MOLGEN 5.0, A MOLECULAR STRUCTURE GENERATOR

RALF GUGISCH, ADALBERT KERBER, AXEL KOHNERT, REINHARD LAUE, MARKUS MERINGER, CHRISTOPH RÜCKER, ALFRED WASSERMANN

ralfg1@gmx.net, Munich; kerber@uni-bayreuth.de, University of Bayreuth; axel.kohnert@uni-bayreuth.de, University of Bayreuth; reinhard.laue@uni-bayreuth.de, University of Bayreuth; markus.meringer@dlr.de, German Aerospace Center (DLR); christoph.ruecker@uni-leuphana.de, Leuphana University of Lüneburg; alfred.wassermann@uni-bayreuth.de, University of Bayreuth

ABSTRACT: MOLGEN 5.x combines the efficiency of the molecular generator MOLGEN 3.5 and the flexibility of MOLGEN 4.x. To achieve this, the software was reimplemented based on a totally new concept. The most visible new features are fuzzy molecular formula input and explicit use of atom state patterns. We describe the version MOLGEN 5.0 of this new series.

Keywords: MOLGEN, structure generation, fuzzy molecular formula, atom state pattern, molecular graph, goodlist, badlist, backtracking, Diophantine equation, orderly generation, molecular libraries, connectivity isomers, constitutions, molecular structure elucidation, substructure restriction, aromaticity detection

Corresponding author: Adalbert Kerber
Telephone: 0049 921 68009
Fax: 0049 921 55 3385
Post address: Department of Mathematics, University of Bayreuth, D-95440 Bayreuth, Germany
Email: kerber@uni-bayreuth.de, adalbert-kerber@t-online.de

1 INTRODUCTION

The program system MOLGEN is devoted to generating all structures (connectivity isomers, constitutions) that correspond to a given molecular formula, with optional further restrictions, e.g. presence or absence of particular substructures.

MOLGEN arose from the idea to provide an efficient and portable tool for molecular structure elucidation in chemical industry, research, and education. Historically, up to version MOLGEN 3.5, the main intention was to generate structures as fast as possible. The result is one of the fastest generators for molecular structures. However, applications showed that generator efficiency is not the only important topic for molecular structure elucidation. Thus in the development of series MOLGEN 4.x [6, 10] the interface was organized in a much more flexible way. Now advanced restrictions can be passed to the generator that are obtained from spectroscopy. MOLGEN–MS and MOLGEN–QSPR [7, 8] are special versions that arose from these efforts. In generating huge libraries without advanced restrictions, the performance of MOLGEN 4.x is not comparable to that of MOLGEN 3.5 . Series MOLGEN 5.x is now intended to combine the advantages of both approaches, i.e. the efficiency of MOLGEN 3.5 and the flexibility of MOLGEN 4.x.

All MOLGEN versions provide the mathematical heart of a program system for structure elucidation, rendering all mathematically possible candidates that correspond to a given set of structural constraints. MOLGEN allows to compute the complete set of structures corresponding to a given molecular formula or a set of molecular formulas. Often the molecular formula is sufficient as input, the generator will then use default values for the valences of all atoms included. Of course, it is possible to override defaults, by e.g. specifying particular atom valences.

The generation is *free of redundance*, i.e. no structure is generated twice within a single run. Moreover, the construction is *complete*, which means that the full set of all possible structures is obtained that correspond to a given molecular formula and, optionally, further restrictions. For example, given the input

$$C_8H_{16}O_2$$

each MOLGEN version will construct exactly 13,190 pairwise different

structures. This example already shows that, in general, the number of structures corresponding to a given molecular formula is very large. Therefore it is often desirable to reduce the output by imposing additional restrictions. For this purpose, together with a molecular formula, substructures may be specified that must be contained in each isomer constructed, or that on the contrary are not allowed. For example, if together with molecular formula $C_8H_{16}O_2$ a carboxyl group is prescribed, exactly 39 structures will be generated. If additionally the isopropyl group is excluded, then out of the 39 structures just 27 will remain.

Sometimes, compounds of interest are not described by a single molecular formula. For example, we may be interested in all chlorinated biphenyls, or even in all halogenated small alkanes with up to 4 carbon atoms. The present version MOLGEN 5.0 was developed to solve such problems. Solutions for these examples are presented in Section 3.

An important issue is, of course, how far MOLGEN 5.0 will reach. The only noteworthy limitations are those of time and hardware, i.e. due to an astronomical number of solutions, the program may not be able to generate the complete set of structures for a molecular formula within a reasonable time or to store all structures on the given harddisk.

MOLGEN 5.0 runs under Microsoft Windows (XP, Vista, 7, ...) and Linux operating systems. Generated structures are written in MDL SDfile (`.sdf`) or in the MOLGEN MB4 (`.mb4`) file format. Details on installation and hardware requirements can be found in the manual, to be obtained from

`http://www.molgen.de`

where the interested reader can also play with a restricted online version of MOLGEN 5.0 and can download further publications related to the MOLGEN series.

MOLGEN is unique in that it serves purposes different from those of other software packages, in particular from those of traditional combinatorial chemistry software. Both input to and output from MOLGEN differ from those of the latter software, a comparison with respect to performance, speed etc. is therefore impossible. From the mathematical point of view, MOLGEN's salient feature is its use of sophisticated algebraic methods, in particular of group theory, in order to avoid the combinatorial explosion as far as possible.

2 METHODS

In describing molecular structure we distinguish several levels of detail:

1. **Fuzzy molecular formula**
   Instead of prescribing exact occurrence numbers for each chemical element (or more exactly for each atom type, cf. Subsection 2.1.1), for broader coverage numerical intervals are allowed here. On the other hand, for each atom its state may be partially prescribed (valence, charge, hybridization, etc., see Subsection 2.1.1) in a fuzzy as in an exact molecular formula.

