Kekulé: OCR—Optical Chemical (Structure) Recognition[†]

Joe R. McDaniel* and Jason R. Balmuth

Fein-Marquart Associates, Inc., 7215 York Road, Baltimore, Maryland 21212

Received March 10, 1992

The Kekulé program was developed to fulfill a need for efficiently converting printed chemical structure diagrams into connection tables and other computer-readable formats suitable for chemical structure database updating and searching. Scanners are used to capture structure diagrams. Kekulé then interprets these diagrams into an internal format using raster-to-vector, optical character recognition, and rule-based logic. Editing of the resulting interpretation or creation of new diagrams is provided. Output is available in ISIS, MOLfile, SMILES, ROSDAL, and internal formats. Interfaces are provided for communicating results to remote database systems. Resulting structure images can be copied to word processors and desktop publishers for publication-quality output.

INTRODUCTION

Chemists communicate structural information for compounds via structural diagrams. However, chemical structure databases understand not structural images but connection tables—lists of the atoms with their connecting bonds, types of bonds, etc.—or structural string notations. An example of a connection table is the MACCS-II[1] MOLfile format. Examples of string notation systems include Daylight's[2] SMILES and Beilstein's[3] ROSDAL formats. While chemical structure database systems can produce structural diagrams, they cannot read such diagrams. What is needed is an equivalent to optical character recognition—optical chemical (structure) recognition that can automatically turn a structural diagram into a structure table.

In the past, there were two basic techniques for creating structural descriptions—connection tables and structural strings—that were suitable for input into chemical structure database: manual compilation of the structural description and "automatic" compilation of the structural description by a software package capable of generating a description from a structural diagram drawn via the facilities of the software package. However, both techniques had serious shortcomings.

Manual compilation of structural descriptions is extremely tedious and highly prone to error. The specialized formats such structure descriptions take and the difficulty of translating a structural image into the appropriate format—without errors or oversights—render this technique little better than a desperation maneuver.

Using a structure drawing program, which translates a drawn image into a structure description, is a far better option. Various programs for drawing chemical structures while *simultaneously* capturing the connectivity information have existed for many years. Examples of this latter capability are SuperStructure,[4] MOLKICK,[5] and STN Express.[6] Warr and Wilkins[7] have compiled a more extensive listing of such software. However, redrawing of structures for the purpose of capturing connectivity is an effective but inefficient method; redrawing a complex molecule such as vitamin B_{12} takes about 20 min. Even redrawing relatively simple structures is time-consuming and error-prone compared to processing scanned images automatically.

The approach taken in Kekulé[8] is to literally read the structure diagram. The basic approaches to this task by various investigators seem to be similar.[9–13] However, Kekulé seems to be the first successful attempt to integrate all of the required elements of image processing, optical character recognition, structure editing, communication, and publication-quality output. None of the other systems we know about is a complete system—some lack the OCR capability, most lack any way of editing interpreted structures or multiple interfaces to widely used database systems.

A scanner is used to capture a printed structural diagram. The captured image is digital, but it contains no directly interpretable information: it is merely a pattern of 0 and 1 bits, corresponding to light and dark areas of the image, such as a FAX machine would generate. Kekulé then processes this scanned image to extract information on characters (atom symbols, charges, masses, repeat counts, etc.), lines (single, double, and triple bonds; brackets, etc.), and other objects (e.g., stereo bonds). This extracted information is then assembled into a connection table format. If errors are detected due to chemically unrecognizable atomic symbols or group formulas, the user is prompted for verification or correction. When interpretation completes, the user may use the editing features of Kekulé for tasks as simple as repositioning nodes or as complex as creating complete new structures.

Kekulé's internal format allows for storing connectivity as well as a snapshot of the user preferences for bond styles and fonts. This native format is currently supported only by the National Cancer Institute's Drug Information System. The format is not proprietary.

The resulting interpreted diagram can be conveyed in ISIS,[14] MOLfile, SMILES (simplified molecular input line entry system),[15] or ROSDAL (representation of structure diagrams arranged linearly)[16] format to chemical structure database systems. Since Kekulé has been implemented using Microsoft Windows, a seamless interface has been created to various database systems by taking advantage of features of Windows and scripts provided by graphic terminal emulation software such as VistaCOM.[17] Interfaces to other databases can be similarly implemented by using the script languages of these emulators. Once the connectivity data has been transferred to the database host computer, the structure of interest can be added to the database or used as a template for substructure searching.

