LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT
TECHNISCHE UNIVERSITÄT MÜNCHEN TUM

Institute for Computer Science
Database and Information Systems

Diploma Thesis
in Bioinformatics

Chemical Structure Recognition via an expert system guided graph exploration

Peter Kral

Supervisor 1:	Prof. Dr. Hans-Peter Kriegel
Supervisor 2:	Dr. Marc Zimmermann
Submission date:	15.03.2007

SCAI
Fraunhofer Institute
Algorithms and
Scientific Computing

Erklärung

Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

. . . . . .

Sankt Augustin, den 15.03.07

Contents

Abstract

The goal of this thesis was to develop and implement a new concept for the automatic reconstruction of chemical structural formulas. Although most of the time chemical drawings are generated with the computer, their machine-readable format gets lost by the publication process. For that reason an automatic reconstruction method is required to close the gap between the increasing amount of non machine-readable depictions and the requirement of molecule representations in the field of chemoinformatics. For this purpose the existing reconstruction software chemoCR of the Fraunhofer-Institute for Algorithms and Scientific Computing (SCAI) was analyzed and its bottlenecks identified.

A new concept was developed which took into account previous findings. The designed reconstruction method represents a completely new technique for the automatic reconstruction of structural formulas. The symbols which are contained in a chemical depiction are represented by a newly developed orientation graph which describes the spatial arrangements between the symbols. The identification of the chemical components of the molecule is accomplished by an expert system which has been specifically developed for this purpose; it applies a newly defined rule language. Here a new kind of constraint based graph exploration is applied for the processing of the symbols.

By implementing a prototype which recognizes two semantic elements (e.g., superatoms and chirals) it was originally planned to provide evidence that the newly developed technique works correctly. As a result of the beneficial concept the program already recognizes all elements which are also discovered by chemoCR. In contrast to that, the created approach is minimally dependent on parameters and, with the help of the new expert system, allows chemical knowledge to be considered in the recognition process. With that it is possible to, in a knowledge-based manner, resolve situations during the recognition process for which there were no clear solutions before. In addition, the complexity of the created method has been reduced considerably by the introduction of the standardized knowledge representation and the application of the orientation graph. This is essential for developing a transparent and extendable reconstruction proceeding which is able to keep pace with the permanently growing structural formula space.

The method has been evaluated by a chemical depiction test set. This evaluation has shown that the method is able to correctly reconstruct most of the molecules by applying only a small set of recognition rules.

Zusammenfassung

Ziel dieser Diplomarbeit war es ein neues Konzept für die automatische Rekonstruktion von chemischen Strukturformeln zu entwickeln und implementieren. Obwohl chemische Zeichnungen meistens am Computer generiert werden, geht ihr maschinell lesbares Format durch den Publikationsprozess verloren. Deshalb ist ein automatisches Rekonstruktionsverfahren notwendig, um die Lücke zwischen der immensen Anzahl von nicht maschinell lesbaren chemischen Abbildungen auf der einen Seite und der Notwendigkeit von lesbaren Molekülrepräsentationen in der Chemoinformatik auf der anderen Seite schließen zu können.

Dazu wurde zunächst die bestehende Rekonstruktionssoftware chemoCR des Fraunhofer-Instituts für Algorithmen und Wissenschaftliches SCAI analysiert und dessen Probleme aufgedeckt. Unter Berücksichtigung der daraus gewonnen Erkenntnisse wurde ein neues Konzept entwickelt.

Dabei stellt das in dieser Arbeit entwickelte Verfahren eine vollkommen neue Vorgehensweise bei der automatischen Erkennung von Strukturformeln dar. Die Symbole innerhalb eines zu rekonstruierenden Bildes werden zunächst durch einen neu entwickelten Orientationsgraphen repräsentiert, der die räumliche Anordnung zwischen den Symbolen beschreibt. Die Identifikation der chemischen Bestandteile des Moleküls erfolgt durch ein dafür speziell entwickeltes Expertensystem mit einer eigens definierten Regelsprache. Hierbei kommt bei der Prozessierung der Symbole eine neue Art von bedingter Graphtraversierung zum Einsatz.

Ursprünglich sollte anhand eines Prototypen, der zwei semantische Elemente (z.B. Chirale und Superatome) erkennen kann, gezeigt werden, dass das entwickelte Verfahren funktioniert. Das neue Konzept erwies sich jedoch als so vorteilhaft, dass das entwickelte Programm bereits alle Elemente erkennen kann, die auch durch chemoCR abgedeckt sind. Im Unterschied dazu ist es kaum abhängig von Parametern und erlaubt aufgrund des eingeführten Expertensystems das Einbringen chemischen Wissens in die Erkennung. Damit ist es möglich, Situationen wissensbasiert aufzulösen, für die sich beim Erkennungsprozess keine eindeutige Lösung ermitteln ließ. Zusätzlich konnte die Vereinheitlichung der Wissensrepräsentation und die Einführung des Orientationsgraphen die Komplexität des Verfahrens gravierend reduzieren. Dies ist vor allem relevant unter dem Apekt, ein transparentes und erweiterbares Vefahren zu entwickeln, das dem permanent zunehmenden Strukturformelraum gewachsen ist.

Die entwickelte Methode wurde auf einem Strukurformel-Datensatz evaluiert und konnte bereits mit wenigen Erkennungsregeln zuverlässig einen Großteil der Formeln korrekt rekonstruieren.

Chapter 1

Introduction

1.1 Motivation

In chemistry the communication of molecules is mainly based on images. The theoretical inherent chemistry of a chemical compound and its molecular properties can invariably be best explained through a pictorial representation of the molecular topology. That is why molecules are visualized in scientific literature in the area of chemistry, patent specifications and internet websites through two dimensional structure diagram pictures (compare 1.1). Although these images were generated by chemical drawing software tools, the computer readable molecule structure information is getting lost through the process of publication. Originally a chemist feeds a corresponding tool with all information about the molecule's atoms, bonds and their spatial arrangement. All these properties are stored in a suited textual chemical file format. If the compound should be published, this file is converted into a graphical file format, like bmp, jpg or gif and is placed into the publication. Through this irreversible transformation, the molecule information containing the structure and composition is reduced to a raster image representation. Instead of a formal molecule description it is now only a collection of pixels and their color intensities. This format can be easily interpreted by a human but a computer cannot deal with it. On the other hand the enormous amount of chemical data generated

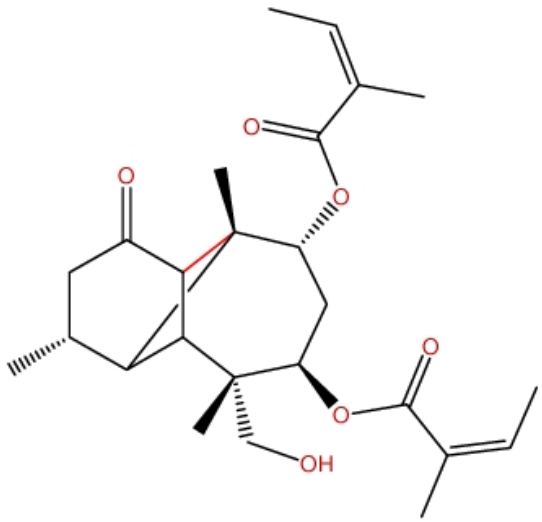


Figure 1.1: Example of a structural formula

by the increasing automated experiments in drug research requires the analysis by computers and informational techniques. The chemical research is already largely impacted by chemoinformatic methods in areas such as data visualization, knowledge discovery, exploratory data analysis, machine learning, simulation and prediction.
All these techniques require machine readable molecule representations. To obtain the full potential of these methodologies it is now essential to close the gap between the massive amount of non-machine readable chemical depictions and the requirement for computer processable molecules in the field of chemoinformatics. This task is called *chemical structure reconstruction (CSR) problem* and can be addressed through different kinds of strategies.

Avoiding the information loss through the publication in advance would be the most obvious and intuitive way. If the release media itself is nondigital, this is difficult to reach. Otherwise most publications about recent scientific developments are mainly published in a digital form, like pdf or postscript format. There exist a few approaches in this electronic publishing field, which try to maintain the full range (image and computer representation) of chemical information embedded in the document. That is realized by enriching the release formats through a chemical metainformation language like CML[1] [35]. Similar proceedings already exist in the computer aided design (CAD) field, where technical drawings can be placed into a document[2]. Such extensions might affect future electronic publications and presume the thoughtful usage of these enrichment techniques by the author. The often not computer affine chemists have to detach themselves from the impression that only the chemistry research community is interested in their scientific results. The current situation is that already now and supposable in the closer future, handwritten drawings, print media like journals, books, digital documents and websites exist, which are full of non computer processable molecules relevant for the field of chemoinformatics.

Another solution is to manually redraw the molecules of interest. This approach has several disadvantages. Manual redraw would come up with a duplication of effort in the sense that the user works from an existing generated drawing. Even a trained operator requires appreciable time to reproduce a digital format of a chemical depiction. This process is error-prone and in large scale only applicable with an enormous financial and temporary effort. Because of this, manually maintained databases, like the Beilstein database[3], are very expensive and often not in the reach of academic research.
It is important to realize that the CSR problem is related to different applications. The organisation of new molecules in databases to keep track of the evolution of the chemical space is without a doubt one of the most important topics. In addition new scenarios might require on-the-fly extraction and reconstruction of molecules and entire reaction schemes.

A new adequate answer dealing with all of these requirements would be an automatic reconstruction software for chemistry similar to the well established Optical Character Recognition (OCR) software which converts typewritten text into machine-editable text. For that purpose the Institute for Algorithms and Scientific Computing[4] of the Fraunhofer

[1] http://www.xml-cml.org/
[2] http://www.adobe.com/svg/
[3] http://www.mdl.com/products/knowledge/crossfire_beilstein/
[4] http://www.scai.fraunhofer.de/

society in Sankt Augustin developed a software called *chemoCR* [33]. It allows the automatic reconstruction of the machine readable format out of a two dimensional chemical representation, regardless if it is embedded in print or electronic media.

1.2 Goal of the thesis

The current chemoCR version is already able to reconstruct a full molecule, covering the most common semantic elements like atoms and different types of bonds. The software works well for diverse chemical depiction test sets, which were partly already available from the beginning of the project. In the meantime a lot of new chemical images were collected from different resources where chemoCR has difficulties in the reconstruction. These drawings contain new chemical semantic elements and often contain variants of the already covered ones. The original conceptual design planned to implement for each chemical semantic element an isolated recognition module. Because a growing chemical space was assumed from the first, a simple updating, modifying and extending of these modules was a basic part of the chemoCR recognition strategy. All modules should work independent from each other to grant exchangeability and modularity. If a certain semantic element cannot be identified correctly, then the error should be backtrackable to the responsible module. Unfortunately this clean module separation and the associated mentioned advantages were not ensureable after certain unavoidable dependencies between the individual recognizers were realized. Several significant bottlenecks of the recognition proceeding, which were not predictable at the stage of the conceptual design led to the requirement of an extension of the current strategy or a conceptual redesign.
This thesis exhibits a producibility study about a completely new way of addressing the CSR problem which does not possess the bottlenecks of the current chemoCR. For that purpose, a new concept for chemical structure reconstruction was developed and implemented.

1.3 Existing approaches

Although pattern recognition in images is not a new field, there exist only a few approaches dealing with the CSR problem. The first projects which addressed the topic of the reconstruction of 2D molecule representations started in the 1990s. In the following a short overview about the important existing approaches in this domain is given.
One of the first projects was Kekulé [34] developed by McDaniel et al. in 1992. The algorithm was mentioned in literature but did not lead to an available software. Because the program was not thought as a full automatic system it involves manual human intervention in the recognition process. Kekulé's recognition results of simple small molecules were promising and required less postprocessing. On the other side, it was nearly impossible to reconstruct more complex molecules and/or images with a lower ($<$ 300 dpi) resolution.
Another project which deals with the CSR problem is CLiDE [24] (Chemical Literature Data Extraction) which was also started in the early 1990's at the University of Leeds, England. This approach is available as the only commercial chemical structure reconstruc-

tion software, distributed from the Canadian company Symbiosys Inc[5]. The recognition accuracy of CLiDE was evaluated in a former master thesis [19] advised from the SCAI Fraunhofer Institute. For that purpose CLiDE was applied on a test set of the 100 most sold drugs of the year 2002. Diverse image sets were generated, which differ in their image quality, bond width and typeface. The reconstructed molecules often contain several kinds of misclassifications. For example, a common error was to interpret a Chlorine (Cl) atom as carbon with an ingoing single bond. The input files often had to be optimized in a time consuming step to better fit into the tool's requirements. In summary the evaluation study has shown that the accuracy rate of CLiDE is not high enough for large scale use.

Two further approaches exist in addition to CLiDE and Kekulé.
The group around Jean-Yves Ramel [39] developed a system for the localization and recognition of graphical entities in handwritten chemical formulas. According to its publication in 1999 the group used a relative small test data set (20 documents) for evaluation and listed the text recognition rate and the graphic part recognition rate separately from each other. On that small test set the individual recognition rates were close to 93%, or rather 97%.
Another prototype [42] was created by the IBM Almaden Research Center in San Jose, USA California (1993). This pattern recognition process follows a sequence of steps. It detects lines and determines their interrelations, recognizes geometric shapes, distinguishes printed characters and encodes them by means of OCR. It is mentioned that "the algorithm is accurate for simple planar diagrams, although large-scale tests have not been conducted to characterize performance" [42].

Besides the identification of the individual symbols, the CSR problem mainly demands the correct assembly to the whole molecule. Accuracy rates concerning the individual chemical elements (e.g. the atom characters) within a molecule do not reflect the accuracy with which an entire molecule can be reconstructed.
This is why it is difficult to make statements about the reconstruction power of the cited tools. Either the evaluation test sets were too small to allow significant conclusions or the tests were focused on the accuracy to identify individual elements or even no theoretical analysis of their performances have been published so far.
Although the IBM workflow led to a patent[6][U.S. Pat. No. 5,157,736], it seems that no more endeavors were invested in this project. The existence of only one available recognition software and its relatively bad evaluation results was the reason to start the chemoCR project at the SCAI Fraunhofer Institute in Sankt Augustin.

[5] http://www.simbiosys.ca/clide/
[6] http://www.patentgenius.com/patent/5157736.html

1.4 Document outline

The CSR problem and the application of chemoCR can be assigned to the field of chemoinformatics. A short overview into this topic is given in the second chapter. Additionally, to the main tasks of chemoinformatics, it also explains how structural formulas emerged and how molecules are identified and described. The reconstruction of chemical depictions requires different branches of image processing. Chapter three provides a brief overview about the involved topics of this field, Segmentation, Vectorization and Optical Character Recognition. Starting point of this thesis is the existing reconstruction software chemoCR, which is analyzed in chapter four. Chapter five presents the new reconstruction approach and explains developed concepts. Here the new orientation graph (see 5.1.2), the developed expert system (refer to 5.2.2) and the so-called constraint based graph exploration (see 5.3.2) are detailed and their motivation elucidated. The achieved results and the evaluation of the novel technique are shown in chapter six. Finally the thesis ends with a discussion about the new reconstruction concept and an outlook is given.

Chapter 2

Chemoinformatics

The algorithmic problem of the automatic chemical structure reconstruction falls into the field of chemoinformatics. Various chemists already used computer methods in the past [46] to manage and make sense of their chemical data. For that reason the relatively new discipline of chemoinformatics [2] actually emerged from several older domains such as computational chemistry, computer chemistry, chemometrics, QSAR, etc. Although several varying definitions exist, which focus on different branches of chemistry, a relatively general one is listed here.

Definition[1]:
'Chem(o)informatics is a generic term that encompasses the design, creation, organization, management, retrieval, analysis, dissemination, visualization, and the use of chemical information.'

Today chemical experiments are mainly performed by applying novel high-throughput screening (HTS) methods [29], based on miniaturization and automation techniques. HTS allows chemists to conduct simultaneously millions of biochemical, genetic or pharmacological tests in a short time period. The amount of the resulting chemical data is enormous and increases rapidly. More than 45 million chemical compounds are known at the moment and this amount is still growing several millions each year. Before data can be transferred into information and information into knowledge, additional considerable technological effort has to be prosecuted. All these results can only be managed by storing them in databases and process them with chemoinformatic methods. Diverse computer techniques were developed to describe chemical compounds in an abstract manner. Being able to map chemistry onto the *in silico* world requires an elementary understanding of the chemistry concepts.

2.1 Introduction to chemistry

Chemistry is the science of matter at the atomic to molecular scale. It primarily deals with collections of atoms like molecules, their properties, as well as their transformations and interactions to form materials encountered in everyday life. Often this discipline is called the "central science" because it connects other sciences, such as physics, material science, nanotechnology, biology, pharmacy, medicine, bioinformatics and geology.
A major task of a chemist is to find or develop new molecules with certain desired properties

[1] G. Paris (August 1999 Meeting of the American Chemical Society)

(Structure-Property Relationship) and to design the corresponding synthesis reaction for it. Chemistry knowledge was gathered by learning from data and experiments. Often the learning process is based on observing common features in experiments, generating a hypothesis about specific molecule and finally confirm, refine or reject the formulated models.
The emergence of the atomic theory in the 19th century helped to gain important insights and simplified the imagination and representation of chemical elements. An element is a class of atoms which have the same chemical properties. The most convenient presentation of elements is based on the periodic table, which contains symbol abbreviations for each known chemical element. An atom is the smallest particle of a chemical element that retains its chemical properties. Atoms of chemical elements itself can form in a fixed ratio composition and particular organization, molecules with new specific chemical features. The particular organization depends on which atoms of the molecule are interconnected through bonds, formed by overlapping atomic orbitals of the participating atoms. Today these insights are used to clearly name and identify molecules. This was not always the case.

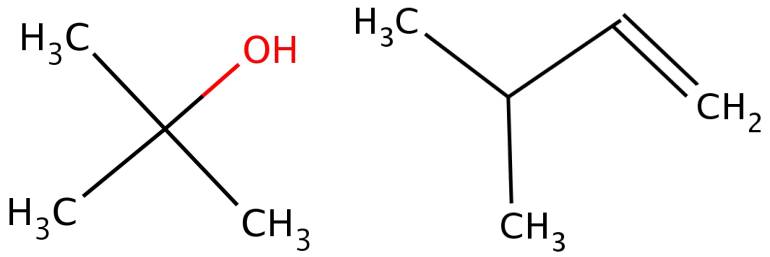
2.2 Identification of molecules

At the beginning of chemistry, chemicals were named according to their origin, properties, or application. Still today these trivial names are often used because they are short and easy to memorize. Besides the lack of a systematic naming proceeding, the main disadvantage of these identifiers is the missing information about the internal chemistry, e.g. the atom ratio or the inherent spatial arrangement of the molecule.
As the number of known organic compounds rapidly increased by the end of the 19th century, there was an urgent need for a systematic nomenclature, which generates names that convey information about the chemistry of a compound. A nomenclature in general, is a system of naming and categorizing objects in a given category. In chemistry a nomenclature has the aim to ensure the unambiguous identification of a specific molecule by its name.
In general the development of a widely accepted nomenclature system requires to consider certain aspects. For minimizing errors, the naming system must be intuitive to learn, simple to apply and interpret. Beyond this it should be flexible to cover all chemistry principles, unique and unambiguous in both directions. So a name should be derivable from its molecule and vice versa.
Scientific efforts resulted in diverse systems [21], which are the basis for identifying, classifying, and naming chemicals. The first known systematic naming system was the so called nomenclature of Geneva (1892). It defines an exact set of rules how to write a chemical formula (also called molecular formula). The molecular formula is a concise way of expressing information about the absolute number of element atoms that participate to a particular chemical molecule. This string representation is shortened through a symbol-based representation. Each atom is typified by its atomic symbols from the periodic system. In addition, the Hill system describes the order of the atoms within a molecular formula, e.g. the number of carbon atoms in a molecule is indicated first, then the number of hydrogen atoms are listed followed by the number of all other chemical elements subsequently, in alphabetical order.

This first systematic naming approach was a step towards more standardization, which was commendable but not sufficient. Scientists realized soon that different molecules can form the same chemical formula (i.e. isomerism problem). Thus, the original form of the nomenclature was not applicable.

2.2.1 IUPAC nomenclature

The way out of that misery was based on new experimental methods which allowed insights of molecule structures. These results led to the concept of two dimensional molecular diagrams, which make molecules more conceivable. Different molecules with the same molecular formula differ in their structural bodies. The deficits of this simple representation can be corrected by enriching it with molecule specific structural units (compare fig 2.2.2). This modified Geneva Nomenclature was developed by the Commission on Nomenclature of Organic Chemistry [21], formed by the IUPAC[2] (International Union of Pure and Applied Chemistry), an international non-governmental organization established in 1919. This IUPAC nomenclature naming system is still in use today and new updates are presented regularly. The inference of a molecule's name starts with determining the longest hydrocarbon skeletal structure of the molecule and identifying all functional groups in the molecule that distinguish it from the parent hydrocarbon (compare figure 2.1). The numbering of the carbons in the main chain is done in such a way that the functional groups belong to small enumerated carbons. The name is generated from the main chain-functional group connection points, followed by the functional group names and finally ends with the name of the hydrocarbon skeletal structure. Although this line notation nomenclature is now systematic, includes stereochemistry and allows the reconstruction of the structural formula, it exhibits several disadvantages. The generated names are complicated, alternative correct names might exist and the application of the complex system makes it prone to errors.



2-methylpropan-2-ol 3-methylbut-1-ene

Figure 2.1: Examples of IUPAC representation codes

2.2.2 SMILES nomenclature

An alternative more recent approach to communicate molecules is the SMILES notation [51]. SMILES is the abbreviation for Simplified Molecular Input Line Entry Specification and was originally specified by Arthur and David Weininger in the late 1980s. Like its name already promise SMILES is relatively simple to understand and only a few rules

[2] http://www.iupac.org/

are required to create and interpret most of the SMILES strings.
Atoms are represented by their atomic symbols. Because SMILES is a hydrogen suppressed notation, no hydrogen atoms are listed in the SMILES string (compare fig 2.2). Single bonds do not have an own symbol representations. Double bonds are typified by '=' and triple bonds by '#'. To determine a SMILE string, the molecule graph is traversed in such a manner, that each vertex is just visited once. Rings are described by allocating digits to the connecting ring atoms. For that purpose these connecting atoms are broken up and marked with the same digit. If a branch point in the molecule is arrived, parentheses are introduced. One use a left-hand bracket which symbolizes a new branch and a right-hand bracket which indicates that all atoms in that branch were visited. Branches itself can be nested to any level necessary.

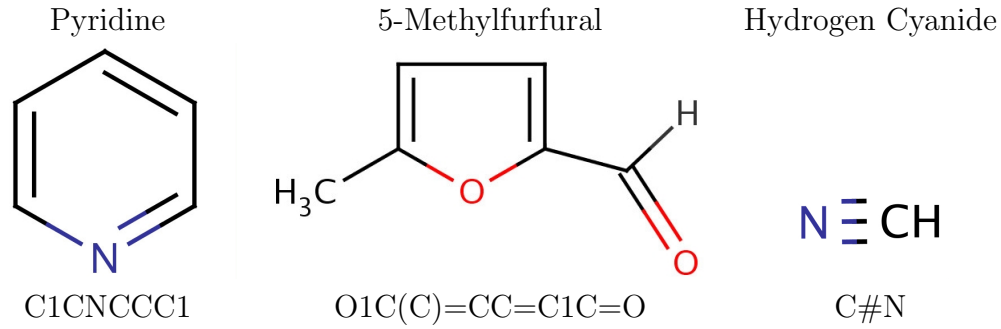


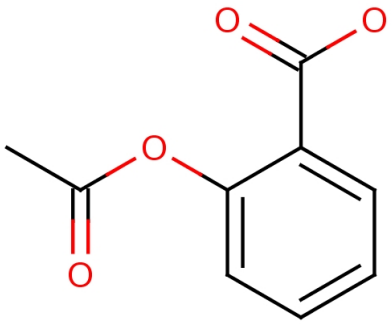
Figure 2.2: Examples of SMILES representation codes

This simplicity of SMILES led to a widespread acceptance as a universal line notation nomenclature for the representation of chemicals. In the meantime the notation has been extended several times, most notably by Daylight Chemical Information Systems Inc[3]. Various forms of the SMILES notation were formulated, differing in their complexity. E.g Unique SMILES refers to a specification that includes rules for ensuring that each distinct molecule has exactly one single unique SMILES representation.

2.3 Computer representation

For the computer based processing of chemical data special approaches to represent, store and retrieve structures are required. After giving a brief introduction about the identification of molecules in section 2.2 the following deals with their machine-readable representation. Between both topics there is a clear overlap. For the well-defined naming of a molecule it's structural properties have to be considered. Although SMILES and IUPAC already exhibits an own representation possibility several other techniques have been developed. They mainly differ in their information content. Which of them to select depends on the application context in which a molecule is used. Standardized representation forms are essential for the exchange of chemical structure information. Molecules have to be submitted to databases, the stored data must be accessible to chemistry software systems and these systems have to be able to transfer data between each other.
At first major efforts were focused on fast data exchange and human readability. For

[3] `http://www.daylight.com`



Trivial name:	Aspirin
Empirical formula:	C9H8O4
Condensed formula:	C6H4(OCOCH3)COOH
IUPAC name:	2-acetyloxybenzoic acid
SMILES name:	CC(=O)OC1=CC=CC=C1C(=O)O

Figure 2.3: Identification of Aspirin

that reason especially the line notations like SMILES, which encode chemical structures as terms (compare figure 2.2.2), experienced a widespread acceptance as a chemical representation form.
When the chemoinformatic analysis tasks became more complex, the strategy to represent structures as alphanumerical strings for computer processing was no longer suited. Today SMILES is mainly used for molecule identification and as query language for textual chemist-computer interaction to find entire molecules or substructures within molecules.

