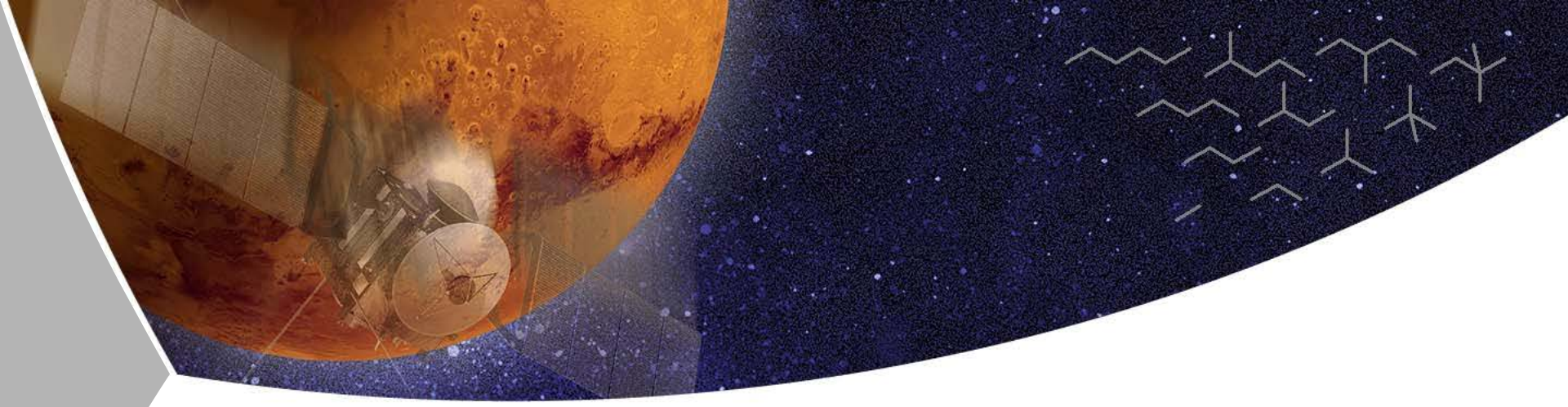


Markus Meringer

Generation of Molecular Graphs and Applications in Chemistry

Navigating Chemical Compound Space for Materials and Bio Design

Workshop II: Optimization, Search and Graph-Theoretical Algorithms for Chemical Compound Space

Institute for Pure and Applied Mathematics, University of California, Los Angeles

April 11 - 15, 2011

Deutsches Zentrum
für Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

Outline

- Introduction
  - What is molecular structure generation?
  - Why is it needed?

- Structure enumeration
  - Enumerating labeled graphs
  - Enumerating unlabeled graphs
  - Introducing constraints
  - From simple graphs to molecular graphs

- Results and Applications
  - Structure elucidation
  - (Inverse QSAR/QSPR)

Introduction: Representing Chemical Compounds

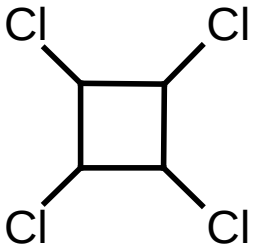
Different levels of abstraction

Composition

$C_4H_4Cl_4$

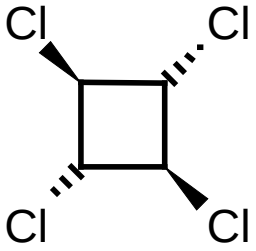
molecular formula

Constitution

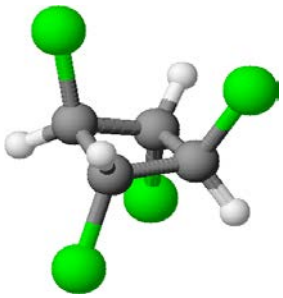


structural formula

Configuration



Conformation

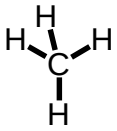


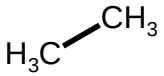
Specialization

Generalization

Introduction: Constitutional Isomers

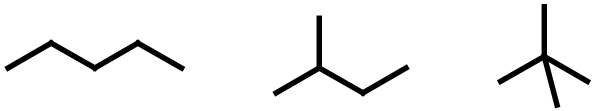
Example: Alkanes C_nH_{2n+2}

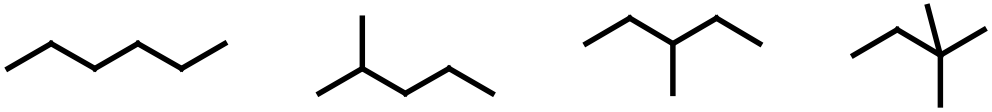
CH_4 

C_2H_6  or simply 

C_3H_8 

C_4H_{10} 

C_5H_{12} 

C_6H_{14} 

C_7H_{16} ... 9 *isomers* (try yourself – it‘s fun!)

Typically there are several, mostly very many structural formulas with the same molecular formula

Applications: Relating Structure and Properties

- From structure to physical, chemical, biological and pharmaceutical properties
  - structure-property relationships, esp. QSAR/QSPR
  - application of such relationships to predict properties of virtual structures (→ inverse QSAR)

- From physical and chemical properties (spectra) to structure

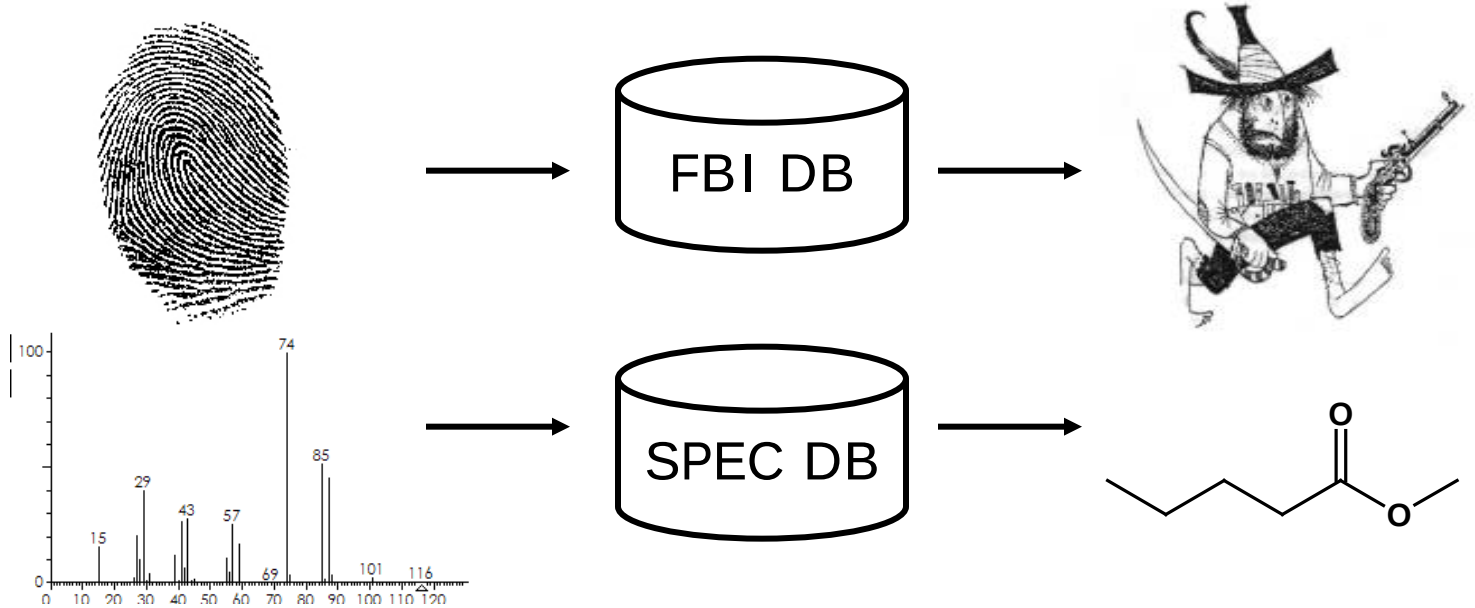
  computer-aided / automated

  molecular structure elucidation

  “CASE”

Structure Elucidation by Database Searching

- Established approach: use spectral data as molecular fingerprint for a database search