While the primary reason for developing Kekulé was to provide a means for automatically interpreting structure diagrams for use with structure databases, other uses of the

[†] Work on this project was supported by the National Cancer Institute under SBIR Grant 5 R44 CA47241.

© 1992 American Chemical Society

program, such as for publishing and documentation, are also available. The editing capability includes complete control over typefaces and fonts, selection of bond representations, and precise positioning of atoms and group formulas in a WYSIWYG (What You See Is What You Get) mode. The resulting on-screen image can be conveyed directly to typesetting equipment such as a Linotronic 630 at 3251 dpi resolution. Other opportunities for using structures are provided by interfaces to desktop publishing and word processors running under Windows from which the resulting text and structures can be produced in publication-quality form. Output can also be saved in PostScript format for inclusion in programs external to the Windows environment.

DESCRIPTION OF THE SYSTEM

The process of interpreting a chemical structure diagram consists of roughly seven steps:

1. Scanning
2. Vectorization
3. Searching for dashed lines and dashed wedges
4. Character recognition
5. Graph compilation
6. Post processing
7. Display and editing

Scanning is integrated into Kekulé but is actually accomplished using software and hardware supplied by second parties. The user of Kekulé initiates the scanning operation simply by choosing an icon in the Kekulé window with a mouse.

Any scanner that can be operated within the Windows environment can be seamlessly interfaced to Kekulé. To date, our experience has been limited to the Hewlett-Packard ScanJet page scanner with its SCANGAL software and the Logitech ScanMan-32 and ScanMan-256 hand-held scanners using the ANSEL and FotoTouch software supplied by Logitech.

The ScanJet provides 300 dpi resolution, while the ScanMan provides 400 dpi resolution. In practice, 300 dpi is sufficient for all but the smallest structural images when 400 dpi works better, and the choice between page and hand-held scanners is primarily one of personal preference. We find that the hand-held scanner is usually preferable when scanning a few structures from a document. If the document format includes several structures per page, page scanners such as the ScanJet may be preferred.

The scanning process, per se, consists of selecting an area of the page that contains a structure diagram, physically scanning that area, cropping the scanned image to eliminate extraneous text, and copying the resulting image to the Windows Clipboard. (The Clipboard provides a simple mechanism for moving data between Windows programs.) The image may be saved as a TIFF[18] file for deferred batch processing, but this step is not required when using Kekulé.

Vectorization consists of reducing the scanned image to line elements only 1 pixel in width by thinning[19] and raster-to-vector translation. The results of this step are lists of vectors associated with the original pictorial elements by coordinates of the vector end points. An adaptive smoothing algorithm developed by the authors eliminates the fine, pixel-level detail inherent in a direct translation from pixels to vectors.

Searching for dashed lines and wedges reduces the later processing requirements for Kekulé by finding such artifacts and converting them to single picture elements instead of unconnected vectors.

Several techniques are available for finding elements of collinear lines, including Hough transforms.[20] The theory behind the Hough transform is that points on a line, transformed from XY into r-Θ space, will result in peaks that can be distinguished from non-collinear data. In practice, we found that this approach was not satisfactory because of the shortness of many of the line segments. For instance, Figure 1 illustrates that some dashed lines consist of only a few pixels.

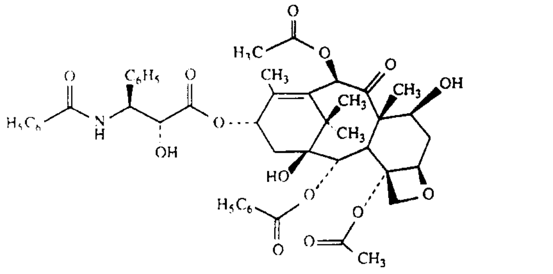


Figure 1. Typical scanned structure (Taxol) illustrating problems encountered in interpreting: dashed lines, wedge bonds, characters ligatured to bonds and adjacent characters, and small fonts.

