

  nops(as),mean,evalf(sqrt(add((i-mean)^2,i=as))/nops(as))
  end:
> for i from 2 to 6 do i=esd(i) od;
```

$$2 = (1, 2., 0)$$

$$3 = (2, 2.500000000, .3535533906)$$

$$4 = (4, 3., .3535533905)$$

$$5 = (8, 3.625000000, .2460627461)$$

$$6 = (17, 4.117647059, .2017630413)$$

We could go up to $i = 9$ in less than 2 minutes.

Alkyls according to their height

Grammar

We define the class $Alkyl\_height_n$ to be the class of alkyls of height *at most* n. An alkyl of height at most n can be viewed as a carbon atom linked to at most three alkyls of height at most $n - 1$, according to the equation

$$Alkyl\_height_n = Carbon\,(\mathrm{E} + Alkyl\_height_{n-1} + Alkyl\_height_{n-1}^{\;2} + Alkyl\_height_{n-1}^{\;3}).$$

```
> gramm_ltd_height:=proc(n) option remember;

  Alkyl_height[n]=Prod(Carbon,Set(Alkyl_height[n-1],card<=3)),gramm_ltd_height(n-1)
  end:
  gramm_ltd_height(1):=Alkyl_height[1]=Prod(Carbon,Epsilon),Carbon=Atom:
  specs_ltd_height:=proc(n) option remember;
       [Alkyl_height[n],{gramm_ltd_height(n)},unlabelled]
  end:
```

The following procedure rewrites an alkyl into a more readable way.

```
> nice:=proc(alk)
  eval(subs({Epsilon=NULL,Carbon=C,Prod=proc() global H;
  [args] end,Set=proc() args end},alk)) end:
```

The following procedures compute the size and height of a given alkyl.

```
> size:=proc(alk) option remember;
  1+convert(map(size,op(2,alk)),`+`) end:
  size(Prod(Carbon,Epsilon)):=1:
```

```
> height:=proc(alk) option remember;
  1+max(op(map(height,op(2,alk)))) end:
  height(Prod(Carbon,Epsilon)):=1:
```

For instance, we compute the height of a random alkyl of size 10 and height at most 5.

```
> alk:=draw(specs_ltd_height(5),size=10):
> nice(alk);
```

$$[C,[C],[C,[C]],[C,[C,[C],[C],[C,[C]]]]]$$

```
> size(alk),height(alk);
```

$$10, 5$$

In this section, we proceed to compute a table of the number of alkyls according to their size and height. The first method is by generating all structures. Next, we use generating functions to extend the table.

Generating all structures

The following procedure remembers all the alkyls of a given size and with bounded height.

```
> list_all_st:=proc(d,s) option remember;
  allstructs(specs_ltd_height(d),size=s) end:
```

An alkyl of height h has a size at most $\frac{3^h-1}{2}$. Therefore, to produce all alkyls with height at most h_{max}, we need to produce all alkyls with size up to $s_{max}=\frac{3^{h_{max}}-1}{2}$.

```
> s[max]:=(3^h[max]-1)/2;
```

$$s_{max} := \frac{1}{2}3^{h_{max}}-\frac{1}{2}$$

To begin with, we enumerate all alkyls with height at most 3.

```
> h[max]:=3;
```

$$h_{max} := 3$$

```
> for i from 1 to s[max] do
  i,nops(list_all_st(h[max],i)),map(nice,list_all_st(h[ma
  x],i)) od;
```

$$1, 1, [[C]]$$

$$2, 1, [[C,[C]]]$$

$$3, 2, [[C,[C],[C]],[C,[C,[C]]]]$$

$$4, 3, [[C,[C],[C,[C]]],[C,[C,[C],[C]]],[C,[C],[C],[C]]]$$

$$5, 4, [[C,[C],[C],[C,[C]]],[C,[C],[C,[C],[C]]],[C,[C,[C],[C],[C]]],$$
$$[C,[C,[C]],[C,[C]]]]$$

$$6, 4, [[C,[C],[C,[C]],[C,[C]]],[C,[C],[C,[C],[C],[C]]],$$
$$[C,[C,[C]],[C,[C],[C]]],[C,[C],[C],[C,[C],[C]]]]$$

$$7, 5, [[C,[C,[C]],[C,[C]],[C,[C]]],[C,[C],[C],[C,[C],[C],[C]]],$$
$$[C,[C],[C,[C]],[C,[C],[C]]],[C,[C,[C],[C]],[C,[C],[C]]],$$
$$[C,[C,[C]],[C,[C],[C],[C]]]]$$

$8, 4, [[C, [C], [C, [C], [C]], [C, [C], [C]]],$
$[C, [C, [C], [C]], [C, [C], [C], [C]]], [C, [C], [C, [C]], [C, [C], [C], [C]]],$
$[C, [C, [C]], [C, [C]], [C, [C], [C]]]]$
$9, 4, [[C, [C], [C, [C], [C]], [C, [C], [C], [C]]],$
$[C, [C, [C]], [C, [C]], [C, [C], [C], [C]]],$
$[C, [C, [C]], [C, [C], [C]], [C, [C], [C]]],$
$[C, [C, [C], [C], [C]], [C, [C], [C], [C]]]]$
$10, 3, [[C, [C, [C]], [C, [C], [C]], [C, [C], [C], [C]]],$
$[C, [C], [C, [C], [C], [C]], [C, [C], [C], [C]]],$
$[C, [C, [C], [C]], [C, [C], [C]], [C, [C], [C]]]]$
$11, 2, [[C, [C, [C]], [C, [C], [C], [C]], [C, [C], [C], [C]]],$
$[C, [C, [C], [C]], [C, [C], [C]], [C, [C], [C], [C]]]]$
$12, 1, [[C, [C, [C], [C]], [C, [C], [C], [C]], [C, [C], [C], [C]]]]$
$13, 1, [[C, [C, [C], [C], [C]], [C, [C], [C], [C]], [C, [C], [C], [C]]]]$

In this way, we have obtained the truncation of the bivariate generating function of alkyls with size marked by z and height by u.