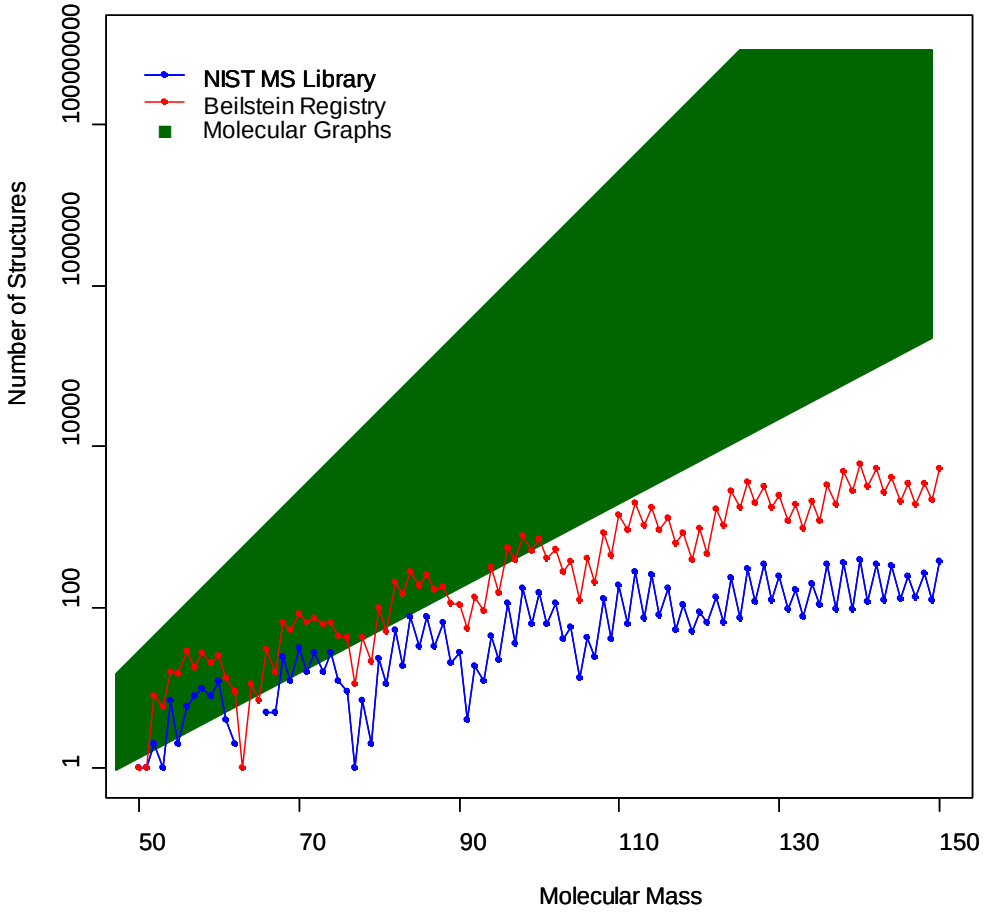


- Problem: only such data can be found that is stored in the database

Sizes of Data Bases

Structures:

- elements C, H, N, O
- at least 1 C-atom
- standard valencies
- no charges
- no radicals
- only connected structures

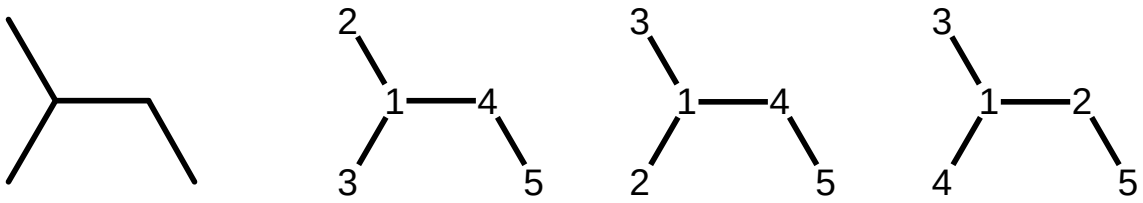


Need for techniques to explore virtual chemical space in silico!

Chemical Componds in Nature and in Silico

Chemical compounds

- in nature: atoms are not labeled
- in a computer: atoms have to be labeled



leads to problems

- deciding whether two labeled structures are isomorphic (isomorphism problem)
- enumerating all unlabeled structures

Discrete mathematics knows solutions!

Structure Counting, Enumeration and Sampling

Different disciplines

- Counting
  - only number of structures
  - non-constructive
- Enumeration
  - constructive
  - exhaustive
  - non-redundant

focus on „Orderly Generation“

- Sampling
  - constructive
  - not necessarily exhaustive
  - maybe redundant

M. Meringer: Structure Enumeration and Sampling. Handbook of Chemoinformatics Algorithms, Edited by J. L. Faulon, A. Bender, CRC/Chapman&Hall, 233-267, 2010.

Order on Edges of Labeled Graphs

Order on edges of graphs:

$e = (x,y)$, $e' = (x',y')$ with $x < y$, $x' < y'$

then $e < e'$, iff

$x < x'$ or ($x = x'$ and $y < y'$)

Examples:

$(1,2) < (2,3)$

$(1,2) < (1,3)$

Order on Labeled Graphs

Lexicographical order on graphs on n nodes

$\gamma = \{e_1,\ldots,e_t\}$ with $e_1 < \ldots < e_t$

$\gamma' = \{e'_1,\ldots,e'_{t'}\}$ with $e'_1 < \ldots < e'_{t'}$

then $\gamma < \gamma'$, iff

(there is an i with $e_i < e'_i$ and for all $j < i$: $e_j = e'_j$) or

($t < t'$ and for all $j \leq t$: $e_j = e'_j$)

Examples: graphs on 3 nodes 1, 2, 3

{(1,2),(1,3)} < {(1,2),(2,3)}

{(1,2),(1,3)} < {(1,2),(1,3),(2,3)}

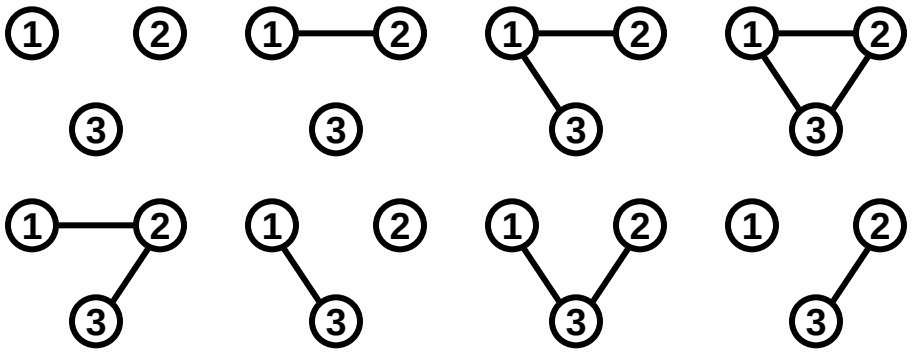
Generation of Labeled Graphs

Algorithm: Labeled Generation (γ)

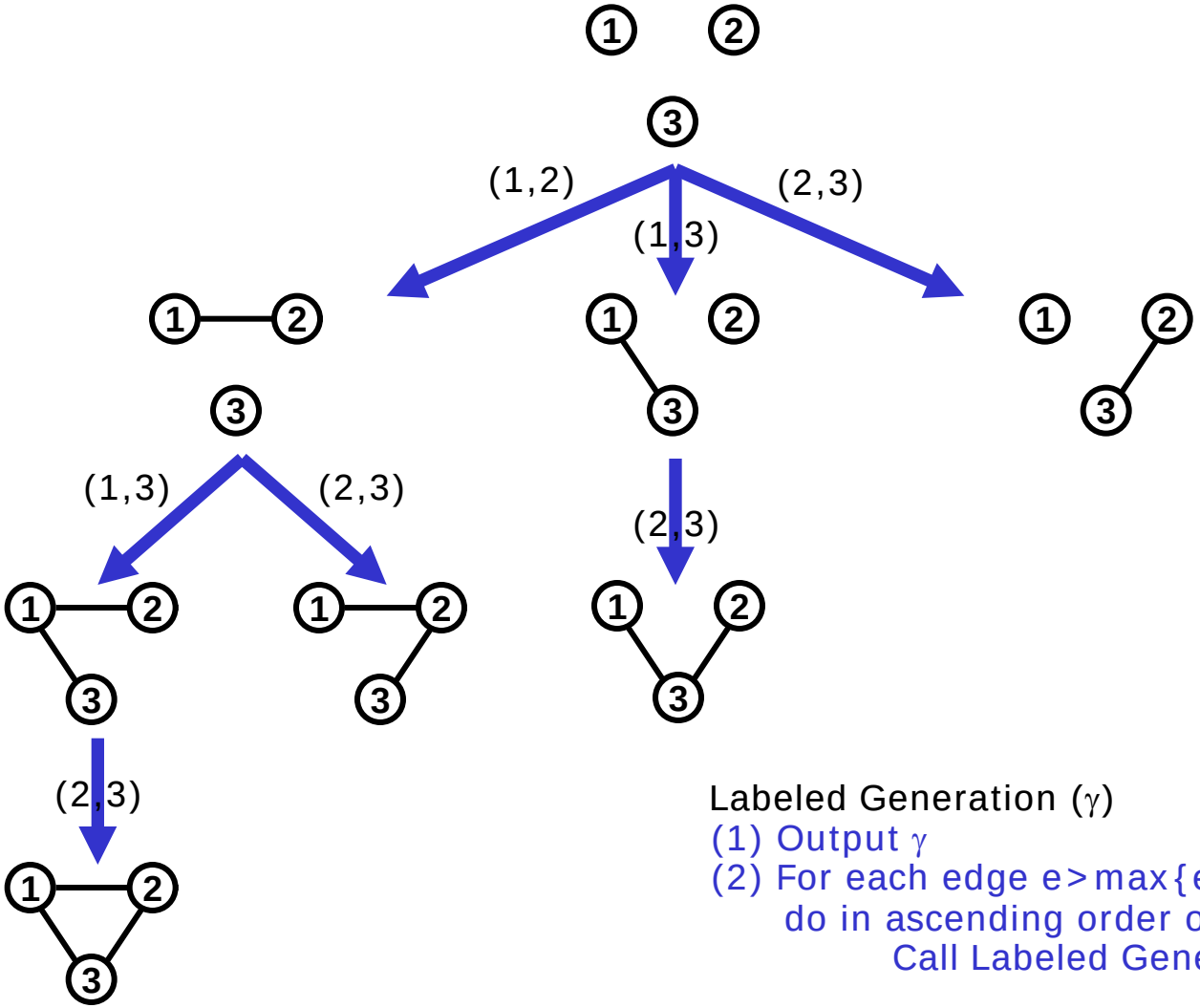
(1) Output γ

(2) For each edge $e > \max\{e' \in \gamma\}$
do in ascending order of e
Call Labeled Generation ($\gamma \cup \{e\}$)

Example: graphs on 3 nodes starting with the empty graph, Labeled Generation ({}) produces the output