2.3.1 Molecules are 3D objects

Initially, it was assumed that a planar molecule representation is sufficient to address the tasks of chemoinformatics. The German organic chemist August Kekulé (1829–1896) already postulated that molecules are complex three dimensional objects. For representing the full complexity of a molecule, a two dimensional representation form is too simplifying and restricting in its expressivness. A planar chemical representation only explains which atoms are interconnected (topology) but not the 3D spatial arrangement (topography) of the atoms. On the other hand many physical, chemical and particularly biological properties of a molecule are determined to a large extent by its three-dimensional structure. In addition, a 3D structure of stable compounds can be obtained by the current experimental methods, like X-ray crystallography, microwave spectroscopy, electron diffraction, or NMR spectroscopy.

2.3.2 Molecular graph based representation

To cover the full three dimensional information content of a molecule, more detailed representation forms were developed. Today molecules are usually stored in a computer as molecular graphs [6].

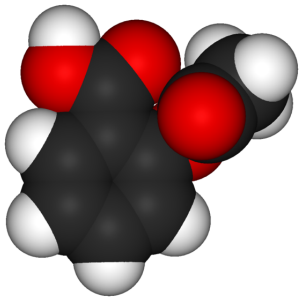


Figure 2.4: 3D structure of Aspirin

Graph theory

From the earliest times, chemists tried to develop visualization techniques to represent their imagination of a molecule. The Scottish chemist William Cullen already started in 1758 to use so-called "affinity diagrams" to represent the supposed forces, existing between pairs of molecules undergoing various chemical reactions. Then, during the 19th century, the way of thinking in chemistry was mainly affected by Newtonian ideas, especially those pertaining to the internal structure of matter and the short range forces existing between particles.

The first attempts were performed to study the spatial arrangement of atoms in molecules. Pioneering work by Dalton and Wollaston led to a greatly improved understanding of the relationships between atoms in space. Based on these newly gained insights new graph-oriented diagram models were developed to make molecules more conceivable. Graph theory [15] represents a very natural formalism for chemistry and has already been applied to a variety of fields. Graph theory is one of the few branches of mathematics that are said to have had a precise starting date. In 1736 Euler's solution of the Koenigsberg bridge problem is considered to be the first theorem of graph theory[4][5].

Graph definition:
An undirected graph G is an ordered pair G: = (V, E)
that is subject to the following conditions:

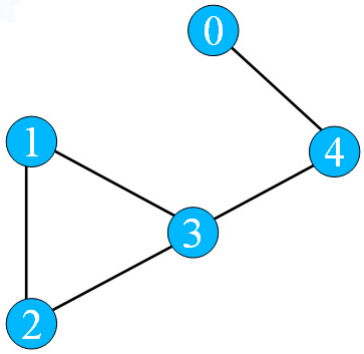
- V is a set of vertices or nodes
- E is a set of pairs (unordered) of distinct vertices, called edges or lines
- The vertices belonging to an edge are called the ends, endpoints, or end vertices of the edge.

[4] http://math.dartmouth.edu/~euler/docs/originals/E053.pdf
[5] http://www.jimloy.com/puzz/konigs.htm

Various mathematical formalism exist to clearly describe a labeled graph.
One very common technique is the use of so-called adjacency matrices (compare figure 2.5) .

Adjacency matrix definition:
Let G be a graph with n vertices that are assumed to be ordered from v_1 to v_n.
The $n \times n$ matrix A, is called adjacency matrix with

- $a_{ij} = 1$ if there exists a path from v_i to v_j
- $a_{ij} = 0$ otherwise



$$A = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Figure 2.5: Labeled undirected graph and its corresponding adjacency matrix

Based on this graph theory, a formal concept for molecule representation [6] can be introduced. A molecule graph [21] is a graph-based description of a molecule, where the vertices are representing the atoms and the edges representing the bonds. Each vertex is labeled by the type (the name of the corresponding element) and each edge has a non-negative weight label, describing the order of the bond (0 for a non-existent bond, 1 for a single bond, 2 for a double bond, and 3 for a triple bond).

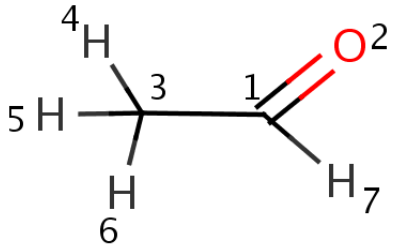
Molecular Graph definition:
A molecule graph is a weighted undirected graph (V, E, w), without multiple edges or self-loops and a weight function w : E $\rightarrow$ N. The nodes are labeled by chemical elements and the valence of a node in a chemical graph is the total weight of the edges incident to it.

Today the concept of molecular graphs is used in all major branches of chemistry and can be considered as natural language between chemists. Because of their formal description possibilities, they were the starting point for the application of computers in structural chemistry. Numerous graph based file formats were developed, which allow the communication of molecules to and from a computer. Although these formats are as a result of the abstract mathematical formalism indeed less readable for a human they are simple and efficient to store and to visualize for a computer. Usually chemists do not have direct contact with this abstract representation. Molecules are drawn in simple

wysiwyg[6] chemical structure drawing tools (ChemDraw[7], ISIS/Draw[8]), which generate the corresponding abstract molecular graph in the background, which can then be stored. In the following the two most common formalism are presented.

Molecular adjacence matrix

Corresponding to the adjacence matrix definition for general graphs, molecular graphs can be described in a similar way. For n atoms a square (n × n) matrix is created, containing all necessary entries to describe all bonds between these atoms. Different enrichments [21] led to a set of new matrices like Distance Matrix, Incidence Matrix, Bond-Matrix and Bond-Electron Matrix. A Bond matrix e.g. is related to the adjacency matrix but gives also information about the bond order of the connected atoms. Elements of the matrix obtain e.g. the value 2 if there is a double bond between atoms, e.g. between atoms 1 and 2 in the shown example 2.6. Otherwise the value can be 0, 1, or 3 for other bonding combinations.



$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 2 & 1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Figure 2.6: Ethanol molecule and its corresponding bond matrix. The number between matrix element i and j shows if there is a bond between atom i and atom j (the number indicates single (1), double (2), triple (3) or no bond (0)).

Connection tables

A significant disadvantage of the matrix representation is that the number of entries increases with the square of the number of atoms in the molecule. For that reason so-called connection table formats [21] prevailed, where the number of entries only increases as a linear function of the number of atoms. This is achieved by listing, in a tabular form only the atoms and bonds which are really present in the molecule. This kind of formats consist of at least two sections. The first contains all indices labeled atoms and their space coordinates. The second one holds all bonds specified as pairs of atom labels. Both tables are linked together by the atom indices.

After various organizations tried to establish own proprietary formats, finally the connection table based MDL Molfile[9] [13] and its derivates (SDF, RDFile) became a de facto

[6] wysiwyg: what you see is what you get
[7] `http://www.cambridgesoft.com/software/ChemDraw/`
[8] `http://www.mdli.com/downloads/`
[9] `http://www.mdli.com/`

standard file format (compare figure 2.7). Most of the chemoinformatic applications can read and create these standard formats. Because there also exists many other formats (Mol2, PDB, CML, ...) the chemical structure conversion tool BABEL[10] can be used to create a required format out of 70 different formats.

```
2.5369   -0.2500    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
3.4030    0.2500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
4.2690   -0.2500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
3.8015    0.7249    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
3.0044    0.7249    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
3.9590   -0.7869    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
4.8059   -0.5600    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
4.5790    0.2869    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
2.0000    0.0600    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
1  2  1  0  0  0  0
1  9  1  0  0  0  0
2  3  1  0  0  0  0
2  4  1  0  0  0  0
2  5  1  0  0  0  0
3  6  1  0  0  0  0
3  7  1  0  0  0  0
3  8  1  0  0  0  0
```

Figure 2.7: SDF file of Ethanol (compare 2.6), consists of two blocks, first block contains atoms and their coordinates; second holds the connection table indicating which bonds are between which atoms (referenced through the position index)

2.4 Chemoinformatic applications

Once a molecule is formally represented various chemoinformatic tasks can be addressed. In the following two example application fields are presented.

2.4.1 QSAR

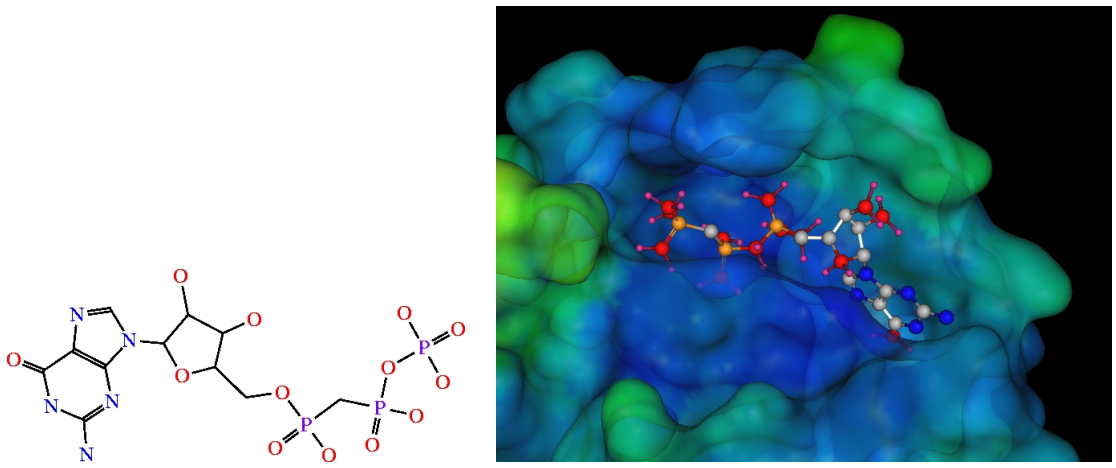
The identification of a new molecule with desired properties often follows a certain iterative scheme. It starts with a chemical compound which possesses an interesting biological/chemical profile. The chemist forms a hypothesis which assigns the chemical features of the molecule to the observed activity effect, without having any comprehension about the underlying inherent chemical process(es) responsible for that. This assumption is successively improved by analyzing the molecular structural similarities and differences for molecules which exhibit the certain effect or not. Usually the molecules which maximize the attendance of functional groups or chemical properties supposed to be responsible for the impact, are selected in each step. Computer based Quantitative Structure Activity

[10] http://openbabel.sourceforge.net/wiki/Main_Page

Relationship (QSAR) [38, 21] can be used to reduce this labor intensive effort. The QSAR theory represents an attempt to correlate structural/property descriptors of molecules with desired activities. If the molecule is available in a computer processable representation, these descriptors can be automatically inferred by the computer. These descriptors collect diverse physicochemical numerical features about hydrophobicity, molecular weight, logP, van der Waals volume, electronic properties and topology. QSAR has already been successfully applied to a range of disciplines, many of them concerning drug design.

2.4.2 Virtual screening

In drug design research the retrieval of new drugs is a very challenging and costly endeavor. Advances in molecular modelling, combinatorial chemistry and molecular biology will substantially influence this drug discovery process in the pharmaceutical industry. Through novel virtual screening (VS) methods [50] it is possible to assess huge chemical compound libraries in order to supervise the selection of drug like candidates. The underlying mechanism behind that is to predict binding affinities between small ligands, the potential drugs, and pharmaceutically interesting target proteins. For both groups the structure must be known and available as a machine readable format. Then a docking algorithm docks sampled molecules against the interesting targets to infer the emerging binding forces. The biological activity of a drug is mainly determined by this affinity value. Consequently VS is a kind of computer-aided filter reducing the number of potential candidates to be screened experimentally.



(a) ATP derivative ligand (b) Docking into an active pocket

Figure 2.8: Example for docking a molecule against a protein

While still limited, the possibilities of chemoinformatics have established a respected place within all branches of chemistry and continue to evolve. There are still many challenging chemical problems waiting to be supported or to be solved by further progresses in chemoinformatics.

Chapter 3

Image Processing

The reconstruction of chemical molecules out of digital images requires diverse algorithms from the image processing field. Although the first methods have been already developed in the 1960s, image processing is still a rapidly growing area of computer science. Its development came from several technological advances in digital imaging, computer processors and mass storage devices. The most popular examples can be found in the character recognition, medical imaging and satellite imagery area. New applications like video phoning became imaginable through the emergence of communication media like internet and mobile phoning. On the other hand the use of image processing algorithms to reconstruct chemical depictions is a relatively unexplored application. Here especially the extraction of coherent graphical objects, character recognition and the vector approximation of a given shape are of interest.

3.1 Segmentation

The human mind can easily divide the content of an image into coherent semantic units, like atom and bond symbols in a chemical depiction. There has been a tremendous amount of effort devoted to achieving a comparable level of performance in computer vision. Before symbols can be recognized, their underlying pixel composed units must be first extracted from the image background. This step is called image segmentation [16]. From these units, feature values can be derived, which are the starting point for further processing steps like pattern recognition and classification. Because all postprocessing steps ared based on the extraction quality, an optimal segmentation of images remains a challenging problem [28] in image processing. Although segmentation of binary images is trivial, grayscale and color images can be in contrast very complex (compare figure 3.1). Here the boundary of a symbol cannot always be defined by means of the gray values only. The trend of the variations of the gray values have to be taken into account.

(a) Original (b) Optimal (c) Not optimal

Figure 3.1: Binarization is an essential step. Distinct symbols are in one connected component if a wrong binarization threshold is selected.

Although chemical depictions can contain binary, grayscale or color information, they exhibit a relatively clean separation of the individual units. Therefore it is often sufficient to convert a grayscale or color image into a binary picture, based on a simple threshold approach. From there it is trivial to infer the desired isolated symbols: A binary image I contains only white (0) or black (1) pixels. $\{I : (x, y) \rightarrow \{0, 1\}\}$, where x and y are coordinates within the image raster.

The segmentation problem here is tackled often through a connected component labeling scan. All pixels of a raster image are traversed from left to right and top to bottom and grouped together into so called connected components (compare figure 3.2). A connected component C represents a maximal subset of black pixels, such that for any pair of that subset there is at least one path of black pixels between them. This can recursively be defined as: Any black pixel is itself a connected component and any black pixel adjacent to another black pixel is part of the latter's connected component. Two pixels are adjacent if they have a common edge or a common corner (compare equation).

Definition of a connected component C:

$$I(x_0, y_0) = 1 \quad \rightarrow \quad (x_0, y_0) \in C \quad (3.1)$$

$$I(x_1, y_1) = 1 \text{ and } (x_0, y_0) \in C \text{ and } |x_1 - x_0| + |y_1 - y_0| = 1 \quad \rightarrow \quad (x_1, y_1) \in C \quad (3.2)$$

$$I(x_1, y_1) = 1 \text{ and } (x_0, y_0) \in C \text{ and } |x_1 - x_0| = |y_1 - y_0| = 1 \quad \rightarrow \quad (x_1, y_1) \in C \quad (3.3)$$

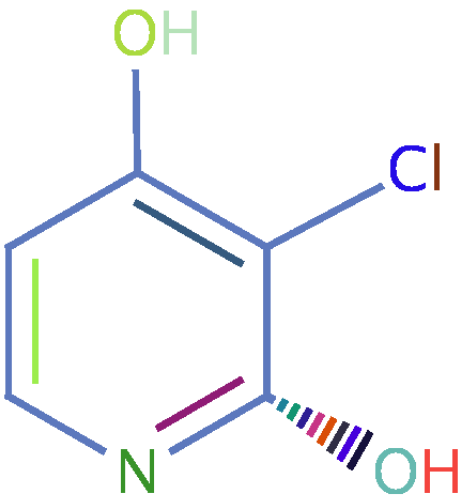


Figure 3.2: Molecule image after segmentation Each identified distinct symbols became a connected component, indicated by the different coloring.

Finally this extraction algorithm for binary images provides a result list of all identified symbols contained in the image. These connected components (denoted CC) are the starting point for all further image analysis tasks which are required in the interpretation procedure to reconstruct the molecule.

3.2 Optical Character Recognition

Molecule images contain different kinds of string oriented chemical symbols, like atoms, SMILES strings and superatoms. For this reason optical character recognition (OCR) algorithms [23] are required to correctly identify all contained character symbols. OCR is one of the most successful applications of automatic pattern recognition. It translates images of hand/type written text, usually captured by a scanner, into machine-editable text. OCR research and development started in the 1950s [44] and is still an active field today. The character recognition procedures find their applications in diverse fields, such as in the home and office use, forms processing (medical claim forms, bank checks), address reading (mail, express) and large scale conversion projects (patents, historical documents, books). The underlying recognition algorithms usually consist of several proceeding steps. After the noise reduction is completed, an image binarization is done. A segmentation algorithm similar to that already mentioned in the previous section, isolates the individual character symbols. These patterns are then recognized through one of the following approaches.

3.2.1 Feature based recognition

Oftentimes characters are identified by through featurebased methods [49]. For that purpose a suited set of numerical features (e.g. for the shape) is extracted from the segmented symbols. These features are the starting point for diverse classification techniques based on learning from examples. This class of methods includes different kinds of machine learning methods such as statistical methods [20], artificial neural networks [5], support vector machines [12, 8], etc. The extracted features are used to assign to each symbol a character label, which fits best into the inherent classification model. Optimal features minimize the within-class pattern variability while enhancing the between-class pattern variability. The selected attributes should be invariant to the expected distortions and variations that the characters may have in a specific application. Invariant implies, that features are independent to transformation operations like translation, scaling, rotating, stretching, skewing and mirroring. Finally, the recognition ends with a character concatenation and a contextual verification.

3.2.2 Template based recognition

Another recognition strategy is structural based [9] and involves template matching. Instead of a feature extraction step, the entire image is used as a feature. Each symbol C_i identified from the segmentation step is compared with all character templates T_j of an alphabet. A suited distance function d (e.g. mean square distance) is used to compute a similarity measure between a character symbol and all templates. The template T_k, which exhibits the highest similarity measure to the symbol, is identified. If this similarity is above a certain threshold, the character is assigned the character class label k. To encounter the invariant problem, different sizes of template characters may be used or the character symbols in the input image can be scaled to suit the template sizes.

3.2.3 OCR problems in the CSR field

A software addressing the CSR problem requires a high accuracy rate of the applied OCR algorithms. Reconstructing an entire molecule relies on identifying all atoms within the chemical depiction. But image defects and several other problems can occur, which might impact the OCR software and therefore the chemical reconstruction algorithm.
Already after the molecule generation with a chemical drawing tool, the first problems can be introduced. The sketched molecule is usually exported as a compressed image file, which is in addition visually optimized for the publication process. Here image processing steps like anti-aliasing are often applied to remove blocky patterns. The compression as well as the image optimization lead to an image information loss where pixels intensities may be modified. This can influence the segmentation and therefore the OCR results. In the worst case, e.g. a characters is completely pixel connected with a bond and cannot be correctly identified as a letter (compare also figure 3.1).
More resources of errors originate from imaging defects in scanned images. During and after the printing different kind of failures can be included. Porous paper causes the ink to spread, leading to a print result which is different to the original digitized characters. More noise arises through the use of copiers or just dirty surfaces. In addition even the scan process is not perfect in separating the print from the background because paper is not a high contrast medium. Hence, the characters of scanned molecule images can contain diverse handicaps, like fragmentation, merges, self touching or broken loops.
In contrast to print material, no physically introduced image defects appear in digital images. The difficulties here arise from other reasons. Digital images are often embedded in journals or patents. It sometimes happens that authors bulge parts of the shown molecules, which are less interesting or are situated at the outer region of the image. This leads to a character shape deformation or to problems similar to the imaging defects of scanned material.
But even if the quality is sufficient and no image defects are present, the OCR still might exhibit uncertainty about the correct classification of a character symbol. Amongst others, these algorithms often use the shape of a character and its associated features. Shape can be defined as outline or characteristic surface configuration of an object. Uncertainty often occurs where characters can not easily be distinguished based on their shape, e.g. the letters u and v, the character g and the number 6 or the symbols 'l', 'j', '(', ')', '/', 'I', '1' can cause confusions in the classification process.
In ordinary text, optical recognition software would effectively address this problems by applying diverse post processing steps. A well-established technique is to verify the concatenated string by searching the recognized word in a dictionary. It is accepted if the reference book contains it. Otherwise the most similar word match is identified. If the similarity between the two strings is over a certain threshold the recognized word is replaced through the corresponding dictionary match. The strings in a molecule are usually very short. Due to the fact there are often only one-character strings to identify, no contextual information like seen in the recognition in documents can be applied. Several character recognition algorithms may be used in parallel, to achieve nevertheless a high character accuracy rate. If there are more character suggestions for one symbol, then the character label with the highest confidences can be selected.

3.3 Vectorization

Bonds, beside atoms, are the most frequently used symbols in chemical images. Among other things, they hold the information which atoms have to be connected. Here it is important to infer a relative accurate determination where the bond is situated in the picture. In addition to the connecting information, bond sets can symbolize themselves a collection of atoms. Although an aromatic ring system contains several carbon atoms, it is represented through a set of connected bonds. For the reconstruction of molecule depictions, it is therefore absolutely required to process these line drawings with an appropriate procedure.
Vectorization is a common starting point in pattern recognition. The automatic conversion from a raster image to a geometric vector [16] representation is a problem that has a long history and has received considerable attention during the last decades. These conversion algorithms were developed for diverse fields, such as the recognition of CAD/CAM drawings [36], schematic diagrams and geographic maps [10]. Instead of using all pixels in a picture, the main graphic objects are described through its contour, represented by a set of vectors. Although this representation form is limited and leads to an information loss, the derived vectors ideally approximate the content still sufficiently to portray the main information within the picture. A possible claim could be to achieve vectors which match the traces of the original drawing as closely as possible(maximal pixel coincidence criteria) [31][41].

3.3.1 Vectorization algorithms

Most of the vectorization methods consist of several processing steps [47]. At the beginning a line detection in a binarized raster image is done. Among several techniques a widely used strategy for that is skeletonization [16] which tries to infer a medial axis for the shape to be vectorized.

This centerline can be computed by diverse approaches. Morphological thinning successively deletes pixels from the boundary, until no more thinning is possible. An alternative approach first computes the distance transform of an image, where the greylevel intensity of points inside foreground(e.g. black) regions are modified. The intensity of a inner pixel depends on the distance to its closest border of the foreground region. Finally, the skeleton lies along the distinctive pixel positions, possessing intensities representing the furthermost distances to the boundary. Through skeletonization the original problem of vectorization is reduced on segmenting a 2D discrete curve into meaningful features and finally into a set of vectors. For the following segmentation step also several techniques exist. The algorithms produce high quality results if the objects to be vectorized are isolated. But unfortunately images often contain complex cases, like crossings and junctions between nontrivial thick shapes. In addition objects intersect or touch each other or contain pixel corruptions coming from image defects. This all can lead to a vector result set, approximating the original object only weak or inadequate. Pattern recognition methods, like for the chemical structure reconstruction, then have difficulties disposing the vector data.
Diverse processing steps after the segmentation step are applied to countersteer these circumstances. This can include simple heuristics, like finding better positions for the

junction points, setting junctions straight, merging those which are close to each other or reconnecting lines split up by a missing pixel. In many cases at this point domain specific contextual knowledge is additionally involved. For a progressive correction it is often required to take the domain dependent nature of a drawing into account. It would be definitely premature to regard vectorization as a solved problem. Vectorization methods work, but none of them covers all application domains.

3.3.2 Vectorization in CSR

The use of vectorization algorithms in the field of chemical structure reconstruction is largely unexplored. Several open source and commercial vectorization algorithms were evaluated for the chemoCR project. All of them emerged not to be qualified for handling sufficiently line drawing in chemical depictions. Often the focus of the algorithms lies on the pixel coincidence between the line drawing and the resulting vector set.

In contrast to that the requirements for an vectorization approach in the field of CSR are different. Here, the extraction of complete high level structures, such as lines and polygones and mainly their connectivity is essential.

A main bottleneck of several approaches was the lack of influence capabilities. They are operating as sealed black boxes, where no domain dependent nature of chemical drawings can be respected. The required effort for correcting the vectors and their connectivities was immense if possible at all and resulted often in a weak approximating vector set.
Contextual expert knowledge appeared mandatory to obtain sufficient vectorization results, required for the chemical pattern recognition. For that reason a new vectorization algorithm was developed at the SCAI Fraunhofer Institute. This algorithm implements a new concept of so called textures which are patterns of variations of the local directions and lengths of segments. Segments are connected black pixel collections in a row of a raster image. A method called discrete direction analysis identifies the segments that belong to each texture and translates them into vectors. The algorithm computes for each identified connected component of a chemical depiction an optimized vector set which fulfills the requirement of connectivity.

