



universität
wien

MASTERARBEIT

Titel der Masterarbeit

„Making Chemical Synthesis Plans Machine Readable“

verfasst von

Lukas Michael Bartonek, BSc

angestrebter akademischer Grad

Master of Science (MSc)

Wien, 2015

Studienkennzahl lt. Studienblatt:

A 066 862

Studienrichtung lt. Studienblatt:

Masterstudium Chemie

Betreut von:

ao. Univ.-Prof. Mag. Dr. Christoph Flamm

A work like this is never possible to be done by one person alone. At this point I want to express my thanks to all the people who supported me in the creation of this work:

My supervisor Christoph Flamm and the leader of the TBI group Ivo Hofacker for their support throughout the work as well as for their thoughts and input on this topic.

My colleague Stefan Hammer for helping me rebuild the webservice presented in this work in modern HTML5 and his input on the design of the user interface.

Roman Ochsenreiter for his help with everything related to Perl.

The whole TBI team for being supportive and helpful throughout my time being there.

And finally my parents for their support both financially and emotionally. Without this the presented work wouldn't exist in this form.

CONTENTS

I	Introduction	9
1	INTRODUCTION	11
II	Theory	13
2	CHEMICAL FILE FORMATS	15
2.1	SMILES	15
2.1.1	Grammars	16
2.1.2	SMILES grammar	18
2.1.3	Descriptive Approach	21
2.1.4	Building up a SMILES string	23
2.2	InChI	25
2.2.1	InChI structure and basic layers	26
2.2.2	Main Layer	26
2.2.3	Charge Layer	27
2.2.4	Stereochemical Layer	28
2.2.5	Further Layers	28
2.3	InChI Key	28
2.4	MOL	29
2.4.1	Header	30
2.4.2	Counts	30
2.4.3	Atoms Block	30
2.4.4	Bonds Block	31
2.4.5	Properties Block	31
2.5	JME	31
3	WALK COUNTS	33
3.1	Atomic Walk Count	33
3.2	Total Walk Count	34
4	GREEN CHEMISTRY	37
4.1	Green Metrics	37
4.1.1	Yield ϵ	38
4.1.2	Environmental Factor	39
4.1.3	Atom economy AE	39
4.1.4	Carbon efficiency	40
4.1.5	Effective mass yield EMY	40
4.1.6	Strategy efficiency metrics	41

III Methods 43

5	SYNTHESIS PLANNING	45
5.1	Development of chemical Synthesis	45
5.2	Problems with classical Synthesis Plans	45
6	INPUT FILE SPECIFICATIONS	49
6.1	Information Header	49
6.1.1	Timestamp	49
6.1.2	Last Author	49
6.1.3	Literature	50
6.1.4	DOI	50
6.1.5	Author	50
6.2	Skeleton	51
6.3	Compounds	51
6.4	Reactions	51
6.4.1	Reaction Meta Data	52
7	DATABASE	53
7.1	Database structure	53
7.2	Frontend	54
8	PERL DANCER	57
8.1	REST	57
8.2	Calls supported by the server	57

IV Results and Discussion 61

9	INPUTSERVER	63
9.1	Compatibility	63
9.2	Usage	67
9.3	Features	71
9.3.1	Editor Features	71
9.3.2	Balance Testing	74
9.4	Synthesis Plan Analysis	79
9.4.1	Phenol synthesis approaches	80

V Future 85

10	SUMMARY	87
10.1	Justification	87
10.2	Product	87
10.3	Usability	88
11	FUTURE	89
11.1	Considerations about the Future of the Project	89
11.1.1	Database interaction	89

11.1.2	Total walk counts	89
11.1.3	Furhter calculations	90
11.2	Applications	90
A	EXAMPLE PLAN	91
B	PHENOL PLANS	99

Part I

Introduction

INTRODUCTION

Synthesis Plans are to the synthetic chemist like a recipe is to a cook. Every aspect of creating a certain compound out of others is written down to ensure that a successful approach is reproducible. Today Synthesis Plans are the backbone of organic chemistry. Every advance ever made in this field is in some way documented by this method.