Example: Labeled Graphs on 3 Nodes



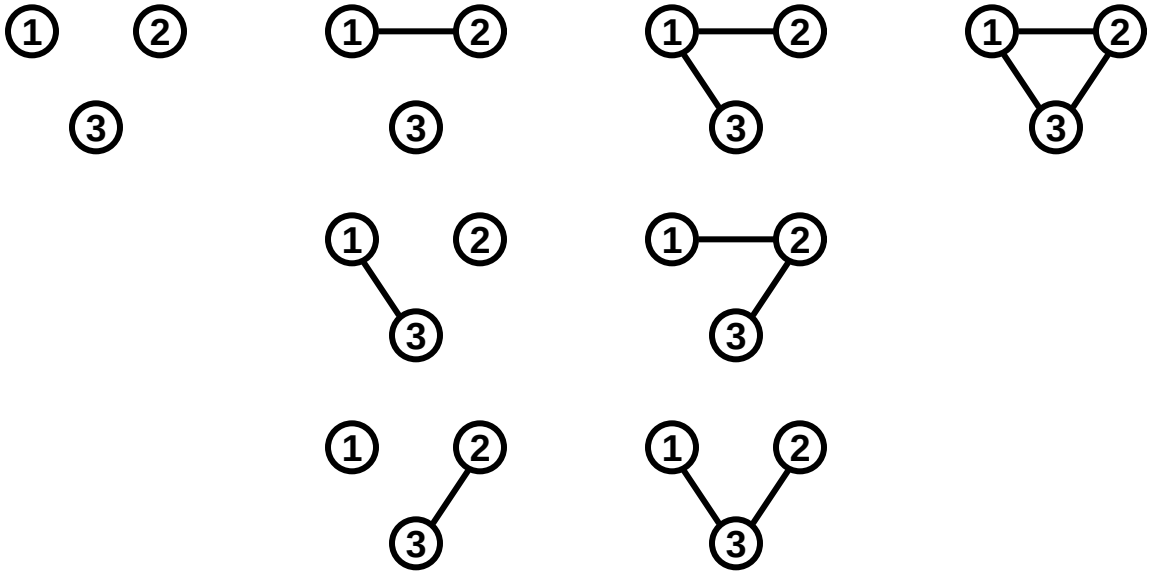
Labeled Generation (γ)

(1) Output γ

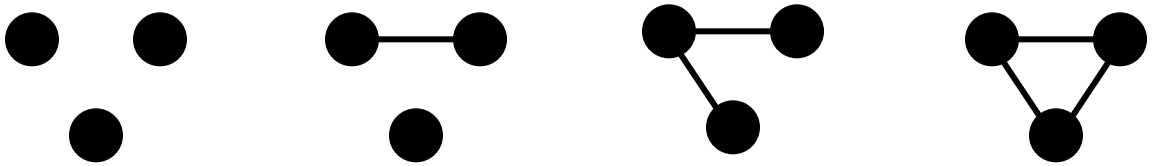
(2) For each edge $e > \max\{e' \in \gamma\}$
do in ascending order of e
Call Labeled Generation ($\gamma \cup \{e\}$)

From Labeled to Unlabeled Graphs

Isomorphism problem: How to obtain from labeled graphs ...

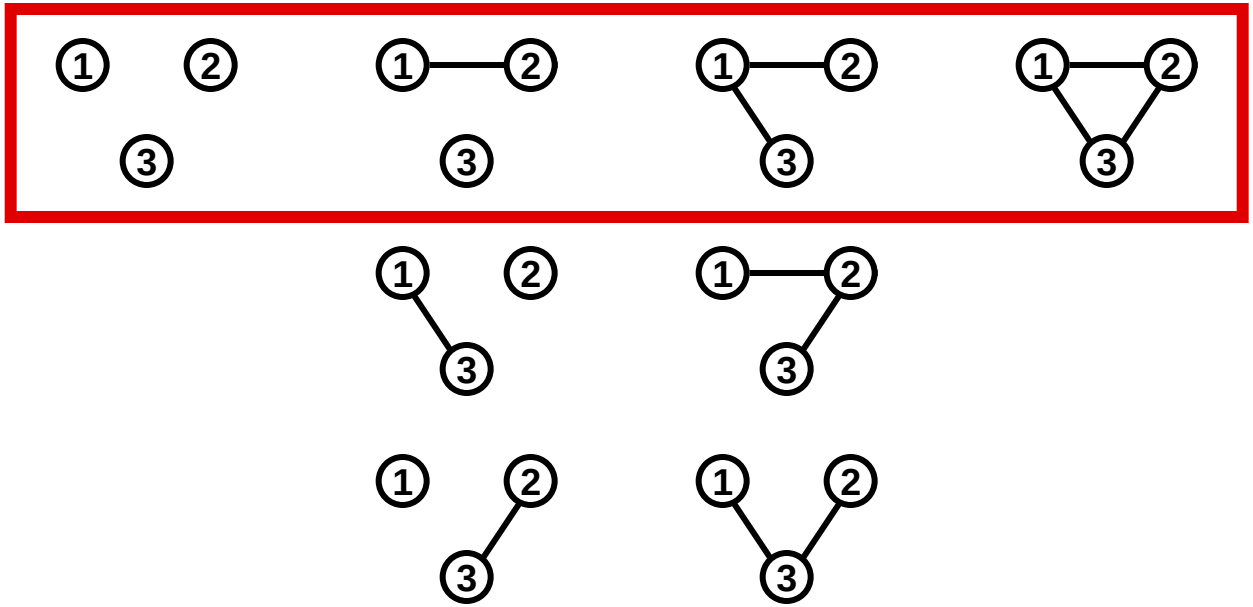


... unlabeled graphs ?



Canonical Orbit Representatives

Solution: Select from each orbit (column) the lexicographically minimal representative



Note: Testing minimality is a rather expensive procedure, up to n! permutations have to be checked

Testing Minimality

γ is minimal, iff

for each permutation of the symmetric group S_n:

$\gamma \leq \pi(\gamma)$

Example:

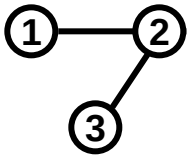
$\pi_3(\{(1,2),(2,3)\})$

$= \{(2,1),(1,3)\}$

$= \{(1,2),(1,3)\}$

$< \{(1,2),(2,3)\}$

$\Rightarrow$ not minimal



$x \rightarrow$	1	2	3
$\pi_1(x)$	1	2	3
$\pi_2(x)$	1	3	2
$\pi_3(x)$	2	1	3
$\pi_4(x)$	2	3	2
$\pi_5(x)$	3	1	2
$\pi_6(x)$	3	2	1

Note: Using algebraic and group-theoretic methods, costs for testing minimality can be reduced considerably

Generation of Unlabeled Graphs

Algorithm: Labeled Generation (γ)

(1) Output γ

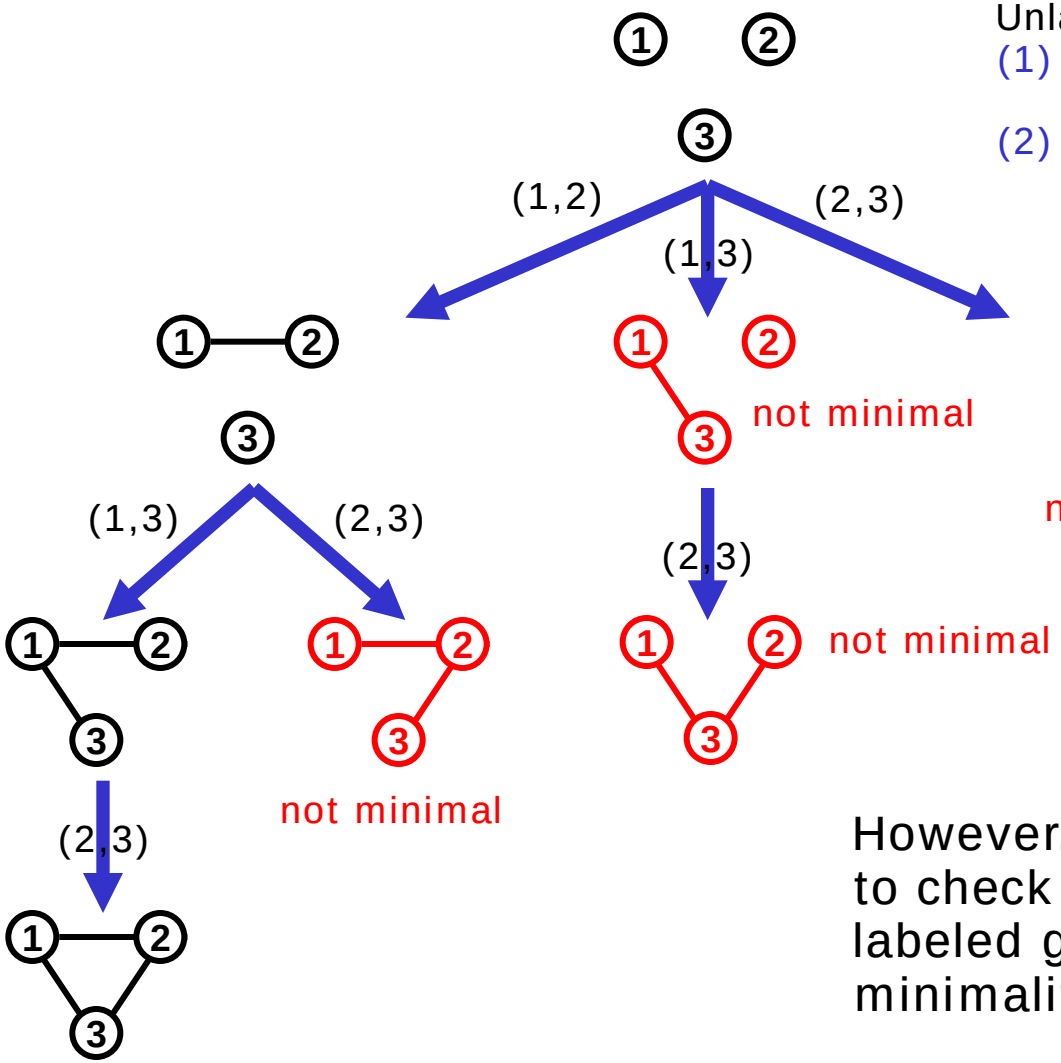
(2) For each edge $e > \max\{e' \in \gamma\}$
do in ascending order of e
Call Labeled Generation ($\gamma \cup \{e\}$)