2. **(Exact) molecular formula**
   For each element symbol with optionally restricted state, its exact occurrence number is given.

3. **Atom state pattern**
   For each non–H atom in the molecular formula, its state is fully defined, including the numbers of bonds of various types and the number of hydrogens attached to it.

4. **Molecular graph**
   The connections between atoms are described as covalent bonds. In mathematical terms, a molecular structure can be understood as a graph, not only with single bonds, but possibly with double, triple or aromatic bonds.

The generation can be started from any of the levels, with a (set of) formula(s) provided by the user. Then, via backtracking, all corresponding molecular graphs are generated.

By choice of the user, the generation can be interrupted on any level, e.g. in order to manually select atom state patterns before generating molecular graphs.

2.1 Structures

2.1.1 Fuzzy and exact molecular formulas

A molecular formula such as $C_5H_{10}SO_2$ is entered as a string, e.g.

```
C5H10SO2.
```

The string contains the following information:

- **Atom types**, which are chemical element symbols,
- optional **atom states**, describing the environment of an atom within the molecular structure (e.g. its valence). For example, the formula above could be entered explicitly specifying the valence of S:

  `C5H10S[val=2]O2,`

- **atom occurences**, i.e. the number of atoms of given type and state occuring in a structure.

  For a fuzzy molecular formula, each atom occurrence number may be replaced by an interval of numbers, e.g. $C_5H_{10}SO_{0-2}$ could be specified by

  `C5H10S[val=2]O0-2.`

Note that an element symbol may occur more than once as input for a formula, i.e. in different atom states, e.g.

`C2H4N[val=3]0-1N[val=5]0-1.`

Exercise. The interested reader is invited to enter these formulas in MOLGEN–online via internet and the address

http://www.molgen.de/?src=documents/molgenonline

For example, enter `C5H10SO2`, click 'Submit' and after a few seconds you will see that this reduced version of MOLGEN 5.0 produced 4,560 structural formulas. Have a few of them displayed.

After that you may enter `C5H10S[val=2]O2` and find out that the same number of isomers is produced, and on inspection you will recognize that the default valence of sulfur used in MOLGEN 5.0 is 2.

Then you may submit `C5H10S[val=2]O0-2` or `C5H10SO0-2`, allowing 0, 1 or 2 oxygen atoms, in which case the online version produces 5,371 molecular graphs.

Atom types (element symbols): An element symbol is one or two letters. Usually an atom type is an element symbol from the Periodic Table of Elements. However, the user may define atom types not yet known to the system. Initially, MOLGEN does not know anything about a user–defined atom type, therefore one has to specify at least its valence as an atom state (see below). As an example, `C4H8Qs[val=2]3O` will produce structures of formula $C_4H_8Qs_3O$, where the user–defined atom type `Qs` has valence 2.

Atom states: Atom states describe the environment of an atom within the molecular structure. The following properties may be described:

- The valence of an atom in the structure. This is the total number of covalent bonds that connect the atom to its neighbors (including bonds to H; a double bond is counted twice, etc.). Default valences are according to the octet rule.
- The charge of an atom in the structure.
- Specification of an atom as a radical center.
- Isotope specification.
- Hybridization (sp3, sp2, sp), where sp2 is further distinguished for atoms in nonaromatic (sp2_n) and aromatic neighborhood (sp2_a), and sp is further distinguished for atoms bearing a single and a triple (sp_st) versus atoms bearing two double bonds (sp_dd).
- Number of H atoms adjacent to an atom.
- Number of single bonds (to non–H atoms) adjacent to an atom.
- Number of double bonds adjacent to an atom.
- Number of triple bonds adjacent to an atom.
- Number of aromatic bonds adjacent to an atom.

2.1.2 Atom state patterns

A state pattern describes a molecular structure by listing the fully defined state of each atom as described in Subsection 2.1.1, including the number of attached hydrogens.

Each atom is listed separately. For coding atom states the following symbols are used:

`Hn` the number of attached hydrogens,

`=n` the number of adjacent double bonds,

`#n` the number of adjacent triple bonds,

`~n` the number of adjacent aromatic bonds.

If *n*=0, the symbol `H`, `=`, `#`, or `~` is omitted; if *n*=1, the numeral 1 is omitted. This information together with an atom's valence defines the number of adjacent single bonds. For example,

```
CH#C#CH=CH=CH2CH
```

is the state pattern corresponding to 3-ethynylcyclobutene, where

- `CH#` codes a C atom bearing one H and a triple bond,
- `C#` is a C atom bearing a triple bond and a single bond to a non–H atom,
- `CH=` is a C atom bearing one H, one double bond and one single bond to a non–H atom,
- `CH2` is a C atom bearing two H and two single bonds to non–H atoms,
- `CH` is a C atom bearing one H and three single bonds to non–H atoms.

For a chemist reader, the notions of atom states and atom state patterns may be new. In earlier versions of MOLGEN they were used internally. In MOLGEN 5.0, they are open to manipulation by the user. This is an advantage in certain situations, providing the opportunity to better specify very large runs or to avoid generation of unwanted isomers stemming from unrequested atom state patterns.

Examples:

- `mgen -sp CH#C#CH=CH=CH2CH`

  generates two structures, 3-ethynylcyclobutene and 3-(2-propynyl)-cyclopropene, while

- `mgen -sp CH#C#C=CH=CH2CH2`

  leads to three structures, 1-ethynylcyclobutene, 1-(2-propynyl)cyclopropene, and (2-propyn-1-ylene)cyclopropane.

2.1.3 Molecular graphs

MOLGEN 5.0 is based on a graphical interaction model of a molecule. Graph nodes represent atoms, lines represent covalent bonds. Element symbol and atom state are stored as node labels, the kind of interaction (single, double, triple, aromatic bond) is stored as bond label.