Almost two centuries ago Wöhler published the first total synthesis^[1] and started this central field of chemistry. Until now these plans are published in journals. The approach publishing on paper may have been the most practical over the years, but as the author wants to show with his work, synthesis plans should be easily search- and analyzeable. Therefore an approach that collects plans in a machine readable manner is preferred.

Several problems with the current way of collecting synthesis plans are developed and different approaches to solving them are discussed. This all leads to a custom database based solution implemented with the help of the author, that allows to do automated analyses of the submitted synthesis plans. A final outlook chapter motivates the reader to try the current implementation for themselves or even write their own analysis software to be used with the database.

Part II

Theory

CHEMICAL FILE FORMATS

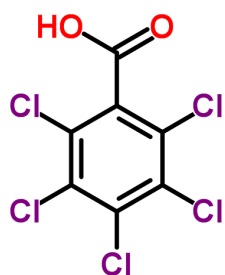
Chemical formulas are usually interpreted by human beings. Once one wants to automate any process involving chemical formulas a machine readable format is needed. This section gives an overview on the most common formats for this purpose with a focus on those that are either very commonly used or were essential in the creation of this work.

Allowing computers to interpret molecules is the first essential step in the task of making synthesis plans accessible for computational approaches. Although the formats shown here vary from a simple coordinate based model like MOL file to the SMILES format that generates a molecule graph - they are all widely used at the moment and succeed in their respective scope of application.

2.1 SMILES

Developed in 1987[2], the SMILES notation tried to improve on the formats used to encode chemical information, that were available at that time. Generally the file formats of this time could be separated into two distinct groups. Those that used a vast amount of disk space for saving a compound but had all the information needed to draw a molecule saved explicitly and those that just tried to save disk space but sacrificed readability for humans to achieve this goal. The first group consisted most notably of the MOL file and the CAS specifications. Both of which depended heavily on complex - and large - connection tables, that had to be saved in a file. This makes it easy to generate a visual depiction of a structure but, because of the complexity of the file, hard to use in a computational environment. Additionally these specifications prove to be hard to be produced by humans.

To tackle the problem of computation line notations were developed. These much shorter notations saved disk space and made it easier for computers to process. Still these notations were nearly impossible to be written by humans without the use of the computational power of computers. One of the more popular of these notations - the Wiswesser Line Notation[3] - depended heavily on complex rules to generate data.



QVR BG CG DG EG FG

Figure 1: Pentachlorobenzoic Acid and the corresponding Wiswesser Line Notation, image generated by ChemSpider[4]

The goal of the SMILES notation was to create a specification that allowed line notation to be written in a way that is possible to be readable by humans while still allowing to generate - at least internally, but also to export - unique identifiers for each structure. The way the SMILES notation is set up there are usually multiple possibilities to encode a certain structure which are all equally correct but for the purpose of generating unique identifiers the canonical SMILES chooses one of the possible structures by following a simple set of rules. To achieve this the SMILES notation interprets all molecular structures as graphs, representing the two dimensional pictures chemists usually draw. Although supporting stereochemistry the three dimensional nature of molecules is ignored in this intended simplification. This has to be kept in mind when trying to understand the SMILES encoding.

Since the SMILES notation is best understood by the underlying formal grammar the concept of a grammar in theoretical computer science is explained in this short section:

2.1.1.1 Grammars

Originally applied to natural languages grammars were soon found to be a nice way of defining the common structure of diverse objects in computer science. In other words, grammars allow to define the common structure of a collection of diverse objects in our case of what a molecule is. Furthermore, they are a formal tool to decide if a given object is well-formed according to the rules of the grammar and therefore belong to the collection or if it should be rejected. To make this concept accessible to the reader a simple example from the English language is chosen to illustrate this:

$$S ::= SP \ VP$$

In this simple model a sentence S is made up of a subject phrase SP and a verb phrase VP . Obviously this model is not able to cover all the

possible sentences in the english language but that is not the goal of this example.

Now that a sentence is defined one needs to define the newly made up signs SP and VP.

```
2 SP ::= Art* N
3 VP ::= V SP*
```

With the subject and the verb phrase defined it is obvious that the subject phrase gets used again in the verb phrase. In this formulation of the grammar an asterisk (*) indicates an optional component of a rule. In regards to the subject phrase the article Art is not needed but may be used.