Algorithm: Unlabeled Generation (γ)

(1) If γ is minimal in its orbit then
Output γ

(2) For each edge $e > \max\{e' \in \gamma\}$
do in ascending order of e
Call Unlabeled Generation ($\gamma \cup \{e\}$)

Example: Unlabeled Graphs on 3 Nodes



Unlabeled Generation (γ)

(1) If γ is minimal in its orbit then
Output γ

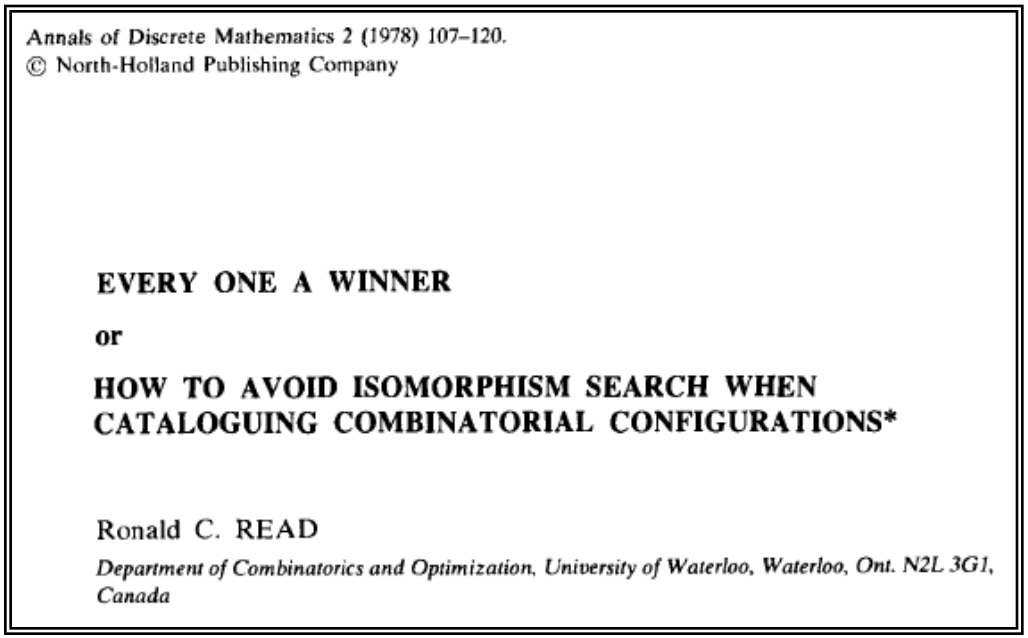
(2) For each edge $e > \max\{e' \in \gamma\}$
do in ascending order of e
Call Unlabeled Generation
($\gamma \cup \{e\}$)

However, this algorithm still has to check all of the $2^{n(n-1)/2}$ labeled graphs on n nodes for minimality

Orderly Generation

Theorem (Read, Faradzev 1978):

Every minimal orbit representative with q edges has a minimal subgraph with q − 1 edges.



Annals of Discrete Mathematics 2 (1978) 107–120.

EVERY ONE A WINNER

or

HOW TO AVOID ISOMORPHISM SEARCH WHEN CATALOGUING COMBINATORIAL CONFIGURATIONS*

Ronald C. READ

Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ont. N2L 3G1, Canada

⇒ non-minimal intermediates do not have to be considered for further augmentation

Orderly Generation of Graphs

Algorithm: Unlabeled Generation (γ)

(1) If γ is minimal in its orbit then
Output γ

(2) For each edge $e > \max\{e' \in \gamma\}$
do in ascending order of e
Call Unlabeled Generation ($\gamma \cup \{e\}$)

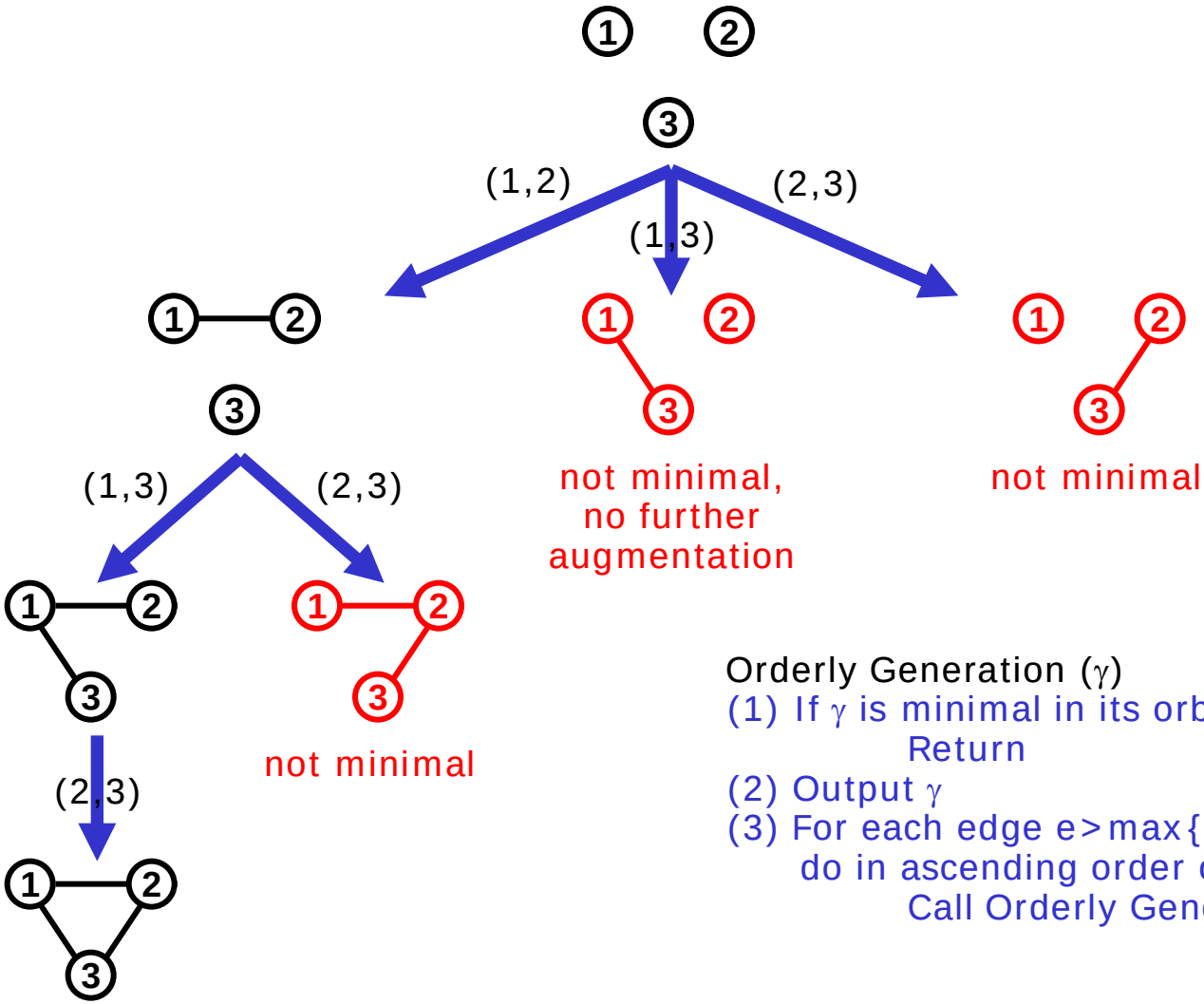
Algorithm: Orderly Generation (γ)

(1) If γ is not minimal in its orbit then
Return

(2) Output γ

(3) For each edge $e > \max\{e' \in \gamma\}$
do in ascending order of e
Call Orderly Generation ($\gamma \cup \{e\}$)

Example: Orderly Generation of Graphs on 3 Nodes



Orderly Generation (γ)

(1) If γ is minimal in its orbit then
Return

(2) Output γ

(3) For each edge $e > \max\{e' \in \gamma\}$
do in ascending order of e
Call Orderly Generation ($\gamma \cup \{e\}$)

Introducing Constraints

Mathematically, a constraint R is a symmetry-invariant mapping from the set of graphs onto boolean values:

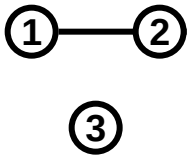
$R(\gamma) = R(\pi(\gamma))$ for each $\pi \in S_n$

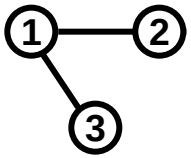
We say

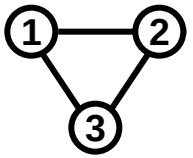
γ fulfills a constraint R, if $R(\gamma) =$ *true* and

γ violates a constraint R, if $R(\gamma) =$ *false*

Examples:







Constraint			
is connected:	*false*	*true*	*true*
has a cycle:	*false*	*false*	*true*
≤ 2 edges:	*true*	*true*	*false*