We interpret bond labels as bond multiplicities. An atom's valence is the sum of its bond multiplicities. However, for an aromatic atom, its valence is composed of the number of single bonds and the number of aromatic bonds plus one. For example, in naphthalene, $C_{10}H_8$, each peripheral C atom bears one hydrogen and is involved in two aromatic bonds, while each of the two central atoms has no hydrogen and is involved in three aromatic bonds.

In a graphical representation, there is no explicit order of atoms specified. In order to handle structures without being restricted to a particular atom numbering, a massive use of group theory is necessary. Details can be found in [1, 5].

Aromaticity. MOLGEN 5.0 has a special bond type 'aromatic' for aromatic bonds. Consequently, cyclically conjugated double bonds forming an aromatic system are not generated. Rather, the corresponding structure is generated with the aromatic ring made of aromatic bonds.

Therefore MOLGEN has a built–in aromaticity detector plus filter that is based on the famous $4n+2$ π electrons rule (Hückel rule). In the current version cyclically conjugated rings of 6, 10, 14, etc. members are considered aromatic. In a future version, additional rings such as pyrrol, furan, thiophen, tropylium, cyclopentadienide etc. will be recognized as aromatic.

For example,

```
mgen C[sp2_n]10H8 -ringsize 6-10
```

results in 6 molecular graphs, none of which corresponds to naphthalene, whereas

```
mgen C[sp2_a]10H8
```

produces 4 structures, among them naphthalene and azulene. Atom states sp2_n and sp2_a therein denote sp2 atoms in nonaromatic or aromatic systems, respectively (see Subsection 2.1.1).

If desired, aromaticity handling may be deactivated. Then, benzene is generated with single and double bonds instead of aromatic bonds. Thus, 1,2-dimethylbenzene (o-xylene) will be generated twice, having either a single or a double bond connecting the substituted ring atoms.

2.2 Restrictions

For each level of generation, several restrictions may be formulated on the set of generated structures.

2.2.1 Restrictions on exact molecular formulas

The following restrictions may be imposed on molecular formulas to be generated from a fuzzy molecular formula. Each number may be restricted by a minimal and maximal allowed value:

- The total number of atoms in a molecular structure (including hydrogens).
- The sum of valences over all atoms. This is double the number of bonds (bonds to H included, double and triple bonds counted as two and three bonds, respectively; aromatic bonds counted as described above).
- The mass of the molecular structure, i.e. the sum over atom masses.
- Charge of the molecular structure, i.e. the sum over all atom charges.
- Sum over all isotopic mass differences.
- Total number of unpaired electrons in the molecular structure.
- Atom sums, i.e. sums of occurrence numbers of atom types/states.

The usage and strength of these restrictions is demonstrated by the following examples.

Examples:

- `mgen C2H0-6F0-6Cl0-6Br0-6I0-6 -atoms 8`

  generates ethane and all halogenated ethanes;

- `mgen C6H0-6Cl0-6 -sum H+Cl=6`

  generates all C_6H_6 hydrocarbons and their chlorinated analogs;

- `mgen C1-10H4-22 -mass 70-80`

  generates all hydrocarbons with a mass between 70 and 80;

- `mgen C1-10H4-22 -sum H-2C=2`

  generates all alkanes up to the decanes;

- `mgen C1-10H4-22 -sum H-2C=0-2`

  generates all alkanes plus monounsaturated alkenes plus saturated monocyclic hydrocarbons of up to ten carbon atoms.

- The atom sum restriction can be used to allow alternative atom states for an element. In the following example generation is restricted to structures containing at most two nitrogen atoms of valence 3 or 5:

  ```
  mgen C2H4N[val=3]0-2N[val=5]0-2 -sum N=0-2.
  ```

2.2.2 Restrictions on atom state patterns

The following restrictions influence the number and type of generated atom state patterns. Again each number may be restricted by a minimal and maximal allowed value:

- Maximal allowed bond multiplicity (i.e. 1, 2, or 3).
- Total number of single bonds (including bonds to hydrogens).
- Total number of double bonds.
- Total number of triple bonds.

- Total number of aromatic bonds.
- Number of bonds between atoms without counting bond multiplicity (including bonds to hydrogens).
- Number of cycles in the molecular structure. This is the number of bonds that have to be broken in order to obtain an acyclic structure, e.g. naphthalene has two, not three cycles, cubane has 5 cycles.
- Number of connected components of the molecular graph. By default connected graphs only are generated.

2.2.3 Restrictions on molecular graphs

In order to reduce the number of isomers generated, the following restriction is useful:

`-ringsize` n[-m] Specify the allowed ring sizes.

Any closed path in the molecular graph is considered a ring. For example, naphthalene contains rings of sizes 6 and 10, cubane has 4-, 6- and 8–membered rings. If a user allows 4–membered rings only, cubane will be missed.

Both power and limitations of the options described hitherto are easily seen in the following example, where we try to restrict the molecular formula $C_6H_5NO_2$ to nitrobenzene.

Example:

- `mgen C6H5NO2`

  results in 444,199 structures, nitrobenzene not among them;
- `mgen C6H5N[val=5]O2`

  gives 1,038,793 structures, among them nitrobenzene;
- `mgen C6H5N[val=5,d=2]O2`

  renders 122,699 structures;
- `mgen C6H5N[val=5,d=2,h=0]O2`

  results in 98,687 structures;

- `mgen C6H5N[val=5,d=2,h=0]O[d=1]2`

  results in 3,893 structures;

- `mgen C6H5N[val=5,d=2,h=0]O[d=1]2 -cycles 1`

  renders 1,436 structures;

- `mgen C6H5N[val=5,d=2,h=0]O[d=1]2 -ringsize 6-9`

  gives 452 structures;

- `mgen C6H5N[val=5,d=2,h=0]O[d=1]2 -cycles 1 -ringsize 6-9`

  results in 140 structures;

- `mgen C6H5N[val=5,d=2,h=0]O[d=1]2 -cycles 1 -ringsize 6`

  produces still 110 structures;

- `mgen C[sp2_n]6H5N[val=5,d=2,h=0]O[d=1]2 -cycles 1 -ringsize 6`

  results in 10 structures, nitrobenzene not among them;

- `mgen C[sp2_n]0-6C[sp2_a]0-6H5N[val=5,d=2,h=0]O[d=1]2 -cycles 1 -ringsize 6 -sum C=6`

  results in 11 structures;

- `mgen C[sp2_a]6H5N[val=5,d=2]O2`

  produces exactly one structure, nitrobenzene.