The verb V the article Art and the noun N now need to be defined. Since this is a very simple grammar that allows only for short sentences a very limited number of words are supplied for these three phrases:

```
4 Art ::= 'a' | 'the'
5 N ::= 'cat' | 'cat food'
6 V ::= 'eats'
```

With the declaration of specific words new symbols are introduced. The single ticks (') indicate that the string they surround should be interpreted as it is and does not refer to another phrase in the grammar. The vertical bar (|) defines a list of choices from which one has to be taken.

To finish this example a sentence is built and the obvious errors of this grammar are shown. Since we want to create a sentence the rule for S is consulted. This rule requires a subject phrase and a verb phrase. Starting with the subject phrase an article can be chosen optionally and a noun has to be decided on. Choosing 'a' as an article and 'cat' as the noun the verb phrase needs to be resolved now.

The rather short verb list doesn't allow for much choice so 'eats' gets chosen. At this point the sentence may be considered as finished or optionally a subject phrase may be added. Doing the later and choosing 'the' as an article and 'cat food' as a noun the sentence is finished:

```
7 a cat eats the cat food
```

But this is not the only sentence that could be built using this grammar. In descending ordering in regards to applicability to the real world the following sentences could also be built:

```
8 the cat eats
9 the cat eats a cat
10 the cat food eats a cat
```

Now the sentences are still valid English sentences but they have taken a rather gruesome turn. While this could be considered a downside of this specific grammar this is intended behaviour since only the English language should be modeled and not the content of the message. But there are still some things left out for simplicity: the spaces between words are not explicitly mentioned, capitalization at the beginning of a sentence is not modeled and punctuation at the end is left out. These are all details that need to be defined when computers are interpreting these rules.

Now that the basics of grammars are explained the reader has the necessary prerequisites to be able to understand the complete grammar of the SMILES encoding with comments added.

2.1.2 SMILES grammar

Very much like the example in section 2.1.1 a SMILES string can be defined by the underlying grammar. Listed here is a comprehensive set of rules that make up the SMILES format. The specific rules here were published in the OpenSMILES project[5] and have been sorted and commented by the author:

```
1 SMILES Strings
2
3 smiles ::= terminator | chain terminator
4 terminator ::= SPACE | TAB | LINEFEED |
   CARRIAGE_RETURN | END_OF_STRING
```

Here the basic rule for SMILES defines the string as either a terminator sign in the case of an empty string or a chain followed by a terminator symbol. The possible terminator symbols are listed in the four as a choice of different whitespace characters.

```
6 Bonds and Chains
7
8 chain ::= branched_atom | chain branched_atom |
   chain bond branched_atom | chain dot
   branched_atom
9 branched_atom ::= atom ringbond* branch*
10 branch ::= '(' chain ')' | '(' bond chain ')' | '('
   dot chain ')'
```

```

11 ringbond ::= bond? DIGIT | bond? '%' DIGIT DIGIT
12 bond ::= '-' | '=' | '#' | '$' | ':' | '/' | '\'
13 dot ::= '.'

```

In line 2 the smiles string gets specified as a chain which is defined in line 8. What can be seen in this line is the concept of chains that is very important in the SMILES encoding. Chains are used to make up the structure of a molecule. As can be derived from the grammar a chain can include additional chains as well as branches. Branches are indicated by parentheses surrounding chains. This allows to translate the structure of a molecule into a spanning tree of the corresponding molecular graph, which, if written down as string, yields the SMILES notation of the molecule. A visual representation of this process can be seen in figure 4. Note that the \$ symbol for quadruple bond in line 12 is only available in the OpenSMILES implementation.

```

15 Atoms
16
17 atom ::= bracket\_atom | aliphatic\_organic |
        aromatic\_organic | '*'
18
19 Organic Subset Atoms
20 aliphatic\_organic ::= 'B' | 'C' | 'N' | 'O' | 'S' |
        'P' | 'F' | 'Cl' | 'Br' | 'I'
21 aromatic\_organic ::= 'b' | 'c' | 'n' | 'o' | 's' |
        'p'