Another approach is to extract line segments and sort them by slope.[21] The expectation is that dashed lines will then group together because they have similar, if not identical, slopes for the individual segments. In practice, the line segments are so short that they may have widely varying slopes, or even no distinguishable slope for dashes approaching the shape of round dots.

We were, therefore, forced to search for other methods for isolating dashed lines. The method that we chose was essentially one of exhaustive search—testing of all possibilities—over only the subset of features that might be possible constituents of a dashed line or dashed wedge feature. In general, we identify all dashes that consist of at least two line segments. This includes most cases where one of the two line segments is attached to other bonds.

Character recognition (OCR) was assumed to be an easily solvable problem when we began working on this project since various techniques[22] for recognizing characters have been developed, some as long ago as 1870. Commercial OCR programs are available from numerous sources. In practice, we discovered three problems with commercial approaches to OCR: (1) They did not work well in an environment including both graphic elements and characters. (2) Recognition accuracy was very low, 80–85%, for the small fonts we found were used for most structures. (3) They would have been relatively expensive to bundle with Kekulé.

The failure to discover a commercial solution to the OCR problem prompted us to investigate alternative approaches. The initial approach was one based on correlation. Unfortunately, this simple approach fails because so many characters are so similar. For example, consider the characters **b** and **h** . These differ only in the few pixels that define the bottoms of the characters. Thus, any correlation approach is going to find a high correlation between these two characters. If serif fonts are considered, as in **b** and **h**, the correlation is even higher.

Another approach that we explored to resolve the OCR difficulty dealt with neutral networks. Neural networks are often touted as a panacea for pattern recognition, and virtually every simulator program includes an OCR example. We found that the traditional fully connected, multilayer, perceptron network does a very good job at learning its training set but tends to do poorly, no better than perhaps 80%, when tested against characters not used for training. In other words, those networks tend not to generalize well.

Part of the problem with generalization is that characters are difficult to accurately place in the input space for neutral

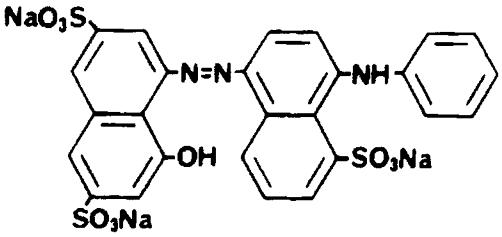


Figure 2. Structure scanned at 400 dip and enlarged here 2× to illustrate typical worst-case capability for the Kekulé OCR. The subscripts were about 3.2 points in the unenlarged original. The skewing was not intentional, but is typical of hand-held scanning, especially. Kekulé interprets (and straightens) this structure perfectly.

networks. Edges of characters are relatively indistinct, especially small fonts printed on inexpensive paper. This introduces translational variance. Similarly, rotational variance is introduced by two routes: scanning misalignment and manual positioning of paper in typewriters to add character notation to structures. Size variance occurs due to differing sources for structures, sub- and superscripts, proportional versus fixed pitch fonts, and, even within a single source for structures such as *The Merck Index*,[23] varying methods for producing structure diagrams. Finally, typeface variance adds to the problems of OCR; although there are only two major typefaces used for structures—Swiss (sanserif) and Times Roman (serif) are exemplars—innumerable variations are found.

Our investigation into OCR's practical problems, which are never discussed in the usual neural network simulator examples nor even in most papers on the subject, led us to try several approaches to achieving invariance in translation, rotation, and size of characters. Among those investigated were Zernike moments,[24-26] Fourier descriptors,[27,28] Gabor transforms,[29] coordinate transforms,[30] and Hough transform-like approaches.[31,32] Introducing such invariances may reduce the dimensionality of the pure pattern recognition problem inherent in OCR, but invariance in rotation, at least, introduces other problems in distinguishing between characters that *are* rotationally variant (such as **b** and **q**; **d** and **p**; **1** and –; E, W, and M; and **6** and **9**, among a few examples). Similarly, size invariance makes distinguishing between upper and lower case as well as **5** versus S and other characters difficult.