Consistent Constraints

A constraint R is called consistent if the violation of a graph γ to R implies that every augmentation γ' of γ violates R:

$$R(\gamma) = \mathit{false} \wedge \gamma \subset \gamma' \Rightarrow R(\gamma') = \mathit{false}$$

Examples:

- consistent: "$\leq$ 2 edges", upper number of edges, a minimal cycle size or graph–theoretical planarity
- inconsistent: "is connected", " has a cycle", presence or absence of a certain subgraph or a maximum ring size

Consistent constraints accelerate structure generation

Incorporating Constraints into Structure Generation

- Consistent constraints: unproblematic
  - check after each insertion of a new edge
  - help to prune the backtracking tree
  - accelerate structure generation
- Inconsistent constraints: more problematic
  - testing only useful, when a graph is complete
- Completeness itself is described by constraints
  - for generating constitutional isomers typically defined as degree sequence

Orderly Generation with Constraints

Algorithm: Orderly Generation with Constraints (γ)

(1) If γ is minimal in its orbit then
Return

(2) If γ violates any consistent constraint then
Return

(3) If γ fulfills all inconsistent constraints then
Output γ

(4) For each edge $e > \max\{e' \in \gamma\}$
do in ascending order of e
Call Orderly Generation with Constraints ($\gamma \cup \{e\}$)

Note: Efficiency is depending on the sequence of tests

Sequence of Tests during Structure Generation

	low	...	high
costs	$	...	$$$
selectivity	*	...	***

- $***: process cheap, selective tests early
- $$$*: process expensive, indiscriminate tests late
- others: find a good trade-off for others

Refinements for Avoiding Minimality Tests

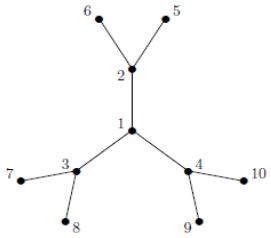
- Semi–canonicity
  - testing mininimality is often replaced by a cheaper, necessary condition for minimality
  - principle: check only for transpositions τ if $\gamma < \tau(\gamma)$
  - full minimality test delayed until the graph is completed

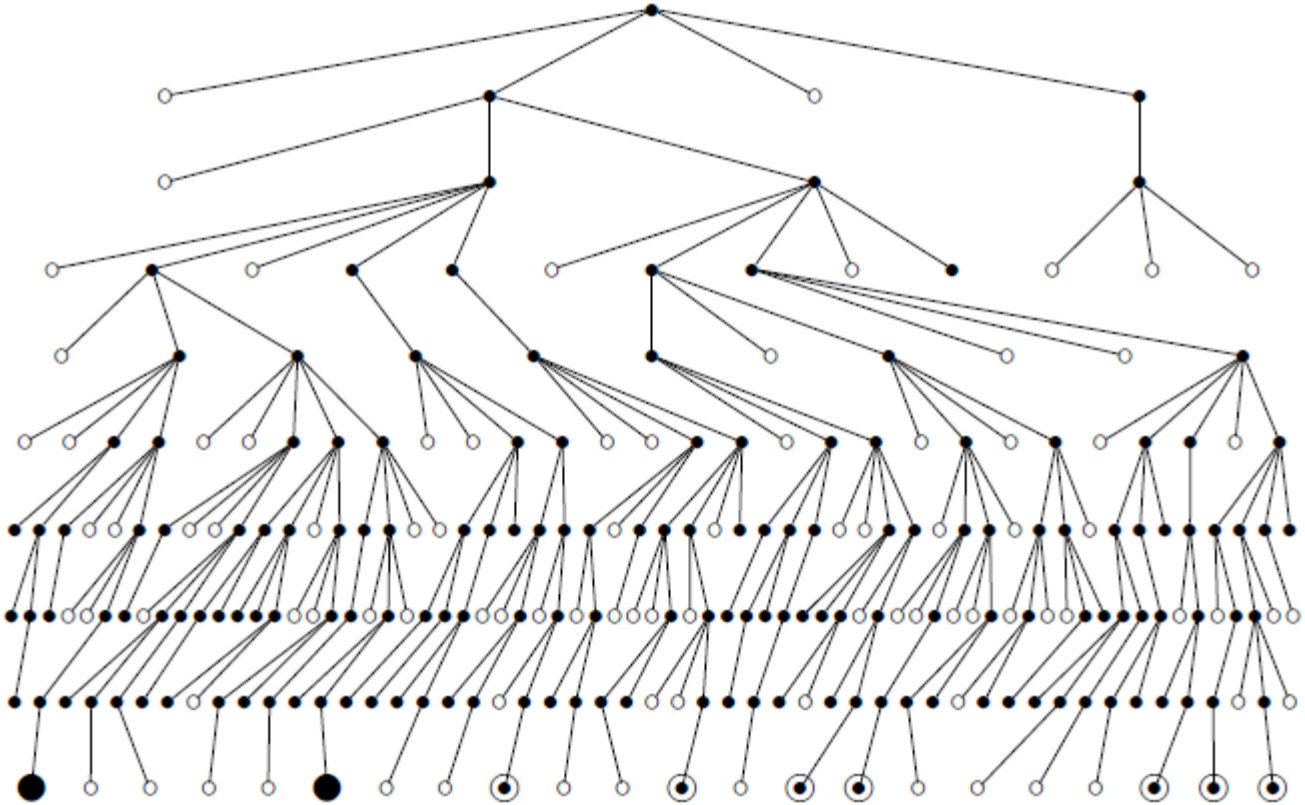
- Learning criterion
  - derives from a non-minimal graph a necessary condition for the minimality of the lexicographic successors
  - determines the earliest extension step where non–minimality could have been detected during generation
  - prunes the backtracking tree

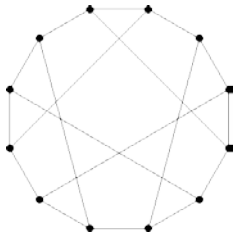
R. Grund: Construction of Molecular Graphs with Given Hybridizations and Non-overlapping Fragments, Bayr. Math. Schriften 49, 1-113, 1995 (in German)
M. Meringer: Fast Generation of Regular Graphs and Construction of Cages. Journal of Graph Theory 30, 137-146, 1999.

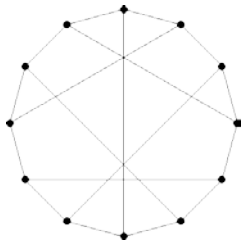
Example of a Backtracking Tree

Regular graphs on 12 nodes, degree 3, girth at least 5









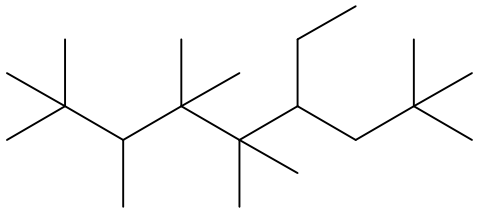
Software:
GENREG
(free)

o: girth criterion failed; ⊙: complete, but not minimal; ●: complete and minimal; •: others

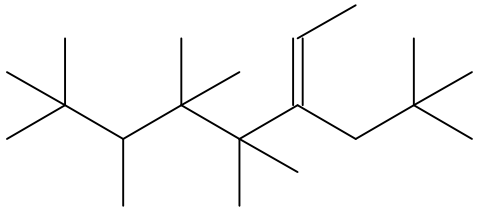
Note: Number of all *labeled* regular graphs on 12 nodes, degree 3: 11,555,272,575

From Simple Graphs to Molecular Graphs

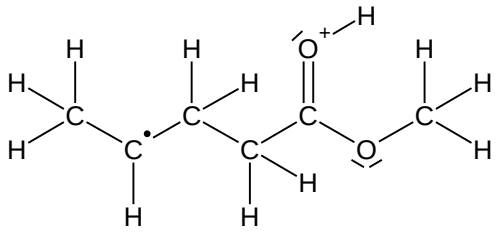
- Simple Graphs
  - nodes and edges



- Multigraphs
  - additionally: edge multiplicities



- Molecular graphs
  - additionally: element & atomic state symbols

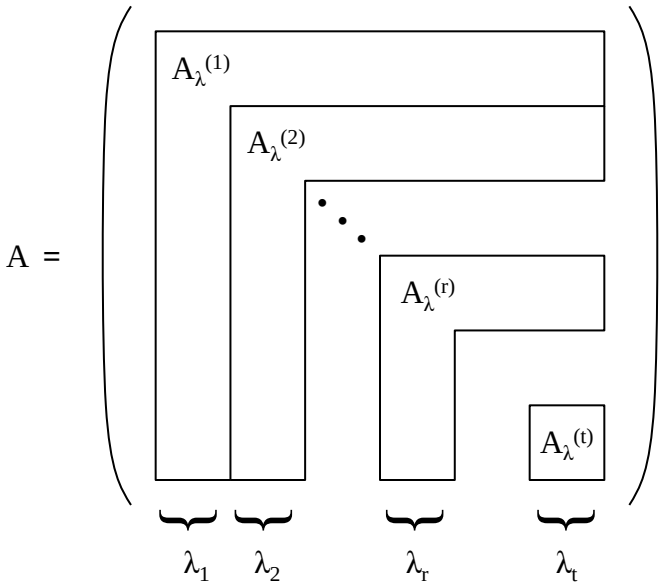


Adaptions for Generating Molecular Graphs

- Use lexicographical order on the adjacency matrix
- Canonical: lexicographically maximal adjacency matrix
- Implicit treatment of hydrogen
- Attributes of atoms:
  - element symbol
  - hydrogen count
  - valency sum
  - charge
  - unpaired electrons
  - bond order distribution
  - ...