```

In this set of rules the atoms get defined. The subset of all possible atoms that makes up the organic subset is treated differently than the others because it is used so often. For this specific set short letter codes were implemented in the SMILES standard that allow to access them more easily than by writing the bracketed atom variant. Lower case variants are used to indicate aromaticity.

```

23 Bracket Atoms
24
25 bracket\_atom ::= '[' isotope? symbol chiral? hcount
        ? charge? class? ']'
26 symbol ::= element\_symbols | aromatic\_symbols | '*'
27 isotope ::= NUMBER
28 element\_symbols ::= 'H' | 'He' | 'Li' | 'Be' | 'B'
        | 'C' | 'N' | 'O' | 'F' | 'Ne' | 'Na' | 'Mg' | '
        Al' | 'Si' | 'P' | 'S' | 'Cl' | 'Ar' | 'K' | 'Ca
        ' | 'Sc' | 'Ti' | 'V' | 'Cr' | 'Mn' | 'Fe' | 'Co
        ' | 'Ni' | 'Cu' | 'Zn' | 'Ga' | 'Ge' | 'As' | '

```

```

Se' | 'Br' | 'Kr' | 'Rb' | 'Sr' | 'Y' | 'Zr' | '
Nb' | 'Mo' | 'Tc' | 'Ru' | 'Rh' | 'Pd' | 'Ag' | '
'Cd' | 'In' | 'Sn' | 'Sb' | 'Te' | 'I' | 'Xe' | '
'Cs' | 'Ba' | 'Hf' | 'Ta' | 'W' | 'Re' | 'Os' | '
'Ir' | 'Pt' | 'Au' | 'Hg' | 'Tl' | 'Pb' | 'Bi' | '
'Po' | 'At' | 'Rn' | 'Fr' | 'Ra' | 'Rf' | 'Db' | '
' | 'Sg' | 'Bh' | 'Hs' | 'Mt' | 'Ds' | 'Rg' | 'Cn'
' | 'Fl' | 'Lv' | 'La' | 'Ce' | 'Pr' | 'Nd' | 'Pm
' | 'Sm' | 'Eu' | 'Gd' | 'Tb' | 'Dy' | 'Ho' | '
Er' | 'Tm' | 'Yb' | 'Lu' | 'Ac' | 'Th' | 'Pa' | '
'U' | 'Np' | 'Pu' | 'Am' | 'Cm' | 'Bk' | 'Cf' | '
'Es' | 'Fm' | 'Md' | 'No' | 'Lr'
29 aromatic_symbols ::= 'b' | 'c' | 'n' | 'o' | 'p' |
's' | 'se' | 'as'

```

This variant can be used for any atom - including those already defined in the organic subset. Always surrounded by square brackets the element symbol may be accompanied by a number indicating the specific isotope, a chirality indicator, a hydrogen count and a charge. In line 28 the complete set of possible elements is listed.

31 Chirality

```

32
33 chiral ::= '@' | '@@' | '@TH1' | '@TH2' | '@AL1' |
'@AL2' | '@SP1' | '@SP2' | '@SP3' | '@TB1' | '
@TB2' | '@TB3' | ... | '@TB20' | '@OH1' | '@OH2'
| '@OH3' | ... | '@OH30' | '@TB' DIGIT DIGIT |
'@OH' DIGIT DIGIT

```

Listed here are all the symbols that can be used to encode stereochemical information. The two most widely used symbols are @ and @@. The first one indicates that the following branches are listed in anticlockwise direction while the second one indicates clockwise direction. This list shows one of the shortcomings of the SMILES standard - stereochemistry is only seen as an atomic property therefore more complex stereochemical concepts like helices cannot be encoded.

35 Hydrogens

```

36
37 hcount ::= 'H' | 'H' DIGIT
38

```

39 Charges

```

40
41 charge ::= '-' | '-' DIGIT? DIGIT | '+' | '+' DIGIT
? DIGIT

```

42

43 Atom Class

44

45 `class ::= ':' NUMBER`

Lastly the hydrogen count, the charge of an atom and the class are defined. All of them are not compulsory for bracketed atoms but assumed to be zero if left out. The last entry 'class' does not define a molecular property but can be used by software to indicate specific atoms as needed. There is no official way how classes should be used.