```
> enum_BGF:=map(series,series(convert(map(proc(s,z,u)
  z^size(s)*u^height(s)
  end,map(op,[seq(list_all_st(h[max],i),i=1..s[max])]),z,
  u),`+`),z,infinity),u,infinity);
```

$$enum\_BGF := (u)\, z + (u^2)\, z^2 + (u^2 + u^3)\, z^3 + (u^2 + 2\, u^3)\, z^4 + (4\, u^3)\, z^5 + (4\, u^3)\, z^6 + (5\, u^3)\, z^7 + (4\, u^3)\, z^8 + (4\, u^3)\, z^9 + (3\, u^3)\, z^{10} + (2\, u^3)\, z^{11} + (u^3)\, z^{12} + (u^3)\, z^{13}$$

Generating functions

combstruct[gfeqns] returns a system of functional equations satisfied by the generating functions of related combinatorial structures. In the case of the alkyls with maximum height above, we get the following triangular system.

```
> gfeqns(op(2..3,specs_ltd_height(4)),z);
```

$$\Big[\mathrm{Carbon}(z) = z, Alkyl\_height_2(z) = \mathrm{Carbon}(z)\Big(1 + Alkyl\_height_1(z) + \frac{1}{2} Alkyl\_height_1(z^2) + \frac{1}{2} Alkyl\_height_1(z)^2 + \frac{1}{3} Alkyl\_height_1(z^3) + \frac{1}{2} Alkyl\_height_1(z)\, Alkyl\_height_1(z^2) + \frac{1}{6} Alkyl\_height_1(z)^3\Big), Alkyl\_height_3(z) = \mathrm{Carbon}(z)\Big(1 + Alkyl\_height_2(z) + \frac{1}{2} Alkyl\_height_2(z^2) + \frac{1}{2} Alkyl\_height_2(z)^2 + \frac{1}{3} Alkyl\_height_2(z^3) + \frac{1}{2} Alkyl\_height_2(z)\, Alkyl\_height_2(z^2)$$

$$+\frac{1}{6}Alkyl\_height_2(z)^3\Bigg), Alkyl\_height_1(z)=\mathrm{Carbon}(z), Alkyl\_height_4(z)=$$

$$\mathrm{Carbon}(z)\Bigg(1+Alkyl\_height_3(z)+\frac{1}{2}Alkyl\_height_3(z^2)+\frac{1}{2}Alkyl\_height_3(z)^2$$

$$+\frac{1}{3}Alkyl\_height_3(z^3)+\frac{1}{2}Alkyl\_height_3(z)\,Alkyl\_height_3(z^2)$$

$$+\frac{1}{6}Alkyl\_height_3(z)^3\Bigg)\Bigg]$$

```
> gfsol:=gfsolve(op(2..3,specs_ltd_height(4)),z);
```

$$gfsol := \{Alkyl\_height_3(z) = z+5z^7+4z^9+3z^{10}+z^{13}+z^2+2z^{11}+z^{12}+4z^6+4z^8+2z^3+3z^4+4z^5,$$
$$Alkyl\_height_4(z)=12z^6+31z^8+47z^9+137z^{12}+z+184z^{13}+70z^{10}+99z^{11}$$
$$+7z^5+20z^7+300z^{15}+498z^{18}+594z^{20}+453z^{26}+570z^{24}+369z^{16}+614z^{21}$$
$$+378z^{27}+181z^{30}+z^{39}+56z^{33}+12z^{36}+624z^{22}+239z^{14}+3z^{38}+6z^{37}$$
$$+37z^{34}+20z^{35}+128z^{31}+312z^{28}+238z^{29}+89z^{32}+601z^{23}+514z^{25}$$
$$+432z^{17}+551z^{19}+z^{40}+z^2+2z^3+4z^4, Alkyl\_height_1(z)=z, \mathrm{Carbon}(z)=z,$$
$$Alkyl\_height_2(z)=z+z^2+z^3+z^4\}$$

In particular, we have obtained a truncation of the bivariate generating function of all alkyls (i.e., with no constraint on height). In this series, u marks the height. It extends the previous truncation *enum_BGF*.