Chapter 4

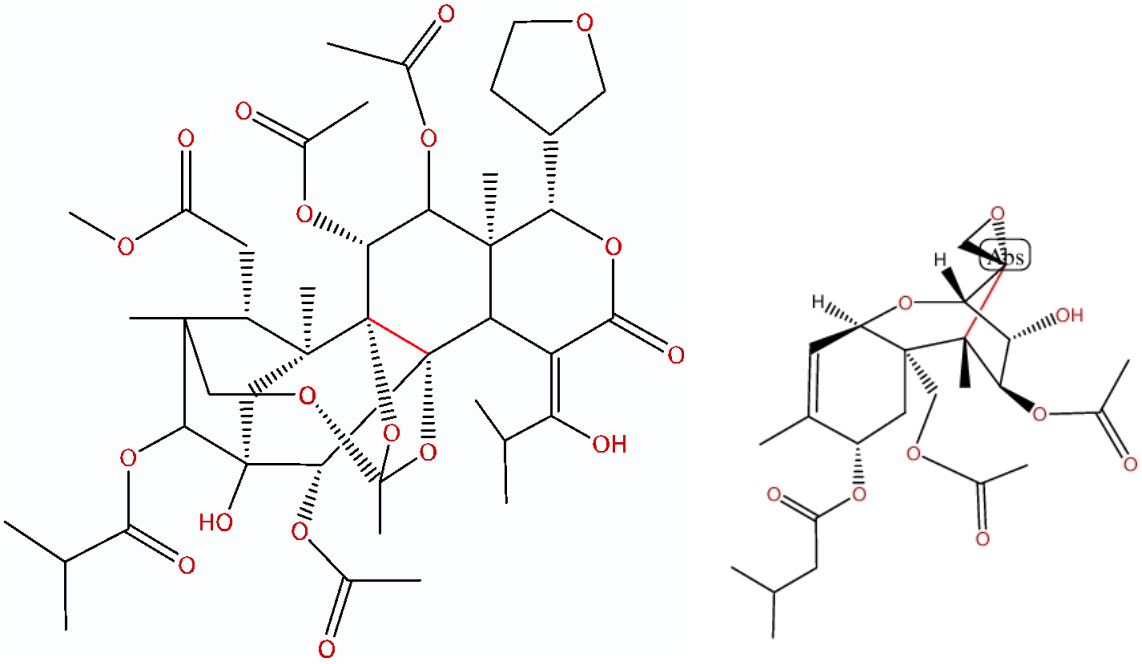
ChemoCR project

4.1 Context of chemoCR

The CSR project started in 2003 at the Algorithms and Scientific Computing Institute of the Fraunhofer society. This applied science organisation deals with the concrete application of scientific research to develop new innovative products.
ChemoCR [33] emerged after elaborating diverse studies about the related work of other CSR projects. Although the relevance of chemoinformatics and the number of non-machine readable chemical depictions is increasing, there exist no approaches addressing the CSR problem sufficiently. Several aspects of chemoCR differ from pure academic research projects. Instead of a limited time frame common for many academic projects, the duration of this project is not restricted. Here, team structure and responsibilities are a dynamic process and can change within months. The participating members possess different knowledge backgrounds, e.g. from machine learning, image processing, chemoinformatics and chemistry. To obtain sustainability for the contribution of each developer, a suited modular architecture must be available. It must be simple to enlarge the covered chemical depiction space as soon as new chemical semantic elements occur. For this reason the software must be able to be extended and maintained to facilitate updating to satisfy new requirements. Structuredness and understandability of the developers code should reduce the project familiarisation effort for new team members. Although this thesis is thought as a producibility study about a new way to address the CSR problem, these aspects should also be included in its conceptual design. The reconstruction of chemical drawings involves various techniques and concepts. Before a detailed overview on the workflow of chemoCR is given, two examples in figures 4.1 should give an impression of how complex their reconstruction can be. These images represent only a restricted cutout. A software addressing the CSR problem must be conscious of a permanent growing structural formula space. New chemical elements or other variances of already covered ones can occur with each novel publication.

4.2 ChemoCR workflow

The workflow of the current chemoCR can be broken down into four main phases. It starts with the preprocessing of the given input image. Then the recognition of the individual chemical patterns is performed. After that the reconstruction of the molecular graph is done. The workflow ends with a validation of the reconstructed molecule.



(a) A lot of dotted chirals

(b) Touching semantic elements

Figure 4.1: Complexity of chemical structure reconstruction

4.2.1 Preprocessing

The preprocessing itself consists of several substeps, which should provide different information for the chemical pattern recognition process. At the beginning the given chemical depiction is read in. For that purpose the mostly gray-scale or colored pictures are first binarized. Here, each square of the raster image is classified as information foreground (black) pixel or as white background one. This 0-1 discretization is then the basis for the segmentation proceeding. Segmentation is the step where image pixels are segregated into black pixel connected units. These so-called connected components represent different molecule symbols like atoms and bonds. An optical character recognition (OCR) program tries to infer from each identified connected component a character label. In a chemical drawing different strings can be found. All atoms are represented through their corresponding periodic system element abbreviation. Some elements, like oxygen(O) and nitrogen(N) consist of only one character. On the other hand elements like chlorine (Cl) exist, which contain two characters in their abbreviations. In addition to superatoms (e.g. 'COOH') and smile strings (e.g. 'C(=O)') hold several characters to encode entire structural patterns of a molecule. The OCR application only processes individually connected components representing single symbols. To get the whole string a character clustering and concatenation of the single characters must be done.
Molecule bonds are represented through line drawings in chemical images. Among other things, they hold the information which atoms have to be connected. Vectorization is a common starting point to deal with line drawings in a picture. There the raster image is converted into a geometric vector representation. For each symbol displayed through its associated connected component a suited vector approximation is calculated through a vectorization algorithm.
After the preprocessing procedures the information about all extracted connected compo-

nents, characters and vectors is available for the chemical pattern identification of the recognition phase.

4.2.2 Recognition

The recognition of the individual chemical semantic entities bases on the analysis of the information inferred through the preprocessing. All atom oriented strings within the molecule are already identified through the OCR and character clustering step. For each of the remaining chemical entities an own recognition procedure is responsible for the correct extraction.
From now on mainly the different types of molecule bondings have to be determined. Thereby the vector information is predominantly used. Each vector is examined, if it has associated similarly oriented vectors in its spatial neighborhood. There, it also can occur that larger bond sets, like an aromatic ring system, contain multibonds. With that proceeding all single, double, and triple bonds within a structural formula can be identified.
In addition to multibonds, chemical drawings can also include steric bond symbols to overcome the spatial restrictions of planar images. Special patterns symbolize that a bond sticks out or goes into the drawing pane. It is possible to differ here into thick chirals and dotted chirals. The symbol of a thick chiral is drawn as wedge. Because the vectorization algorithm must also calculate a suited vector approximation for such symbols, it also handles the recognition of these patterns. Such vectors contain the thick chiral label as well as the orientation indicating where the wider end is situated. This information is used during the thick chiral identification.

The recognition of dotted chirals is more laborious. Such patterns consist of several similar oriented vectors, following one by one. In length oriented dotted chirals the next vector follows after the end of a vector. In contrast to that, cross oriented dotted chirals contain several parallel vectors. In addition, through increasing sizes of the vectors an orientation can be encoded, similar to that in thick chirals. Unfortunately there are no predefined conventions for the number of the participating vectors, their distances and sizes. A chiral bond represents a single bond with particular spatial properties. That is why all chirals are substituted through a suited single bond. For that purpose all involved symbols are used to calculate the coordinates of a single bond vector. The listed chemical elements can be extracted from a given chemical drawing and can be further processed in the reconstruction phase. These recognition methods must be extended as soon as unknown entities or variants of already covered elements occur in new pictures. After having recognized all chemical elements of a structural formula, the individual entities must be now assembled to the entire molecule.

4.2.3 Reconstruction

Before the molecular graph (see section 2.3.2) can be generated and stored in a chemical file format (e.g. SDF), different processing steps are required. The graph consists of vertices and edges, symbolizing atoms and bonds of a molecule. For that all recognized atom strings must be replaced through their atom collections. Single and double character chemical elements can be directly assigned to a vertex. Superatoms and SMILES strings must be interpreted first. They encode several atoms and spatial arrangement to each

other. Each of these atoms must be assigned to its own newly introduced vertex. Besides the atom label information vertices also contain space coordinates for encoding the topology of a molecule. These coordinates correspond with the positions of the atoms in the planar image. This necessary information can be extracted from the associated connected components. In addition to their underlying pixel segments they contain a minimal bounding rectangle (*MBR*). A connected component's bounding box is the smallest box enclosing all pixels of the segmented pattern. The center of such a rectangle exhibits a dedicated coordinate for the atom symbol it represents. The identification of the space position for a single character atom is consequently a straight forward process. Double character atoms makes it necessary that both boxes of the participating characters have to be taken into account to calculate qualified coordinates. The application of precalculated spatial templates was found to be useful for dealing with superatoms and SMILES strings. Such strings are replaced through a collection of atom vertices with suited coordinates and well defined edges. This mini graph is then included in the existing chemical graph. Bonds itself can also encode atoms. There is no explicit carbon atom listing in a connected bond set. Sometimes it occurs that even single bonds have no connected atoms at the end. For that reason new atom vertices and their corresponding edges have to be introduced for each vector ending. Here the coordinates of each vector can be used to approximate the atom positions. At this point it can not yet be decided if the bond vector really represents two connected carbon atoms. In the meanwhile the chemical graph contains unconnected vertices for all explicitly drawn atoms. Furthermore it includes carbon atom vertices which are introduced through the bonds. Now the connecting of the individual vertices can be performed by importing the recognized topologic information into the graph. For that it is examined if there are any identified bonds in the spatial neighborhood of each vertex. In this case it is checked if these bonds connect the vertex with other vertices. If so, the corresponding bond information is set in the chemical graph. With this proceeding it can also be decided if the carbon atoms which are artificially introduced through the bonds must be deleted again. If there is an atom close to the end of a vector, then the carbon atom vertex of this vector end must be removed from the chemical graph. Sometimes determining if a carbon vertex has to be dropped can be complex. Like the usage of chirals, there exist more drawing possibilities to simulate three dimensional information. A commonly used technique is the drawing of bridged bonds. Here an artificial carbon vertex must be also dropped, although there are no atoms close to the end of a bond vector. The reason for that is that this bond continues on the other side of another bond. According to that, vertices must be deleted and new bonding information propagated into the graph. Another case of the complex treatment of artificial vertices are cross bonds. Here, several interconnected bonds share a common artificial atom. The other introduced nodes have to be removed.
After entering all inferred atom vertices, updating all bonding information and interpreting every critical bond situation, the chemical graph should finally contain all atom and topology information of a given structural formula.

4.2.4 Validation

Before the generated chemical graph can now be exported as a chemical file format, an evaluation of the entire recognition/reconstruction process is done. The workflow of chemoCR allows to perform a fully automatic or a semi automatic reconstruction of a

chemical drawing. If an absolutely correct recognition of a molecule must be guaranteed, a kind of numerical score is necessary, which informs about the quality of the reconstruction. The validation of a reconstructed molecule involves several aspects, which all influence the final evaluation score. For that molecule fragmentation, number of occurred errors are recorded. In addition to that, chemical knowledge like valence information and atom string occurrence probabilities are considered. With such a detailed reconstruction assessment a chemist can decide faster if a recognized molecule requires manual correction. In this case the molecule can be automatically loaded into a chemical editor where the problematic image positions are marked.
A method addressing the CSR problem can not and must not be an out-of-the box solution. Problems mainly occur with new chemical elements or new variations of already considered ones. Instead of the ignorance of occurred errors, the detection of these problems is absolutely required to push the development of the reconstruction proceeding and to provide chemists a valid supporting reconstruction tool.

4.3 Bottlenecks

The current chemoCR approach works well for diverse chemical depiction test sets, which were already partly available at the beginning of the project. Nevertheless, the project has still to deal with several problems in the preprocessing, recognition and the reconstruction step. Some significant bottlenecks emerged which were not predictable at the stage of the conceptual design of chemOCR.

4.3.1 Preprocessing

Several aspects in the preprocessing in the current chemOCR project can cause problems.

Binarization threshold

The binarization of a grayscale or color image is a crucial step before the segmentation into isolated units can be performed. Although different binarization algorithms exist, a relatively simple method is applied to infer the required binarization threshold. Depending on the pixel intensity this threshold indicates when a pixel in the raster image is classified as information pixel (black) or as background pixel (white). A less qualified binarization result can lead to a less optimal segmentation result, which thereby influences the quality of the vectorization and OCR algorithm output. Problem situations like connections between atoms close bonds may occur, where the vectorization as well the OCR algorithm would fail to resolve.

Color usage

At the present time all grayscale and color images are reduced on a 0-1 representation. It may be important to also involve color information in the future. In colored chemical pictures there often exists a color code for atoms (like oxygens are often red). This information could be used for several purposes. In the previous paragraph the problem of pixel connected characters was mentioned. Many color pictures would enable a clean separating of connected colored letters from black bonds. In addition, a chemically specific OCR could be implemented, which also considers the color of a recognized character to infer the certainty about it.

Application of conventional OCR

The appliance of conventional OCR in chemoCR is also a problem. In text, these algorithms can correct a wrong identified character through contextual verification. In text, the recognition of entire strings is of interest. That is why the OCR does not provide a functionality which allows the access to several character suggestions and confidence values for each symbol. Instead of that it provides only the best hit. Here it would be more reasonable to provide all suggestions and delay the selection into the reconstruction phase, where a suited chemical atom character could be chosen.

4.3.2 Knowledge representation

Although the recognition of the existing test sets is still not perfect, a lot of new chemical images have been already collected from different resources. These drawings contain new chemical semantic entities and often exhibit drawing variants of the already covered ones. Because a growing structural formula space was assumed from the first, a simple updating, modifying and extending of the recognition modules was a basic part of the recognition strategy. But several conceptual bottlenecks complicate these essential requirements. The complexity of error tracking, recognition module updating and the familiarisation effort for the developer increase with each new introduced chemical entity.
The main problem results from the lack of a knowledge representation concept in the chemoCR project. Each chemical entity recognition module contains a restricted implicit form of chemical knowledge. Every recognition method must be able to interpret structural patterns to identify its target. Beyond that the identified patterns must be extracted and correctly substituted through an atom bond representation. As a result of an autonomous recognition module design different knowledge representations were developed.
Up to a certain degree, the drawing of structural formulas is well defined. For that reason the usage of a rule oriented formalism to encode knowledge has been established.
Although nearly every recognition module codes its recognition and extraction knowledge with rules, there is no concept for a common rule language. This has several consequences. The familiarisation effort for each recognition module is very high. For the maintenance and the extension of a module, it is necessary to get familiar into its individual knowledge representation. In addition, there is no central rule repository, where the entire project knowledge is administrated. Instead of that, different rule encodings are scattered over the entire program. Because of that it is not possible to keep track of the type of chemical knowledge which is already covered in the chemoCR project. Therefore the reutilization of already existing knowledge becomes unfeasible.
All these listed disadvantages emerge as a result of the absence of a suited knowledge representation concept. The maintainability, extendability and the associated accuracy deficits challenge the claim to develop a reconstruction algorithm which copes with the permanently increasing structural formula space.

4.3.3 Recognition strategy

The current conceptual design plans to implement for each chemical semantic element its own recognition module. All modules should work independent from each other so that exchangeability and modularity can be granted. If a certain semantic entity cannot be identified correctly the error should be backtrackable to the corresponding module.

But after certain dependencies between the individual modules were realized, this clean module separation was no longer assured.

Due to so-called *overloaded symbols*, multiple recognition modules try to recognize and extract the same connected component symbols. This problem situation mainly evolves from simple single lines, which can occur in several different kind of chemical entities, such as single and multibond, length and cross oriented dotted chirals as well in several atom oriented characters like 'I', 'l', '(' and ')'.

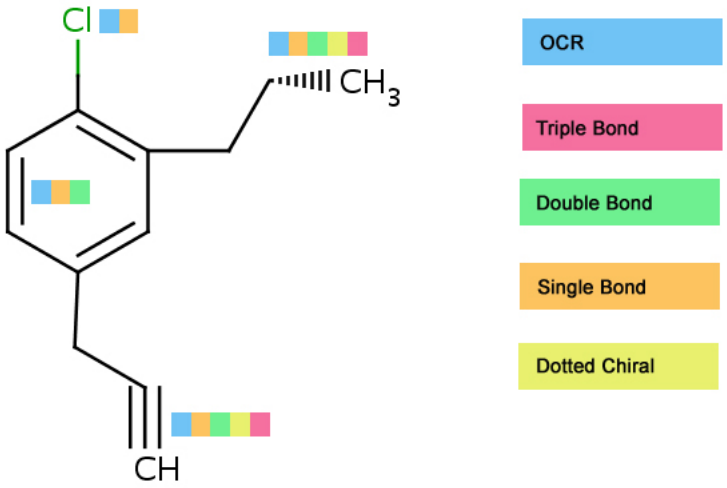


Figure 4.2: A simple line symbol can be element of different semantic elements, like dotted chirals, atoms and bonds. So several recognition modules (indicated by the color code) can claim the same image symbol.

The *multi conflict*, which is the consequence of this symbol ambiguity, is very difficult to resolve if possible at all. This clashing makes up the major problem chemoCR is faced with and it is certain that even more ambiguity of overloaded symbols appear in new images. The main reason for the multi conflict problem is that the full potential of available knowledge about chemistry and structural formulas is not used.
Limited chemistry knowledge comes only after the recognition process into operation. It is applied to curate infrequent atom strings like Ci into Cl and to calculate the evaluation score of reconstructed molecules. In the current approach each recognition module possesses only the structural formula knowledge to identify its pattern. Only the pattern itself is required to decide if it can be assigned to a certain chemical entity. Which entities are situated in the chemical context of a pattern is not involved in the recognition proceeding. With this detection strategy no chemical context and no extended knowledge based recognition can be realized.
In place of a chemical context, chemoCR only requires a spatial context for recognition and reconstruction. Structural patterns like dotted chirals consist of several closely arranged line symbols which share certain features (size, parallelism, common axis). To define what is close, several so-called soft parameters have been introduced. Instead of applying fix values for distances and sizes, these values are approximated through the distances

and sizes detected in the chemical drawing. For instance the line length of a dotted chiral symbol can be used to define the spatial context, in which the next participating line symbol has to occur. This parameter-based context is also required for the reconstruction process. After all individual semantic elements are recognized they have to be associated. To determine if two elements (e.g a bond and an atom) are related and should be linked in the chemical graph, a parameter spanned context space is analyzed. It should be noted that these parameters are just approximations. These manually assigned values are also a form of implicitly defined knowledge. Because the parameter values were mainly empirically derived from the available chemical depictions sets, there is no warranty that they cover all chemical images. Even if the parameter approach allows a suitable spanning of an analysis context, the assembly to a molecule can pose more challenges. For a correct reconstruction, spatial examination is sometimes not sufficient. The so-called *semantical physical distance problem* emerges if chemical entities would be physically close enough to be interconnected, but this is not the case according to their chemically semantics.

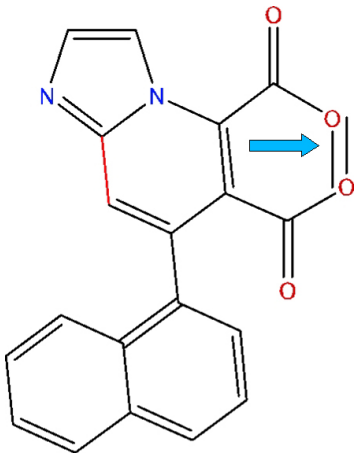


Figure 4.3: The shown molecule has a very difficult position to recognize, indicated by the arrow. Although bonds and atoms would be close enough to be combined from the molecule assembly routine, the reconstructed molecule would be semantically wrong. Due to the valence of oxygen (2) it would be chemical wrong to connect this atom with the close bond.

Complex problems like the multi conflict and the semantical physical distance problem makes it necessary a reconsidering of the current context concept and the requirement for a more context specific knowledge based recognition strategy. Instead of using scattered parameters, a new homogeneous spatial context concept is essential for the recognition and the reconstruction. In the best case this concept should be completely independent of any parameter. The ambiguity of overloaded symbols and the semantical physical distance problem can only be resolved through an advanced context analysis and the appliance of more structural formula and chemistry knowledge.

Chapter 5

New reconstruction concept

In the last chapter several significant bottlenecks of the current chemoCR project were described and their underlying conceptual reasons shown. A main deficit arose from the fact that each chemical entity (see 4.2.2) recognition is delegated to an own module. These modules are in most instances based on elementary geometric analysis. To identify an entity, a structural pattern of image symbols is searched, which satisfies required symbol properties as well as certain spatial arrangement restrictions. Because there exists no general homogeneous knowledge representation concept, each module largely formulates these requirements in different rule encodings. The cross dotted chiral module (see 4.2.2) e.g. encodes its entity as a set of parallel lines which are close enough. For that purpose the module implements like other modules its own perception of closeness. This is realized through the specification of diverse parameters in each module, leading to a wild scattering of values, which have a great impact on the system's performance.
In addition, these various implementations of closeness complicate the application of a more chemical knowledge based recognition, which is essential to deal with complex difficulties like the *multi conflict* and *physical semantical problem* (see 4.3.3).
In contrast to the applied spatial pattern rules a chemical knowledge based recognition procedure is able to introduce advanced knowledge about structural formula drawings and real chemistry into the identification process. Here not only the pattern itself is of interest, also the spatial context which determines a kind of chemical environment of the pattern bias the inference decision. To clearly define this spatial context an unique concept of neighborhood is necessary, which is valid for all methods depending on space information.
In the following a new chemical structure reconstruction approach is presented, which builds on three new concepts. First a new information level is introduced, which allows the unique description of how symbols are arranged in an image. This technique denoted as orientation graph (OG) is based on a Relative Neighborhood Graph (RNG), which will be shortly specified in section 5.1.1. After the definition of the OG in 5.1.2 an algorithm will be shown, which enables an efficient computation of this graph.
Beside the OG the new approach is based on two further concepts. To realize an advanced reconstruction a new distinct knowledge representation and clear inference procedure was developed, which is based on the concept of expert systems. These artificial intelligence systems are detailed in section 5.2.1; their concrete application for the CSR domain are explained in 5.2.2.
The new expert system must be applied to each symbol in the picture, which is done through the novel concept of constraint based graph exploration presented in 5.3.2.

5.1 Spatial arrangement approximation

As mentioned in the introduction of this chapter, the application of the required chemical knowledge based recognition is complicated through the various module implementations of closeness. For that reason the spatial arrangement concept of *OG* has been developed, which possesses a clearly defined description of how the individually connected components are arranged in the image. In contrast to the numerous context definitions of the chemoCR, the representation does not require any parameters. Before specifying the *OG* the Relative Neighborhood Graph is explained on which the new graph builds upon.

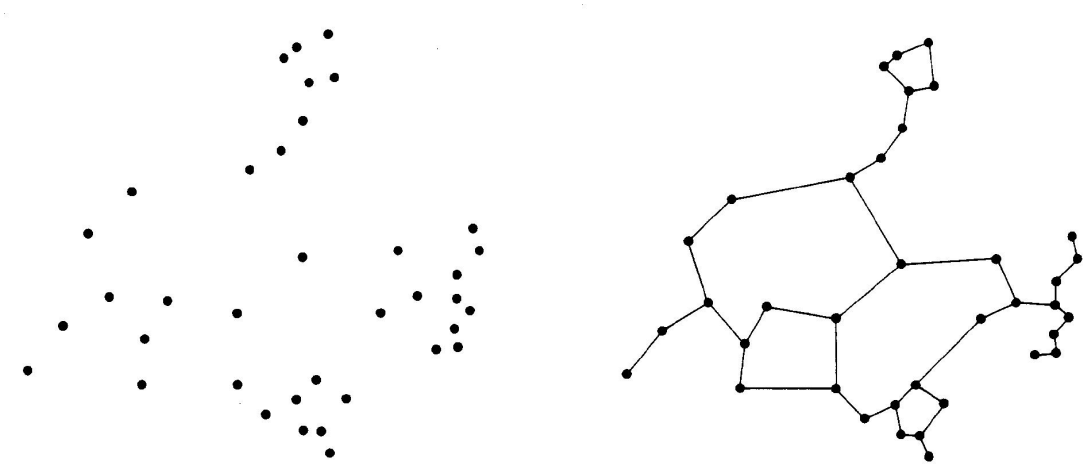
5.1.1 Relative neighborhood graph

Several computer science problems exist, which require to detect an inherent structure in a finite point set.
The field of Clustering [17] for instance tries to separate a point set to distinct clusters, so that the points within a cluster are very close/similar to each other. Points between different clusters in contrast are relatively distant. Clustering algorithms [4] operate on a point set to approximate several neighborhood subsets.
In contrast to that, another structure discovering field exists dealing with the representation of the overall neighborhood of a finite point set. Many definitions have been proposed in the scientific literature, to define if two points are close enough to be neighbors. Lankford [27] declares two points as being relatively close if they are at least as close to each other as they are to any other point.
This relatively close specification was in turn applied by Toussaint et al. [48] to develop the Relative Neighborhood Graph. The *RNG* (compare figure 5.1) approximates the neighborhood of a point set through a graph, whose vertices represent the points and the edges indicate the neighborhood of two points. Two points are considered neighbors if they are relatively close.