Our solution to the OCR problem was to solve the rotational variance problem by rotating the original scanned image to correct for scanning error and ignore the other rotational variations (since they caused few problems). Size and translation variance were solved by combinations of scaling, under sampling, and contrast and density adjustments of the scanned characters. In operation, the normalized characters are then presented to a multilayer perceptron neutral network for recognition; the network was trained on exemplars of characters from numerous serif and sanserif fonts to achieve font invariance.

The resulting OCR system typically achieves a raw accuracy of about 96%. While this accuracy may seem low compared to claims of 99.5+% from commercial OCR systems, their claims are for large fonts (10 points and larger) while our typical fonts consist of characters no larger than 7 points with subscripts as small as 3.2 points (Figures 2 and 3). Our experience in testing commercial OCR systems on small fonts produced recognition rates of about 85%. Also, we combine our raw data on characters with contextual (chemistry) rules to "spell check" and achieve a significantly higher effective recognition rate.

Neural networks can be configured so that their outputs represent some measure of the probability of a match for the character presented to the input of the network. We took advantage of this by ranking the outputs and keeping those above an arbitrary threshold that was derived experimentally for continued processing. Thus, if the actual character was a poorly formed **5**, the outputs might be relatively strong for both 5 and S, and we would retain both, determining which was correct from context.

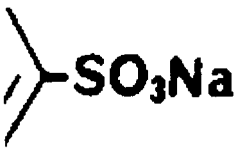


Figure 3. Enlargement (4×) of Figure 2 showing the lack of detail in subscripts typical of some images. These characters were recognized correctly by Kekulé.

Objects that are not recognized as characters are further analyzed to determine whether they are ligatured characters—characters attached to other characters or to bonds. If they are determined to be characters, the individual characters recognized are retained for later processing along with the above-matched characters. Depending on user setup options, Kekulé may prompt for characters that are not valid (above a threshold) at this stage in processing. Alternatively, prompting can be delayed until the post-processing stage when "chemical spell checking" has indicated that characters are invalid.

Individual characters are assembled into character strings based on *XY* coordinates; that is, the *XY* positions of various individual characters are compared, and character strings—formulas, say, or an atom symbol and subscript—are assembled based primarily on adjacency of the coordinates. This is an adaptive process because strings in the original scanned documents are often separated by less space than one might expect in "normal" OCR processing. The prevalence of subscripts and superscripts and the frequency with which one encounters separate symbols that are not well separated spatially introduce additional problems, all of which are dealt with in this process.

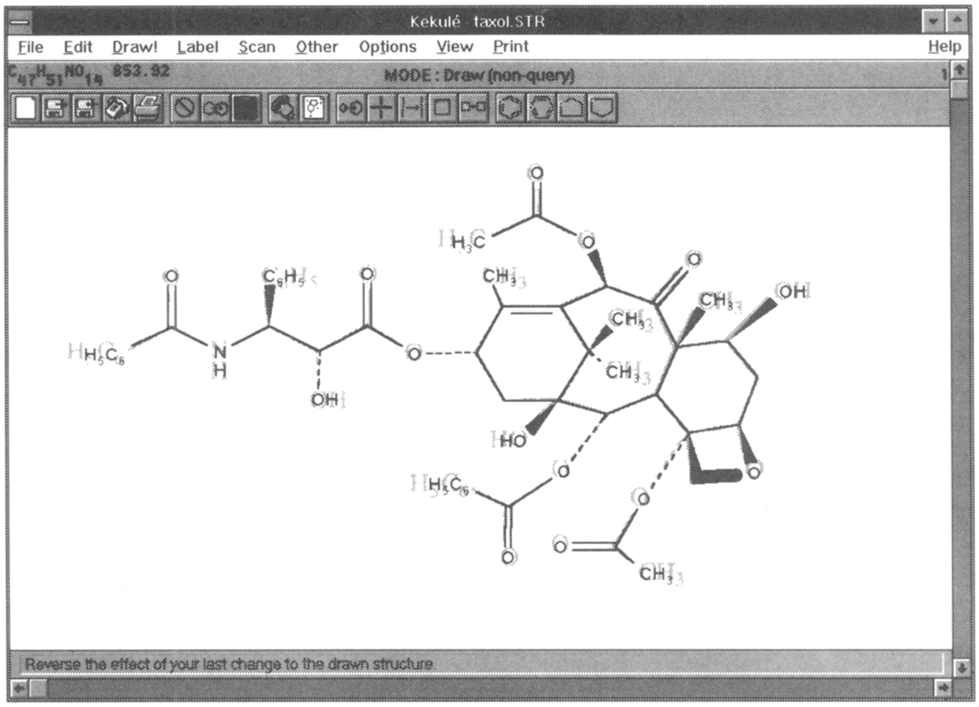
Graph compilation is the process of interpreting the remaining vector data—after eliminating those vectors associated with characters or character strings identified in the preceding step—into a connection table or graph. In reality, the process of computing a connection table is somewhat iterative since the interpretation of some features depends on the computed heights of characters and determining what is really a character ("1" versus a vertical bond, for instance) is partly based on bond lengths, etc.