```
> BGF:=map(series,series(eval(subs(Alkyl_height[0]=0,gfso
  l,add(u^h*(Alkyl_height[h]-Alkyl_height[h-1])(z),h=1..4
  ))),z,infinity),u,infinity);
```

$$BGF := (u)z+(u^2)z^2+(u^2+u^3)z^3+(u^2+2u^3+u^4)z^4+(4u^3+3u^4)z^5+$$
$$(4u^3+8u^4)z^6+(5u^3+15u^4)z^7+(4u^3+27u^4)z^8+(4u^3+43u^4)z^9+$$
$$(3u^3+67u^4)z^{10}+(2u^3+97u^4)z^{11}+(u^3+136u^4)z^{12}+(u^3+183u^4)z^{13}+$$
$$(239u^4)z^{14}+(300u^4)z^{15}+(369u^4)z^{16}+(432u^4)z^{17}+(498u^4)z^{18}+(551u^4)$$
$$z^{19}+(594u^4)z^{20}+(614u^4)z^{21}+(624u^4)z^{22}+(601u^4)z^{23}+(570u^4)z^{24}+$$
$$(514u^4)z^{25}+(453u^4)z^{26}+(378u^4)z^{27}+(312u^4)z^{28}+(238u^4)z^{29}+(181u^4)$$
$$z^{30}+(128u^4)z^{31}+(89u^4)z^{32}+(56u^4)z^{33}+(37u^4)z^{34}+(20u^4)z^{35}+(12u^4)$$
$$z^{36}+(6u^4)z^{37}+(3u^4)z^{38}+(u^4)z^{39}+(u^4)z^{40}$$

This is made explicit on the following normalized difference: each entry starts with a term in u^4, denoting alkyls with height at least 4.

```
> map(series,series(BGF-enum_BGF,z,infinity),u,infinity);
```

$$(u^4)z^4+(3u^4)z^5+(8u^4)z^6+(15u^4)z^7+(27u^4)z^8+(43u^4)z^9+(67u^4)z^{10}$$
$$+(97u^4)z^{11}+(136u^4)z^{12}+(183u^4)z^{13}+(239u^4)z^{14}+(300u^4)z^{15}+$$
$$(369u^4)z^{16}+(432u^4)z^{17}+(498u^4)z^{18}+(551u^4)z^{19}+(594u^4)z^{20}+(614u^4)$$

$z^{21} + (624\, u^4)\, z^{22} + (601\, u^4)\, z^{23} + (570\, u^4)\, z^{24} + (514\, u^4)\, z^{25} + (453\, u^4)\, z^{26} + (378\, u^4)\, z^{27} + (312\, u^4)\, z^{28} + (238\, u^4)\, z^{29} + (181\, u^4)\, z^{30} + (128\, u^4)\, z^{31} + (89\, u^4)\, z^{32} + (56\, u^4)\, z^{33} + (37\, u^4)\, z^{34} + (20\, u^4)\, z^{35} + (12\, u^4)\, z^{36} + (6\, u^4)\, z^{37} + (3\, u^4)\, z^{38} + (u^4)\, z^{39} + (u^4)\, z^{40}$

Table of the number of alkyls according to size and height

Calculations with respect to different heights are much more efficient than the method of exhaustive enumeration. This makes it possible for us to set up the table of the number of alkyls according to size and height in a few minutes:

```
> h[max]:=5;
```

$$h_{max} := 5$$

```
> gfsol:=gfsolve(op(2..3,specs_ltd_height(h[max])),z);
```

$$gfsol := \{ Alkyl\_height_5(z) = 16\, z^6 + 63\, z^8 + 121\, z^9 + 749\, z^{12} + z + 1344\, z^{13} + 225\, z^{10} + 415\, z^{11} + 8\, z^5 + 33\, z^7 + 39922778\, z^{93} + 552046535\, z^{84} + 261713408\, z^{87} + 12677964\, z^{96} + 4129\, z^{15} + 20354\, z^{18} + 55706\, z^{20} + 872727\, z^{26} + 364555\, z^{24} + 7106\, z^{16} + 90628\, z^{21} + 1328545\, z^{27} + 4393287\, z^{30} + 89755449\, z^{39} + 13204526\, z^{33} + 36095102\, z^{36} + 145729\, z^{22} + 2365\, z^{14} + 66951451\, z^{38} + 49418998\, z^{37} + 18657905\, z^{34} + 26088244\, z^{35} + 6407683\, z^{31} + 2000536\, z^{28} + 2980554\, z^{29} + 9246830\, z^{32} + 231801\, z^{23} + 567206\, z^{25} + 12104\, z^{17} + 33883\, z^{19} + 119063149\, z^{40} + z^2 + 2\, z^3 + z^{121} + 3994067586\, z^{69} + 2532213546\, z^{75} + 1323523938\, z^{51} + 2848892771\, z^{57} + z^{120} + 3993437445\, z^{62} + 2576241555\, z^{56} + 3116054839\, z^{58} + 4227813312\, z^{64} + 519559381\, z^{46} + 1121537006\, z^{50} + 1235964517\, z^{80} + 4132169661\, z^{63} + 1030602778\, z^{81} + 109250394\, z^{90} + 27\, z^{117} + 3469135\, z^{99} + 25584\, z^{108} + 338\, z^{114} + 3353\, z^{111} + 808547\, z^{102} + 417359802\, z^{45} + 158476\, z^{105} + 1545167948\, z^{52} + 779843029\, z^{48} + 202984042\, z^{42} + 2038928979\, z^{54} + 3606730433\, z^{60} + 1710204982\, z^{78} + 4277663720\, z^{66} + 3356383912\, z^{72} + 331715843\, z^{44} + 2249523074\, z^{76} + 2815570112\, z^{74} + 4134086103\, z^{68} + 3813878148\, z^{70} + 4\, z^4 + 848181768\, z^{82} + 688882553\, z^{83} + 436439448\, z^{85} + 340324807\, z^{86} + 198431393\, z^{88} + 148315264\, z^{89} + 79298004\, z^{91} + 56695196\, z^{92} + 27675367\, z^{94} + 18885015\, z^{95} + 8372800\, z^{97} + 5435626\, z^{98} + 2174395\, z^{100} + 1338790\, z^{101} + 479339\, z^{103} + 278280\, z^{104} + 88204\, z^{106} + 48126\, z^{107} + 13348\, z^{109} + 6744\, z^{110} + 1604\, z^{112} + 758\, z^{113} + 154\, z^{115} + 62\, z^{116} + 10\, z^{118} + 4\, z^{119} + 3371001341\, z^{59} + 3816383212\, z^{61} + 4276971739\, z^{65} + 4229607116\, z^{67} + 3599155257\, z^{71} + 3092740855\, z^{73} + 1973755292\, z^{77} + 1463215476\, z^{79} + 156284730\, z^{41} + 260865858\, z^{43} + 639939517\, z^{47} + 940236752\, z^{49} + 1784589063\, z^{53} + 2304400735\, z^{55},$$

$Alkyl\_height_3(z) =$

$z+5z^7+4z^9+3z^{10}+z^{13}+z^2+2z^{11}+z^{12}+4z^6+4z^8+2z^3+3z^4+4z^5,$

$Alkyl\_height_4(z) = 12z^6+31z^8+47z^9+137z^{12}+z+184z^{13}+70z^{10}+99z^{11}$
$+7z^5+20z^7+300z^{15}+498z^{18}+594z^{20}+453z^{26}+570z^{24}+369z^{16}+614z^{21}$
$+378z^{27}+181z^{30}+z^{39}+56z^{33}+12z^{36}+624z^{22}+239z^{14}+3z^{38}+6z^{37}$
$+37z^{34}+20z^{35}+128z^{31}+312z^{28}+238z^{29}+89z^{32}+601z^{23}+514z^{25}$
$+432z^{17}+551z^{19}+z^{40}+z^2+2z^3+4z^4$, $Alkyl\_height_1(z)=z$, $\mathrm{Carbon}(z)=z$,

$Alkyl\_height_2(z)=z+z^2+z^3+z^4\}$

```
> BGF:=map(series,series(eval(subs(Alkyl_height[0]=0,gfso
  l,add(u^hh*(Alkyl_height[hh]-Alkyl_height[hh-1])(z),hh=
  1..h[max]))),z,infinity),u,infinity);
```

$$BGF := (u)z+(u^2)z^2+(u^2+u^3)z^3+(u^2+2u^3+u^4)z^4+(4u^3+3u^4+u^5)z^5+(4u^3+8u^4+4u^5)z^6+(5u^3+15u^4+13u^5)z^7+(4u^3+27u^4+32u^5)z^8+(4u^3+43u^4+74u^5)z^9+(3u^3+67u^4+155u^5)z^{10}+(2u^3+97u^4+316u^5)z^{11}+(u^3+136u^4+612u^5)z^{12}+(u^3+183u^4+1160u^5)z^{13}+(239u^4+2126u^5)z^{14}+(300u^4+3829u^5)z^{15}+(369u^4+6737u^5)z^{16}+(432u^4+11672u^5)z^{17}+(498u^4+19856u^5)z^{18}+(551u^4+33332u^5)z^{19}+(594u^4+55112u^5)z^{20}+(614u^4+90014u^5)z^{21}+(624u^4+145105u^5)z^{22}+(601u^4+231200u^5)z^{23}+(570u^4+363985u^5)z^{24}+(514u^4+566692u^5)z^{25}+(453u^4+872274u^5)z^{26}+(378u^4+1328167u^5)z^{27}+(312u^4+2000224u^5)z^{28}+(238u^4+2980316u^5)z^{29}+(181u^4+4393106u^5)z^{30}+(128u^4+6407555u^5)z^{31}+(89u^4+9246741u^5)z^{32}+(56u^4+13204470u^5)z^{33}+(37u^4+18657868u^5)z^{34}+(20u^4+26088224u^5)z^{35}+(12u^4+36095090u^5)z^{36}+(6u^4+49418992u^5)z^{37}+(3u^4+66951448u^5)z^{38}+(u^4+89755448u^5)z^{39}+(u^4+119063148u^5)z^{40}+(156284730u^5)z^{41}+(202984042u^5)z^{42}+(260865858u^5)z^{43}+(331715843u^5)z^{44}+(417359802u^5)z^{45}+(519559381u^5)z^{46}+(639939517u^5)z^{47}+(779843029u^5)z^{48}+(940236752u^5)z^{49}+(1121537006u^5)z^{50}+(1323523938u^5)z^{51}+(1545167948u^5)z^{52}+(1784589063u^5)z^{53}+(2038928979u^5)z^{54}+(2304400735u^5)z^{55}+(2576241555u^5)z^{56}+(2848892771u^5)z^{57}+(3116054839u^5)z^{58}+(3371001341u^5)z^{59}+(3606730433u^5)z^{60}+(3816383212u^5)z^{61}+(3993437445u^5)z^{62}+(4132169661u^5)z^{63}+(4227813312u^5)z^{64}+(4276971739u^5)z^{65}+(4277663720u^5)z^{66}+(4229607116u^5)z^{67}+(4134086103u^5)z^{68}+(3994067586u^5)z^{69}+(3813878148u^5)z^{70}+(3599155257u^5)z^{71}+$$

$(3356383912\, u^5)\, z^{72} + (3092740855\, u^5)\, z^{73} + (2815570112\, u^5)\, z^{74} + (2532213546\, u^5)\, z^{75} + (2249523074\, u^5)\, z^{76} + (1973755292\, u^5)\, z^{77} + (1710204982\, u^5)\, z^{78} + (1463215476\, u^5)\, z^{79} + (1235964517\, u^5)\, z^{80} + (1030602778\, u^5)\, z^{81} + (848181768\, u^5)\, z^{82} + (688882553\, u^5)\, z^{83} + (552046535\, u^5)\, z^{84} + (436439448\, u^5)\, z^{85} + (340324807\, u^5)\, z^{86} + (261713408\, u^5)\, z^{87} + (198431393\, u^5)\, z^{88} + (148315264\, u^5)\, z^{89} + (109250394\, u^5)\, z^{90} + (79298004\, u^5)\, z^{91} + (56695196\, u^5)\, z^{92} + (39922778\, u^5)\, z^{93} + (27675367\, u^5)\, z^{94} + (18885015\, u^5)\, z^{95} + (12677964\, u^5)\, z^{96} + (8372800\, u^5)\, z^{97} + (5435626\, u^5)\, z^{98} + (3469135\, u^5)\, z^{99} + (2174395\, u^5)\, z^{100} + (1338790\, u^5)\, z^{101} + (808547\, u^5)\, z^{102} + (479339\, u^5)\, z^{103} + (278280\, u^5)\, z^{104} + (158476\, u^5)\, z^{105} + (88204\, u^5)\, z^{106} + (48126\, u^5)\, z^{107} + (25584\, u^5)\, z^{108} + (13348\, u^5)\, z^{109} + (6744\, u^5)\, z^{110} + (3353\, u^5)\, z^{111} + (1604\, u^5)\, z^{112} + (758\, u^5)\, z^{113} + (338\, u^5)\, z^{114} + (154\, u^5)\, z^{115} + (62\, u^5)\, z^{116} + (27\, u^5)\, z^{117} + (10\, u^5)\, z^{118} + (4\, u^5)\, z^{119} + (u^5)\, z^{120} + (u^5)\, z^{121}$

In the following table, the entry at row r and column c is the number of alkyls of size r and height c:

```
> matrix([[` `,seq(`height =
  `.hh,hh=1..h[max])],seq([`size =
  `.ss,seq(coeff(coeff(BGF,z,ss),u,hh),hh=1..h[max])],ss=
  1..s[max])]);
```

	height = 1	*height = 2*	*height = 3*	*height = 4*	*height = 5*
size = 1	1	0	0	0	0
size = 2	0	1	0	0	0
size = 3	0	1	1	0	0
size = 4	0	1	2	1	0
size = 5	0	0	4	3	1
size = 6	0	0	4	8	4
size = 7	0	0	5	15	13
size = 8	0	0	4	27	32
size = 9	0	0	4	43	74
size = 10	0	0	3	67	155
size = 11	0	0	2	97	316
size = 12	0	0	1	136	612
size = 13	0	0	1	183	1160
size = 14	0	0	0	239	2126
size = 15	0	0	0	300	3829
size = 16	0	0	0	369	6737
size = 17	0	0	0	432	11672
size = 18	0	0	0	498	19856
size = 19	0	0	0	551	33332
size = 20	0	0	0	594	55112
size = 21	0	0	0	614	90014
size = 22	0	0	0	624	145105
size = 23	0	0	0	601	231200
size = 24	0	0	0	570	363985
size = 25	0	0	0	514	566692
size = 26	0	0	0	453	872274
size = 27	0	0	0	378	1328167
size = 28	0	0	0	312	2000224
size = 29	0	0	0	238	2980316
size = 30	0	0	0	181	4393106
size = 31	0	0	0	128	6407555
size = 32	0	0	0	89	9246741
size = 33	0	0	0	56	13204470
size = 34	0	0	0	37	18657868
size = 35	0	0	0	20	26088224
size = 36	0	0	0	12	36095090
size = 37	0	0	0	6	49418992
size = 38	0	0	0	3	66951448
size = 39	0	0	0	1	89755448
size = 40	0	0	0	1	119063148
size = 41	0	0	0	0	156284730
size = 42	0	0	0	0	202984042
size = 43	0	0	0	0	260865858
size = 44	0	0	0	0	331715843

A (huge) table for $h_{max} = 7$ could be computed in less than 10 minutes.

Disubstituted alkanes, $C_n H_{2n} X Y$

Enumerating disubstituted alkanes $C_n H_{2n} X Y$ is equivalent to enumerating monosubstituted alkyls $C_n H_{2n} X$. The latter can generically be viewed as a carbon atom linked to one monosubstituted alkyl and at least 2 nonsubstituted alkyls. This yields the class equation $S1\_Alkyl_X = Carbon\ S1\_Alkyl_X\,(\mathrm{E} + Alkyl) + Carbon\ X\,(\mathrm{E} + Alkyl + Alkyl^2)$.