Definition:
A *Relative Neighborhood Graph* is an undirected graph with vertices $v \in V$ in a metric space X with a distance function d, such that there is an edge between points p and q if and only if $d(p,q) \leq \max[d(p,z), d(q,z)] : z \in V \backslash \{p,q\}$ (relatively close condition)

A distance function on a set X is a mapping $d : X \times X \rightarrow \mathbb{R}_0^+$.
For all $x, y, z \in X$, this function is required to satisfy the following conditions:

- $d(x,y) \geq 0$ (Non-negativity)
- $d(x,y) = 0 \Leftrightarrow x = y$
- $d(x,y) = d(y,x)$ (Symmetry)
- $d(x,z) \leq d(x,y) + d(y,z)$ (Triangle inequality)



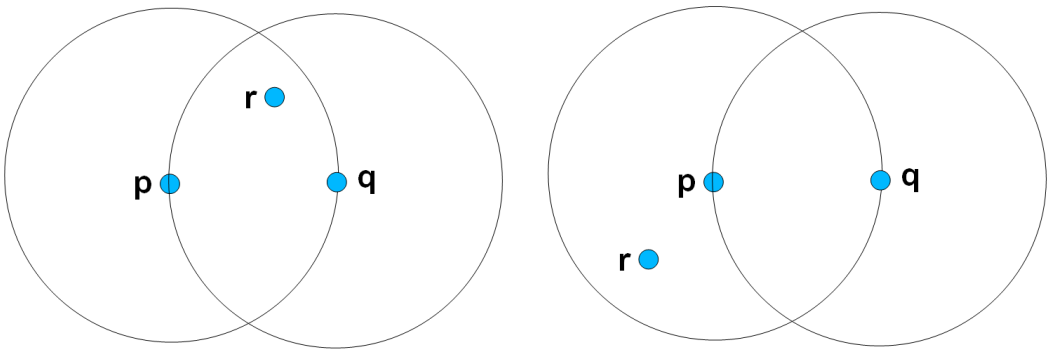
(a) Planar point set

(b) *RNG* for given point set

Figure 5.1: Example of a Relative Neighborhood Graph (RNG). The left image shows a planar point set. The right figure illustrates its corresponding *RNG*. Each point becomes a vertex of the graph and an edge is drawn if two associated points are relatively close to each other (compare figure 5.2).

Although this graph was originally defined for a planar point set, it has been extended for the use in multidimensional space [1]. It should be mentioned that there is a great difference between '$<$' and '$\leq$' in the definition of relative closeness. In a degenerate situation such as three points lying equidistant from each other, all three points are considered relative neighbors of each other, whereas with a '$<$' none of the three points would be connected.
As a consequence of the definition each vertex can be reached through a path beginning at an arbitrary starting point. This as connectivity denoted feature of the *RNG* guarantees that all vertices belong to the same graph.
For the generation of the *RNG* a relative close test must be applied for each vertex pair from the point set P. An edge is drawn between two points p and q, if there are no other points in the *lune* of these two points. The *lune* [48] is defined as the intersection of two spheres of radius $d(p, q)$, one of the spheres centered at p and the other at q. In figure 5.2 two cases are demonstrated. The point p would be connected with q if r would not be present in the *lune* otherwise there would be no edge. The shape and volume of the *lune* depend on the applied distance function such as Minkowski and the Euclidean distance function [37].

The *RNG* seemed qualified as new spatial arrangement information level for the new recognition concept, because it allows the description of the overall spatial neighborhood for given objects. Why this graph approach cannot be applied directly in the CSR field, is detailed in the following sections. There, a new so-called *orientation graph* (*OG*) is presented, which enables to address this requirement.



(a) Points p and q are not relative close (b) Points and q are relative close

Figure 5.2: Two points p and q are relatively close to each other, if there is no further point in the intersection of two spheres of radius $d(p,\ q)$, where one sphere is centered at p and the other at point q.

5.1.2 Orientation graph

Although the already in section 5.1.1 defined relative neighborhood graph seems to be a qualified representation for the required spatial arrangement approximation level, it possesses a significant bottleneck. In place of a neighborhood graph for a planar point set, it is desired to obtain such a graph for connected components, representing the symbols in a chemical drawing. Each of these components is in turn itself a set of pixel points. Because the *RNG* is only defined for one pixel set the *OG* was developed, to overcome this bottleneck. It mainly based on the extension of the relative closeness concept of the *RNG*. The decision if two vertices are edge-connected in the *RNG* only requires the analysis of the *lune* (5.1.1), whose shape is determined by the two corresponding points and the applied distance function. For the relative close test for connected components there must be also an intersection area examined, but in contrast to the simple *lune* shape, the form of the intersection can be very complex in this case.

This results from the different distance functions, which are used to measure distances between point sets. Two amongst other proposed functions in literature [43, 18] are the *single link* and the *complete link* distance function.

The single link distance between two point sets is the minimal detectable point pair distance. In contrast to that indicates the complete link distance the maximal detectable point pair distance.

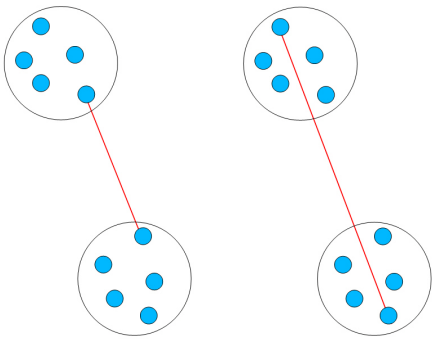
Point Set Distance Definition:
Let P and Q be two point sets.

$$\text{Single link distance: } \min\{d(p,q) : p \in P,\ q \in Q\} \tag{5.1}$$

$$\text{Complete link distance: } \max\{d(p,q) : p \in P,\ q \in Q\} \tag{5.2}$$

Both definitions are based on an usual distance function $d(x, y)$ for points, such as the Euclidean distance. Figure 5.3 illustrates the two specified distance functions for point sets.

For the generation of the orientation graph the application of the single link definition seemed qualified to calculate the distances between its connected component vertices.



(a) Single link (b) Complete link

Figure 5.3: Distance measurements for point sets. The figure a illustrates the single link distance, defined as the minimal detectable point pair distance. Figure b shows an example of complete link distance, defined as the maximal detectable point pair distance.

Due to the function definition it is possible that several point pairs (p, q) might hold the same single link distance.

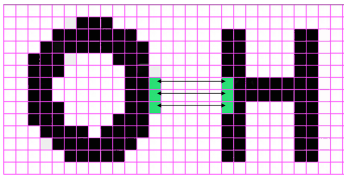


Figure 5.4: Several point pairs in the connected components might exist, which hold the same single link distance (marked as green pixels connected by thin lines).

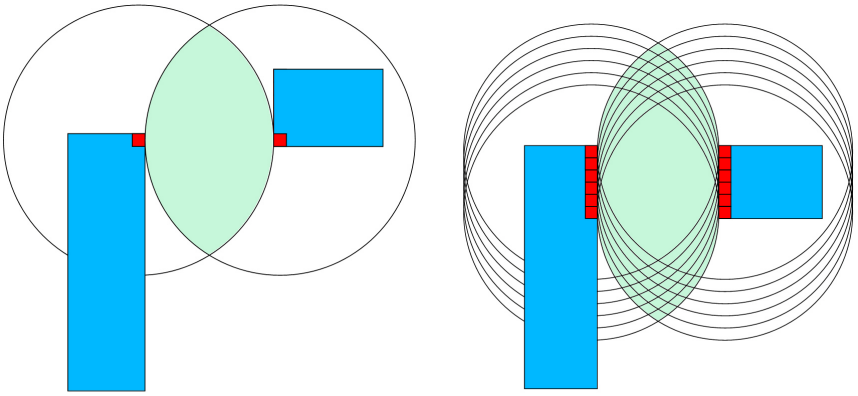
After defined distance functions for point sets, the orientation graph can be now defined.

Orientation Graph Definition:
An *orientation graph* is an undirected labeled graph with vertices $v \in V$ in a metric space X with a distance function d, with the following properties:

- each vertex P is a point set
- an edge is drawn if two vertices P and Q are relative close
- two vertices are relatively close if there is no other vertex in their intersection area, defined through: intersection(P, Q) = $\bigcup_{i=1}^{n}$(sphere(p_{sl_i},r) $\cap$ (sphere(q_{sl_i}, r))) with r= $d(P,\ Q)$ and n= #(point pairs (p_{sl}, q_{sl}) with single link distance)

Although several algorithms [26] already exist to generate a Relative Neighborhood Graph, the OG definition yields to such substantial differences in the graph inference

process that a new proceeding was required. In contrast to the simple *RNG lune* the shape of the intersection area to analyze might be very complex. For each single link point pair, the intersection of the two spheres defined through the single link distance and the two point centers must be calculated.

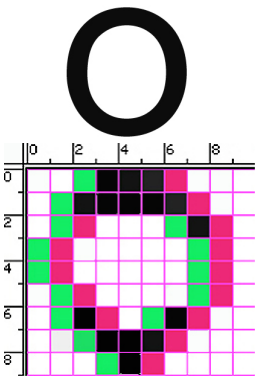


(a) No extended intersection (b) Extended intersection

Figure 5.5: Inferring relative closeness for connected components (blue boxes) requires the analysis of intersection areas (green), which might have a complex shape. Left example shows the trivial case where a unique single link point pair exist. Right example represents the intersection area of two components having six single link point pairs.

Next a two substep algorithm (compare algorithm 1) is described which enables to create an orientation graph from a set of connected components.
At the beginning all distances between all connected components are being computed, while all point pairs with single link distance are being stored. To speed up this step, some properties of the connected component data structure are exploited. A component consists of a collection of continuous black pixel segments, which are represented through their starting and ending raster image coordinates (compare figure 5.6).

Instead of deriving the distance between all against all pixel pairs, it is sufficient to consider only the border pixels of the connected component. Because of the continuity property of the segments, these border points can be easily derived and dependent on the relative position of the components it can be enough to compute only the distances between the start and stop coordinates of the segments. If a smaller distance has already been observed, a further processing of the segment is not required. For each connected component pair a so-called *MinDistResult* is calculated (compare algorithm 1), which holds the single link distance value and all point pairs exhibiting this distance. This MinDistResults is then the starting point for the relative close tests in substep 2, which decide if two connected components cc_i and cc_j are linked in the OG or not. For that the intersection area of cc_i and cc_j must be analyzed, which is clearly determined by their single link distance and the single link point pairs.
The algorithm calculates for each point pair the spheres with radius $d(cc_i, cc_j)$ centered in $p_{cc_i x}$ and $p_{cc_j x}$ and tests if there is any cc in their intersection (compare figure 5.5). If a cc has been found, a further extension of the intersection area is not required and the



CC Data:

Line	Segments
0	(2,0)–(6,0)
1	(1,1)–(7,1)
2	(1,2)–(2,2); (6,2)–(8,2)
3	(0,3)–(1,3); (7,3)–(8,3)
4	(0,4)–(1,4); (7,4)–(8,4)
5	(1,5)–(2,5); (7,5)–(8,5)
6	(1,6)–(3,6); (5,6)–(7,6)
7	(2,7)–(6,7)
8	(3,8)–(5,8)

Figure 5.6: A connected component is a collection of line segments, representing connected black pixels. This character 'O' e.g. can be described through 14 segments, shown in the right table. Because each segment is continuous it is only required to store the start (green box) and the stop coordinate (red box) of each segment.

two vertices cc_i and cc_j are left unconnected in the graph. Otherwise the proceeding is repeated with the next spheres intersection until the complete intersection area is fully analyzed. An edge is drawn if no *cc* was found in this area.

Checking all connected components to see if they fall within or are partly contained in the intersection area would lead to unnecessary pixel distance calculations. This is avoided by the use of a spatial index structure for multi-dimensional data called *RTree* [3, 32], which splits space with hierarchically nested minimum bounding rectangles. All segments of a connected component can be described through a minimal bounding rectangle (denoted *MBR*). Already during the distance computation in step 1, all *MBRs* and the corresponding component identifiers are inserted into the *RTree*.

Like every tree, it consists of inner nodes and and leaf nodes. In every entry within an inner node the identifier of a child node and the bounding box of all entries within this child node is stored. In every entry within a leaf node the *cc* identifier and its bounding box is hold. The fact that "nearby" elements are placed in the same or in a close leaf node is a crucial property of the *RTree*, which allows an efficient search in this index structure. The bounding boxes can be used for containment queries to decide whether or not it is required to search inside a child node. In this way, most of the nodes in the tree are not analyzed at all during a search. For the *OG* generation the *RTree* implementation of Wolfgang Baer was used, which is part the Deegree software package[1]. Deegree is a free software initiative founded by the Geographical Information Systems and Remote Sensing unit of the Department of Geography at the University of Bonn. The *RTree* package was extended by the implementation of a range query algorithm.

After the *RTree* contains all *MBRs* of all molecule image symbols, the required relative close tests for the orientation graph can be done. For both points of each single link point pair of two connected components, a range query with radius d(cc_i,cc_j)is applied on the

[1] `http://deegree.sourceforge.net/`

spatial index structure. Only the components in the intersection of both result lists have to be checked on pixel distance level. Having analyzed all required intersection areas, the orientation graph can be visualized whereas the vertices can be approximated through the centers of the individual *MBRs* and edges are drawn if the corresponding *ccs* are relatively close in the chemical depiction. Figure 5.8 shows an example of an orientation graph of a chemical depiction.

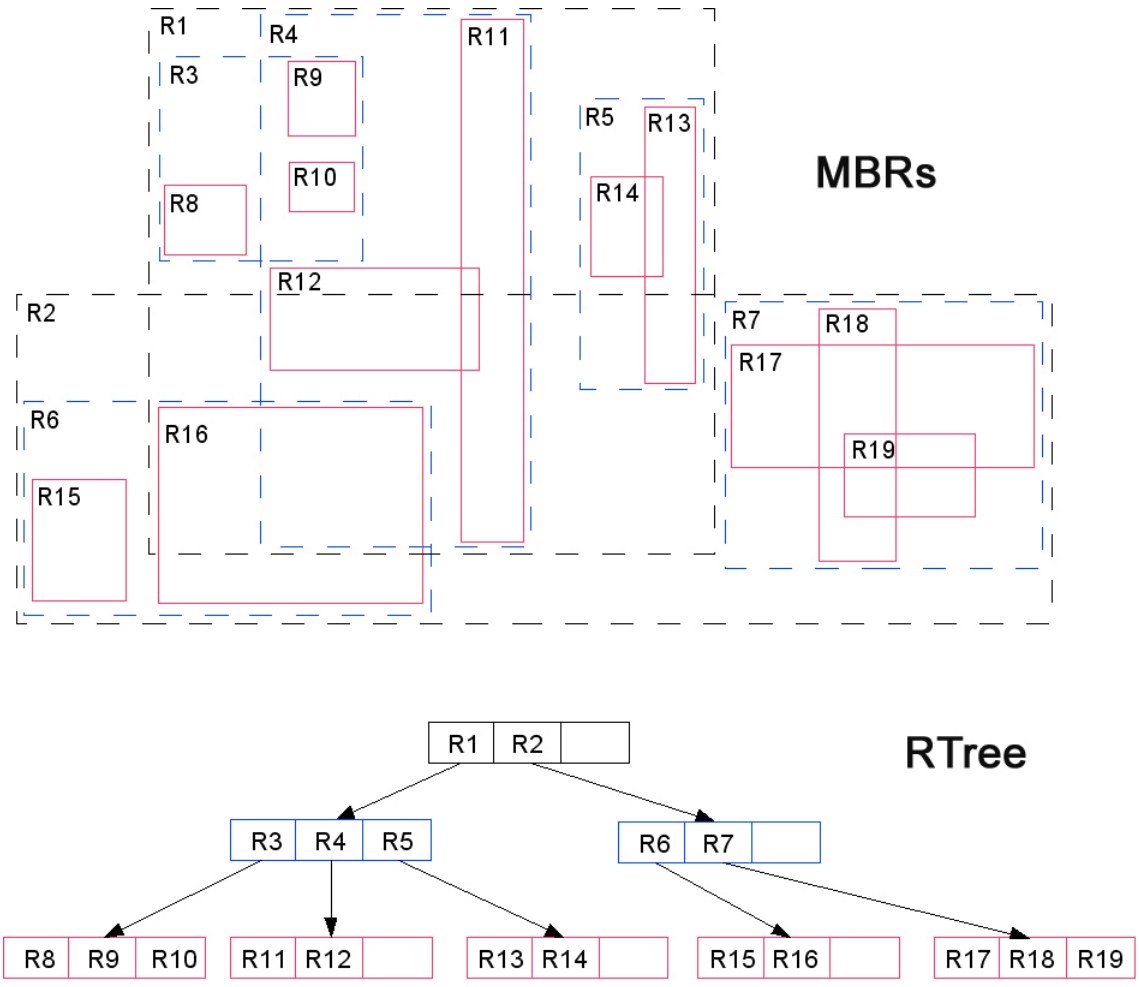


Figure 5.7: Symbols within an image can be described by their minimal bounding rectangles. These *MBRs* can be organized in an *RTree*, which allows spatial queries. The dotted lines in the above picture indicate various bounding rectangles, whereas the coloring represents different sizes of boxes. E.g. the black dotted *MBRs* are the largest ones which contain the other smaller red and blue boxes. For that reason the black rectangle R1 and R2 is situated in the root of the *RTree* (compare tree image).

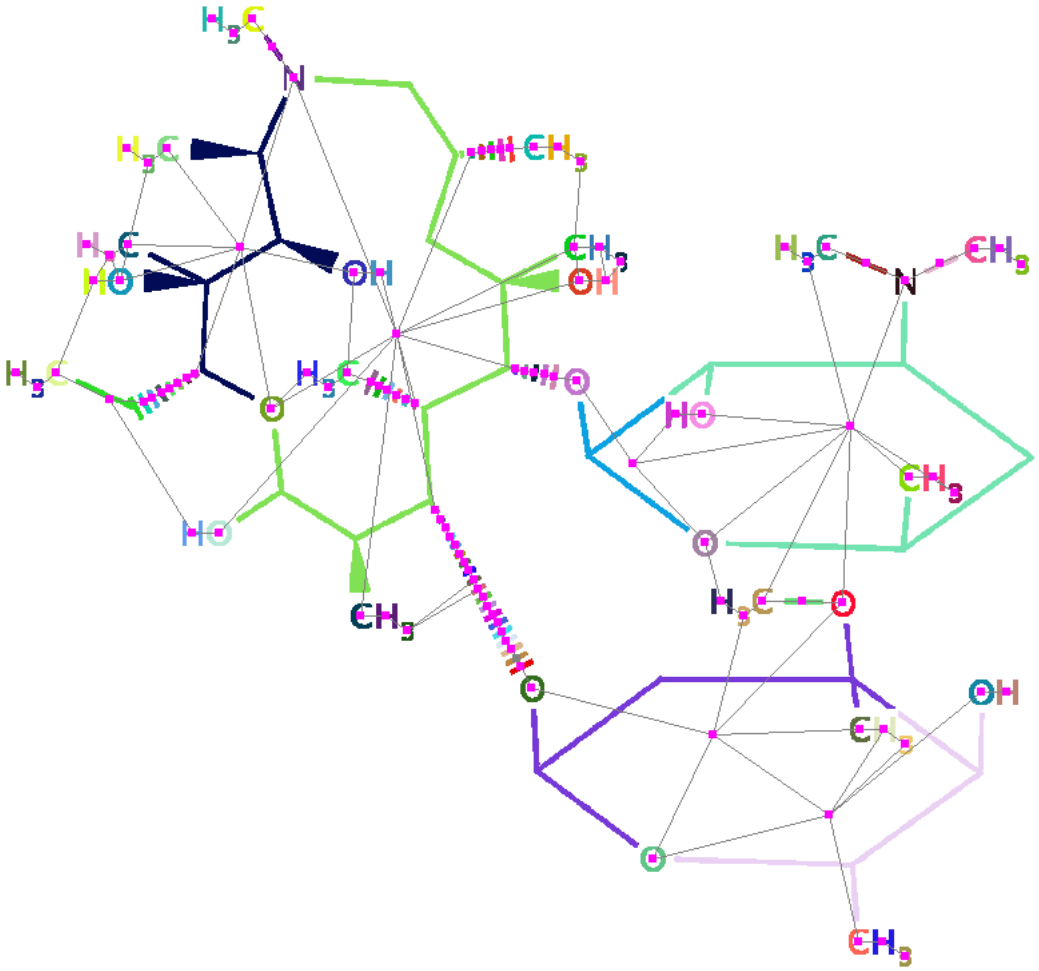


Figure 5.8: Orientation graph describes the relative closeness of the image symbols. Each connected component of a symbol has an own coloring. The vertices of the graph are indicated by small purple boxes which are placed in the center of the minimum bounding rectangles of each component. Thin gray lines between these boxes indicate that the underlying vertices are relatively close.

Algorithm 1: Orientation graph generation algorithm

```
   input  : ConnectedComponentCollection ccList
   output: OrientationGraph og (= AdjacenceMatrix)
 1 begin
 2     rtree ← initEmptyRtree();
 3     og ← initOGMatrix[ccList.size][ccList.size];
 4     minDistResultMatrix ← initMinDistResultMatrix[ccList.size][ccList.size];
 5     for i ← 0 to ccList.getSize() do
 6         box ← ccList[i].getBoundingBox();
 7         rtree ← rtree.add(ccList[i].id, box);
 8         for j ← (i+1) to ccList.getSize() do
 9             minDistResult ← calculateMinDist(ccList[i], ccList[j]);
10             minDistResultMatrix[i][j]= minDistResult;
11         end
12     end
13     for i ← 0 to ccList.getSize() do
14         for j ← (i+1) to ccList.getSize() do
15             minDistResult ← minDistResultMatrix[i][j];
16             pointPairList ← minDistResult.getMinDistPointPairList();
17             numberToTest ← pointPairList.getSize();
18             testList ← initEmptyList();
19             relativeClose ← true;
20             for k ← 0 to numberToTest do
21                 pointSet = pointPairList.get(k);
22                 queryResultsList1 ← rtree.rangeQuery(pointSet.p1,
                   minDistResult.getMindist());
23                 queryResultsList2 ← rtree.rangeQuery(pointSet.p2,
                   minDistResult.getMindist());
24                 testList ← intersection(queryResultsList1, queryResultsList2);
25                 testList ← testList.remove(ccList[i]);
26                 testList ← testList.remove(ccList[j]);
27                 for l ← 0 to testList.getSize() do
28                     if (isInIntersection(testList[l], minDistResult)) then
29                         relativeClose ← false;
30                         break;
31                     end
32                 end
33                 if (not relativeClose) then
34                     break;
35                 end
36             end
37             if (relativeClose) then
38                 og[i][j] ← 1;
39                 og[j][i] ← 1;
40             end
41         end
42     end
43     return og;
44 end
```

5.2 Knowledge Representation

After extracting several information levels from an image, these information must be interpreted to gain a perception about the role of each symbol in the chemical drawing. Several pattern recognition modules must be developed, which analyze the existing symbols on the basis of the information levels and assign to each symbol a particular chemical meaning.
One great bottleneck of the current chemoCR software (refer to 4.3) is the intransparency which arose by the implementation of own pattern recognition strategies for each of these modules. Although nearly every module encodes its own recognition and extraction knowledge with rules, there is no concept for a common rule language. Instead of that, mainly structural pattern knowledge is dispersed in form of different geometric analysis methods over the entire program.
On the other side the previous chapter 4 showed that a more chemical knowledge based recognition is essential for a high quality automatic reconstruction of structural formulas (compare 4.3.3).
For that reason a new unique knowledge representation must be developed, which allows a more comprehensive chemical knowledge formulation and application. To keep track of the covered knowledge, it must be placed in a central rule repository where it can be administrated and reused from each module in the project. Goal of the thesis was to develop a new knowledge based recognition concept for the CSR problem. After identified several parallels, the idea arose to implement the new concept as expert system [25], which have been already successfully applied to a broad range of fields. In the following a short overview about expert systems is given.

5.2.1 Introduction of expert systems

Expert systems [25] form a distinct and identifiable class within artificial intelligence [30]. This is a branch of computer science dealing with the design and implementation of computer programs which are able to emulate human cognitive skills such as problem solving, visual perception and language understanding. These expert systems have already been successfully applied to a various range of scientific domains, including object recognition in computer vision, medical diagnosis [45], bioinformatics and many more [40]. In the next sections the characteristic features and the main differences between expert systems and conventional artificial intelligence programs will be presented.

Characteristics of an expert system

Although there is no precise definition of an expert system, there are several basic features which all expert systems exhibit to a certain degree.
They simulate human reasoning about the problem domain, rather than simulating the domain itself. This reasoning is performed over a suited representation of human knowledge. Knowledge is the theoretical or practical understanding of a subject or a domain. It can be seen as the sum of what is currently known by human experts.
The knowledge in the program, called *knowledge base*, is usually kept separate from the inference engine, the code that is responsible for the reasoning. Problems tend to be solved by using heuristics or approximate methods which, unlike algorithmic solutions, are not guaranteed to result in a correct or optimal solution. Instead of that the system

can propose solutions with varying degrees of certainty.

Expert systems need to provide explanations and justifications of their solutions. After each inference proceeding they must be able to defend their decisions by proving that their reasoning is correct.