The example demonstrates the demand for more powerful restrictions, i.e. for substructure restrictions.

2.2.4 Substructure restrictions

You can specify substructures as restrictions to MOLGEN.

MOLGEN substructures support 'Any' atom type (element symbol A) and extended bond types like 'single or aromatic', 'double or aromatic', 'single or double', or 'any bond'. For creating and editing substructures,

any standard molecule editor supporting MOL files is suitable, for example Symyx Draw or ACD Chemsketch.

MOLGEN distinguishes 'open' and 'induced' substructures. In the **induced** case, if free valences on different atoms in a given substructure get connected to each other, this is considered a non–match. Thus, additional zero–length bridges within a substructure, or higher bond multiplicities, will cause a non–match. In the **open** case, however, such variations are recognized as a match. In mathematical terms, an induced substructure is an induced subgraph of the molecular graph, while an open substructure is a subgraph in general.

Consider for example a substructure `general_cyclohexane.mol` consisting of a 6–membered ring of A atoms ('Any' type), all bonds are single.

Using `general_cyclohexane.mol` as open substructure, e.g. cyclohexane, cyclohexene, cyclohexa-1,3-diene, cyclohexa-1,4-diene, benzene, benzyne, piperidine, pyridine, bicyclo[2.2.0]hexane substructures, etc., will be considered matches of the substructure.

Using `general_cyclohexane.mol` as induced substructure, e.g. cyclohexene, cyclohexa-1,3-diene, cyclohexa-1,4-diene, benzene, benzyne, pyridine, bicyclo[2.2.0]hexane substructures will be considered as non-matches. Piperidine and of course cyclohexane are recognized as matches.

Given a substructure, you can restrict its occurrence number in the generated molecular graphs to a specific range.

Examples:

- `mgen C8H11N -cycles 1-4 -ringsize 5-9`

  results in 11,586 compounds, among them being substituted pyridines, dihydro- and tetrahydropyridines, piperidines, benzenes, cyclohexadienes, cyclohexenes, and cyclohexanes;

- `mgen C8H11N -cycles 1-4 -ringsize 5-9`
  `-substr open 0 general_cyclohexane.mol`

  generates 6,290 compounds, none of which contains any 6–membered ring;

- `mgen C8H11N -cycles 1-4 -ringsize 5-9`
  `-substr induced 0 general_cyclohexane.mol`

leads to 10,857 compounds, among them pyridines, dihydro- and tetrahydropyridines, benzenes, cyclohexadienes and cyclohexenes, but no piperidines or cyclohexanes. So the piperidines and cyclohexanes filtered out amount to 729;

- `mgen C8H11N -cycles 1-4 -ringsize 5-9 -substr induced 1-4 general_cyclohexane.mol`

  produces exactly 729 substituted piperidines and cyclohexanes, and this set is identical to the set filtered out above.

Having another substructure `benzene.mol` consisting of a 6–membered ring of carbon atoms, all bonds specified as aromatic, we can use it to restrict our generation to structures having at least one benzene substructure.

However, using `benzene.mol` as induced substructure, dehydrobenzene (benzyne) or a zero–bridged benzene ring will not be considered a match, and consequently structures containing a benzyne but not a benzene will not be generated. Of course, structures containing both a benzene and a benzyne may occur.

Using `benzene.mol` as open substructure, benzyne or a zero–bridged benzene ring will be considered a match, and consequently structures containing a benzyne but no benzene substructure will be generated.

- `mgen C6H5N[val=5]O2 -substr induced 1 benzene.mol`

  results in 143 structures, each containing a benzene substructure, and nitrobenzene being among them;

- `mgen C6H5N[val=5]O2 -substr open 1 benzene.mol`

  results in 312 structures, many of which contain a (presumably undesired) zero–bridged benzene ring;

- `mgen C6H5N[val=5,h=0]O2 -substr induced 1 benzene.mol`

  renders 7 structures;

- `mgen C6H5N[val=5,d=2]O2 -substr induced 1 benzene.mol`

  generates nitrobenzene as the only structure;

- `mgen C6H5N[val=5]O2 -substr induced 1 nitro.mol`

  results in 685 structures, among them nitrobenzene;

- `mgen C6H5N[val=5]O2 -substr induced 1 nitro.mol -cycles 1`

  gives 197 structures;

- `mgen C6H5N[val=5]O2 -substr induced 1 nitro.mol -cycles 1 -ringsize 6`

  renders 14 structures;

- `mgen C6H5N[val=5]O2 -substr induced 1 nitro.mol -substr induced 1 benzene.mol`

  of course delivers nitrobenzene as the only structure.

Recall that for the examples to work appropriately it is important that the bonds in benzene.mol are of type 'aromatic' and that the nitrogen in nitro.mol has valence 5.

Two SDfiles of 'bad' open substructures are shipped together with MOLGEN, named `badlist.sdf` and `badlist2.sdf`. The former contains 39 highly strained saturated and unsaturated small mono-, bi-, and polycyclic structures that we consider 'not viable' (Fig. (1)). The latter is a collection of 14 'not viable' bridged aromatic structures, shown in Fig. (2). Though such lists are, of course, somewhat arbitrary, they are useful for removing obviously unwanted structures, as demonstrated in the following examples.