The first step in graph compilation consists of defining each character string resulting from the OCR processing above as a node. Note that a character string might be as short as one character (an atom symbol).

Next, a pass is made through the list of vectors remaining after subtraction of the character string and dashed line vectors. These are generally assumed to represent bonds, and the pass through these vectors is to determine whether either end of any vector should be attached to one of the character-string-defined nodes. If either end point of a given vector is too far from a character-string-defined node to be attached to the node, a new node is created at the end point of the vector.

When a new connection is made between nodes, the pixels in the original image for that line are examined to determine the line's width at its end points to determine whether it is a filled wedge and, therefore, a stereo-up bond.

If, while connecting vector ends to nodes, it is determined that this would result in two coincident connections between



Reverse the effect of your last change to the drawn structure

Figure 4. Kekulé window with original image of Taxol in gray, interpreted image in black, and molecular formula and weight near the upper left corner of the window. The second "line" of the window contains the menu list heads. The icons just above the structure area are buttons that, when chosen, provide immediate access to menu commands and common ring templates. The bars to the right and below the structure area are scroll bars for moving around in structures too large to display in their entirety. The original image was in color; the reproduction above deemphasizes the visual impact the user actually sees. Also, this image was captured as displayed so resolution is limited. The font for recognized characters was chosen to minimize the hiding of the original image; a larger font would be more appropriate for actual use.

nodes, the connection is upgraded from a single bond (initial assumption) to a double bond, and finally, to a triple bond.

Stereo-down bonds, dashed collinear, and dashed wedges, were previously interpreted, and their data is added to the graph being compiled.

Post processing uses current graph information and the partially processed character string data—which may consist of single atom symbols, atom symbols with charge and mass, group formulas, or moiety (fragment) formulas—to determine the character string's chemical meaning. Atom symbols are verified against a list. Special symbols such as Ph (phenyl), Me (methyl), or Pr (propyl) are included in the search. Complex strings that are really group or moiety formulas are then processed to interpret them into graph format. At this stage, Kekulé will typically prompt for any nodes that are not understood as valid chemical constructs. In some cases, the reason for prompting will be errors in the original OCR processing; other reasons for prompting are usually caused by incorrect or misunderstood bonding to group formulas.

It is at this stage that we analyze the existing graph to look for circles (and convert them to alternating single–double bonds), nodes that are really bond crossings, and large parentheses and brackets. Circles are recognized by searching for rings with no external bonds within rings. Bond crossings are initially recognized as nodes. Subsequent analysis of ring systems is used to determine which nodes are really bond crossings. Large brackets and parentheses are recognized by testing the configuration of their line segments.

Display and editing processing consists of several operations including rotating the entire graph to adjust for scanning error and cleaning up bond angles to standard values when possible. The final graph is then displayed on the video screen in black superimposed on the original, scanned image in gray (Figure 4). With both images present simultaneously, it is easy for the user to verify whether the interpretation is correct. In addition to the interpreted graph, Kekulé computes the molecular formula and molecular weight of the interpreted graph—both useful for verifying the interpretation.

INTERPRETATION CAPABILITIES

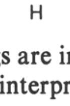
Kekulé will correctly interpret most chemical notation. The following list represents the highlights of Kekulé's ability to read chemical structures:

Single, double, or triple lines connecting atom symbols or intersecting other lines are interpreted as single, double, or triple bonding of atoms with carbon assumed if no atom symbol is present.