Knowledge acquisition

During the knowledge acquisition process the domain specific knowledge of an expert is introduced into the expert system. This is realized in several principal stages (see figure 5.9). In the knowledge elicitation step the communication between the knowledge engineer and the expert leads to the knowledge extraction is some systematic way. For that purpose as much information as possible about the given domain, the existing problems and their solving strategies is collected. Then a categorisation of the types of reasoning and problem solving tasks the expert system has to support is done. At the beginning the received expertise is formulated in some trivial human friendly intermediate representation. Afterwards this form is then translated into an executable representation (e.g. production rules), which can be interpreted from an inference engine. In practice many refining and incremental iterations are usually required to be able to model expert knowledge.

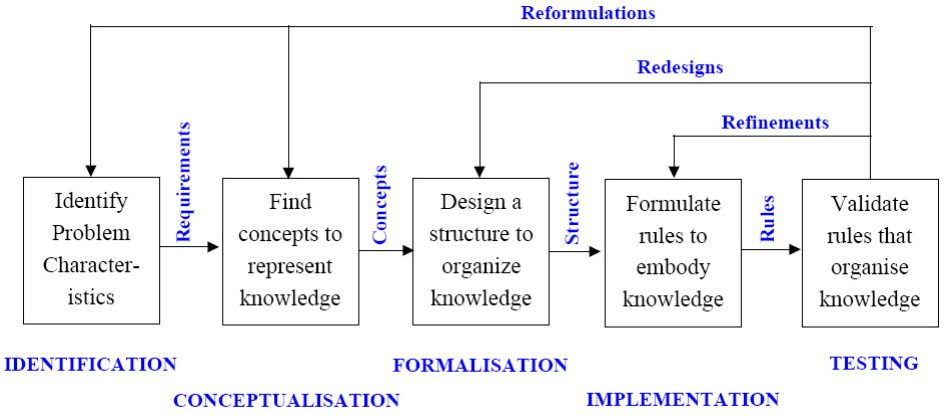


Figure 5.9: Image describes the iterative process of knowledge acquisition, including all required steps to achieve a knowledge base for an expert system.

Difference between expert systems and other artificial intelligence programs

Many non-expert system based artificial intelligence programs require a simplification of real world problems, because they reduce them to abstract mathematical problems. Then, real objects are often described by a set of features, projected in an Euclidean feature space. For the classification of these objects only these feature points are used. Instead of explicit knowledge modelling, the intelligence consists of detecting a kind of inherent structure in the data, which simulates knowledge. The SVM classification [8] e.g. is based on the inference of a multi-dimensional hyperplane, which is able to separate objects of different classes at the best. With that data driven strategy a lot of problems can be solved. Domain expertise mainly lies in identifying suited features of the objects of interest. The mathematically based reasoning itself is then often hidden in a black box and is domain-independent, which allows the reusage of algorithms for diverse fields.

In contrast to that, expert systems deal with particular problems of realistic complexity that normally requires human expertise to infer a solution. They offer elaborated facilities to introduce domain specific knowledge, are more transparent during the reasoning process and provide an explanation for their decisions. For each problem domain a suited knowledge representation must be found to formally describe this field specific knowledge. With formal description an unambiguous language is meant, which has a well-defined syntax determining the form of expressions in the language, and a well-defined semantics which uncover the meaning of such expressions. As a result of the concept of clear separation between knowledge base and inference engine, several knowledge independent frameworks were published. Here the domain knowledge can be inserted separately to produce a working expert system.

5.2.2 Expert systems and CSR

A more chemical knowledge based reconstruction of molecule images presumes a consistent and standardized way to formulate chemist expert knowledge. After the initial conceptual groundwork of expert systems was laid in section 5.2, this subsection proceeds to detailing the usage of these systems for the CSR domain.
Finding a qualified knowledge representation is the most complex task in developing an expert system in general. Before these systems allow deductive reasoning, domain specific knowledge must be engineered into a form that can be embedded in the program.

Chemistry Knowledge Representation

Before the chemistry expert system was developed, all thinkable molecule reconstruction strategies were collected and categorized in two knowledge branches.
On the one hand the knowledge about how structural formulas are drawn is essential. The widely spread acceptance of structural formulas resulted from the clear and intuitive definition how a semantic entity has to look like and in which drawing context it can be used. Although there can be large variances in these restrictions, the overall requirements must always be satisfied. That is why these structural constraints form a crucial part of the defined knowledge of the expert system.
On the other hand, this knowledge is often not sufficient to identify all parts of a molecule correctly (compare section 4.3.3), because it does not consider the underlying chemistry. For that, a second knowledge branch was introduced which tries to represent the basic chemistry like valences and chirality.

After collected the most important facts and requirements of both branches, a fundamental design decision was required: how this knowledge can be represented for the purpose of automated reasoning? For the representation of knowledge, there exist numerous possibilities, like semantic networks [14], taxonomic classifications [7] and relation systems [7].
For the CSR expert system, the application of so-called conditional rules seemed to be suitable to express the required chemical and structural knowledge.
A rule has always the following form:

'IF A and B are true THEN conclude that there is evidence that C is true'
or abbreviated through $A \wedge B \Rightarrow C$

The precedent of the rule is called the *premise* or *left-hand side (LHS)*, whereas the other part is denoted as *consequence* or *right-hand side (RHS)* of the rule. The consequence is inferrable from the premise, which is in turn a conjunction of predicates. In general a predicate is an operator if applied on some input returns a boolean value. A rule was successfully if all predicate have been true, otherwise the rule failed.

This knowledge representation was chosen because it implicates several advantages.
The cognitive psychology differs in different forms of knowledge, such as knowledge about facts (e.g. the chemical element abbreviation of Iodine is 'I') or declarative knowledge (e.g how to draw a structural formula). The application of conditional rules for specifying these types of knowledge is very similar to the human thinking and proceeding. At first certain requirements of an image symbol and its context are checked. If assumed requirements were correct, the human perceives a symbol's meaning for the molecule. For this reason, rules are intuitive to interpret and formalize, which simplifies the development and maintenance of the required CSR knowledge.
It is intended that each rule is a distinct entity of a specific chemical knowledge fact. The collection of several rules form the so-called *knowledge fact base*, representing all knowledge which is covered about the reconstruction domain. Through the central administration in an unique rule repository it is simple to keep track of already covered knowledge and to maintain the rules. Because of the inherent modularity of the knowledge base, it is relatively easy to update. Individual rules can be inserted, deleted, or modified without drastically influencing the overall performance of the expert system.
As mentioned in the introduction part 5.2.1, expert systems need to provide explanations and justifications for their inferred solutions. Through the application of rules this is easy to realize. Each rule, as well as each included predicate is associated with an unique identifier, which can be used to log each single decision step in the reasoning process. With this proceeding the successful rules as well as the failed rules can be collected. So it is easy to follow, why the computer assigned an image symbol to this chemical entity and not to another. Based on this clear rule identification, statistics can be calculated, which allow the improvement of the existing knowledge fact base. Therewith, it is straight forward to detect weak rules, which are not often used and often applied rules, which enable strong discrimination. Through the unique rule identification it also can be avoided that a user has to work with the abstract rule formalism. Instead of that, a human readable text translation can be retrieved, describing on which idea the rule is based on and which predicates have been applied in the premise.
All these reasons led to the decision, to apply rules as a qualified representation form for the chemical knowledge in the CSR expert system.

Implementation CSR expert system

After evaluated some public available expert system frameworks, like Drools[2] [52] and Jess[3] an own CSR expert system was developed and implemented. The existing approaches required a relatively high degree of familiarisation effort and would have been time consuming to adapt for the required purpose. The implementation of the CSR expert system mainly consists of the knowledge base containing the CSR knowledge and the inference engine, responsible for the reasoning (compare section 5.2.1). Both concepts and their implementations are presented in the following two subsections.

KnowledgeBase

The knowledge base is the program's store of facts and associations. It "knows" about the chemical structure reconstruction subject area. These facts are formulized through rules, which have different so called granularity levels. It can be differentiated between approximation and itemization rules. The idea behind that is the following: To obtain a maintainable knowledge base, it is necessary to keep the number of rules and their complexity as small as possible. It is not reasonable to develop embracing rules which fit perhaps perfectly to a certain depiction set but will fail as soon some variations occur. This overfitting effect can only be avoided through the development of simple but general rules, based on as less predicates as possible. Their discrimination power is achieved by the application of the most significant properties of the individual chemical semantic entities. Often already simple structural considerations are sufficient to discriminate ambivalent possibilities. That is why different granular rule levels have been introduced. The so-called *approximation rules* allow a preclassification of possible entity environments, where a symbol can occur. Although numerous entities have to be recognized, each entity can be assigned to a distinct so-called *structural family*. For instance superatoms, SMILES, charges, markush groups, abbreviations, captions and solvents have all in common that they are all string associated. Thus coarse approximation rules try at first to assign each image symbol to one of these structural families. In addition to the string associated approximation, in can be differed into bond (single bond), bond set (larger vector set containing several bonds), bond associated (dotted chirals) and external (reaction arrows) symbols. For each family, a set of rules was defined, which are mainly structural based. If no rule of these listed approximations was successful for a symbol, an unknown approximation label is assigned.
This kind of preclassification is then the starting point for a more detailed so-called *itemization procedure*, where e.g. a superatom can be distinguished from a SMILES string. Also for this detailed recognition, itemization rules have been formulated, which are applied for the extraction of the concrete chemical entities. In contrast to the recognition and reconstruction separation of chemoCR (refer chapter 4), the new approach additionally includes an extraction level. That is realized to keep approximation as well as itemization rules as simple as possible, because they do not have to cover recognition and extraction knowledge. One the other side, the extractors consuming the itemization rules can still incorporate the structural family knowledge of the more coarse approximation rules. This process is explained in section 5.4, where the entire workflow of the concept is described.

[2] `http://labs.jboss.com/portal/jbossrules`
[3] `http://herzberg.ca.sandia.gov/jess/index.shtml`

The rules for the CSR expert system were implemented in a logically consistent manner. The left-hand side of a rule contains a conjunction of predicates, which lead in case they are proven to be true to the consequence. The body of the rule is actually a piece of JAVA code, and "evaluating" a rule entails the executing of the implemented predicate methods. The consequence of an approximation rule assigns a provided symbol to a possible structural family. Based on the associated structural family, specific extractors try to extend the symbol to a full semantic entity, where it can occur. An extractor uses itemization rules to decide if adjacent symbols belong to the same entity and are therefore accepted for the extension. Itemization rules can be separated into extension and validation rules. The first group indicates if an adjacent symbol is an extension symbol or not. After the extension has been finished, validation rules are applied to decide if the extracted symbol set can form a valid chemical semantic entity.
For instance, there exist different string associated entities, like SMILES and superatoms. A character symbol can appear in both entities. Before being able to clearly decide to which exact entity the current symbol belongs, it is first necessary to obtain the entire string. The symbol set derived from the extractors, is then the starting point for validation rules. They check the validity of the entities by testing requirements like, does such a superatom exist or is the extracted string in a valid SMILES notation.

It has already been mentioned, that the rule implementation is based on executable JAVA predicate methods, which consider all kind of information for their reasoning process.
For testing diverse symbol properties and symbol arrangements, a central management and access system for all input information was required. A so-called *analysis system* (*AS*), was implemented, which administrates the existing information level of connected components, vectors and characters, as well as the new introduced information components *RTree* and orientation graph.
During the elaboration of the reconstruction system, it was soon realized, that information which is derived during a recognition process could be very helpful for a more stable recognition. For instance the string associated entities *solvent* and *atoms* can be identical strings, although they encode two complete different meanings for the molecule. The first one is an additional information about the experimental conditions, the other one represents an essential part of the molecule. The only way to discriminate such ambiguities, is to introduce metainformation, like distances and character sizes derived during the recognition process. A separate level called metainformation was additionally implemented in the *AS* because this kind of discrimination information is often required. Here all size and distance values can be updated, which have been observed during runtime.
Additionally to its administration functionality the *AS* provides comprehensive spanning information retrieval methods, which allows to set the individual levels into context.
In the following a short overview about the developed rules, their predicates and their underlying technique is given. A CSR rule consists of several attributes. Beside an unique integer identifier, it holds a name, a native language description, a priority value and a collection of predicates, which are assigned to so-called *requirements*. Each CSR predicate is an elementary JAVA method, which can operate on different information levels. It matches a defined precondition against the dynamic information resources of the analysis system. A predicate can be easily reused by several rules, because it evaluates a

well-defined condition and is administrated in a central PredicateBase, which is also part of the expert system.
An approximation rule assigns to each symbol a structural family, based on three kinds of requirements. Each of these contains in turn a selection of predicates. On the one hand a symbol must fulfil itself already certain properties to be part of a certain semantic entity. A symbol requires at least an OCR character match of its corresponding connected component, to fall into the string association class. Such constraints are formulated in a symbol requirement, which exist in each rule exactly once.
Symbols are often only a part of a larger entity, like a character in a superatom or a line in a dotted chiral. So it is still necessary to test also the properties adjacent symbols. Severe problems mainly arise through overloaded symbols (4.3.3), which can occur in different structural families. A 'I' symbol e.g. can be string associated, could be a bond but also only bond associated. To resolve this difficulty it is required to take also the neighbors into account. Within a rule this is done by so-called *context requirements*. It is similar to a symbol requirement, because it also tests the properties of a symbol. But in contrast to that, it is clearly associated to a certain reference symbol. Each rule requirement holds an unique identifier, with which it is possible to refer to a specific already tested symbol. The 'I' symbol can be e.g. a character if it is part of a Chlorine atom. Here a context symbol requirement can be defined, which expects a character 'C' close to a symbol with 'I' properties.
A third so-called *relation requirement* was introduced, because this character must be in a certain arrangement to the 'I' and the distance between the two symbols should be under a certain threshold. Relation requirements consider the relation properties between two symbols, which satisfies already symbol or context symbol requirements of a rule. By the introduction of the relative position concept, arrangement between symbols is simple to specify. The surrounding boxes of two connected components can be used to measure the angle between the line connecting their centers and the x-axis. This degree value can be discretized into eight distinct compass values such as SOUTH and SOUTHEAST. Although this concept does not work properly for large components containing e.g. bond sets, it is quite optimal dealing with character symbols. With that, a chlorine rule can be defined, specifying that an 'I' symbol requires a character 'C' match symbol in its context, which is situated in the WEST and is sufficiently close.
With a similar proceeding also Iodine can clearly distinguished from other structural families. A corresponding rule assigns a symbol to the string associated class, if no other symbols are in the local neighborhood. This might discriminate a Chlorine assignment, but is still not enough to throw out the structural families of bond association and bond. Chirals in turn can be discriminated by avoiding adjacent parallel lines. The exclusion of the bond family is based on a combination of meta information and real valence chemistry. It is tried to use typical character sizes in the Iodine inference. If the the 'l' symbol is significant larger as already observed characters, it cannot be an atom character under the assumption that characters do not change the size. Unfortunately also chemical drawings exist, where bonds and characters have a similar size. In this case the usage of valence information turned out to be a powerful recognition criterion. Valence information indicate the number of chemical bonds formed by the atoms of a given element. Iodine is a so-called halogen, which accepts through its electron configuration only one bond. This fact can be used to test, if there is exactly one close bond or bond set in the 'I' symbol neighborhood, where the extension of the closest vector intersects the line character.

```
IF
{
SR(1){+isOneVector AND +hasOrientation(RelativePosition.TWELVE)} AND
CR(2,1){+isCharacter('C')} AND
RR(1,2){+distanceSmallerThan(METAINFO.AverageCharacterSize) AND
    +relativePosition(RelativePosition.EAST)}
}
THEN Approximation.StringAssociation
```

Figure 5.10: Example of a Chlorine approximation rule, identifying an overloaded line symbol as string associated. SR is the abbreviation for symbol requirement, which determines the requirements of the symbol, which has to be assigned to a certain structural family. The '+' in the rule indicates that a predicate must be true to be correct. Here the symbol must be a line which is drawn from the top to the bottom (= 12 o'clock orientation). In the context of the line symbol (the second number in CR(2,1) reference requirement 1) there must be a character 'C'. If such a neighbor exists, the relation of the two symbols must fulfill certain conditions (expressed by the corresponding reference identifiers in RR(1,2)). The distance must be smaller than the average character size of all observed characters and the 'C' must be in the East of the line symbol. If all three requirements were correct the line symbol is assigned to the approximation StringAssociation.

With this symbol-context-relation requirement concept advanced structural as well as chemical knowledge can be formulized and applied to resolve all kind of ambiguities. By the usage of logical operators like And (conjunction), Or (disjunction) and negation predicates as well as requirements can combined to form powerful discrimination rules. In the moment there are 15 mainly structural predicates implemented, which are not all applied by the rules. The current CSR expert system manages to assign each symbol to one structural family based on seven approximation rules. The extraction and the validation incorporate again round about eight rules.
A number of constraints influenced the design of the expert system. One important claim was the simple extension and maintenance of the system's knowledge, because a permanent increasing structural formula space is assumed. In the best case an expert in chemistry should be able to simplly specify new recognition rules based on the provided rule language. These requirements were addressed by implementing the JAVA reflection API[4] for the inference system and the rule specification. This API [5] allows the invoking of JAVA classes and methods during run time. A chemist e.g. just needs to know which predicate methods are available or have to specify new ones for the programmer. Then the expert can easily formulate rules (compare 5.11) in a textual XML[6] editor and load

[4] http://java.sun.com/docs/books/tutorial/reflect/index.html
[5] Application Programming Interface
[6] Extensible Markup Language

the specified rules into the experts system knowledge base, which can be automatically considered in the next inference process.
After explained the structure of the rules and their implementation details, the following section considers the reasoning with them.

Inference engine

This section details the implemented inference process of the CSR expert system. In general, inference is the process of deriving a conclusion based solely on what is already known. Here, a so-called deductive reasoning is applied, where the unique evaluation of the rule is reached from previously known facts (the premises). In the case that these premises are true, the conclusion must be also true. Other inference concepts, such as abductive and inductive reasoning seemed not suited here, because their premises may only predict a higher probability of the conclusion, but do not ensure that it is true.
The evaluation of the rules is done by an inference engine, which can base on different reasoning strategies. Two common methods for reasoning with rules are forward and backward chaining [25].
Forward-chaining scans the rule base until a rule is found where the left-hand side is true. Its consequence is in turn again new fact knowledge for a next scan to detect more succeeding rules. This process is repeated until no further facts can be inferred. Forward-chaining inference is often called data driven, because the starting point is the data which applied on the rules produce conclusion results, which are unpredictable at the beginning. All successful rules must be analyzed, to detect if a certain consequence was true.
In contrast to that the goal driven Backward-chaining works exactly the other way round. It starts with a list of goals (e.g. hypothesis about data) and works backwards to see if there are data available that will support any of these goals. So, to the beginning it selects the rules, which contain a desired goal in their THEN clause. If the IF clause of the corresponding rules are not known to be true, they are added to the list of goals. In order to confirm a goal, this procedure is repeated until no further goals to add are available or data confirmed the LHS of a rule.
Which of the two strategies is chosen, often depends on the field of application. Forward and Backward chaining may vary in their number of inspections at each state (branching factor). In applications where the data is cached "in-memory" there might be no performance impact. On the other side if voluminous data must be first collected from a database, considerations about efficiency are useful.
The choice of course also depends on the problem, desired to solve with the expert system. Different tasks can be simpler to address with a forward chaining system that are complicated with a backward chaining system and vice versa. Forward chaining engines e.g. are more qualified for process monitoring, because they relies on the application bringing in new facts. A backward chaining engine would in contrast query for facts, which might not yet be available.

For this new reconstruction concept an own basic reasoning algorithm was developed and implemented. It realizes a data driven inference similar to forward chaining, whereas some aspects of this technique have not been incorporated. Rules are considered "chained" if they share conditions between each other. Chaining in turn allows that theses conditions are evaluated only once for all rules. Because the current expert system is based on only

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Rule id="4"
      name="chlorineRule"
      description="Identification of the l of chlorine"
      priority="100">
    <Requirements>
        <SymbolRequirement identifier="1">
            <Predicate>
                <Name>isOneVector</Name>
                <Sign>true</Sign>
            </Predicate>
            <Predicate>
                <Name>hasOrientation</Name>
                <Attribute>Twelve</Attribute>
            <Sign>true</Sign>
            </Predicate>
        </SymbolRequirement>
        <ContextSymbolRequirement identifier="2">
            <ContextId>1</ContextId>
            <Predicate>
                <Name>isCharacter</Name>
                <Attribute>C</Attribute>
                <Sign>true</Sign>
            </Predicate>
        </ContextSymbolRequirement>
        <RelationRequirement>
            <SymbolId1>1</SymbolId1>
            <SymbolId2>2</SymbolId2>
            <Predicate>
                <Name>distanceSmallerThan</Name>
                <Attribute>Metainfo.AvCharSize</Attribute>
                <Sign>true</Sign>
            </Predicate>
        <Predicate>
            <Name>relativePosition</Name>
            <Attribute>RelativePosition.West</Attribute>
            <Sign>true</Sign>
        </Predicate>
        </RelationRequirement>
    </Requirements>
    <Consequence>Approximation.StringAssociation</Consequence>
</Rule>
```

Figure 5.11: Chlorine rule specified in XML based rule language (compare figure 5.2.2)

on a few rules, consisting of a small set of predicates and no recursive rule definition is allowed, chaining has not been included in the implementation. The decision for a data driven inference resulted from the classification task to assign each image symbol to a structural family. The goal driven backward chaining seemed not qualified here, because priori there is no bias for a certain label.
The knowledge base contains for every structural family and for each itemization entity a set of rules, which covers various drawing cases. At first the coarse structural family assignment is inferred, by examining if at least one family set in the KnowlegdeBase exists, which contains a successful approximation rule. A rule succeeds if the requirements specified in the left-hand side, matches with the properties of the symbol to recognize, its context symbols and their relation properties between each other.

In the following the algorithm 2 is described, which checks if a symbol x satisfies a certain rule r. The inference process starts with the evaluation of all predicates of the symbol requirement, because each rule minimally consists of such a requirement.
Although there exist three kind of requirements, it exists only one unique method called 'checkRequirement', which is responsible to evaluate all predicates of a specified requirement. It loads for each predicate its associated JAVA method and based on the provided symbol parameter(s), the invokation of the method allows the testing of the defined condition against the corresponding information of the analysis system. A predicate e.g. might test if there is a character match for a symbol. Then the provided symbol identifier enables the invoked method to retrieve the corresponding OCR results stored in the *AS*. If there is a match the method return true, otherwise false.
If all conditions of the symbol requirement were true, the requirement id and the symbol to assign is stored in the associative array *acceptedMap*. In this array all processed requirements ids are keys, which are associated with arrays containing the symbols, which passed their corresponding predicate tests. The algorithm terminates if the *acceptedMap* is empty, indicating that all requirements of the rule were processed or stops at an earlier stage if a requirement evaluation was not successful. The next requirement to process, is determined through the first key entry called *reqId* of *acceptedMap*. Based on this *reqId*, all associated context requirements can be determined and can be hold in a list *contextReqlist*. So for instance if the first key was the id of the symbol requirement SR(1) of the ChlorineRule, mentioned in 5.2.2, the contextRequirement CR(2,1) can be identified and enqueued in *contextReqlist*. Before evaluating the constraints of the contextRequirement, the symbols which passed the requirement *reqId* are loaded from the *acceptedMap* and are stored in *symbolList*. They are necessary to retrieve the neighbor symbols, which are required to apply the context requirements of *contextReqlist*. So if more symbols were associated with *reqId*, various neighbors have to be loaded. Then for each *cr* of the requirements in *contextReqlist* a fitting neighbor symbol is tried to detect. The pseudo code of the shown algorithm shortens this procedure with the method 'inferAcceptedSymbols', which is applied on all neighbors at the same time. The method return the *acceptedSymbolList*, containing the symbols which succeeded to pass the predicates of the applied context requirement *cr*. If no symbol in the neighborhood could be found, which fulfill all claims of the context requirement *cr*, the inference method returns false. Otherwise it is checked, if there is a relation requirement considering the current processed context requirement *cr* and *reqId*. Back to the chlorineRule example, after CR(2,1) is processed, also the RR(1,2) can be evaluated, testing if the character 'C' is

correctly arranged to its reference point the symbol fulfilling the requirement SR(1). Is this the case the checkRequirement method is again applied on this requirement testing the relation between the symbol s belonging to *symbolList* and the current symbol t from the *acceptedSymbolList*. If the symbol t also passes this test it is added into the *acceptedMap*. The command 'insert(*acceptedMap*,*cr*,*t*)' checks if requirement *cr* is already a key in *acceptedMap*. Then t is added into the associated array. Otherwise a new entry with key *cr* and a new array with entry t is enqueued into *acceptedMap*. After all symbols from *acceptedSymbolList* were processed, it is checked if there are any entries at all in the *acceptedMap* for the current analyzed requirement *cr*. If there is no entry with *cr* as key the algorithm terminates and return false, because for the processed requirements no qualified symbol could be found. If there is an entry, there must be at least one detected symbol which fulfils the the processed requirements. Then the algorithm can continue with the next requirement. For that the already fully analyzed first entry in *acceptedMap* can be dropped. If there are no more requirements to process, the evaluation method can return true, because symbols with the claimed properties have been found in the neighborhood of the symbol x, which was to be assigned.
This rule inference is done for every structural family rule set. As soon a successful rule within a set was detected, the other rules of the set are no longer considered, because the symbol has already been assigned to the corresponding approximation. Rules from other structural families are still evaluated. Due to the fact that several rules of different approximations can be successful, a priority value is specified in each rule. In literature various strategies [52] for rule conflicts are proposed. So it is common that e.g. the most general rule or the first introduced rule prevails. For this approach the relatively simple strategy of rule dominance was chosen. If several rules with contradictory consequence occur, the result with the highest priority dominates. The priority value is an empirical value, based on the statistics how often a rule was the right decision in such a conflict situation. The CSR expert system has not yet made use of this procedure, because the current structural families are sufficiently different. If for a symbol no successful rule at all could be found an 'UNKNOWN' approximation label is assigned.

Algorithm 2: Rule evaluation algorithm