2.1.3 Descriptive Approach

The above defined description as a formal grammar are great when creating computer programs to read or write SMILES but not necessarily easy for humans to read. In the following section the same content is provided in a more descriptive form following the original papers that did not include grammars[2],[6].

As noted before SMILES tries to represent a molecule as a graph. Therefore the two essential elements a SMILES encoded molecule is made of nodes (atoms) and vertices (bonds):

Atoms

The SMILES notation accepts atoms as the letter(s) representing the atom type in the periodic table. Atoms may be specified in square brackets or in the case of the so called "organic subset" consisting of B, C, N, O, P, S, F, Cl, Br and I without brackets, implicitly including hydrogen atoms according to the open valences of the atoms. A simple SMILES string may be made up of only one letter as seen in table 1.

SMILES	MOLECULE
C	methane (CH ₄)
N	ammonia (NH ₃)
O	water (H ₂ O)
Cl	hydrogen chloride (HCl)

Table 1: Simple examples for valid SMILES strings

Elements that are declared with square brackets need to have hydrogens added explicitly. Therefore this accounts for all the elements not included in the organic subset as well as atoms from the organic subset declared that way. Hydrogen atoms may be added by adding a uppercase H to the element followed by a number declaring the amount of hydrogen atoms. For atoms in square brackets charges may be added in the same way as hydrogen atoms.

SMILES	MOLECULE
[He]	singular helium atom
[CH ₄]	methane (CH ₄)
[OH ₃ ⁺]	hydronium cation
[Cl ⁻]	chlorine anion

Table 2: Example of element declaration in SMILES strings

Bonds

The SMILES notation is able to include four different bond types: single (-), double (=), triple (#) and aromatic (:) bonds. The symbols are added between two atoms to represent a bond between the two. Single bonds may be omitted and aromatic bonds may be omitted if a special notation is used as seen in section 2.1.

Side Chains

So far all the tools to produce SMILES strings for single chains have been discussed. Retaining the idea of making this notation easy to write and read adding side chains is easy. Side chains are added following to the atom branch in form of parentheses. This concept is visualized in table 3.

SMILES	MOLECULE
CC(C)C	Isobutane
CCC(C)N	2-butanamine
CC(C(C)O)N	3-amino-2-butanol

Table 3: Simple examples of molecules with sidechains

Cycles

SMILES considers all structures as linear with intermolecular connections to form cyclic structures. The two connection points are marked with numbers following the atom letters. The two connecting atoms share the same number, this way it is possible to create multiple cycles, by using different numbers for each cycle. Although it is not necessary the numbers start usually with 1 and ascend up to 9. Ring closure numbers are reusable once a cycle is closed again therefore structures with more than 9 cycles can easily be constructed. In the rare cases this doesn't suffice double digit numbers may be used if a leading percentage sign (%) is added. The examples in table 4 on the facing page should help the reader to understand the discussed concepts.

SMILES	MOLECULE
<chem>C1CCCCC1</chem>	Cyclohexane (C ₆ H ₁₂)
<chem>BrC1CCCCC1</chem>	Bromocyclohexane C ₆ H ₁₁ Br)
<chem>C1CC1</chem>	Cyclopropane (C ₃ H ₆)
<chem>C1CCC2CCCCC2C1</chem>	Decahydronaphtalene (C ₁₀ H ₁₈)

Table 4: Several cyclic structures - including Decaline with two cycles - encoded as SMILES strings.

Aromaticity

The last concept of the SMILES notation discussed in detail here is aromaticity. As already mentioned in the bond section, aromatic bonds don't have to be added explicitly but may be omitted. If an aromatic structure only consists of C, N, O, P, S, As and/or Se automatic aromaticity detection works in SMILES. There are multiple ways aromatic compounds may be defined so they are treated internally like that. Firstly the aromatic bonds may be added with $\cdot$. Secondly a cycle with alternating single and double bonds is treated as aromatic if it fulfills the Hückel's Rule. Thirdly and most widely used aromatic atoms may be declared by writing them in lowercase letters. For all three mentioned ways of defining aromatic structures the implicit hydrogen adding works as expected, even for more complicated aromatic systems - with some exceptions - both shown in table 5.