Examples:

- `mgen C6H6`

  generates all 217 mathematically possible benzene isomers;

- `mgen C6H6 -badlist badlist.sdf`

  results in no more than 66 isomers.

Though 151 isomers are removed thereby, the remaining set still contains those isomers that are known compounds either themselves or as more or

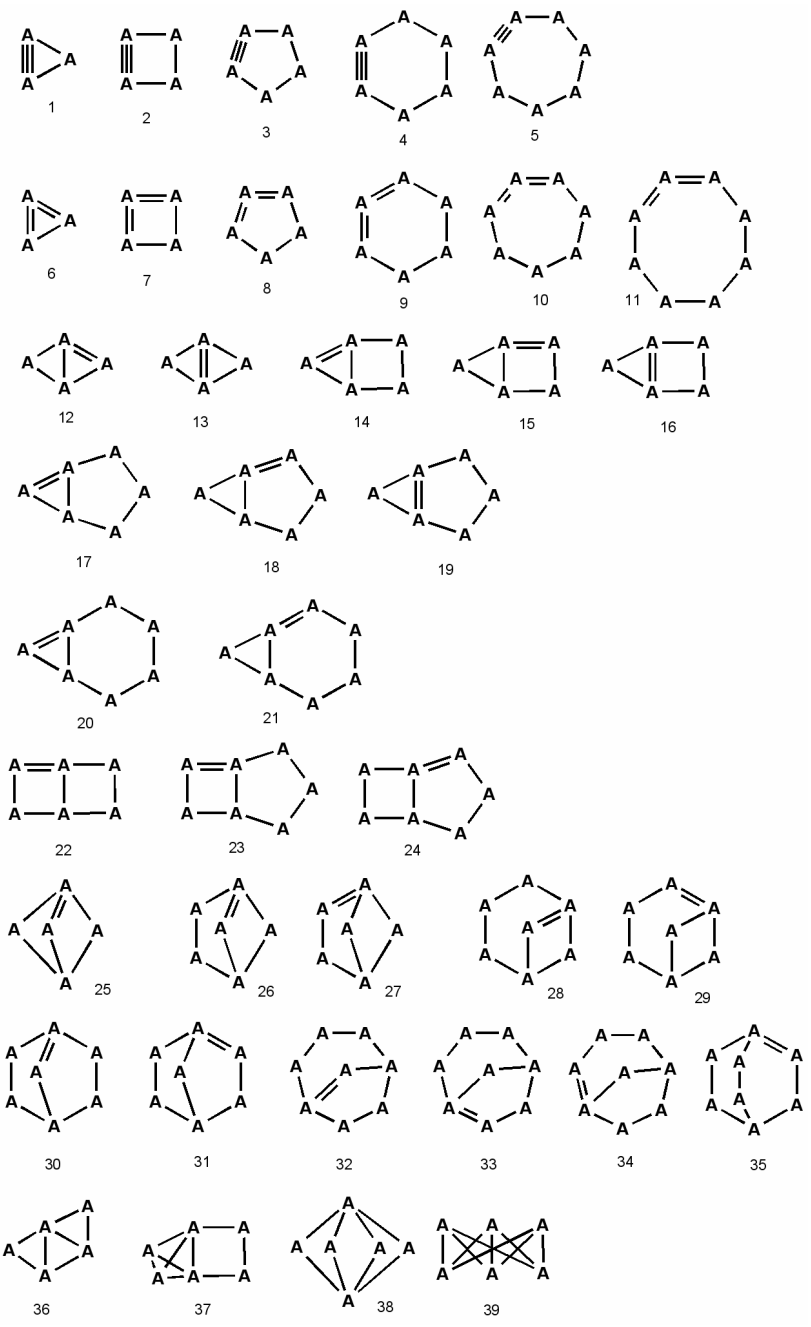

Fig. (1). 'Bad' cyclic and unsaturated substructures contained in `badlist.sdf`.