```
   input  : Rule r, Symbol x
   output: Boolean indicating if symbol x fit into rule r
 1 begin
 2   if (!checkRequirement(r.sr, x)) then
 3     return false;
 4   end
 5   acceptedMap ← initEmptyAssociativeArray();
 6   acceptedMap ← acceptedMap.put(r.sr.id, [x]);
 7   while acceptedList not empty do
 8     reqId ← acceptedMap.getFirstEntry();
 9     contextReqlist ← r.getContextRequirementsFor(reqId);
10     symbolList ← acceptedMap(reqId);
11     forall Symbols s of symbolList do
12       // symbol neighbors come from the orientation graph
         neighborList ← s.getNeighbors();
13       forall ContextRequirements cr of contextReqlist do
14         // inferAcceptedSymbols applies checkRequirement for symbol list
           acceptedSymbolList ← inferAcceptedSymbols(cr, neighborList);
15         relationReqList ← r.getRelationRequirements(cr.id);
16         forall Symbols t of acceptedSymbols do
17           if (relationReqList not empty) then
18             forall RelationRequirements rr of relationReqList do
19               if (checkRequirement(rr, s, t)) then
20                 // insert symbol t in associated arraylist of key cr
                   acceptedMap ← insert(acceptedMap, cr, t);
21               end
22             end
23           else
24             acceptedMap ← insert(acceptedMap, cr, t);
25           end
26         end
27         if (acceptedList.contains(cr.id)) then
28           acceptedList.removeFirstEntry();
29         else
30           return false;
31         end
32       end
33     end
34   end
35   return true;
36 end
```

5.3 Graph exploration based reconstruction

In the previous section an expert system was shown, which is able to assign to each symbol in a chemical depiction an approximation label, indicating to which semantic element the symbol belongs. For the reconstruction of the entire *molecule graph* it is additionally required to identify all semantic elements with all participating symbols as well as a semantic meaningful linkage between these elements.
The chemoCR project recognizes each semantic entity in an own isolated global walk through the picture. Instead of doing this overall processing, the new approach implements a specific manner how the symbols are explored. Which token is visited next depends on the current analyzed symbol and the properties of its neighborhood.
This context specific analysis is enabled by a so called constraint based exploration of the orientation graph (refer to 5.1.2). Before this new exploration is explained in section 5.3.2, a general overview about graph traversal is given.

5.3.1 Introduction graph traversal

Traversing a graph in a systematic manner is a fundamental problem of graph theory [11]. Exploration algorithms find numerous applications in diverse fields. This can vary from the simple tasks like enumerating the contents of each vertex to more complex problems like identifying shortest paths [22] between two vertices which is a typical scenario for GPS routing.
An important requirement for the correctness of a graph exploration is that it leads through every edge and vertex in the graph. Here the algorithms often assume that the graph is connected, so each vertex is reachable from another. For an efficient systematic traversal it must be avoided to visit vertices several times.
The most graph exploration algorithms base on the idea to label a visited vertex and to hold not yet fully explored nodes in a processing list. It is a common way to use different vertex states which all vertices pass. If the vertex has not yet been visited it is labeled as unexplored. After it has been encountered it exchanges its state into explored if not yet all incident edges have been visited. Otherwise the vertex change the state into fully-explored. The traversal algorithms starts with an initial start vertex, which is marked as explored. At the beginning all other nodes are marked as unexplored. Before a vertex can change its state to fully-processed, all adjacent neighbor vertices must be visited. If a vertex has not yet been discovered, it is marked as explored and added to the processing list. The neighbor is ignored if it has already been visited or fully-explored.
Different traversal algorithms mainly distinguish in their vertex processing order, which depends on the applied data structure to store the explored nodes. The two most common used methods [11] are breadth-first (BF) and depth-first (DF) traversal. The first is based on a first in, first out (FIFO) queue, whereby the oldest unexplored vertices are visited first. Consequently the exploration radiate out slowly from the starting vertex. In contrast to that the depth-first traversal implements a last in, first out (LIFO) stack, where the youngest vertex is explored next. With that strategy the exploration quickly leads away from the original starting point, by discovering a new neighbor if available, and reseting only when no further vertices are available to visit or the surrounded vertices have been already explored.

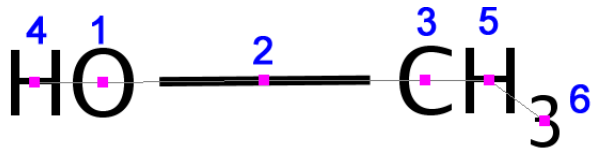
5.3.2 Constraint based graph traversal

The current chemoCR software explores all symbols of the image several times. For each chemical entity an own run through the entire picture is performed.
The new reconstruction approach builds on the application of a more chemical knowledge based recognition process. This is realized by the advanced analysis of the spatial context of each image symbol, which is matched against the rule sets of the expert system. A new strategy to explore the symbols of a chemical depiction was required, because the overall proceeding of chemoCR does not support this analysis. Spatial context is specified in the new developed reconstruction approach through the *OG* (refer to 5.1.2). For that reason, the task of visiting all image symbols for the reconstruction, can be reduced on the exploration of the orientation graph.
In section 5.3.1 the two common used breadth-first and depth-first traversal algorithms were described. As a result of their traversal behaviour (compare 5.3.1), the application of these established graph exploration methods are not suitable here. Where the breadth-first exploration radiates out slowly, the depth-first traversal quickly leads away from the starting vertex.
To achieve a context specific analysis for a knowledge based recognition the neighborhood of a vertex has to influence the visiting order of the *OG*. A so called constraint based graph exploration algorithm was implemented which satisfies this requirement.
In comparison to breadth-first and depth-first exploration the vertex processing order as well as the insertion proceeding is more complex. The traversal order of the new algorithm is determined through so called constraints. Instead of visiting the vertices in a fixed order, each vertex just represents a seed for one or several semantic entity extractors. It is tried to fully extend the seed to its complete semantic entity where it occur. This extension is based on the itemization rules of the expert system (refer 5.2.2) and also use the orientation graph. With that, the extension is a kind of subgraph exploration. For that reason, the further exploration of the graph is constrained through the resulting extracted symbol superset, because some vertices have then already been visited during the extraction process. So discovering supersets of symbols which belong together at first before continuing with the neighborhood of their vertices is the main idea of the constraint based exploration.
The basic principle of the algorithm is similar to the proceedings mentioned in the graph introduction section 5.3.1. In each exploration step a seed vertex is dequeued from a *seedlist*, which administrates the vertices still to process. This associative array holds entries which consist of single vertex identifiers as keys, associated with arraylists of neighbor vertices as values. The simple arraylist *seedlist* administrates the order of insertion of the seedlist entries, because these cannot be reproduced from an associative array. A *visitedlist* additionally stores all vertices which have been visited so far.
The exploration of the *OG* starts with an arbitrary vertex, which represents an easy identifiable non-overloaded image symbol (see section 4.3.3). The vertex is queued in the *seedlist* and as value in the *seedlist*, whereby an artificial key 0 is used here. After this initialization the iterative travsersal of the *OG* can start. Each exploration step consists of two routines.
The first routine is responsible for selecting and extending the next seed vertex from the *seedlist*. This vertex is always the first list value of the entry with the key equal the first item of the *seedlist*. It is tried to maximally extend the seed to its superset, containing

several vertices. This node set can contain only neighbors of the seed vertex as well as more distant ones, which do not have a direct edge with the seed. To be a valid update step, the set must contain at least one neighbor of the seed. How the extension to the superset by the extractors is realized will be described in section 5.4. If they are not yet contained in the three data structures, all superset vertices are propagated to the *seedlist* and *visitedlist*. In addition each superset vertex is pasted with its corresponding neighbors into the *seedlist*.
The second routine transforms the data structures in a consistent form, enabling an efficient and correct traversal of the rest of the graph. Already explored vertices, indicated by the *visitedlist*, are removed from the value arrays of the *seedlist*. Then all entries are removed from the *seedlist* and the *seedlist*, whose key is associated with an empty arraylist, which results from the fact that all connected neighbors have been already visited.
These two routines are repeated until the *seedlist* is empty, indicating that the complete OG has been explored.

Algorithm 3: Constraint based exploration algorithm

```
   Data: Graph g
 1 begin
 2     // find optimal start vertex in g and init data structures
       pointerList, seedList ← initStartConfiguration();
 3     visitedList ← initEmptyList();
 4     while (pointerList not empty) do
 5         seedvertex ← seedList.getFirstValue(pointerList.getFirst()) ;
 6         // get constraint which includes several vertices of g
           superset ← generateConstraint(seedvertex);
 7         visitedList, seedList, pointerList ← propagateToList(superset) ;
 8         seedList ← seedList.removeAllValues(visitedList);
 9         removeList ← seedList.getEmptyValueEntries();
10         seedList ← seedList.removeEntries(removeList);
11         pointerList ← seedList.removeEntries(removeList);
12     end
13 end
```



Step	Seed	VisitedList	PointerList	SeedList
0	4			0:(4)
1.1		0, 4, 1	0, 4, 1	0:(4); 1:(2, 4); 4:(1)
1.2		0, 4, 1	1	1:(2)
2.1	2	0, 4, 1, 2	1, 2	1:(2); 2:(1, 3)
2.2		0, 4, 1, 2	2	2:(3)
3.1	3	0, 4, 1, 2	2, 3, 5, 6	2:(3); 3:(2, 5); 5:(3, 6); 6:(5)
3.2		0, 4, 1, 2		

Figure 5.12: Example of a constraint based exploration of the orientation graph of the simple structural formula shown in the above figure. Before the traversal can begin the exploration datastructures must be initialized. For that a suited start vertex is choosen. This is usually a symbol which is easy to identify and not overloaded (see 4.3.3). Here, the vertex 4 (which represents the character H) is selected. By the help of an artificial key 0, which is only used for the initialization, this vertex 4 can be included in the seedlist. In addition, the artificial key is added to the pointerlist. With this initialization the exploration can now be started. In each step a well-defined vertex (compare algorithm 3) is selected from the seedlist and propagated to a qualified extractor (see 5.4). Here, the seed is extended to vertex super set until all underlying symbols, wich belong to same semantic entity, were extracted. For this reason the seed 4 (character H) results in the superset (1, 2), because the characters O and H belong together in the shown molecule. Which extractor is selected, is determined by the expert system which is introduced in section 5.2.2. So the traversal is being constrained due to the expert system and the extraction process. The vertices in the detected superset are updated into the corresponding data structures and the exploration continues with the next seed vertex. For a better understanding it is distinguished between the insertion and deletion routine (compare the section 5.3.2), where e.g. 1.1 indicates the status of the data structures in step 1, after the extracted superset vertices have been included and 1.2 after visited vertices have been deleted.

5.4 Implemented reconstruction workflow

After having introduced the orientation graph, the CSR expert system as well as the constraint based graph exploration, this section explains their application in the implemented workflow of the new reconstruction concept.

5.4.1 Initialization of the analysis system

All required information levels of the analysis system have to be calculated, before the pattern recognition can start. This initialization begins with the loading of a chemical depiction from the GUI or the command line interface. At first the image is binarized (refer to section 3.1), where grayscale or color information is reduced on a simple 0-1 image representation. If a pixel intensity value is greater than a certain threshold it becomes a foreground (1) otherwise a background(0) pixel. Based on this data the segmentation process (3.1) extract connected components, representing the individual symbols in the drawing. Each component is send to an OCR procedure (see 3.2) to infer all possible character matches and to a vectorization algorithm (3.3) deriving a vector representation of the symbol. For the implemented workflow the available image processing modules of chemoCR have been used to compute these three information levels. Then the orientation graph and the spatial index structure *RTree* is generated, which both are applied on the already calculated connected components. The *RTree* (see 5.1.2) allows querying for symbol neighbors and the *OG* (refer to section 3.2) describes in general the spatial arrangement of the image symbols. In addition to the already mentioned information levels of the *AS*, an empty annotation container and explanation component is created. During the recognition, the annotation data structure logs all identified semantic elements and their assigned symbols. The explanation data structure is updated during runtime, with all applied successful rules. The last component of the analysis system is the meta info container. It is initialized with all values, which can be already computed. The average character size is inferred, by iterating through all character matches. Overloaded symbols (refer 4.3.3), numbers and and special signs like '+' are excluded from the calculation, to consider only atom character sizes. A similar routine derives the average size of isolated single vectors, which do not have a 12 o'clock orientation (compare 5.2.2). All other values of the meta info container are updated during runtime.

5.4.2 Initialization of graph explorer

The reconstruction starts with the completion of the initialization of the analysis system. A so-called *GraphExplorer* (*GE*) is responsible to explore the *OG* in a constraint manner. At first the *GE* tries to identify a qualified start vertex within all connected components. An optimal vertex would be a trivial non-overloaded character which has only a few neighbor nodes in the *OG*. With that the *GE* can initialize its inherent exploration data structures (compare 5.3.2) and can start the traversal. The exploration continues as long as the *poinertlist* of the *GE* holds vertices to process. It is empty, if all symbols of the image have been visited and recognized. In each iteration step a seed vertex is selected, is analyzed by the expert system and extended to a vertex superset by diverse extractors. These steps are more explained in the following subsections.

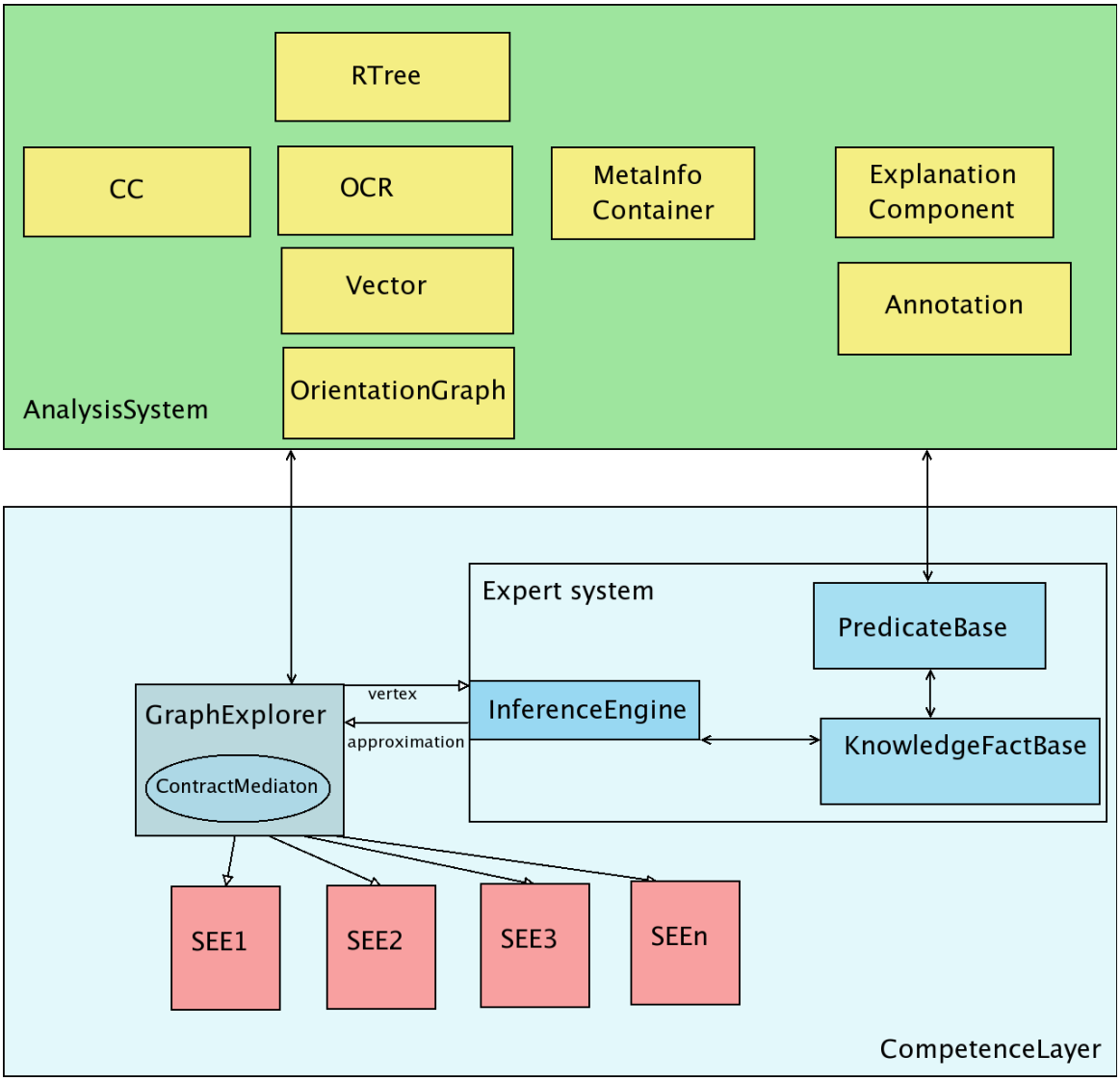


Figure 5.13: The architecture of the new reconstruction concept. In can be distinguished into two layers. The analysis system holds all extracted information levels which are required for the pattern analysis. In the competence layer the entire recognition and extraction take place. The graph explorer is responsible to visit all connected component vertices of the orientation graph. Each vertex is forwarded to the expert system, which tries to recognize the structural family in which the underlying symbol can occur. Based on the approximation of the expert system, specific semantic entity extractors (SEE) are instructed. They extend the seed vertex to a vertex superset, representing the whole semantic entity in which the seed occur. All vertices of the superset are propagated to the data structures of the graph explorer and the exploration continues.

Algorithm 4: Algorithm to recognize and extract all SEs in a chemcial drawing