SMILES	MOLECULE
<chem>c1ccccc1</chem>	Benzene (C ₆ H ₆)
<chem>c1ccncc1</chem>	Pyridine C ₅ H ₅ N)
<chem>c1ccc2ccccc2c1</chem>	Naphthalene (C ₁₀ H ₈)
<chem>C1CCC2CCCCC2C1</chem>	Decahydronaphtalene (C ₁₀ H ₁₈)

Table 5: Several cyclic structures - including Decaline with two cycles - encoded as SMILES strings.

Further Concepts

Some other features only mentioned here are declaration of the absolute configuration of stereocenters, declaration of isotopes and the very important feature of generating a canonical SMILES string that serves as a unique identifier for a chemical compound[6].

2.1.4 Building up a SMILES string

This short section build up on the last two sections and explains with the use of an example how a SMILES string is generated. For this example a rather simple molecule only consisting of carbon atoms is chosen. Although simple in its atomic composition branches and cycles are present.

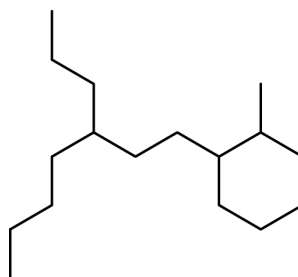


Figure 2: Sample molecule used to explain the build up process of a SMILES string. Note that this molecule contains branches as well as a cycle.

In a first step all the rings in a molecule have to be detected. Once this is done one of the ring bonds is chosen to be deleted. Deleting one bond in a ring creates a linear structure and can therefore be represented by a simple string of atomic symbols. To make sure the information about the ring is not lost labels (numbers) are added to the atoms that boarder the deleted bond. Any bond in the ring may be chosen which is one of the reasons the standard SMILES may generate different strings for the same molecule.

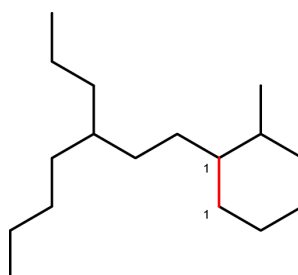


Figure 3: One ring was detected and linearized by deleting a bond inside the ring. The deleted bond is marked in red and numbers have been added to the atoms surrounding the former bond.

Once the molecule is linearized the main chain is detected. This may be any chain in the molecule - another reason why different strings may be generated. Once this chain is defined all the branches are added in parantheses following the branching atom.

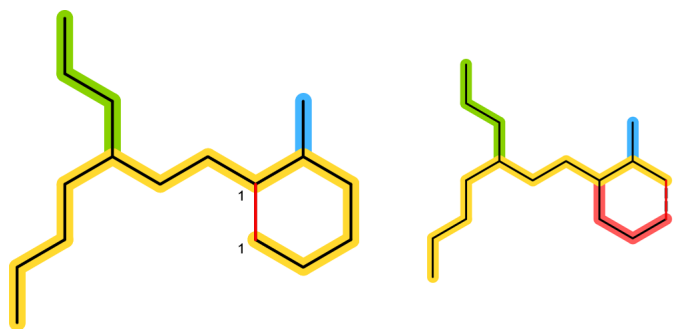


Figure 4: The main chain (yellow) and branches (green and blue) are colored in one possible way. For the shown molecule several combinations of ring bond breaking and branch assignment are possible resulting in different SMILES string representations for the same molecule, as seen on the right side of the figure.

Once the bond breaking and branching is done the SMILES string may be created by following the main chain (highlighted in yellow) starting at one of the two sides and writing down all the atoms encountered on the way to the other end. Whenever a branch is encountered the branch is written down before the main chain continues. In this case this leads to the following SMILES string (branches marked in the respective colours). Note that the most common type of bond in organic molecules, the single bond is not recorded in the SMILES string, to keep the representation compact and short.

1	CCCCC	(CCC)	CCC ₁ C	(C)	CCCC ₁
2	CCCCC	(CCC)	CCC	(CCC ₁)	C (C) C ₁

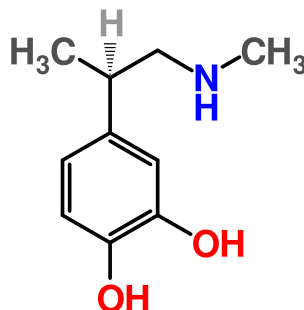
Figure 5: Two of the possible SMILES strings for the molecule shown in the previous figures. One shows the left branching in figure 4, two shows the right variant. The main chain is colored with a yellow background while the branches are colored in green, blue and red. The numbers indicate the ring closure numbers.