```
> gramm_S1_Alkyl:=S1_Alkyl[X]=Union(Prod(Carbon,S1_Alkyl[X],Set
  (Alkyl,card<=2)),Prod(Prod(Carbon,X),Set(Alkyl,card<=2))),X=E
  psilon:
> specs_S1_Alkyl:=[S1_Alkyl[X],{gramm_S1_Alkyl,gramm_Alkyl},unl
  abelled]:
> seq(count(specs_S1_Alkyl,size=i),i=0..50);
```

0, 1, 2, 5, 12, 31, 80, 210, 555, 1479, 3959, 10652, 28760, 77910, 211624, 576221, 1572210, 4297733, 11767328, 32266801, 88594626, 243544919, 670228623, 1846283937, 5090605118, 14047668068, 38794922293, 107215238057, 296501478704, 820476261295, 2271726458263, 6293333029156, 17443168163416, 48370062636654, 134190690985978, 372435833881578, 1034078866908394, 2872232726571749, 7980695109514561, 22182422656423849, 61676117449283837, 171537091915110029, 477227744594009504, 1328048856698095447, 3696729316849207130, 10292748327630264925, 28664895623718825161, 79849131533514081701, 222477780725979665937, 620005805241494744835, 1728199288005906578667

This series appears as the entry **M1418** ("paraffins with *n* carbon atoms") in the book by N. J. A. Sloane and S. Plouffe [*The Encyclopedia of Integer Sequences*, (1995), Academic Press].

Of course, there are more monosubstituted alkyls than nonsubstituted ones. We give the ratios number of monosubstituted alkyls/number of alkyls for small sizes:

```
> seq([i=evalf(count(specs_S1_Alkyl,size=i)/count(specs_Alkyl,s
  ize=i))],i=1..50);
```

[1 = 1.], [2 = 2.], [3 = 2.500000000], [4 = 3.], [5 = 3.875000000], [6 = 4.705882353], [7 = 5.384615385], [8 = 6.235955056], [9 = 7.009478673], [10 = 7.808678501], [11 = 8.604200323], [12 = 9.407916258], [13 = 10.19897892], [14 = 10.99859675], [15 = 11.79210069], [16 = 12.58714553], [17 = 13.38032304], [18 = 14.17376379], [19 = 14.96597929], [20 = 15.75825478], [21 = 16.54987862], [22 = 17.34135391], [23 = 18.13247884], [24 = 18.92344500], [25 = 19.71418351], [26 = 20.50478793], [27 = 21.29523810], [28 = 22.08557742], [29 = 22.87580605], [30 = 23.66594591], [31 = 24.45600192], [32 = 25.24598688], [33 = 26.03590630], [34 = 26.82576826], [35 = 27.61557771], [36 = 28.40534018], [37 = 29.19505975], [38 = 29.98474048], [39 = 30.77438565], [40 = 31.56399832], [41 = 32.35358114], [42 = 33.14313651], [43 = 33.93266655], [44 = 34.72217319], [45 = 35.51165815], [46 = 36.30112299], [47 = 37.09056911], [48 = 37.87999780], [49 = 38.66941024], [50 = 39.45880747]

Here is an example of a monosubstituted alkyl with 6 carbon atoms, obtained by the command combstruct[draw].

```
> alk:=draw(specs_S1_Alkyl,size=6);
```

$alk := \mathrm{Prod}(Carbon, \mathrm{Prod}(Carbon,$
$\mathrm{Prod}(\mathrm{Prod}(Carbon, X), \mathrm{Set}(\mathrm{Prod}(Carbon, \mathrm{Set}(\mathrm{Prod}(Carbon, \mathrm{E}))))),$
$\mathrm{Set}(\mathrm{Prod}(Carbon, \mathrm{E}))), \mathrm{E})$

The following procedure rewrites a monosubstituted alkyl into a more readable way.

```
> nice:=proc(alk)
  subs([C,X]=CX,eval(subs({Epsilon=NULL,Carbon=C,Prod=proc()
  [args] end,Set=proc() args end},alk))) end:
> nice(alk);
```

$$[C, [C, [CX, [C, [C]]], [C]]]$$

The following procedures compute the size and height of a given monosubstituted alkyl.

```
> size:=proc(alk) option remember;
  1+convert(map(op,map2(map,size,[op(2..-1,alk)])),`+`) end:
  size(Carbon):=1:
  size(X):=0:
  size(Epsilon):=0:
> height:=proc(alk) option remember;
  `if`(nops(alk)=2,1+max(op(map(height,op(2,alk)))),1+max(heigh
  t(op(2,alk)),op(map(height,op(3,alk))))) end:
  height(Carbon):=1:
  height(X):=0:
  height(Epsilon):=0:
```

Here is a monosubstituted alkyl with 50 carbon atoms, its nice representation and height.