Characters representing atom symbols, subscripts, and superscripts are found and interpreted. (IUPAC conventions are assumed: Charge to the upper right of the symbol, mass to the upper left, and repeat counts at the lower right.)

Group formulas are interpreted into their appropriate connection table structure while retaining the on-screen character display complete with subscripts, etc. Group formulas may be multiply bonded to the remainder of the structure.

Vertical character strings such as



on the outside of rings are interpreted as if bonded.

Circles within rings are interpreted as ring-alternating bonds.

Double bonds within rings are positioned automatically for pleasing display.

Wedged bonds, either filled or as a series of parallel dashes, are interpreted as stereo-up bonds. "Thick" bonds between two wedges are automatically generated.

Collinear dashed lines are interpreted as stereo-down bonds.

π bonding (such as in ferrocene) is recognized.

Stubs can be interpreted as either hydrogen or methyl groups at the user's option.

Parentheses and brackets surrounding structures, with specific or indefinite multipliers or charges, are recognized.

Misalignment of scanning will be corrected to produce structures square to the page. Bond angles are further corrected to the nearest standard angle. (These adjustments can be canceled for special cases.)

Multiple moieties, whether as structures or formulas, are recognized and preserved.

Bond crossings are recognized.

Locants, node numbers used for nomenclature amplification, are interpreted and discarded.

STRUCTURE EDITING CAPABILITIES

What you see is what you get (WYSIWYG)—The structure on your screen is what you can print or incorporate in any Windows-based word processor or desktop publishing system.

Resolution is limited only by the output device used—this could be 3251 dpi on a Linotronic 630.

Fonts used can be any supported by Windows.

Bond styles are user selectable—includes all forms of stereo bonds and metal bonds. Double bond positioning is user adjustable.

Straightening can be applied to user-drawn images to clean them up without complete loss of control over the resulting image.

Group and moiety (fragment) formulas can be entered in normal notation with internal conversion to connection table format.

Query structure features include multivalued atoms and bonds with automatic conversion to the closest approximation supported by SMILES, ROSDAL, MOLfile, and ISIS formats.

Customizable button bar is provided for instant selection of commands.

DISCUSSION AND CONCLUSIONS

The current version of Kekulé was tested on 444 chemical structures obtained from a wide variety of sources.[33] These were chosen to test Kekulé's limits and are, therefore, not necessarily typical of structures in general. Of those structures, 98.9% were processed with an average of .74 prompts per structure for verification of interpretation or correction. The average time for this test of Kekulé's abilities was 9 s per structure on an 80486 at 33 mHz.

Five of the 444 structures tested required more than 30 s[34] of editing to correct interpretation errors. Figure 5 shows an example of one of the five structures. The basic problem with this structure is the presence of numerous broken characters. However, Kekulé processes even such difficult structures well—only the OCR processing slows down the program. Figure 6 shows the same structure after processing that took about 90 s—45 s to interpret and 45 s to edit.

The results from Kekulé, whether in ISIS, MOLfile, SMILES, or ROSDAL format (see supplementary material), can be used for building chemical structure database records and for building search inputs for searching such databases using terminal emulation programs to provide virtually seamless interfaces to MACCS-II, Daylight's DAYMENUS (and other systems), the Beilstein Database at Dialog, STN, etc. The publication-quality images printed by Kekulé can be produced directly or cut and pasted into documents and reports.

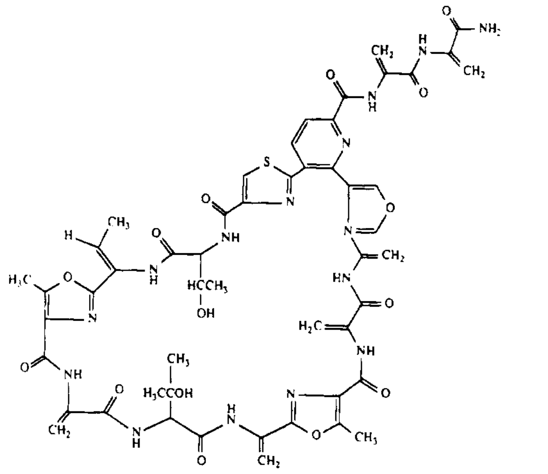


Figure 5. Scanned structure image[35] which Kekulé processes slowly due to the large number of broken characters. Many of the Hs and Ns have breaks in the cross bar of the character. Also, note that this structure has a published error in the bonding to the nitrogen in the upper right five-membered ring.