2.2 INCHI

InChI intended to provide a unique representation of chemical compounds[7] the IUPAC International Chemical Identifier format (short InChI) was developed in an IUPAC project from 2001 to 2005[8]. To make it easier for the reader to understand further concepts discussed in this thesis a short introduction in the technical aspects of the InChI encoding is provided at this point. To make this information as ac-

cessible as possible examples will be provided for every feature the InChI format provides.

The InChI encoding is based on several layers that contain different aspects of the described molecule. In the following section the most important layers shall be discussed and illustrated by examples.



InChI=1S/C9H13NO3/c1-10-5-9(13)6-2-3-7(11)8(12)4-6/h2-4,9-13H,5H2,1H3/t9-/mo/s1

Figure 6: Example of an InChI code - this string encodes the structural information of epinephrine. Note the different layers separated by slashes.

2.2.1 InChI structure and basic layers

The principle structure of an InChI is illustrated here by gradually adding more information to a molecule. For this example L-Alanine as an anion will be used. This part is based on the current version of the technical manual to provide an up-to-date description of the InChI standard[7].

Each InChI code has to start by declaring that the following code will be in InChI. This is done by the prefix **InChI=**. Next the InChI version needs to be added. The current version number is **1** and to show that the code was generated in compliance with the InChI standard a **S** is added.

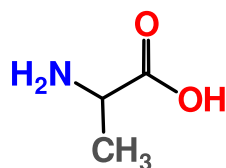
InChI=1S

Figure 7: Basic prefix of any InChI code.

2.2.2 Main Layer

Following the declaration of the file format and the specification of the version the first chemical information is added. In the main layer only basic information about the chemical compound is found. This includes the chemical formula, the connection of all the atoms **/c** and the specific declaration how many hydrogen atoms are connected **/h** to each atom. Sticking with the example of an L-Alanine anion, in the

main layer only alanine, without the stereochemistry at the α -Carbon or the charges on the acid- and amino-group, could be encoded.

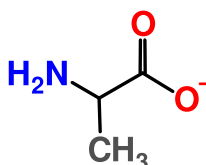


InChI=1S/C₃H₇NO₂/c1-2(4)3(5)6/h2H₄H2,1H₃,(H,5,6)

Figure 8: **Alanine without defined stereocenters** The different parts of the main layer are highlighted by the use of different colors: In **red** is the chemical formula of the molecule. The declaration of the connection follows in **green** and lastly the amount of hydrogens connected to each atom is displayed in **blue**

2.2.3 Charge Layer

To make up for the fact that chemicals may appear in the form of charged compounds the InChI specifications allow to note this in the second major layer. This charge layer consists of two sublayers, each of them may be omitted if the entry was equal zero. The two sublayers are the proton layer /**p** that denotes how many hydrogen atoms have to be added or removed from the chemical formula to yield the target structure. The second component charge layer /**q** declares the total charge of the component encoded in the InChI string. This sublayer is not necessary if the total charge is only defined by the addition or removal of hydrogen atoms.

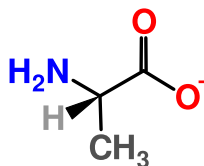


InChI=1S/C₃H₇NO₂/c1-2(4)3(5)6/h2H₄H2,1H₃,(H,5,6)/p-1

Figure 9: **Alanine as an anion without defined stereocenters** Note that the main layer in **grey** is still the same as before, although we removed a hydrogen atom when adding the negative charge. This difference is resolved by mentioning in the charge layer in **red** there is one hydrogen missing. As can be seen in the string here the localization of the charge is not defined in the InChI itself. Only the total amount of positive and negative charges is included.