```
> alk:=draw(specs_S1_Alkyl,size=50):
> nice(alk);
```

$[C, [C,$
$[C, [C, [C, [CX], [C], [C, [C], [C, [C, [C, [C, [C]]], [C, [C, [C]], [C, [C]]]]]]]],$
$[C]], [C, [C], [C, [C], [C], [C, [C]]]], [C, [C, [C], [C, [C], [C], [C, [C]]],$
$[C, [C, [C, [C, [C]]]], [C, [C, [C, [C]], [C, [C], [C]], [C, [C], [C]]]]]]]]$

```
> height(alk);
```

$$11$$

Trisubstituted alkanes, $C_n\, H_{2\,n-1}\, X\, Y\, Z$

Enumerating trisubstituted alkanes $C_n\, H_{2\,n-1}\, X\, Y\, Z$ is equivalent to enumerating disubstituted alkyls $C_n\, H_{2\,n-1}\, X\, Y$. In this section, we assume X, Y and Z to be distinct. The grammar is more involved than in the disubstituted case: we have to distinguish several cases, according to which of X and Y go into subtrees, and into which subtrees. The corresponding class equation is $S2\_Alkyl_{X,\,Y} = Carbon\; S2\_Alkyl_{X,\,Y}\,(\mathrm{E} + Alkyl + Alkyl^2)$
$+ Carbon\,(X + S1\_Alkyl_X)\,(Y + S1\_Alkyl_Y)\,(\mathrm{E} + Alkyl)$.

```
> gramm_S2_Alkyl:=S2_Alkyl[X,Y]=Union(Prod(Carbon,S2_Alkyl[X,Y]
  ,Set(Alkyl,card<=2)),Prod(Carbon,Union(S1_Alkyl[X],X),Union(S
  1_Alkyl[Y],Y),Set(Alkyl,card<=1))):
```

```
> specs_S2_Alkyl:=[S2_Alkyl[X,Y],{gramm_S2_Alkyl,gramm_S1_Alkyl
,op(subs(X=Y,[gramm_S1_Alkyl])),gramm_Alkyl},unlabelled]:
> seq(count(specs_S2_Alkyl,size=i),i=0..50);
```

0, 1, 4, 13, 42, 131, 402, 1218, 3657, 10899, 32298, 95257, 279844, 819390, 2392392, 6967956, 20250974, 58744089, 170118980, 491913999, 1420493862, 4096940530, 11803172152, 33970257473, 97678027311, 280624328431, 805587723862, 2310919999992, 6624670101196, 18978908257258, 54340562045429, 155503251237194, 444766664162993, 1271498880014923, 3633315536811959, 10377791606909654, 29630012602096393, 84565516906270186, 241267111088603729, 688104455854536297, 1961866816555420391, 5591785495039589114, 15933226976278838204, 45387390825197706053, 129255934882453809489, 368005885500527402163, 1047497232155611335438, 2980913638772831767811, 8481029051770110331198, 24124279361703485318308, 68607333785448471672444

This series extends the entry **M3466** ("paraffins with *n* carbon atoms") in the book by N. J. A. Sloane and S. Plouffe [*The Encyclopedia of Integer Sequences*, (1995), Academic Press].

Here is an example of a disubstituted alkyl with 6 carbon atoms, obtained by the command combstruct[draw].

```
> alk:=draw(specs_S2_Alkyl,size=6);
```

alk := Prod($Carbon$, Prod($Carbon$, X, Prod($Carbon$, Prod(Prod($Carbon$, Y), Set(Prod($Carbon$, E))), Set(Prod($Carbon$, E))), E , E)

The following procedure rewrites a disubstituted alkyl into a more readable way.

```
> nice:=proc(alk)
  subs({[C,X]=CX,[C,Y]=CY,[C,X,Y]=CXY},eval(subs({Epsilon=NULL,
  Carbon=C,Prod=proc() [args] end,Set=proc() args end},alk)))
  end:
> nice(alk);
```

$$[C, [C, X, [C, [CY, [C]], [C]]]]$$

The following procedures compute the size and height of a given disubstituted alkyl.

```
> size:=proc(alk) option remember;
  `if`(nops(alk)=2,1+convert(map(size,op(2,alk)),`+`),1+convert
  (map(size,[op(2..-2,alk)]),`+`)+convert(map(size,op(-1,alk)),
  `+`)) end:
  size(Carbon):=1:
  size(X):=0:
  size(Y):=0:
  size(Epsilon):=0:
> height:=proc(alk) option remember;
  `if`(nops(alk)=2,1+max(op(map(height,op(2,alk)))),1+max(op(ma
  p(height,[op(2..-2,alk)])),op(map(height,op(-1,alk))))) end:
  height(Carbon):=1:
  height(X):=0:
  height(Y):=0:
  height(Epsilon):=0:
```

Here is a disubstituted alkyl with 50 carbon atoms, its nice representation and height.