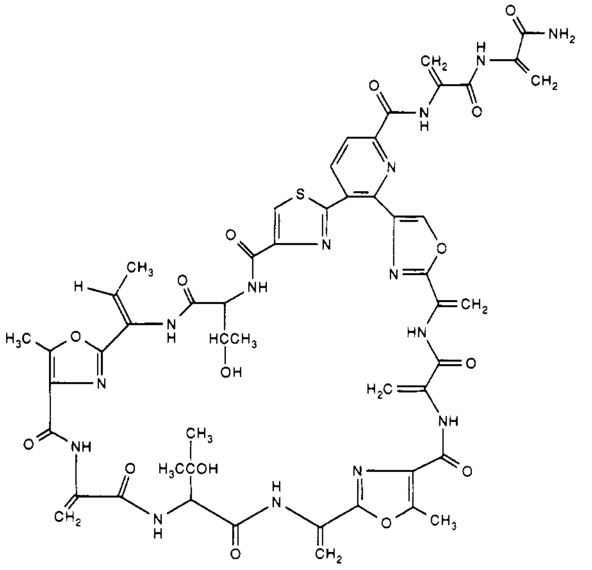


Figure 6. Structure of Figure 5 as interpreted and edited by Kekulé. Because of problems with broken characters in the original image, this structure took 45 s to interpret and 45 s to edit—more than the (arbitrary) limit of 30 s for editing to be considered as being "good". Another 45 s were required to correct the publishing errors.

Future work will include upgrading Kekulé's ability to process hand-drawn structures. Although no effort was made to provide this capability in the current version, Kekulé, nonetheless, does a marginally good job in processing such structures with the most frequent source of errors being interpretation of the characters. By adjusting internal parameters and retraining the OCR for hand-drawn characters, it is expected that hand-drawn structures can be interpreted reliably—the major caveat being that only block-printed characters are likely to be recognizable, not cursive script characters.

Additional work to be performed will expand the number of structure formats that Kekulé can write (currently ISIS, MOLfile, SMILES, ROSDAL, and Kekulé's native format) and read (Kekulé's, ISIS, MOLfile, and ROSDAL).

SUPPLEMENTARY MATERIAL

Five tables showing the Kekulé connection table for Figure 6 and the MOLfile, ISIS, SMILES, and ROSDAL formats for Figure 6 (37 pages). Ordering information is on any current masthead page.

REFERENCES AND NOTES

(1) Trademark of Molecular Design Ltd, San Leandro, CA.

(2) Daylight Chemical Information Systems, Inc., Irvine, CA.

(3) Springer-Verlag New York, Inc., New York, NY.

(4) Fein-Marquart Associates, Inc., Baltimore, MD.

(5) Springer-Verlag New York, Inc., New York, NY.

(6) STN International, Columbus, OH.

(7) Warr, W. A.; Wilkins, M. P. Front End Software for Chemical Structure Searching: A State-of-the-Art Review. *ONLINE* **1992**, *16* (1), 48–55.

(8) Named for Friedrich August Kekulé von Stradonitz (1829–1896), who invented chemical structure notation as well as elucidated the structure of benzene.

(9) Rozas, R.; Fernandez, H. Automatic Processing of Graphics for Image Databases in Science. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 7–12.

(10) Contreras, M. L.; Allendes, C.; Alvarez, L. T.; Rozas, R. Computational Perception and Recognition of Digitized Molecular Structures. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 302–307.

(11) IBM Almaden Research Laboratories is reported to have a program similar to Kekulé, but no published reports exist.

(12) Eastman Research Laboratories, Rochester, NY, in a private communication, reported having some portions of the capabilities necessary to assemble a program similar to Kekulé.

(13) Fraser-Williams (Poynton, Cheshire, England) has a program for converting structure images having a constant format.

(14) Trademark of Molecular Design Ltd, San Leandro, CA.

(15) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.

(16) Welford, S. *ROSDAL Manual for Users of the Beilstein Database at Dialog*; Springer: New York, 1989.