Refinements for Generating Molecular Graphs

- Atoms with identical attributes define t blocks of the adjagency matrix
- If attributes cannot be deduced directly from input, iterate through all possibilities
- Fill adjagency matrix block-wise
- Test canonicity after a block is filled
- Complexity of canonicity test decreases from n! to $\lambda_1!\cdot\ldots\cdot\lambda_t!$
- For canonicity testing of block r only automorphisms of blocks 1,...,r-1 need to be considered



Implementations and Examples

- MOLGEN 3.5 (1997)
- MOLGEN 4.0 (1998), MOLGEN-MS, MOLGEN-QSPR
- MOLGEN 5.0 (2007, freely accessible online version)
- others, e.g. Assemble

Computational example with restrictions

Restrictions	no. of isomers	CPU-time
Chemical formula $C_6H_8O_6$ only	2,558,517	838 s
no triple bonds	2,434,123	703 s
hydrogen distribution $1CH_2$,$2CH_1$,3C,4OH	79,831	25 s
no substructure -O-O-	35,058	97 s
hybridization 1Csp3-2H,2Csp3-1H,3Csp2-OH,1Osp2-OH	990	8 s
minimal size of rings =5	348	5 s
contains at least one CO_3 branch	15	11 s

www.molgen.de

T. Grüner, A. Kerber, R. Laue, M. Meringer: MOLGEN 4.0. MATCH Communications in Mathematical and in Computer Chemistry 37, 205-208, 1998.

Example: Constitutional Spaces

Molecular formula	Structural formulae	CPU time	Beilstein database	NIST MS database
$CH_2N_6O_3$	76720	0.2	0	0
CH_6N_8O	97234	0.3	0	0
$C_2H_2N_4O_4$	216893	0.6	0	0
$C_2H_6N_6O_2$	971399	2.4	1	0
$C_2H_{10}N_8$	57508	0.2	0	0
$C_3H_2N_2O_5$	137656	0.4	0	0
$C_3H_6N_4O_3$	2429018	6.2	10	1
$C_3H_{10}N_6O$	749873	2.1	0	0
$C_4H_2O_6$	9986	0.1	1	0
$C_4H_6N_2O_4$	1432731	3.9	22	0
$C_4H_{10}N_4O_2$	2125930	5.9	33	1
$C_4H_{14}N_6$	68990	0.2	0	0
$C_5H_2N_6$	7055345	14.8	1	0
$C_5H_6O_5$	95870	0.3	28	2
$C_5H_{10}N_2O_3$	1360645	3.8	153	9
$C_5H_{14}N_4O$	311390	1.0	6	0
$C_6H_2N_4O$	26123593	49.9	3	0
$C_6H_{10}O_4$	97394	0.3	345	25
$C_6H_{14}N_2O_2$	257122	0.8	249	3
$C_6H_{18}N_4$	6742	0.0	7	2
$C_7H_2N_2O_2$	17388955	34.1	0	0
$C_7H_6N_4$	96024197	196.1	94	10
$C_7H_{14}O_3$	22151	0.1	672	36
$C_7H_{18}N_2O$	9780	0.0	52	2
$C_8H_2O_3$	1187784	2.7	2	0
$C_8H_6N_2O$	109240025	217.7	177	14
$C_8H_{18}O_2$	1225	0.0	334	28
$C_9H_6O_2$	9660231	20.4	45	4
$C_9H_{10}N_2$	46024195	98.6	411	22
$C_{10}H_{10}O$	7288733	17.2	421	34
$C_{11}H_{14}$	950064	2.7	450	52
$C_{12}H_2$	3571212	65.0	1	0

- molecular mass 146
- elements C, H, N, O
- at least 1 C-atom
- standard valencies
- no charges
- no radicals
- only connected structures

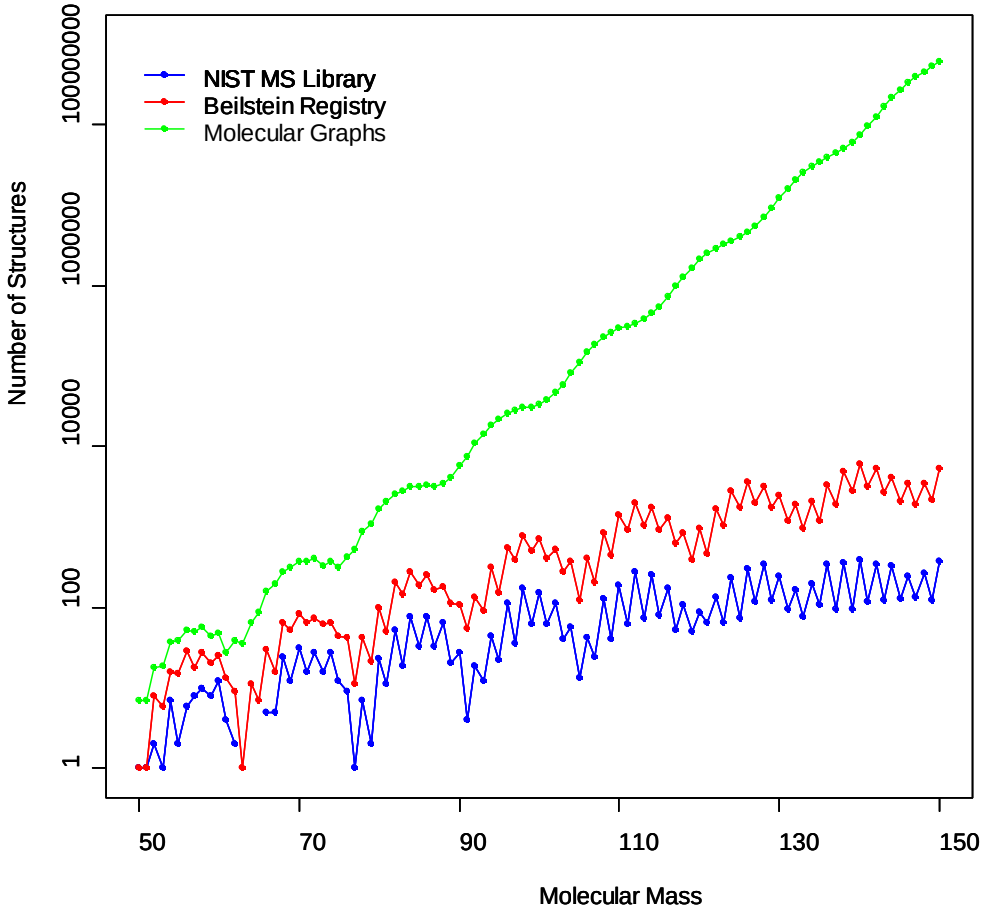
M. Meringer: Mathematische Modelle für die kombinatorische Chemie und die molekulare Strukturaufklärung. Doctoral thesis, University of Bayreuth, May 2004. Published by Logos-Verlag, Berlin.

A. Kerber, R. Laue, M. Meringer, C. Rücker: Molecules in Silico: The Generation of Structural Formulae and Applications. Journal of Computer Chemistry, Japan 3, 85-96, 2004.

Sizes of Data Bases and Compound Spaces

Structures:

- elements C, H, N, O
- at least 1 C-atom
- standard valencies
- no charges
- no radicals
- no stereoisomers
- only connected structures



A. Kerber, R. Laue, M. Meringer, C. Rücker: Molecules in Silico: Potential versus Known Organic Compounds. MATCH 54 (2), 301-312, 2005.

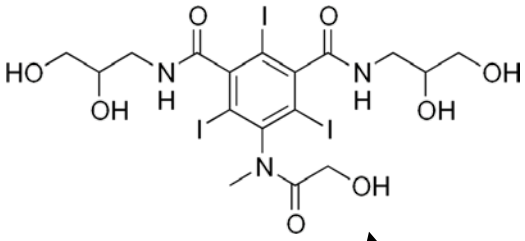
Application: Molecular Structure Elucidation

- What?

  structural characterization of

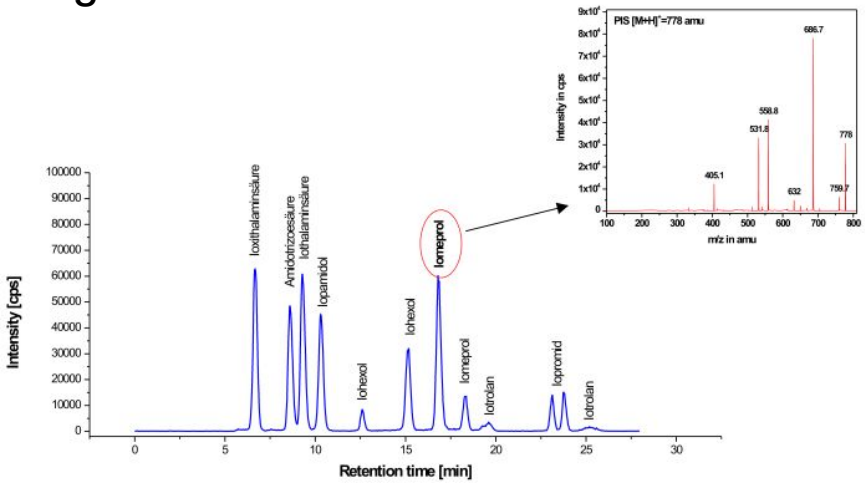
  unknown chemical compounds

- Why?
  - environmental chemistry: toxic substances
  - natural products chemistry: drugs …



- How?