```
> alk:=draw(specs_S2_Alkyl,size=50):
> nice(alk);
```

$[C, [C, [C, [C, [C, [C, [C, [C, [C, [C, [C,$
$[C, [C, [CX], [C, [C, [C], [C], [C]]], [C, [C, [C, [C]]]]], [C, [C]], [C, [C], [C]]],$
$[C], [C, [C]]]]]], [C, [C], [C]]], [C, [C, [C]], [C, [C], [C]]]]]], [C], [C]]],$
$[C, [CY, [C, [C, [C, [C]]]]], [C, [C]]]]]$

```
> height(alk);
```

$$17$$

Trisubstituted alkanes, $C_n\,H_{2\,n-1}\,X_2\,Y$

Enumerating trisubstituted alkanes $C_n\,H_{2\,n-1}\,X_2\,Y$ is equivalent to enumerating disubstituted alkyls $C_n\,H_{2\,n-1}\,X_2$. In this section, we assume X and Y to be distinct. The class equation is

$$S2\_Alkyl_{X,X}$$
$$= Carbon\ S2\_Alkyl_{X,X}\,(\mathrm{E} + Alkyl + Alkyl^2)$$
$$+ Carbon\,(S1\_Alkyl_X^{\ 2} + S1\_Alkyl_X\,X + X^2)\,(\mathrm{E} + Alkyl).$$

```
> gramm_S2b_Alkyl:=S2_Alkyl[X,X]=Union(Prod(Carbon,S2_Alkyl[X,X
  ],Set(Alkyl,card<=2)),Prod(Carbon,Union(Prod(S1_Alkyl[X],S1_A
  lkyl[X]),Prod(S1_Alkyl[X],X),Prod(X,X)),Set(Alkyl,card<=1))):
> specs_S2b_Alkyl:=[S2_Alkyl[X,X],{gramm_S2b_Alkyl,gramm_S1_Alk
  yl,gramm_Alkyl},unlabelled]:
> seq(count(specs_S2b_Alkyl,size=i),i=0..50);
```

0, 1, 3, 9, 29, 91, 282, 864, 2624, 7905, 23665, 70458, 208813, 616400, 1813354, 5318673, 15559072, 45410239, 132258752, 384493609, 1115907585, 3233782193, 9358293913, 27048189854, 78087437216, 225198736949, 648826102943, 1867669679538, 5371673739750, 15437695238654, 44334467825097, 127235245949015, 364921130790146, 1046003809731397, 2996578878003288, 8580064056382390, 24555042887500969, 70240586991176211, 200837035211971305, 574007829828207160, 1639908132470162451, 4683360193435350915, 13370263334951108655, 38156995934449004558, 108859702271798238723, 310474064975088473530, 885227243624407302887, 2523254862611498946312, 7190343827493770143147, 20484498978131112323407, 58343548311292541719012

This series does not appear in the book by N. J. A. Sloane and S. Plouffe [*The Encyclopedia of Integer Sequences*, (1995), Academic Press], nor do its first four differences. Comparing to the entry **M2838** ("tertiary alcohols with n carbon atoms") in this book, we check that there are always more compounds $C_n\,H_{2\,n-1}\,X_2\,Y$ than compounds $C_n\,H_{2\,n-1}\,X_3$ (for $X \neq Y$).

Here is an example of a disubstituted alkyl with 6 carbon atoms, obtained by the command combstruct[draw].

```
> alk:=draw(specs_S2b_Alkyl,size=6);
```

$alk := \mathrm{Prod}(Carbon, \mathrm{Prod}(\mathrm{Prod}(Carbon, \mathrm{Prod}(\mathrm{Prod}(Carbon, X), \mathrm{Set}(\mathrm{Prod}(Carbon, \mathrm{E}))),$

Set(Prod(*Carbon*, E), Prod(*Carbon*, E))), *X*), E)

The following procedure rewrites a disubstituted alkyl into a more readable way.

```
> nice:=proc(alk)
  subs({[C,X]=CX,[C,X,X]=CX[2]},eval(subs({Epsilon=NULL,Carbon=
  C,Prod=proc() [args] end,Set=proc() args end},alk))) end:
> nice(alk);
```

$$[C, [[C, [CX, [C]], [C], [C]], X]]$$

Here are the 9 disubstituted compounds $C_n H_{2n-1} X_2 Y$ for $n = 3$:

```
> map(nice,allstructs(specs_S2b_Alkyl,size=3));
```

$[[C, [[CX], X], [C]], [C, [C, [X, X]], [C]], [C, [C, [X, X], [C]]], [C, [[CX, [C]], X]],$
$[C, [[C, [CX]], X]], [C, [C, [C, [X, X]]]], [C, [C, [[CX], X]]], [C, [X, X], [C, [C]]],$
$[C, [[CX], [CX]]]]$

Conclusion: multiply substituted alkyls

In the previous sections, we have enumerated the substituted compounds $C_n H_{2n+1} X$, $C_n H_{2n} X Y$, $C_n H_{2n-1} X Y Z$ and $C_n H_{2n-1} X_2 Y$. We could in principle enumerate the class $S_{p_1, \ldots, p_t}$ of compounds obtained after substituting p_1 hydrogen atoms by $X^{(1)}$ atoms, p_2 hydrogen atoms by $X^{(2)}$ atoms, ..., p_t hydrogen atoms by $X^{(t)}$ atoms, and one hydrogen atom by Y (so as to plant the trees). Doing so would require to define the class $S_{q_1, \ldots, q_t}$ for each $q_1 \le p_1$, ..., $q_t \le p_t$, and for each $q_1 \le p_1$, ..., $q_t \le p_t$, to write a recursion involving partitions into 4 parts of the multiset $\{X^{(1)}$ (q_1 times), ..., $X^{(t)}$ (q_t times)$\}$. When the q_i's are given, those partitions can be computed by a call to combstruct[allstructs]. It follows that we would describe and generate the grammar for multiply substituted alkyls in terms of the grammar for partitions into 4 parts!