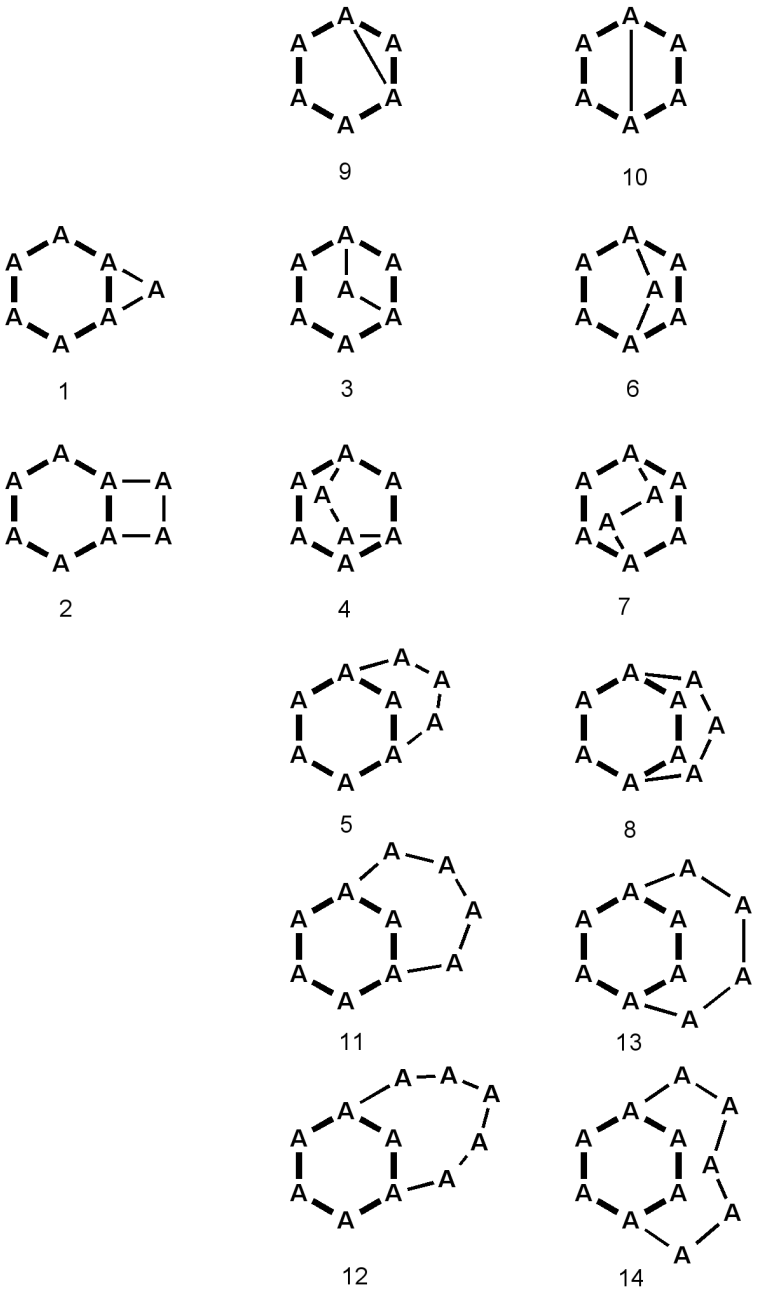

Fig. (2). 'Bad' bridged aromatic substructures contained in `badlist2.sdf`. Aromatic bonds are symbolized here by thick lines.

less substituted derivatives, such as prismane, Dewar benzene, benzvalene, fulvene, bi-cyclopropenyl, etc.

- `mgen C6H5N[val=5]O2 -substr open 1 benzene.mol`
  generates 312 structures (see above);
- `mgen C6H5N[val=5]O2 -substr open 1 benzene.mol -badlist badlist2.sdf`
  results in nitrobenzene as the only product.

Obviously, the user may edit these badlists or create one himself.

Required and forbidden substructures are used in other structure generators as well, see for example [12].

2.3 The backtracking algorithm

2.3.1 Restriction sharpening

Given, for example, a fuzzy molecular formula, a couple of restrictions are induced by simple logic. For example, the number of atoms may not get larger than the sum of maximal occurence numbers of each element symbol, and it may not get less than the sum of minimal occurence numbers. Or, if a substructure is prescribed to occur at least once, several minimal bounds are induced, e.g. on the number of single bonds, etc. in the molecule. Before starting the generation, such induced restrictions are automatically added to the set of restrictions.

Further, the restrictions are highly intercorrelated. For example, the following formula holds for any molecular graph.

$$\text{atoms} + \text{cycles} = \text{bonds} + \text{connected components}$$

Thus, if two of the three quantities number of atoms, of bonds, and of cycles are prescribed e.g. for a connected molecular graph, there is no choice for the third. If there are minimal and/or maximal bounds on the numbers, some of the other bounds may be sharpened by applying this formula.

A couple of graph–theoretic intercorrelations are checked by MOLGEN 5.0 at several stages during the generation in order to keep the restrictions as sharp as possible.

During each level of backtracking, a couple of new properties get fixed. For example, when an exact molecular graph was generated starting from a fuzzy molecular formula, the number of atoms gets fixed. Each time after some properties of the molecule get fixed, the graph–theoretic intercorellations are checked again in order to sharpen the remaining restrictions.

Whenever an inconsistency is recognized, for example if a lower bound gets larger than its corresponding upper bound, the current backtrack subtree is pruned.

2.3.2 From fuzzy formula to exact formulas

For a given fuzzy formula the generator runs through all corresponding exact formulas and the restrictions are tested.

Generating exact formulas implements the following mathematical problem: Generate all partitions of n, which is the maximum allowed nominal molecular mass, into $m+1$ blocks, where m equals the number of different atom types in the fuzzy formula. Blocks correspond to the atom types, weighted by the corresponding nominal atomic mass. An additional block is for technical purposes to allow generation of formulas not only for a fixed atom weight, but for a range of allowed atom weights.

Example: For the fuzzy formula $\mathsf{C}_{1-10}\mathsf{H}_{4-22}$ with molecular mass restricted to the range $70-80$, all number partitions of 80 into three blocks are generated. The first block with weight 12 defines the number of carbon atoms, the second block with weight 1 defines the number of H atoms, and the third block with weight 1 fills the gap between the actual molecular weight and the maximal weight 80.

The first block is restricted to appear 1 to 10 times in the partition, the second block is restricted to appear 4 to 22 times and the third block to appear 0 to 10 times (as the difference between maximal and minimal molecular weight is 10).

The implementation is straightforward, via backtracking. A couple of tests are executed before a molecular formula is written to the output or passed to the next level, they follow directly from graph theory and chemistry:

- The sum of valences must be even.

Let a denote the number of atoms including H atoms and b be half of the sum of valences, i.e. the sum of all bond multiplicities in any graph corresponding to the formula. Then

- b must be greater than or equal to the maximum valence occurring in the formula,
- $a - b \leq c_{max}$ must be fulfilled. c_{max} is the maximal allowed number of connected components (default is 1).

Further, all user–given restrictions on molecular formulas must be fulfilled:

- all restrictions on the number of atoms,
- all restrictions on the sum of valences,
- all restrictions on charge, isotopes, unpaired electrons, and
- all atom sum restrictions.

If all above tests are passed, the exact molecular formula is accepted and in turn used as input for the generation of state patterns.

2.3.3 From exact formula to atom state patterns

A system of linear equations is established, where the variables are restricted to nonnegative integer values. Usually, problems of this kind are hard to solve. However, MOLGEN contains its own algorithm called 'solvediophant' to solve these systems of equations. It is based on the mathematical concept of *lattice basis reduction* [17, 18].