```
   input  : AnalysisSystem as, ExpertSystem expert
 1 begin
 2     ge ← initGraphExplorer(as.getOrientationGraph);
 3     while (ge.pointerList.size > 0) do
 4         key ← ge.pointerList.get(0);
 5         vertex ← ge.seedList.get(key).get(0);
 6         approximationResultList ← expert.inferApproximations(vertex);
 7         suggestions ← initEmptyList;
 8         foreach approx in approximationResultList.getKeySet() do
 9             extractorList ← ge.getExtractorsFor(approx);
10             foreach extractor in extractorList do
11                 suggestions ← suggestions.add(extractor.inferEntity(vertex);
12             end
13         end
14         if (suggestions.getSize() == 1) then
15             annotation ← suggestions.get(0).getAnnotation();
16             superset ← annotation.getSuperset();
17             updateGraphExplorer(superset);
18             updateAnalysisSystem(suggestions.get(0));
19
20         else if (suggestions.getSize() > 1) then
21             //conflict handling if supersets differ
22
23     end
24 end
```

5.4.3 Expert system approximation

For each selected seed vertex the CSR expert system tries to approximate a structural family (compare figure 5.14), in which the corresponding image symbol would fit. This is realized through the matching of the properties of the symbol and its neighborhood against the rule sets of the expert system (refer to section 5.2.2). A rule consists of a collection of requirements and predicates, which are all evaluated by the inference engine applied on the vertex. If the premise of the rule was successful, the structural family approximation of the rule's consequence is assigned to the underlying symbol. For each approximation a set of rules exist. As soon as a rule has been true, the other rules of the same structural family are no longer processed, because the their consequence has already been assigned to the vertex. In contrast to that, rules from other structural families are still evaluated. A rule conflict emerges, if the symbol satisfies several rules from different approximation classes. Two strategies can be followed, to resolve this ambiguity. A more plain approach continues with that approximation, whose successful rule possesses the highest priority (see 5.2.2). The other strategy becomes clearer if the further workflow is more detailed. Due to the fact that every expert system needs to provide explanations and justifications of their solutions the identifiers of all successful rules are logged in the explanation component of the analysis system.

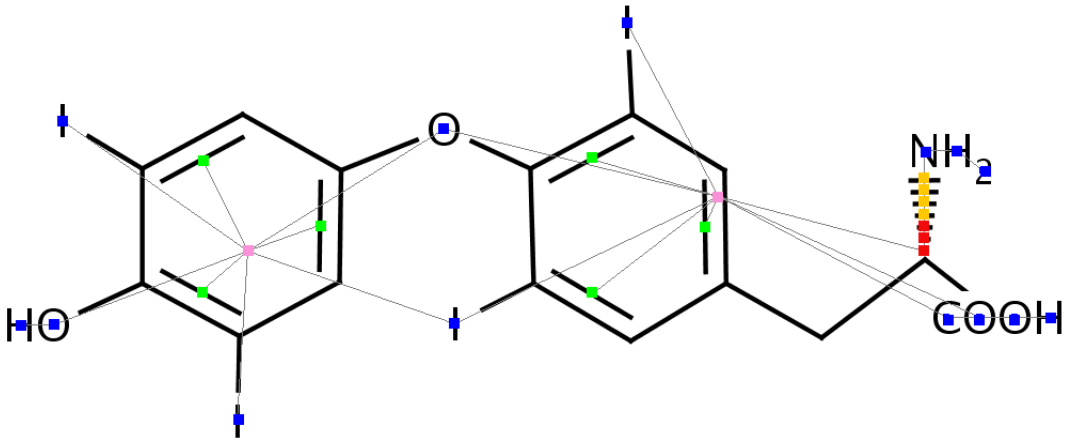


Figure 5.14: Orientation graph of a molecule with labeled vertices. The expert system recognized each connected component in the image and assigned a corresponding approximation label to its associated graph vertex. The color of the vertex indicates the assigned structural family approximation: Blue: string associated, green: isolated bond, purple: bond set, orange: bond associated, red: unknown.

5.4.4 Mediation to the extractors

The expert system inferred a preclassification of the seed vertex into a structural family, indicating if a symbol can be a bond, a bond set, might be bond associated, string associated, external or unknown (compare 5.2.2). It is still just an approximation, because the expert system analyzes a symbol's neighborhood only within a certain depth. In the optimal case, the structural family already discriminates enough to avoid contradictory entity assignments (refer to section 4.3.3). Now the itemization process is responsible to recognize the exact semantic entity of the symbol. For each approximation, a set of semantic entity extractors (*SEE*) exist. The GraphExplorer instructs the associated extractors to recognize the entire entity in which the symbol can occur, based on the expert system's approximation. In the case of the string associated approximation e.g. the vertex have to be mediated to numerous extractors, such for SMILES, superatoms, atoms, solvents, captions, R-groups, enumerations and abbreviations.

5.4.5 Semantic element extraction

Through the mediation of the vertex to appropriate extractors the itemization process begin. Each instructed *SEE* tries to recognize its supposed semantic entity, based on specified itemization rules. The vertex itself can already be such an entity (e.g. a single bond) or it can participate in a multi symbol entity like a dotted chiral. In the second case it is tempted to maximally extend the seed to a vertex superset, containing all symbols the entity is composed of. Thereto the *SEE* can systematically explore the neighborhood of the *OG*. Depending on the entity to identify, the context of the superset vertices and the specified extension rules this exploration can look different.
Assume that the seed is a character within a larger superatom string. It is reasonable to

process all remaining characters at first, instead of continuing with the other neighbors, belonging to other chemical entities. Not all superatom characters might have a connecting edge with the seed in the *OG*. For instance if the seed vertex is the character C in the superatom COOH then only the neighbor character O is relative close to the C. To obtain the entire string the neighborhood of this character in turn must be analyzed and so on.

The great advantage here is that for each symbol planed to add to the superset, the expert system can be asked if it fits into the assumed entity. Based on the approximation rules, the system will return the structural family of the candidate symbol. If this approximation does not match into the semantic entity, the symbol cannot be included.
The extension is continued until there are no more possibilities in the environment of the seed and the assigned superset vertices.
After that, the *SEE* applies diverse validation rules of the expert system on the derived symbol superset to decides if it was a successful extraction or not. For instance a SMILES extractor has an opening bracket in its superset, but no closing one. Then a validation rule would assess the extraction as failed, because the corresponding recognized string has no valid SMILES notation.
The contract of a *SEE* ends with submitting all collected and inferred information to the GraphExplorer in form of a *Suggestion*. A *Suggestion* contains a flag indicating if the extraction was successfully, all to the entity participating symbols, the semantic entity label as well as all pre-defined observed metainformation. In addition it contains all during the extraction or validation applied rules, which are later required for the explanation component of the *AS*. Several *Suggestions* might be forwarded to the GraphExplorer, because more than one extractor succeeded.

5.4.6 Suggestion propagation

Therewith the GraphExplorer possesses a collection of different *Suggestions*. Due to the high discrimination power of the expert system rules and the *SEEs* usually only one successful *Suggestion* remains. The *GE* is responsible to propagate all contained information into all affected data structures.
On the one hand it has to refresh its inherent exploration status. For that the extracted superset vertices are inserted into the *seedlist*, *seedlist* and *visitedlist* (compare section 5.3.2). Depending on which entity the seed vertex belonged to, the number of vertices in the extracted superset as well as the position of the corresponding vertices in the *OG* vary and will therefore bias the further exploration of the graph.
In addition to the inherent status of the *GE* the infered results of the *Suggestion* must be also propagated into the analysis system. An annotation, including all identifiers of the superset vertices as well as the assigned entity label is stored in the annotation container, which administrates all inferred annotations.
Supplemental to that the local explanation and the collected meta information is transferred into the explanation component and the meta info container of the analysis system.
In section 5.4.3 it was mentioned that there exist beside the rule dominance proceeding another strategy to resolve a multi approximation situation. The other strategy plans to mediate the seed vertex to all extractors of all approximation assignments, in the hope that one class of extractors completely fail. If the expert system inferred e.g. for a line symbol in the image a string associated and a bond associated approximation, different *SEEs* (like

length dotted chiral, Superatom, ...) would be instructed to compute their *Suggestion*. In the case the symbol is part of a dotted chiral, the string associated extractors will fail to extract their entities. With this procedure the original multi approximation problem is solved in an uncomplex manner.
If more than one *SEE* extracted a valid entity, their *Suggestions* have to be analyzed if all of them concern the same connected components. Although ambiguously all corresponding annotations can be in this case propagated into the analysis system and the conflict is resolved to a later point, when more knowledge is available or the user selects a solution. If the supersets concern different components, it is a complex problem which must be handled by manual intervention. Conflicts are at the most expectable from new chemical depictions, which contain situations where the discrimination power of the existing *SEEs* and the expert system is not sufficient. In this case, the intuitive applied rule language of the CSR expert system allows an appropriate addressing of these deficits.

5.4.7 Molecule reconstruction

If all symbol vertices in the orientation graph have been visited, the recognition and extraction by the expert system and the extractors are finished. Now the annotation container of the analysis system hold all information, which are required to reconstruct the molecule representation. Before the final result can be stored in a SDF file, several processing steps must be done previously.
The first reconstruction step is the assembly to the so-called *chemical graph*, which holds all recognized semantic entities and their linkage between each other.

In contrast to the molecular graph (see section 2.3.2), which only holds the atom and bond information of a single molecule, the chemical graph can handle all identified entities in the image. This is essential, because a chemical depiction might include more than just a molecule. Beside several molecules also external symbols, like captions, enumerations, solvents and reaction arrows might be in the image.
Here it is important to consider, that even the final SDF format do not cover all recognized entities. Although several molecules can be included and solvents and captions information can be stored as remarks, it does not support more advanced concepts like reaction schemes. New formats still have to be developed, where several molecules can be annotated (e.g. as educts and products) and not only atoms but also molecules can possess space coordinates.
For generating the chemical graph, it is required to infer which entities have to be connected with each other. For this analysis the recognized entities can be easily retrieved from the AnnotationContainer of the analysis system. Beside spatial closeness their connecting also have to consider chemical correctness to avoid problems like the semantical-physical distance problem explained in section 4.3.3.
After the chemical graph is fully established, it can be started to extract the desired molecular graph. Now the mapped entities of the chemical graph have to be converted into corresponding space coordinates and atom labels. Thereby the associated bonds have to be considered. Some entities still have to be replaced by a more suited representation form. For instance a dotted chiral annotation, still just contain the individual line symbols in its superset. In the conversion step, these symbols are substituted through a single vector, approximating the entire drawn chiral. Beside this annotation, also the bonds and bond sets must be further processed, because there can also encode atom information (see 4.2). For dealing with laborious string associated entities like superatoms and SMILES

the application of precalculated spatial templates turned out to be useful. So the strings are displaced by a kind of minigraphs, containing a collection of atom vertices with suited coordinates and well-defined edges. If all interpreted and substituted annotations of the AnnotationContainer have been incorporated the molecular graph finished and can be stored in a qualified molecule representation format.

Chapter 6

Results

The starting point for new concept was the existing reconstruction software chemoCR (see chapter 4), which is currently confronted with several significant bottlenecks, concerning robustness, extendability and mainly reconstruction accuracy. At the beginning an intensive study of the existing software and its underlying concept was required to enlighten the exact reasons for these bottlenecks. Here it comes clear that a lot of problems already emerge due to the simple recognition strategy (compare 4.2) of chemoCR.

For the development of the new concept it was tried to avoid the identified deficits. Originally it was planned to create a prove-of-concept by the implementation of a JAVA prototype which recognizes two semantic entities, e.g. dotted chirals and atoms. As a result of the beneficial concept the effort to recognize a chemical entity decreased remarkably and it was possible to create more than these two modules. In the meantime the new approach is able to identify all elements which can be also recognized by chemoCR. It can deal with atoms, superatoms, single and multi bonds, dotted chirals and larger bond sets. In addition, existing recognizers for thick chirals, cross bonds and bridges have been adapted and included in the new approach.

A main component of the novel strategy is the developed orientation graph (refer to 5.1.2), which is able to describe the spatial arrangement of the symbols contained in a picture. Therefore the idea of a relative neighborhood graph (see 5.1.1) had to be extended because it is originally defined only for a single point set. In contrast to that, an image of a molecule contains several symbols, whereas every symbol consists of a collection of pixels. So a new definition for relative closeness has been specified, which is able to deal with several point sets. For the calculation of the orientation graph an algorithm has been implemented which uses a *RTree* (see section 5.1.2) as spatial index structure. This tree finds in the new approach several applications, e.g. to infer the underlying symbol of the current mouse pointer position in the graphical user interface and in the extraction process. The derived orientation graph is the basis of nearly every proceeding step in the created recognition procedure.

Another essential component of the novel technique is the applied expert system (see 5.2.2), which allows the consideration of advanced chemical knowledge in the recognition process. After evaluated two existing open source projects, it was decided to design and realize an own knowledge representation (compare 4.3.2) and a suited inference engine (refer to section 5.2.2). Due to its implementation the expert system allows the outsourcing of the applied knowledge. For that purpose an own XML based format has been specified which enables the storage of rules e.g. in external flat files and databases. Aim of the rules is to recognize in which structural family a symbol in the chemical

depiction can occur. These rules are interpreted by the implemented inference algorithm which decides e.g. if a symbol can be part of a dotted chiral.

The new concept separates into the recognition step which is realized by the expert system and the extraction process which is performed by diverse extractors. Several of these modules have been implemented, which are able to extract atoms (one char and multi char), dotted chirals (length and cross dotted), bond sets, single and multi bonds.

A constraint based graph exploration algorithm (see section 5.3.2) has been developed and implemented, which allows a context specific processing of the symbols in the image. For that, the orientation graph is traversed and depending on the constraints provided by the expert system and the extractors the exploration can change. The reconstruction of the chemical drawing is finished if all vertices in the orientation graph have been visited.

The developed technique has been evaluated on a test set with 100 images. A structural formula required between 10-20 seconds to be reconstructed, whereas the recognition itself took less than five seconds. In the current workflow the OCR is the most time-consuming part. Each structural formula in the test set possesses at least four different entities and more than 30 atoms. The images were collected from different chemical depiction resources to obtain an impression how stable the reconstruction process is against drawing variants. Amongst others the test set contained molecular structures composed of the most sold drugs[1] of the year 2002. The whole recognition is done by the expert system (see section 5.2.2), which should also avoid contradictory recognition results between different extractors. For the current prototype 15 predicates have been created, which are applied in seven specified rules for the recognition of Iodine (1), Chlorine (1), bonds (2), bond sets (1) and chirals (2).

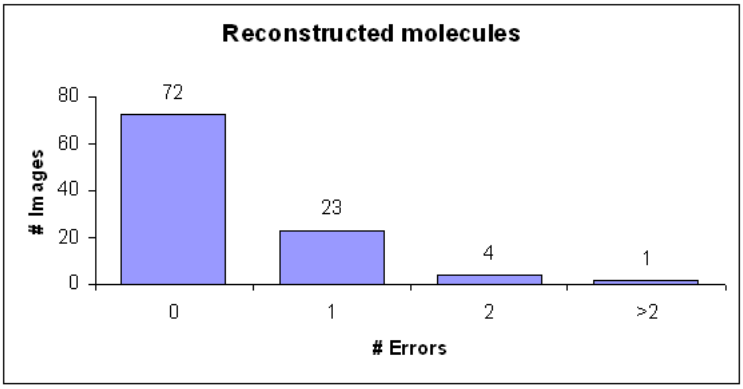


Figure 6.1: The diagram shows the of number of correct reconstructed molecules.

The accuracy rate of the new approach is quite promising (see figure 6.1). Of all images 72 molecules have been reconstructed completely correct. In 23 reconstructions exactly one error and in four cases two errors occurred. For the evaluation of the results, it was distinguished in four different error classes (compare figure 6.2): wrong clustering of close characters, missed bonds (single or multi), missed dotted chirals and confused single Iodine classification. It is to mention that the OCR, the vectorization process as well as the molecule assembly (see section 4.2) did not produce any errors, because all of them have been already optimized for the evaluated test set.

[1] http://www.rxlist.com

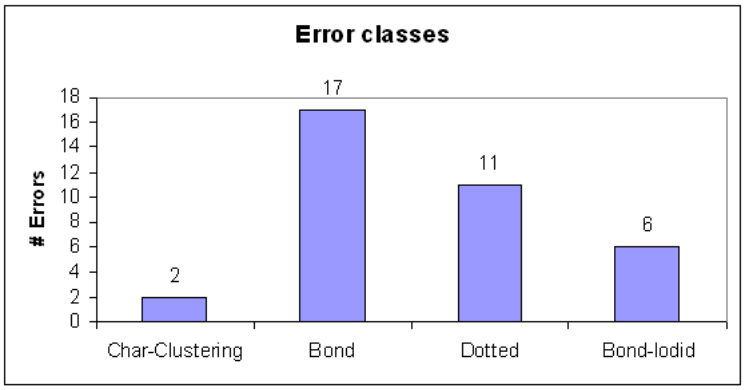


Figure 6.2: The diagram contains the number of observed recognition errors.
Here it is distinguished in four different error classes:
Char-Clustering: character from different atom strings were falsely combined
Bond: a single or a multi bond were not recognized
Dotted: the extraction of a dotted chiral failed
Bond-Iodine: confused a single bond with an Iodine or the other way round

In the following some wrong but also correct reconstructed images are presented and their particularities explained.

The most frequent error which occurred in the recognition process were missed bonds. Although the expert system correctly identified them as bonds the extraction process was sometimes not able to derive the corresponding multi bond. Figure 6.3 shows an example where the arrows indicate, which bonds were not correctly extracted as double bonds. This failure mainly occurred in images where different bond sizes have been observed. In this shown example the missed bonds are shorter due to the close characters of the heteroaromats. Hence the vector orientation as indicator for parallelism works less robust.

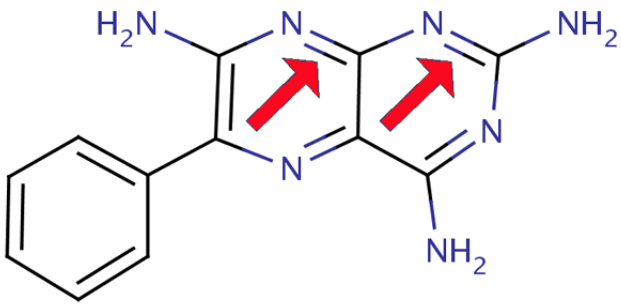


Figure 6.3: In the reconstructed molecule two double bonds (see red arrow) were missing

Another error which happened several times (7) was the confusion between an overloaded Iodine (compare 4.3.3) and a single bond assignment. The reason for that is the relatively weak recognition rule for Iodine which is currently applied. It does not consider advanced context analysis in its predicates and is mainly based on size and vector orientation values. In nearly all images where this error occurred, the bond had a similar size like seen in

characters (compare figure 6.4) and the majority of the already identified bonds were significantly larger. For that reason the Iodine rule falsely accepted the symbol as valid character.

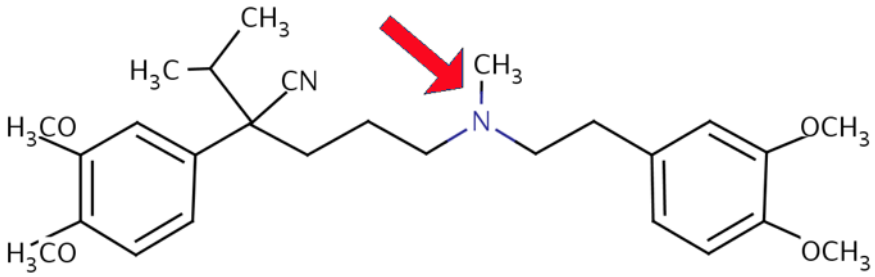


Figure 6.4: In the reconstructed molecule a single bond was recognized as Iodine

In contrast that the rule for Chlorine is less dependent on size occurrences and is mainly centered on context analysis. Therewith the rule works properly and its recognition accuracy is quite recognition accuracy is quite promising.

In chemical drawings often characters of different atoms are relatively close to each other. The problem of clustering characters belonging to a common string is addressed in the new approach by the consideration of distances and relative positions of the affected characters. The string associated extractor determines for a seed symbol the relative position of the closest adjacent character. With that, it is possible to derive if the symbol occurs in a vertical or a horizontal string. This direction influences the further extension of the superset and avoids the acceptance of characters which do not belong to the current processed entity. The clustering problem becomes even more difficult if overloaded line symbols (compare 4.3.3) emerge within close strings. Figure 6.5 illustrates an example, where the chlorine rule of the expert system and the new character extraction process led to a correct reconstruction of the molecule. In comparison with that, the other figure 6.6 shows the reconstruction result of chemoCR, which had difficulties with this molecule image.

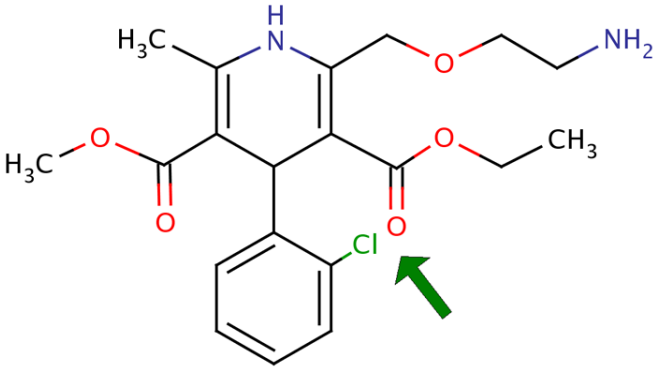


Figure 6.5: By the prototype reconstructed molecule. The expert system recognized the overloaded symbol 'l' in Chlorine correctly as string associated. Although several character of different strings are very close, the string extraction process combined the correct characters.

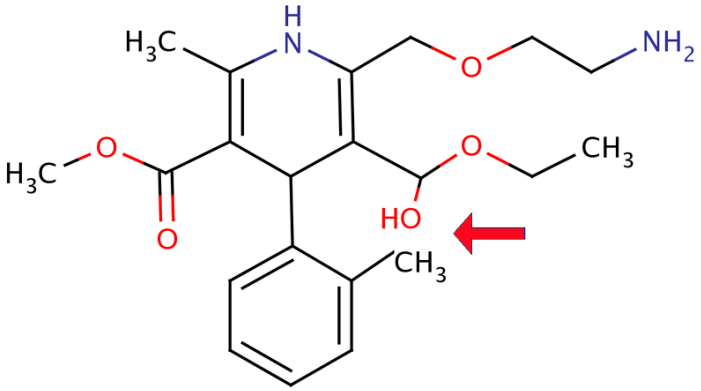


Figure 6.6: Incorrect reconstructed molecule of chemoCR. Method did not recognize the Chlorine and clustered close characters wrong.

The clean extraction of dotted chirals is often a difficult problem. They can strongly vary in the amount of symbols, their size and their arrangement (cross or length dotted). It can occur that they touch other entities, cross bonds or other dotted chirals. Initially rules have been specified which expected that all symbols within a dotted chiral hold a similar vector orientation and a certain relative position to each other. Unfortunately the participating line symbols often contain only a few pixels. Therefore both criteria are no reliable indicators for the resulting short vectors, because their stability decreases as the connected components become smaller. On the one hand rules of the expert system should be able to clearly differentiate these symbols from other semantic entities like bonds and characters. Otherwise the required fuzziness to cover all small dotted chiral symbols would increase the probability of false positives. For that reason another strategy is followed. All symbols which cannot be recognized by the expert system are flagged as unknown. Although unknown symbols cannot become a seed element for the extractors they can nevertheless be included in their semantic entity. The margin line symbols of dotted chirals are large enough to be clearly identified by the expert system (compare figure 6.7). So they can become a seed symbol of the dotted chiral extractor. If during the extension process the extractor encounters a symbol whose structural family is unknown but its relative position approximately fits into the current extension direction it is included into the collected superset. Thus the environment of a symbol influences its recognition and the molecule reconstruction becomes more stable.

The number of missed dotted chirals in the evaluated test set is nevertheless relatively high. This results mainly from the restrictiveness of the extraction process in its selection of the symbols which are included in the superset. The symbols which participate in a dotted chiral can be tiny and strange oriented so that their connectivity pattern in the orientation graph makes it difficult for the processing.

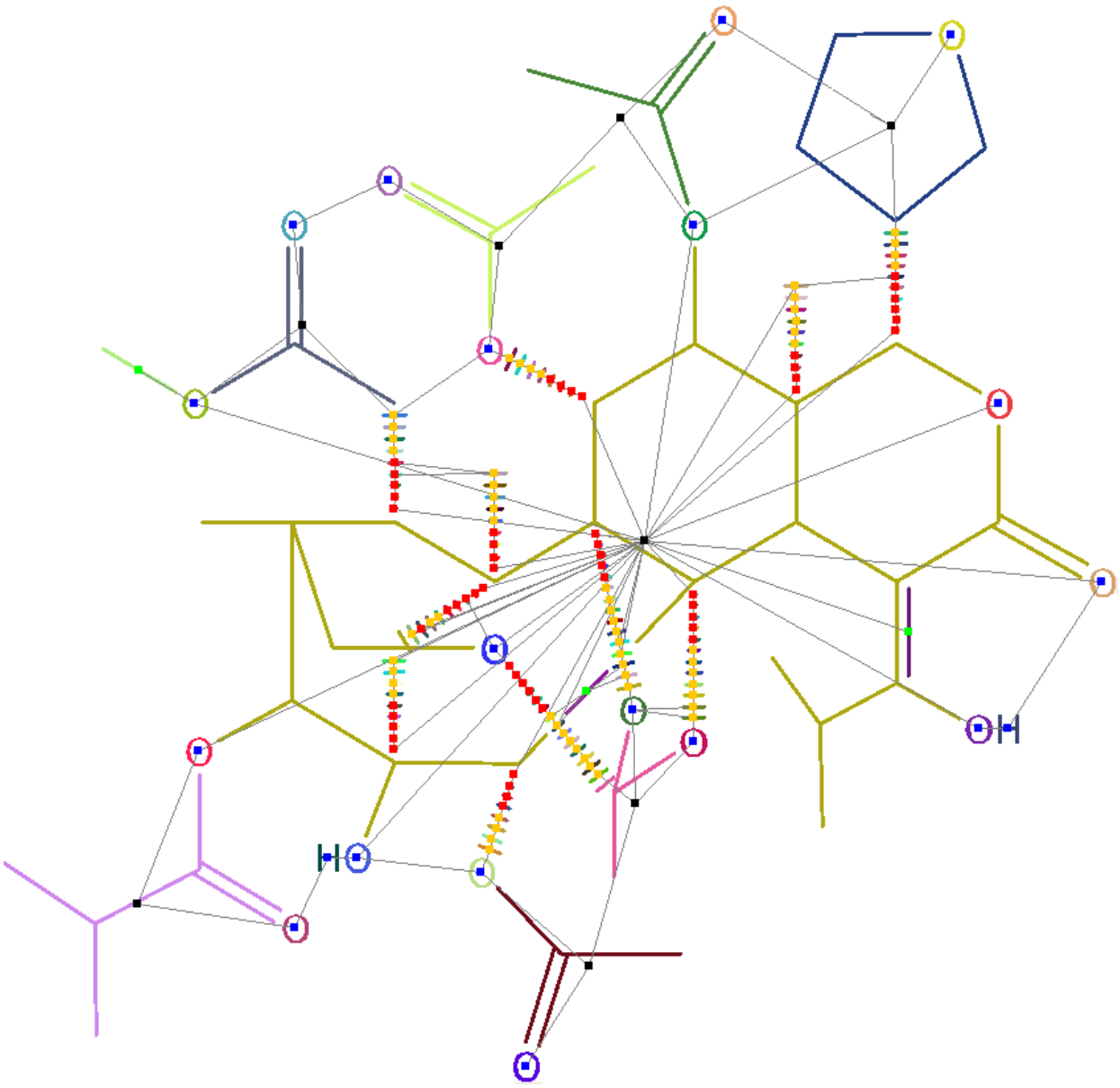


Figure 6.7: The image shows an orientation graph and the underlying structural formula. The coloring of the symbols indicates the distinct connected components. Each vertex in the graph is represented by a small box situated in the center of the minimum bounding rectangle of each component. The gray lines between the boxes exhibits the edges of the graph. The color code of the vertices shows the approximated structural family, which is derived from the expert system (blue: character, green: isolated bond, black: bond set, orange: chiral, red: unknown). Only symbols which have a clear family assignment can become a seed vertex for the extractors. Nevertheless, also an unknown symbol can be included into its corresponding semantic element. An unknown symbol which fit e.g. into the extension process of a dotted chiral extractor can be simple included in its superset. Thus the environment of a symbol influences its recognition.

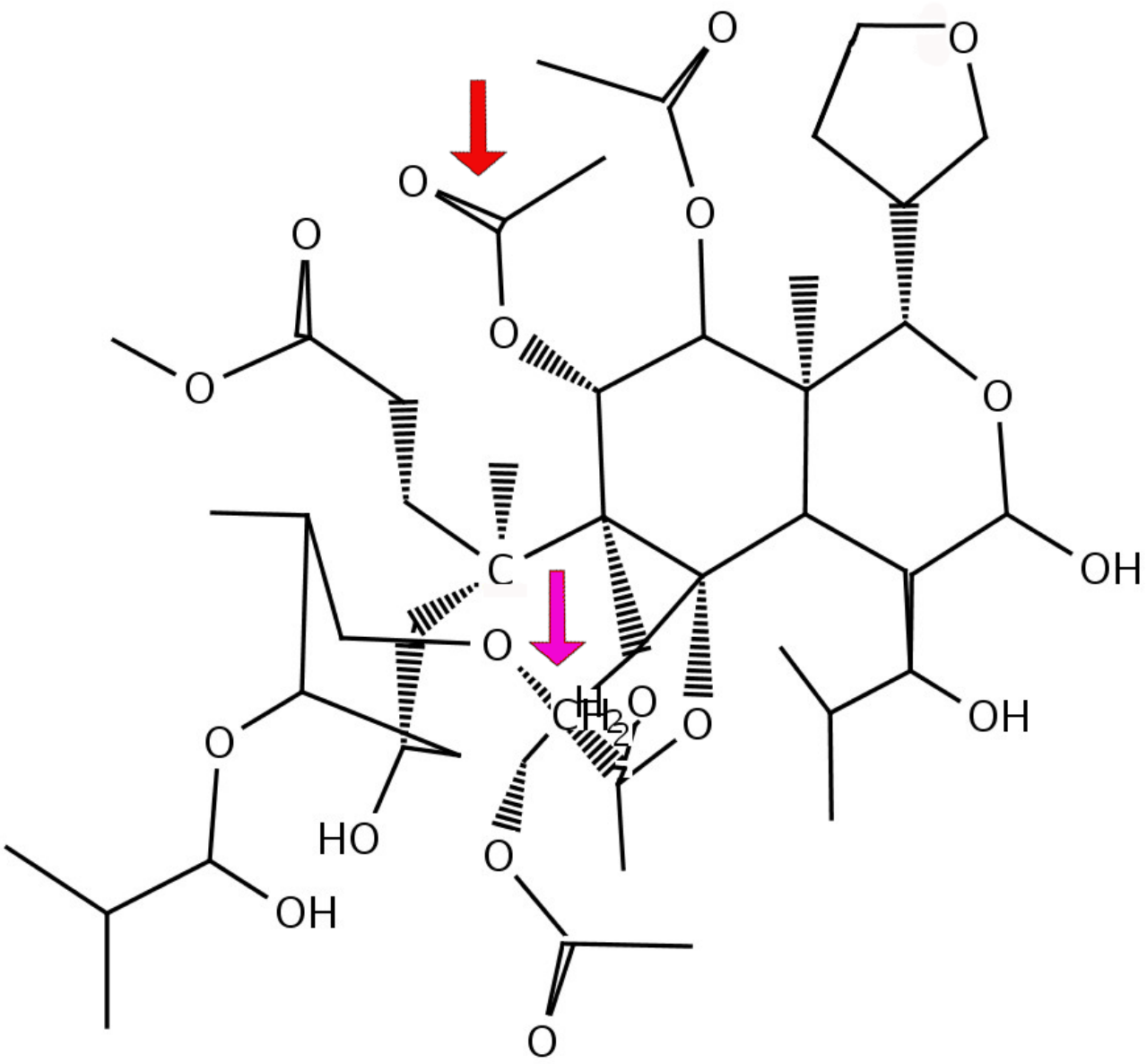
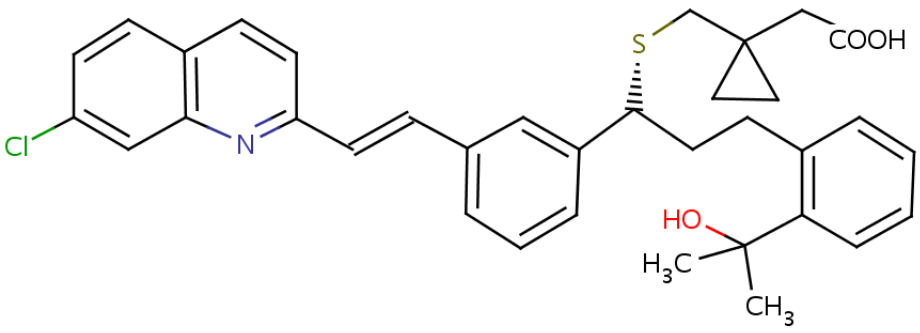
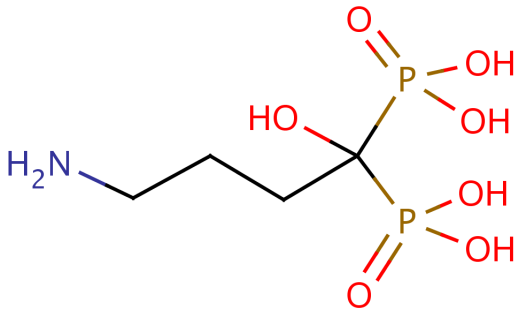


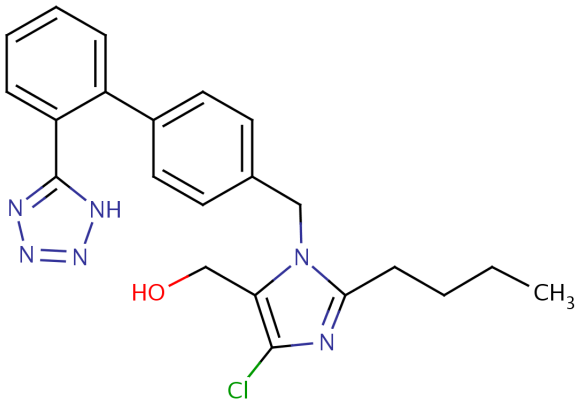