(17) Control Data Corp., Landover, MD.

(18) Tagged Image File Format—standard for storing images developed by Aldus, Microsoft, Hewlett-Packard.

(19) Numerous sources are available. A good reference is Pavlidis, T. *Algorithms for Graphics and Image Processing*; Computer Science Press: Rockville, MD, 1982; pp 201–206.

(20) Rosenfeld, A.; Kak, A. C. *Digital Picture Processing*; Academic Press: Orlando, FL, 1982; Vol. 2, pp 121–126.

(21) Kasturi, R.; Alemany, J. Information Extraction from Images of Paper-Based Maps. *IEEE Trans. Software Eng.* **1988**, *15* (5), 671–675.

(22) Govindan, V. K. Character Recognition—A Review. *Pattern Recognit.* **1990**, *23* (7), 671–683.

(23) Merck & Co., Inc., Rahway, NJ.

(24) Hu, M. K. Visual Pattern Recognition by Moment Invariants. *IRE Trans. Inf. Theory* **1962**, *2*, 179–187.

(25) Khotanzad, A.; Hong, Y. H. Invariant Image Recognition by Zernike Moments. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12* (5), 489–497.

(26) Teague, M. R. Image analysis via the general theory of moments. *J. Opt. Soc. Am.* **1980**, *70* (8), 920–930.

(27) Persoon, E. Shape Discrimination Using Fourier Descriptors. *IEEE Trans. Syst. Man, Cybern.* **1977**, *SMC-7* (3), 170–179.

(28) Zahn, C. T.; Roskies, R. Z. Fourier Descriptors for Plane Closed Curves. *IEEE Trans. Comput.* **1972**, *C-21* (3), 269–281.

(29) Korpel, A. Gabor: frequency, time, and memory. *Appl. Opt.* **1982**, *21* (20), 3624–3632.

(30) Reber, W. L.; Lyman, J. An Artificial Neural System Design for Rotation and Scale Invariant Pattern Recognition. *Proc. IEEE 1st Int. Conf. Neural Networks* **1987**, *4*, 277–283.

(31) Lashas, A.; Shurna, R.; Verikas, A.; Dosinas, A. Optical Character Recognition Based on Analogue Preprocessing and Automatic Feature Extraction. *Comput. Vision, Graphics, Image Process.* **1985**, *32*, 191–207.

(32) Kahan, S.; Pavlidis, T.; Baird, H. S. On the Recognition of Printed Characters of Any Font and Size. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *PAMI-9* (2), 274–288.

(33) Sources include *ACS Nomenclature Guide*; *Aldrich Handbook of Fine Chemicals*; *Alfa Catalog*; *CTFA Cosmetic Ingredient Dictionary*; *CRC Handbook of Chemistry and Physics*; *AHFS Drug Information*; *Eastman Organic Chemicals*; *Analytical Reference Standards and Supplemental Data for Pesticides and Other Organic Compounds*; *Handbook of Reactive Chemical Hazards*; *Handbook of Existing Chemical Substances*; *Identification and Analysis of Organic Pollutants in Water*; *Desk Book of Infrared Spectra*; *Nomenclature of Organic Chemistry*; *Kodak Laboratory Chemicals*; *Merck Index*; *EPA/NIH Mass Spectral Data Base*; *Modern Synthetic Reactions*; *National Institute on Drug Abuse Research Monograph*; *Nomenclature of Organic Compounds*; *Structure Indexed Literature of Organic Mass Spectra*; *Organic Chemistry*; *Parent Compound Handbook*; *A Primer of Drug Action*; *Pesticide Index*; *Reagents for Organic Synthesis*; *Spectrometric Identification of Organic Compounds*; *Encyclopedia of Chemical Terminology*; *Toxic and Hazardous Industrial Chemicals Safety Manual*; *USAN and the USP Dictionary of Drug Names*; and *CA Index Nomenclature Workbook*.

(34) We imposed a limit of 30 s for editing to include a structure in the prior group of 98.9%. This was an arbitrary limit, and all but one of the five structures failing this test could be corrected in no more than 60 s.

(35) *The Merck Index*; 11th ed.; Merck & Co., Inc.: Rahway, NJ, 1989; p 181.