Let a_i, t_i, and d_i be the numbers of aromatic, triple, and double bonds incident with non–H atom i, s_i its number of single bonds to non–H atoms, and h_i the number of H atoms attached to it. Then the number of bonds in the molecule is equal to half of

$$\sum_i \left(a_i + t_i + d_i + s_i + 2h_i\right).$$

The following restrictions are formulated as diophantine equations (all sums are over the non–H atoms):

- The numbers of aromatic, triple, double, single bonds fulfill the corresponding restrictions.
- The number of bonds, rings and connected components fulfill their restrictions.
- The sum $\sum_i \left(a_i + t_i + d_i + s_i + 2h_i\right)$ is even (as it is twice the number of bonds).
- The sums $\sum_i a_i$, $\sum_i t_i$, $\sum_i d_i$, $\sum_i s_i$ are all even (as they are twice the number of aromatic, triple, double or single bonds between non–H atoms).
- The sum $\sum_i h_i$ is equal to the number of hydrogens.
- If there are any aromatic atoms, then there are at least six aromatic atoms and six aromatic bonds. The number of aromatic atoms has to be even.[1]
- The following equation must be fulfilled:

  $$\text{atoms (incl. H)} + \text{cycles} = \text{bonds} + \text{connected components}.$$

- For each non–H atom, the sum of valences needs to be consistent with its valence v_i: Set $a_i^* = 0$ if and only if $a_i = 0$ and put $a_i^* = a_i + 1$ else. Then
  $$a_i^* + 3t_i + 2d_i + 1s_i + 1h_i = v_i.$$
- In particular cases there are further constraints to be fulfilled.
- A system of equations ensures that each state pattern is produced only once by the diophantic solver. We allow only such state patterns in which the list of atom states is sorted in lexicographically decreasing order.

2.3.4 From state pattern to molecular graphs

The construction of all molecular graphs corresponding to a state pattern is done mainly using the same techniques as in MOLGEN 3.5, by orderly generation [3, 13]. More details on how orderly generation is applied to molecular graphs can be found in [4] and were recently discussed in [11].

[1] Some details on the restrictions concerning aromaticity are omitted here.

mass	MF	MG	MGNAD	BS	MS
20	0	0	0	0	0
30	2	2	2	2	2
40	1	5	5	5	1
50	1	7	7	1	1
60	6	47	47	25	12
70	6	380	380	84	31
80	7	1,645	1,644	100	23
90	11	5,849	5,818	107	28
100	16	33,627	33,537	710	154

Table 1). Numbers of molecular and structural formulas for several molecular masses.

3 APPLICATIONS

3.1 Molecular libraries

An interesting problem where we can sometimes take advantage of a fuzzy molecular formula is the generation of molecular libraries. The use of MOLGEN 5.0 makes life easy when we want, for example, to get information on the total set of structural formulas of molecular mass 100, atoms in $\{\mathsf{C},\mathsf{H},\mathsf{N},\mathsf{O}\}$ and containing at least one carbon atom. Enter the fuzzy formula together with the mass constraint

```
mgen C1-8H0-16N0-6O0-4 -mass 100
```

to quickly obtain 33,537 structural formulas. In Table 1 you find numbers of structures that correspond to the various molecular formulas, for several molecular masses ≤ 100.

Column MF contains the number of molecular formulas corresponding to the mass and the fuzzy formula. MG means the numbers of corresponding molecular graphs, the structural formulas. The filter for aromatic duplicates was turned off when these entries were calculated, so that, for example, the total number of structures of mass 100 turned out to be 33,627. In the online version this filter is on, resulting in 33,537 structural formulas. Therefore we give in column MGNAD the numbers of structural formulas without aromatic duplicates. Column BS contains the number of structures that are contained in the Beilstein database, while column MS refers to the

NIST mass spectral library. The table is part of tables published in [9], and so these numbers found in the databases are snapshots, they may have changed in the meantime. Nevertheless they are of interest in order to show the enormous difference between the mathematically possible numbers of compounds and the numbers of existing compounds, and the number of existing compounds whose mass spectra were recorded and made publicly available.

Exercise. Refine this table by manually evaluating the molecular formulas corresponding to mass 100, and obtain the isomer numbers online. Look up these molecular formulas in a database such as SciFinder or Reaxys to find out how many corresponding compounds are contained therein. Comparing the numbers keep in mind that database compounds may include stereoisomers, isotopomers, radical ions and various other compound categories that are not included in MOLGEN counts.

3.2 Generate all chlorinated biphenyls

Often a search space cannot be defined by a single molecular formula, but by a range of several related molecular formulas (a fuzzy molecular formula). A typical example is the generation of congeners. In MOLGEN 5.0 the generation of all chlorinated biphenyls is solved as follows:

```
mgen C12H10 -bonds3 0 -bonds2 0 -bonds1 11
     -cycles 2 -ringsize 6
```

produces a single molecule, biphenyl, within about a second on a standard PC.

```
mgen C12H0-10Cl0-10 -sum H+Cl=10 -bonds3 0
     -bonds2 0 -bonds1 11 -cycles 2 -ringsize 6
```

results in 210 molecules within 3 sec, i.e. the non–chlorinated parent biphenyl and the fully chlorinated decachlorobiphenyl, 3 mono- and 3 nonachlorinated, 12 di- and 12 octachlorinated, 24 tri- and 24 heptachlorinated, 42 tetra- and 42 hexachlorinated, and 46 pentachlorinated biphenyls.

In this example, of course, alternatively eleven runs on a well–defined molecular formula each could be performed, e.g. in MOLGEN 3.5 . In the next example, however, such a semi–manual procedure would be a tedious exercise, to say the least.