Figure 6.8: A dotted chiral molecule (compare 6.7) which was reconstructed with the new concept. Although still several errors occur the overall recognition is quite promissing. The recognition procedure does noy yet cover crossing dotted chirals (see purple arrow). When this picture was generated, the recognition of multi bonds in bond sets (compare dark red arrow) have not yet exist. In the meantime these elements can be recognized properly.



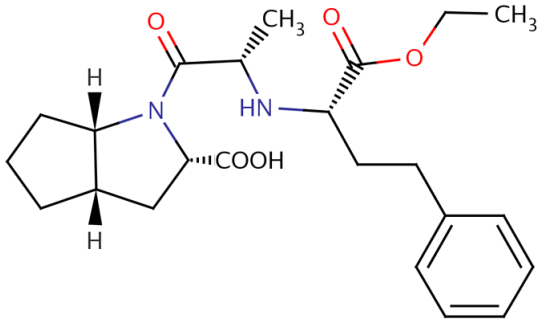
(a) Molecule 1



(b) Molecule 2



(c) Molecule 3



(d) Molecule 4

Figure 6.9: Some examples of the test set, which have been reconstructed without any error. The images contain a broad range of challenging recognition problems such as overload symbols (1 and 3), close characters (1, 2 and 4), multi bonds (1– 4), nontypical bond sets (1 and 4), dotted chirals (1 and 4) and thick chirals (4).

Chapter 7

Discussion and outlook

7.1 Discussion

In the following the reconstruction strategies of chemoCR and the new concept are compared and existing bottlenecks are elucidated.

The reconstruction of chemical drawings is mainly based on the recognition of structural patterns. Although there exists similar proceedings in the pattern identification of chemical entities, chemoCR implements heterogeneous extraction modules. Each module has the ability to recognize and extract a certain structural pattern, whereas there is no well-defined representation how this knowledge can be specified. The complex and intransparent modules often depend on parameters which considerably impact the accuracy rate of the system. Conflicts between different modules can occur because the individual recognizers work isolated and no knowledge based instance exists at an early stage which could avoid this dilemma.

On contrast to that the new approach bundles everything which extractors have in common. Redundant geometric calculations are eliminated by the application of the orientation graph. With that, a clear and overall definition of closeness for all extractors is specified without using different and scattered parameters. In addition, the modules complexity is even more reduced by the conceptual separation into recognition and extraction. Hence, extractors do not require functionalities such as the detection of conflicts and the decision if derived results are correct. In the new technique, these advanced recognition problems have to be solved by the expert system. Here, all recognition knowledge and the sparse remaining parameters are centralized.

Extractors only have to extend a provided seed symbol, whereby all available symbol candidates are determined by the orientation graph. Which of the symbols can be accepted is decided by the expert system. With this strategy the obtained extractors become very slim, e.g. the routine which is responsible for the extraction of multi bonds consists of less than 150 lines of programming code (in comparison with 900 in chemoCR).

Due to the rule based representation, the knowledge of the expert system can be clearly and consistently described. The designed language (refer to 5.2.2) is intuitive, comprehensive and easy to extend. This is essential for an advanced knowledge based approach which makes the recognition of the chemical entities more robust. The expert system has the beneficial ability to explain each recognition decision. Amongst other advantages this e.g. can be used to detect and remove weak discrimination rules to obtain an improved knowledge base.

Instead of an arbitrary processing of the symbols within a chemical depiction like seen in chemoCR a constraint based graph exploration (compare 5.3.2) is performed. This proceeding has two advantages: First it allows to consider all identified neighbor symbols in the recognition process. In addition, conflicts between different extractors are reduced by the cooperation between the constraint based graph traversal and the expert system. This enables the addressing of extractors which are in the current context reasonable in the chemical sense.

Although the developed approach already holds great benefits, its concept and its reconstruction results are far away from perfect.
When this thesis started there were no specifications at all for a new reconstruction concept. Though the idea of the orientation graph and the constraint based traversal emerged quite soon, some aspects like the recognition via an expert system have been added relatively late. The developed knowledge representation allows to describe all kinds of spatial patterns but the applied rules become complex as soon as the context analysis depth is enlarged. Including more neighbors of a symbol into the recognition process would be particularly in ambiguous cases very helpful. The increasing rule complexity can be avoided by the application of more powerful predicates. Such predicates examine extensive context conditions but bypass the symbol-context-relation requirement mechanism. A more advanced rule language may allow to enlarge the context depth without becoming too complex.

Also the developed orientation graph is not free of bottlenecks. Although this graph significantly facilitates the spatial analysis in the recognition and extraction process, drawing situations occur where the graph is not sufficient for the correct identification of all semantic entities. Due to its clear relatively close definition (refer to 5.1.2) it occurs that symbols which belong to the same semantic entity are not linked in the graph. Figure 7.1 shows an example where two line symbols of a double bond are not directly connected in the orientation graph. This results from a close character neighbor which lies in the intersection area of their relative close test. The extension candidates for the extraction process depend on the graph neighborhood of the seed symbol. Such connectivity patterns make the extraction problematic if only the graph is involved.

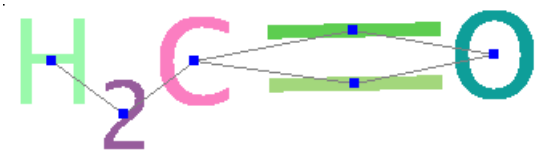


Figure 7.1: The orientation graph sometimes contains deficits which complicate the identification of semantic entities if their extraction is only based on the graph traversal. The image shows two line symbols of a multi bond, which are not connected in the orientation graph (indicated by the little boxes and the lines between them). The reason for that is a close character 'O' which lies in the intersection area of their relative close test (refer to 5.1.2).

Instead of complicating the extraction process by enlarging the context depth another strategy is being applied. For the generation of the orientation graph the single link

distances between all connected components have been calculated. These values can now be used to simply analyze successive one neighbor after another. Before the extractor can include a candidate into its superset, it has to to consult the expert system. If all qualified symbols were extracted, the affected data structures (compare section 5.3.2) of the graph traversal can be updated and the exploration can continue in the usual manner.

The most significant bottleneck of new approach, but also of chemoCR, is the high dependency to the image processing algorithms. If already errors in the image processing occur the recognition process has often no chance to correctly identify a molecule.

The worst case are errors in the segmentation process (see 3.1) which impact all succeeding image processing steps. If a wrong binarization threshold is selected too less border pixels of the image symbols are removed and two usually distinct symbols remain in one connected component. So it can happen e.g. that a character 'O' is included in a connected component representing a bond set. As consequence the OCR cannot recognize the character and the vectorized bond set holds strange vectors.

Errors in the reconstructed molecule can also occur if the character recognition (refer to 3.2) does not work properly. The current workflow of chemoCR and the new approach do not try to recognize each image symbol as character. Because the applied OCR is optimized for the character recognition in documents, the underlying algorithm assumes that the majority of the provided symbols are characters. If this is not the case the recognition rate significantly declines. For that reason a parameter based selection is applied and only potential characters are processed by the character recognition software. Here it can occur that a character is completely discarded because it does not fit to the assumed parameters. In addition to a lost character this also has consequences on the string extraction process of the new technique. This procedure is based on the extension of a seed character to a maximum character set (compare 5.4). Although e.g. the 'H' in 'H_2N' is in the orientation graph connected with '2' which is in turn linked with 'N' also the clustering of 'H' and 'N' will fail because there is no character between them. But even if the all character symbols are send to the OCR it is not sure that they can be recognized correctly (compare 3.2.3).

The main reason why the correct reconstruction of a molecule can fail is the error prone vectorization process (refer to 3.3). A wrong identified character does not change the overall appearance of the molecule, whereas a wrong bond can influence the entire topology. Numerous fuzziness factors of the recognition process have been included which allow a certain tolerance to vectorization errors. Due to this workaround the complexity of the extractors and the applied rules increase. If complete vectors are missing (compare 7.2) the recognition just fails.

Even if everything has been recognized correctly, errors can still occur in the molecule assembly. After all semantic entities in the image have been identified, the chemical graph and the molecule representation must be generated. The current implementation builds on the existing molecule assembly process (see section 4.2) of chemoCR because the reconstruction itself was topic of the thesis. Although a more stable molecule generation might have been reached through the application of the orientation graph the calculation of the molecular graph (see 2.3.2) is performed by the less robust parameter-based method of chemoCR.

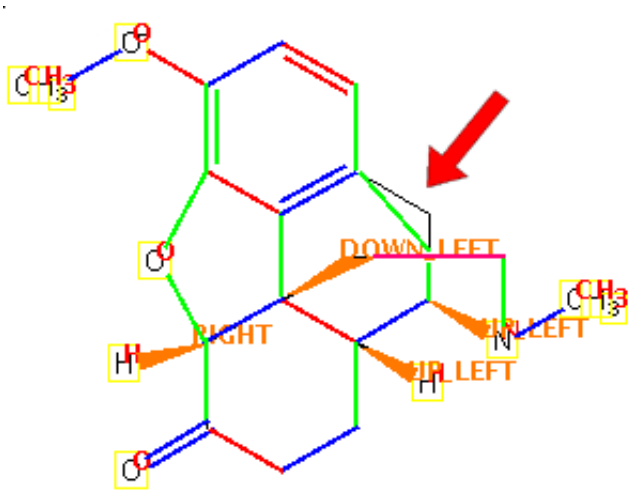


Figure 7.2: The image illustrates a severe vectorization error. The red arrow shows the position of a complete missed bond. Although all other elements have been identified correctly (compare color coverage of underlying black molecule), the recognition procedure cannot handle such errors and the reconstructed molecule will be wrong.

7.2 Outlook

The reconstruction of chemical depictions works quite well but is still far away of being perfect. For obtaining a reliable recognition software which is able to keep pace to the growing structural formula space still several bottlenecks have to be eliminated.

Especially the endeavour for image processing algorithms must be intensified. If the quality of the input data increases, less faults must be handled and the chemical recognition becomes significantly less complex. It turned out that established regular OCR and vectorization software do not work properly for the identification of atoms and bonds in a chemical drawing. For that reason the Fraunhofer-Institute for Algorithms and Scientific Computing implemented a new vectorization algorithm. This method yields to better results but is still very sensitive to small variations because it depends on a set of parameters.

Developing an OCR and a vectoriziation algorithm which considers chemical knowledge in its recognition would be reasonable but very laborious. Another solution might be the combination of the preprocessing steps and the chemical recognition procedure. In the current chemoCR and the new developed approach the segmentation, character recognition and the vectorization must be completed before the recognition of the chemical entities can start. Otherwise the recognition procedure is the only instance in the reconstruction workflow, which can decide if obtained results are reasonable in a chemical sense.
For that an extension of the developed concept is thinkable, where the chemical recognition procedure instructs directly specific image processing algorithms and even adjusts theirs parameters. Here the starting point could also be the orientation graph, which is explored similar to the developed concept. The only requirement to obtain such a graph are

connected components of the segmentation step. The new workflow might look like the following: Depending on the current traversal status a symbol might be send e.g. to a vectorization routine. If the acquired vector result cannot be interpreted in a chemical sense a vectorization error might occur. So the vectorization can again be started with different parameters and provide therefore another result. This proceeding can also be applied with other image processing algorithms. If for the current processed connected component e.g. no results can be inferred the reason might be a segmentation error. So the segmentation for this component can be repeated with another binarization threshold. In the case that the component decomposes, the resulting sub components can be analyzed. If they can be recognized the topology of the orientation graph has to be updated by performing a corresponding vertex split and a new edge calculation. Then the graph exploration can continue and more symbols can be identified until the whole molecule is reconstructed.

It can be assumed that chemical structure reconstruction will be an inherent part of information preparation and information retrieval process. Diverse text mining approaches already exist, which have been successfully applied on scientific literature in biology and medicine. In contrast to that the information extraction in medicinal chemistry and pharmacology often failed because these areas are mainly centred on chemical compounds and their structures. With a reliable reconstruction method new fields for application can emerge which will support public and commercial research in medicinal chemistry, pharmacology and toxicology.

Acknowledgment

At the end of my thesis I would like to thank all the people who made this thesis possible. The work on this chemoinformatic topic has been an exciting, challenging and an enjoyable experience for me. I would like to express my sincere gratitude to Prof. Dr. Hans-Peter Kriegel, head of the Department of Database and Information Systems, Institute for Computer Science, University of Munich (*). Prof. Kriegel gave me the opportunity to work on this interesting topic of my thesis. I also thank Karsten Borgwardt from this institue(*), who supported me with his encouragement and provided useful suggestions for improvement on my work.
I am very grateful to Prof. Dr. Martin Hofmann-Apitius who gave me the opportunity to participate at the chemoCR project at the Fraunhofer-Institute for Algorithms and Scientific Computing (SCAI). Special thanks go to my direct supervisor Dr. Marc Zimmermann who guided this work and offered his help when needed. We had extensive discussions about my work, which were very informative to me and had an important impact on the final form and quality of this thesis. He allowed me great freedom in developing a new way to address the problem of CSR and helped me with his invaluable experience and advice to stay on the right track.
My sincere thanks also go to all members of the chemoCR project from whom I always received a great assistance. Many thanks go to Prof. Dr. Maria-Elena Algorri for her patient and very helpful support and advice. Thanks also to Angelika Weihermueller for her contributions to the development of the graphical user interface. The thesis has also been accompanied and supported by many other people.
I am especially grateful to Roman Klinger who always supported me through extensive discussions during long evenings in the institute as well during our common bus rides. It is also a pleasure to mention my office colleagues Antje Wolf, Marco Hülsmann and Christian Tsatedem who created an outstanding working atmosphere. Thanks also to those who volunteered to read drafts and provide their feedback, and especially for revising the English of my manuscript.

I warmly thank my girlfriend Mirjam whose dedication, love and persistent confidence in me always have supported me. In addition she had the energy to proof-read these chapters again and again. My special gratitude goes to my family for their continuous support, encouragement, and gentle love. They always believe in me and stand by me in any difficult situation in my life.

Sankt Augustin, Germany, March 2007

Peter Kral

Abbreviations and Glossary

Approximation	Process to derive the structural family of a symbol
AS	AnalysisSystem
CC	Connected Component
Chemical Graph	Connection of all semantic entities
CSR	Chemical structure reconstruction
Itemization	Process to derive the exact semantic entity of a symbol
Knowledge Base	Contains the rules for expert system
MBR	Minimum bounding rectangle
Molecule Graph	Atom-Bond representation of the chemical graph
Multi-Conflict	Several extractors claim the same symbol
OCR	Optical character recognition
OG	Orientation Graph
Orientation Graph	RNG for connected components
Overloaded symbol	Symbol which can occur in different chemical entities
Physical-semantic-distance	Very close symbols which do not belong together
Predicate Base	Contains testing methods for expert system
RNG	Relative neighborhood graph
RTree	Spatial index structure
SDF	Structure Data Format
SEE	Semantic entity extractor
Segmentation	Extraction of connected components
SMILES	Simplified molecular input line entry specification
Vectoriziation	Conversion raster graphics to vector graphics

List of Figures

Index

Bibliography

[1] Pankaj K. Agarwal. Relative neighborhood graphs in three dimensions. In *SODA '92: Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, pages 58–65, Philadelphia, PA, USA, 1992. Society for Industrial and Applied Mathematics.

[2] A. R. Leach Andrew R. Leach, Valerie J. Gillet. *An Introduction to Chemoinformatics*. Springer; 1 edition, May 2, 2006.

[3] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The r*-tree: An efficient and robust access method for points and rectangles. In Hector Garcia-Molina and H. V. Jagadish, editors, *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, May 23-25*, pages 322–331. ACM Press, 1990.

[4] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[5] C.M. Bishop. *Neural Networks for Pattern Recognition (paperback)*. Oxford: Oxford University Press, 1995.

[6] Rouvray D. H Bonchev, D., editor. *Chemical Graph Theory: Introduction and Fundamentals*. Taylor & Francis, 1987.

[7] Ronald J. Brachman and Hector J. Levesque, editors. *Readings in Knowledge Representation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1985.

[8] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[9] A. Chianese, L. P. Cordella, M. De Santo, A. Marcelli, and M. Vento. A structural method for handprinted character recognition. pages 289–302, 1989.

[10] Kyong-Ho Lee; Sung-Bae Cho; Yoon-Chul Choy;. A knowledge-based automated vectorizing system for geographic information system. *Proceedings. Fourteenth International Conference on Pattern Recognition*, 2:1546 – 1548, 1998.

[11] Thomas H. Cormen. *Introduction to Algorithms, second edition*. MIT Press and McGraw-Hill, 2001.

[12] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, March 2000.

[13] Arthur Dalby, James G. Nourse, W. Douglas Hounshell, Ann K. I. Gushurst, David L. Grier, Burton A. Leland, and John Laufer. Description of several chemical structure file formats used by computer programs developed at molecular design limited. *Journal of Chemical Information and Computer Sciences*, 32(3):244–255, 1992.

[14] Amaryllis Deliyanni and Robert A. Kowalski. Logic and semantic networks. *Commun. ACM*, 22(3):184–192, 1979.

[15] Reinhard Diestel, editor. *Graph Theory (Graduate Texts in Mathematics) Third Edition*. Springer, Berlin, 2006.

[16] D.S. Doerman. An introduction to vectorization and segmentation. In *Workshop on Graphics Recognition, Algorithms and Systems Lecture Notes in Computer Science, pp. 1-8*, 1997.

[17] Richard C. Dubes. *Cluster analysis and related issues*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1993.

[18] Thomas Eiter and Heikki Mannila. Distance measures for point sets and their computation. *Acta Informatica*, 34(2):109–133, 1997.

[19] T. Fey. Validation of chemical structure recognition software for 2d drawings and a following graphical error curing. Master's thesis, Fachhochschule Bonn-Rhein-Sieg, 2004.

[20] Keinosuke Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[21] Johann Gasteiger. *Handbook of Chemoinformatics 1st edition*. Wiley-VCH, 2003.

[22] Andrew V. Goldberg and Chris Harrelson. Computing the shortest path: A search meets graph theory. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 156–165, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[23] P. S. P. Wang H. Bunke (Author). *Document Image Analysis (Machine Perception and Artificial Intelligence, Vol 16)*. World Scientific Publishing Company, 1994.

[24] P. Ibison, M. Jacquot, F. Kam, A. G. Neville, Richard W. Simpson, Christian A. G. Tonnelier, T. Venczel, and A. Peter Johnson. Chemical literature data extraction: The clide project. *Journal of Chemical Information and Computer Sciences*, 33(3):338–344, 1993.

[25] Peter Jackson. *Introduction to Expert Systems, 3rd Edition*. Longman Addison Wesley, December 23, 1998.

[26] J. W. Jaromczyk and M. Kowaluk. A note on relative neighborhood graphs. In *SCG '87: Proceedings of the third annual symposium on Computational geometry*, pages 233–241, New York, NY, USA, 1987. ACM Press.

[27] P. M. Lankford, editor. *Regionalization: theory and alternative algorithms Geographical Analysis, 1(2):196-212*, April 1969.

[28] Sang Uk Lee, Seok Yoon Chung, and Rae-Hong Park. A comparative performance study of several global thresholding techniques for segmentation. *Computer Vision, Graphics, and Image Processing*, 52(2):171–190, 1990.

[29] B. Liu, S. Li, and J. Hu. Technological advances in high-throughput screening. *Am J Pharmacogenomics*, 4(4):263–276, 2004.

[30] George F. Luger. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving (5th Edition)*. Addison Wesley, 2004.

[31] G. Monagan M. Roosli. Adding geometric constraints to the vectorization of line drawings. *First International Workshop on Graphics recognition, Methods and Applications, Lecture Notes in Computer Science, vol. 1072, pp. 49-56*, 1995.

[32] Nanopoulos A. Papadopoulos A.N. Theodoridis-Y. Manolopoulos, Y. *R-Trees: Theory and Applications Series: Advanced Information and Knowledge Processing*. Springer, 2006.

[33] Le Thuy Bui Thi Marc Zimmermann and Martin Hofmann. Combating illiteracy in chemistry: Towards computer-based chemical structure reconstruction. *ERCIM News*, 60:40–41, 2005.

[34] R. McDaniel and Jason R. Balmuth. Kekulé: Ocr-optical chemical (structure) recognition. *Journal of Chemical Information and Computer Sciences, 32(4):373–378*, 1992.

[35] Peter Murray-Rust. Chemical markup language. *World Wide Web J.*, 2(4):135–147, 1997.

[36] V. Nagasamy and N.A. Langrana. Engineering drawing processing and vectorization system. *Computer Vision, Graphics and Image Processing*, 49(3):379–397, 1990.

[37] Sybil P. Parker, editor. *McGraw-Hill Dictionary of Scientific and Technical Terms*. McGraw-Hill Companies; 5th edition, 1993.

[38] V. Consonni R. Todeschini. *Handbook of Molecular Descriptors*. Wiley-VCH, 2000.

[39] Jean-Yves Ramel, Guillaume Boissier, and Hubert Emptoz. Automatic reading of handwritten chemical formulas from a structural representation of the image. In *ICDAR '99: Proceedings of the Fifth International Conference on Document Analysis and Recognition*, page 83, Washington, DC, USA, 1999. IEEE Computer Society.

[40] Max Bramer (Editor) Robert W. Milne, Ann Macintosh, editor. *Applications and Innovations in Expert Systems VI: Proceedings of ES98, the Eighteenth Annual International Conference of the British Computer Society Computer Society Conference Series.)*. Springer, 1999.

[41] Markus Roosli and Gladys Monagan. Adding geometric constraints to the vectorization of line drawings. In *Selected Papers from the First International Workshop on Graphics Recognition, Methods and Applications*, pages 49–56, London, UK, 1996. Springer-Verlag.

[42] Boyer S. Optical recognition of chemical graphics. *Document Analysis and Recognition, Proceedings of the Second International Conference on Publication*, pages 627–631, 20-22 Oct 1993.

[43] Kramer S. *Lecture: Analysis of structural data in bioinformatics, Distance Measures Between Point Sets*. TU Munich, 2006.

[44] Herbert F. Schantz. *History of OCR, Optical Character Recognition*. Recognition Technologies Users Association, 1982.

[45] Edward H. Shortliffe, editor. *Computer-based Medical Consultations: MYCIN*. Elsevier, 1976.

[46] A.J. Stuper, W.E. Brugger, and P.C. Jurs. *Computer Assisted Studies Of Chemical Structure And Biological Function*. Wiley, New York, 1979.

[47] Karl Tombre and Salvatore Tabbone. Vectorization in graphics recognition: To thin or not to thin. *icpr*, 02:2091, 2000.

[48] G. Toussaint. The relative neighbourhood graph of a finite planar set, 1980.

[49] O. Trier, A. Jain, and T. Taxt. Feature extraction methods for character recognition - a survey, 1996.

[50] W. P. Walters, M. T. Stahl, and M. A. Murcko. Virtual screening - an overview. *Drug Discovery Today*, 3(4):160–178, April 1998.

[51] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, 1988.

[52] Lars Wunderlich. *Java Rules Engines. Entwicklung von regelbasierten Systemen*. Entwickler.Press; Auflage: 1, 2006.