  methods and devices of analytical chemistry:
  - chromatography
  - spectroscopy



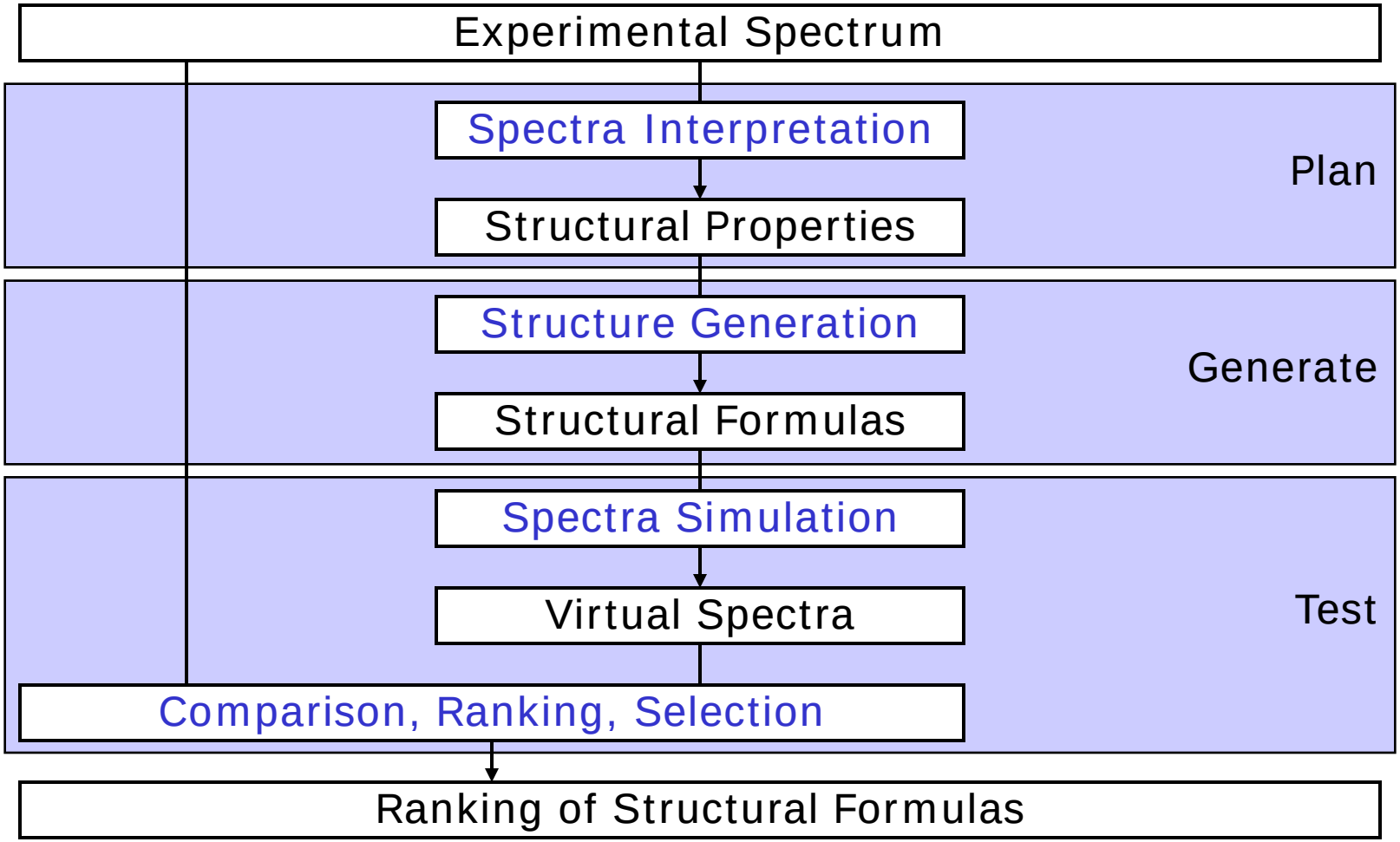
- Data analysis:
  - library searching
  - improvements desired (→ "de novo" structure elucidation)

The DENDRAL Project

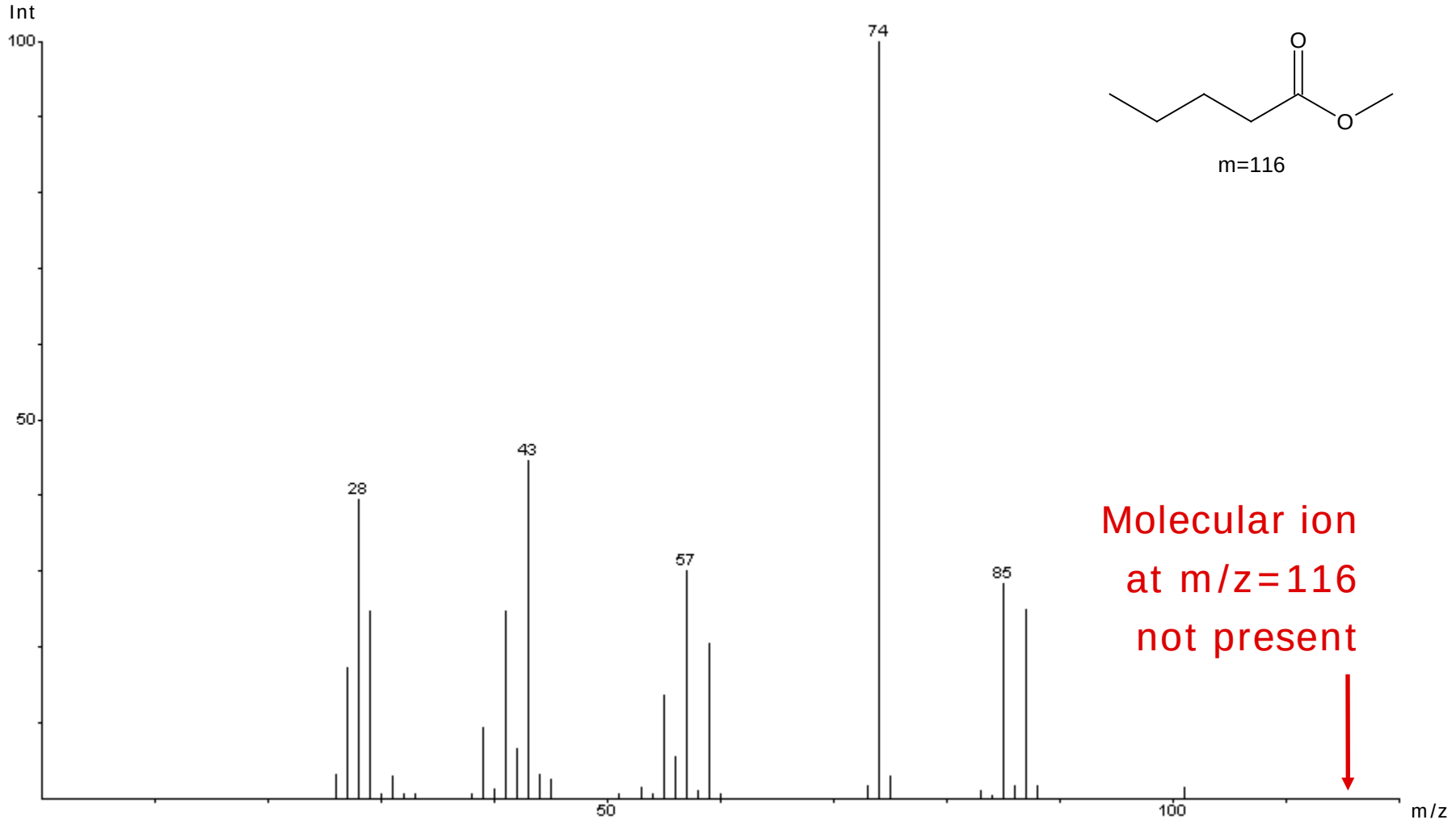
- short for DENDritic ALgorithm
- mid 1960s – early 1970s
- pioneer project in artificial intelligence
- first expert system
- aim: identifying unknown organic molecules by analyzing their mass spectra automatically
- perspective: onboard processing (structure elucidation) of mass spectra on mars missions
- first attempt to construct chemical compound space
- based on the plan-generate-test paradigm

R.K. Lindsay, B.G. Buchanan, E.A. Feigenbaum, J. Lederberg. Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project. McGraw-Hill Book Company, 1980.