3.3 Halogenated alkanes

Generate all halogenated (as well as nonhalogenated) alkanes C_1-C_4, where halogenated means bearing at least one F, Cl, Br, or I substituent.

```
mgen C1-4H0-10F0-10Cl0-10Br0-10I0-10
     -sum H+F+Cl+Br+I-2C=2
```

generates 187,075 compounds, i.e. the alkanes methane, ethane, propane, butane, isobutane, and all their halogen derivatives, corresponding to altogether 1,776 molecular formulas. This takes 35 sec on a standard PC.

3.4 Molecular structure elucidation

An important real case use of MOLGEN is molecular structure elucidation based on mass spectra. Molecular structure generation is crucial whenever the unknown chemical compound considered is not contained in the available databases. This kind of problem is carefully discussed in all detail in a PhD thesis [15], see also [16, 14]. The role of MOLGEN–MS is described and additional software that is useful in this context is mentioned. In particular, Section 6 contains examples of tentative identification of contaminants in groundwater of Bitterfeld, Germany. Mass spectra of 150 contaminants were obtained, of which 42 could be tentatively identified using the NIST database search alone. 32 of these compounds identified using NIST were confirmed using structure generation techniques. In addition, 20 further peaks were tentatively identified using structure generation techniques alone, resulting in a total of 62 tentative identifications. In another case, an unknown spectrum had the molecular formula $C_{13}H_{10}ClNO$ that has more than 10^9 connectivity isomers, but substructures derived from the spectrum and generation using MOLGEN–MS reduced this number to just 36 candidates. Literature search on diclofenac and additional confirmation analysis further reduced this set to a known diclofenac phototransformation product that was also identified as the one responsible for the enhanced toxicity of the transformed diclofenac towards the green algae *S. vacuolatus*.

For molecular structure elucidation based mainly on NMR spectra see [2] and later papers by these authors.

COMPETING INTEREST

All authors together are the MOLGEN team which distributes MOLGEN software at a nominal fee.

REFERENCES

[1] Braun, J.; Gugisch, R.; Kerber, A.; Laue, R.; Meringer, M.; Rücker, C. MOLGEN–CID, a canonizer for molecules and graphs accessible through the Internet. *J. Chem. Inf. Comput. Sci.*, **2004**, 44, 542–548.

[2] Elyasberg, M. E.; Blinov, K. A.; Martirosian, E. R. A new approach to computer-aided molecular structure elucidation: the expert system Structure Elucidator. *Lab. Autom. Inf. Manag.*, **1999**, 34, 15–30.

[3] Faradzhev, I. A. Generation of nonisomorphic graphs with a given degree sequence. *Algorithmic Studies in Combinatorics*, NAUKA, Moscow, **1978**, 11–19 (in Russian).

[4] Grund. R. Konstruktion molekularer Graphen mit gegebenen Hybridisierungen und überlappungsfreien Fragmenten. *Bayreuther Mathematische Schriften*, **1995**, 49, 1–113.

[5] Gugisch, R.; Kerber A.; Laue, R.; Meringer, M.; Rücker, C.; Schymanski, E. *Molecules in Silico, Applications to Computer Chemistry and Chemoinformatics* (in preparation).

[6] Kerber, A.; Laue, R.; Grüner, T.; Meringer, M. MOLGEN 4.0. *MATCH Commun. Math. Comput. Chem.*, **1998**, 37, 205–208.

[7] Kerber, A.; Laue, R.; Meringer, M.; Rücker, C. MOLGEN–QSPR, a software package for the search of quantitative structure property relationships. *MATCH Commun. Math. Comput. Chem.*, **2004**, 51, 187–204.

[8] Kerber, A.; Laue, R.; Meringer, M.; Varmuza, K. MOLGEN–MS: Evaluation of low resolution electron impact mass spectra with MS classification and exhaustive structure generation. *Adv. Mass Spectrom.*, **2001**, 15, 939–940.

[9] Kerber, A.; Laue, R.; Meringer, M.; Rücker, C. Molecules in silico: Potential versus known organic compounds. *MATCH Commun. Math. Comput. Chem.*, **2005**, 54, 301–312.

[10] Laue, R.; Grüner, T.; Meringer, M.; Kerber, A. Constrained generation of molecular graphs. In: *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol 69, American Mathematical Society, **2005**, 319–332.

[11] Meringer, M. Structure enumeration and sampling. In: *Handbook of Chemoinformatics Algorithms*. Eds. J.-L. Faulon and A. Bender. Chapman & Hall/CRC Mathematical & Computational Biology, **2010**, Chapter 8, 233–267.

[12] Molodtsov, S. G. The generation of molecular graphs with obligatory, forbidden and desirable fragments. *MATCH Commun. Math. Comput. Chem.*, **1998**, 37, 157–162.

[13] Read, R. C. Everyone a winner. *Ann. Discr. Math.*, **1978**, 2, 107–120.

[14] Schulze, T.; Weiss, S.; Schymanski, E.; von der Ohe, P. C.; Schmitt-Jansen, M.; Altenburger, R.; Streck, G.; Brack, W. Identification of a phytotoxic phototransformation product of diclofenac using effect-directed analysis. *Environmental Pollution*, **2010**, 158, 1461–1466.

[15] Schymanski, E. L. *Integrated analytical and computer tools for toxicant identification in effect–directed analysis*. PhD thesis 07/2011, Helmholtz Centre for Environmental Reseach – UFZ.

[16] Schymanski, E. L.; Meinert, C.; Meringer, M.; Brack, W. The use of MS classifiers and structure generation to assist in the identification of unknowns in effect–directed analysis. *Analytica Chimica Acta*, **2008**, 615 (2), 136–147 .

[17] Wassermann, A. Finding simple t-designs with enumeration techniques. *J. Comb. Designs*, **1998**, 6, 79–90.

[18] Wassermann. A. Attacking the market split problem with lattice point enumeration. *J. Comb. Optimization*, **2002**, 6, 5–16.