From Spectra to Structure
Flowchart: Plan – Generate – Test



Example: (LR) EI-MS of an ‚Unknown' Compound

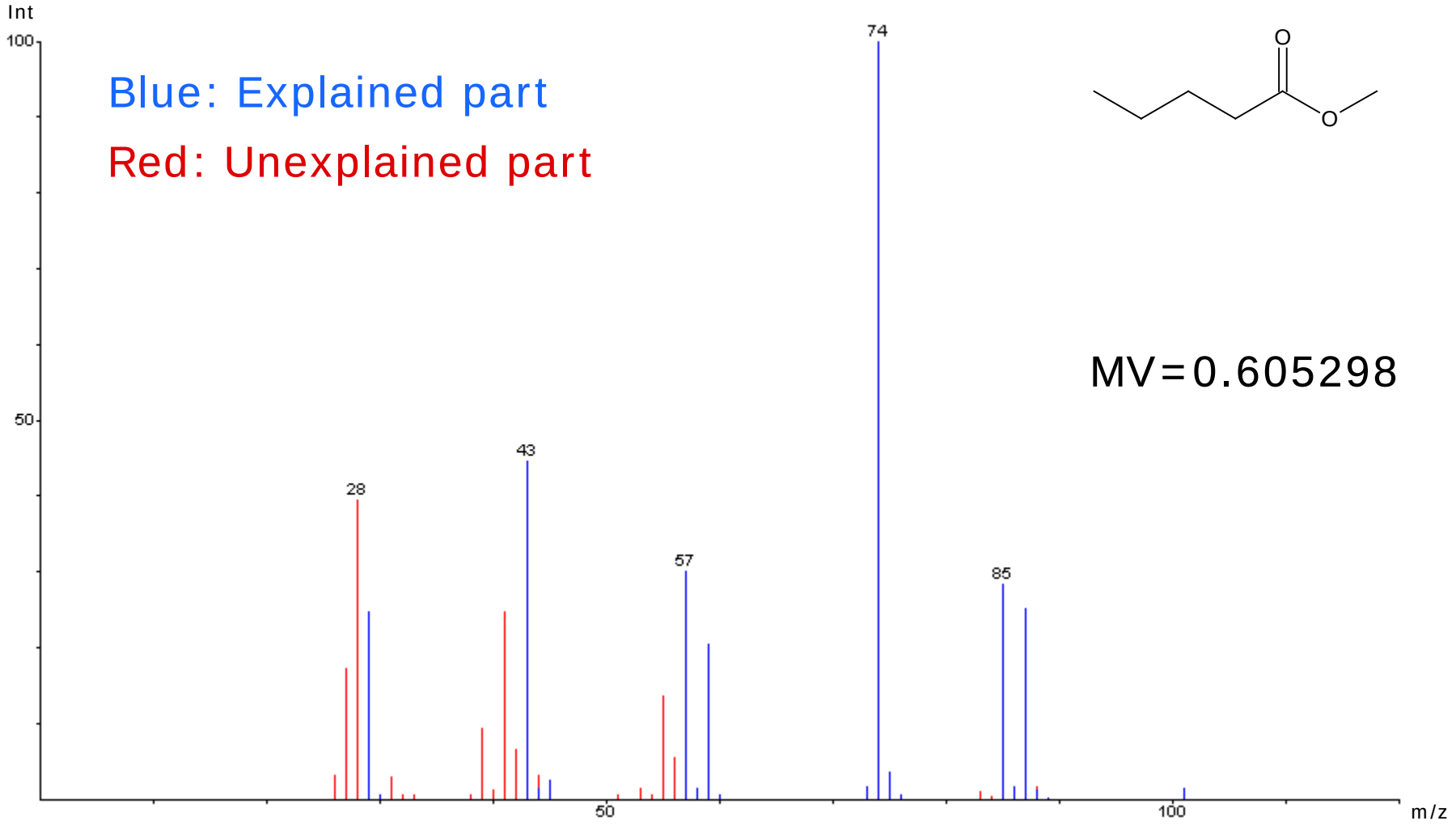


Example: Plan – Generate – Test

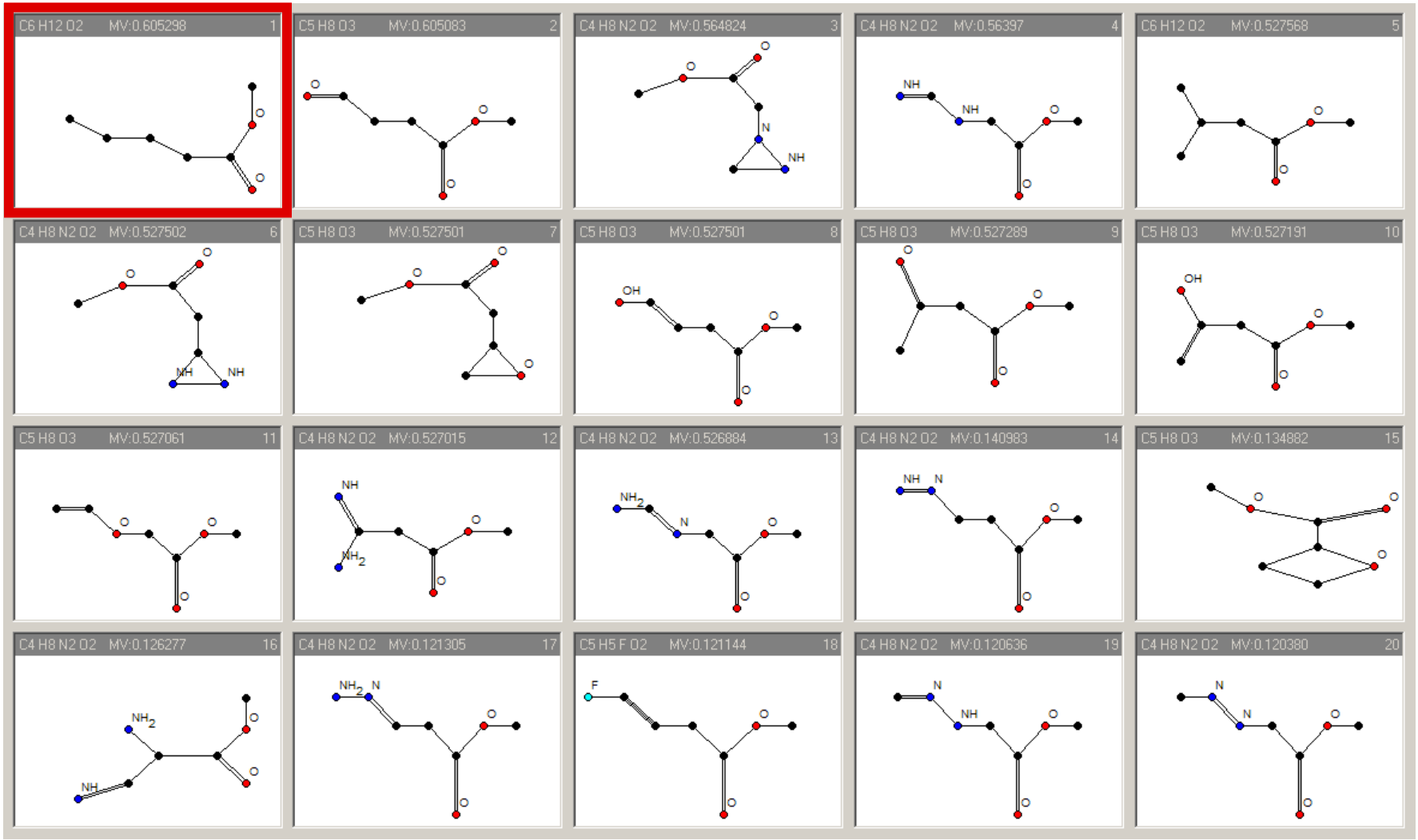
- Plan
  - MS Classifier me-est says “YES” with precision of 98%
  - Functional group –C(=O)–O–CH_3 is likely to be present
- Generate
  - 8 Molecular formulas of mass 116 including $C_2O_2H_3$
  - 131 structural formulas including –C(=O)–O–CH_3
- Test
  - simulated spectrum for each structural formula
  - compare, rank, select …

K. Varmuza, W.Werther: Mass Spectral Classifiers for Supporting Systematic Structure Elucidation. J. Chem. Inf. Comput. Sci., 36, 323-333, 1996.
A. Kerber, M. Meringer, C. Rücker: CASE via MS: Ranking Structure Candidates by Mass Spectra. Croatica Chemica Acta 79, 449-464, 2006.
E. L. Schymanski, C. Meinert, M. Meringer, W. Brack: The Use of MS Classifiers and Structure Generation to Assist in the Identification of Unknowns in Effect-Directed Analysis. Analytica Chimica Acta 615 (2), 136-147, 2008.
E. L. Schymanski, M. Meringer, W. Brack : Matching Structures to Mass Spectra Using Fragmentation Patterns - Are the Results as Good as they Look? Anal. Chem. 81, 3608-3617, 2009.
E. L. Schymanski, M. Meringer, W. Brack: Automated Strategies To Identify Compounds on the Basis of GC/EI-MS and Calculated Properties. Anal. Chem. 83, 903-912, 2011.
M. Meringer, S. Reinker, J. Zhang, A. Muller: MS/MS Data Improves Automated Determination of Molecular Formulas by Mass Spectrometry. MATCH 65, 259-290, 2011.

Example: Explained Part of the Spectrum



Example: Ranked Structural Formulas



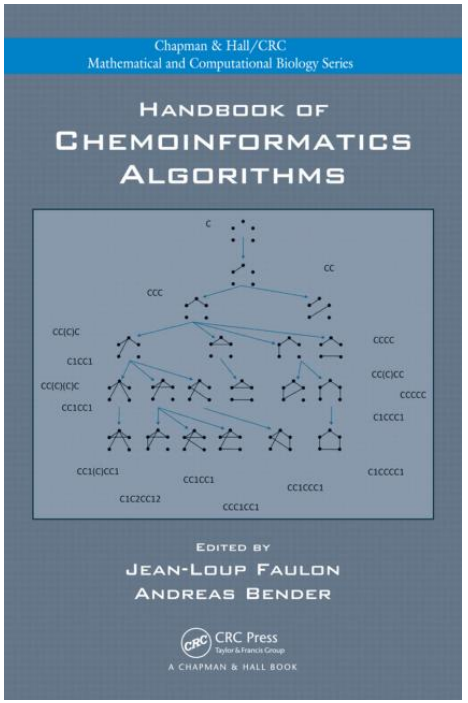
Conclusions

- Structure generation
  - solved: mathematical-algorithmic description
  - open: combinatorial explosion

- Applications in chemistry
  - solved: principles for relating structure and properties
  - open: precision, accuracy

Acknowledgements

- My Colleagues
  Department of Atmospheric Processors,
  Remote Sensing Technology Institute,
  German Aerospace Center
- Profs. A. Kerber and R. Laue et al
  Department of Mathematics,
  University of Bayreuth
- Emma Schymanski, Dr. Werner Brack
  Department of Effect-Directed Analysis,
  UFZ Center for Environmental Research
- Prof. Jean-Loup Foulon
  for inspiring me to write Chapter 8 for the
  “Handbook of Chemoinformatics Algorithms"
- IPAM/UCLA
  for the invitation to talk here



THANKS FOR YOUR ATTENTION!