2.2.4 Stereochemical Layer

The third major layer describes the spacial characteristics of the encoded compound. It can generally be discussed as two independent aspects - sp^2 and sp^3 symmetry. Separating these two cases allows to include the more often specified E/Z stereochemistry layer **/b** and ignore the more rarely defined R/S configuration **/t** and **/m**. This makes it possible to strip away the sp^3 layer and compare structures in databases or other sources only by their E/Z configurations. A third sublayer is included to define the type of stereoinformation **/s**.



InChI=1S/C₃H₇NO₂/c1-2(4)3(5)6/h2H,4H2,1H3,(H,5,6)/p-1/**t2-**
/m0/s1

Figure 10: **L-Alanine anion** Although we defined now a stereocenter at the α -Carbon the first two layers (here in grey) still look the same as before. The additional information is only added in the third layer, marked here in red. As there is no E/Z configuration present the **/b** layer is not shown here. The **/t** layer signifies that the atom with number 2 is present in the L configuration. The **/m** layer conveys the information that the encoded structure is exactly the one entered as input, while **/s1** means that exact stereochemistry was requested when encoding the compound.

2.2.5 Further Layers

The only other layer that is allowed in the standard InChI encoding that was not discussed above is the isotopic layer. It specifies the information about isotopes present in a certain compound. This is extremely important for instance in applications where NMR is used as an analytical instrument. Other layers that may be used in a non standard environment are a fixed-H layer and a reconnected layer that specifies the structure of compounds with with coordinated bonds like metal complexes.

2.3 INCHI KEY

The InChI key is a 27 symbols long hashed version of the standard InChI encoding using the SHA-256[9] algorithm. The main purpose of this hashed version is to generate a shortened representation of the InChI for use in modern computation environments. The full

InChI is easier to read for humans, but also longer so searching for this string is more resource demanding in computer only applications. One notable example of InChI Key usage is searching for molecules with the Google search engine.[10]

Because the InChI Key is generated using a hashing function there is a very low but nonzero chance, that two different molecules could generate the same InChI Key. Another implication of the hashing procedure is that an InChI Key can not be reversed to its original InChI form, and therefore the character encoding itself does not convey any structural information of the underlying molecule.

QNAYBMKLOCPYGJ-UHFFFAOYSA-N

Figure 11: InChI Key representation of Alanine without defined stereocenters as seen in figure 8

2.4 MOL

The MOL file format was developed at Molecular Design Ltd. and published in 1992[11]. It provides a way to specify molecules by only using the spatial coordinates of atoms as well as which atoms are connected. This feature is closely related to the older connection table which was also developed at Molecular Design Ltd. This approach adds a header block to the connection table that allows to specify additional information about the molecule as well as the software used to create the MOL file.

In the following section the MOL file specifications will be explained - the definitions were published in[11]:

```

1 pyridine
2 OpenBabel01071512592D
3
4 6 6 0 0 0 0 0 0 0 0 0999 V2000
5 -0.8660 -0.5000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
6 -1.7321 -0.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
7 -1.7321 1.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
8 -0.8660 1.5000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
9 -0.0000 1.0000 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0
10 0.0000 0.0000 0.0000 N 0 0 0 0 0 0 0 0 0 0 0 0
11 1 6 1 0 0 0 0
12 1 2 2 0 0 0 0
13 2 3 1 0 0 0 0
14 3 4 2 0 0 0 0
15 4 5 1 0 0 0 0
16 5 6 2 0 0 0 0
17 M END

```

Snippet 1: MOL file for Pyridine - generated with OpenBabel

In Snippet 1 an example of a MOL file is shown. This specific file was generated using OpenBabel[12] and a SMILES string of pyridine as an input. The molecule name was added manually.

2.4.1 Header

The header file consists of three lines. In the first line the name of the molecule may be added optionally. This line may not be longer than 80 symbols. The second line allows for a broad set of attributes. Usually the name of the generating program is added. Other attributes are user initials, date and time of generation, dimensional codes, scaling factors and energies for modeling programs. This line may also be left blank. The third line is reserved for comments but must be present, even if left blank.

```
1 pyridine
2 OpenBabel01071512592D
3
```

Snippet 2: "Header of a MOL file"

2.4.2 Counts

This line defines several parameters and mentions the amount of atoms, bonds or in case of reaction, components. The most important values are the first two which represent the amount of atoms and the amount of bonds in the molecule. Also very important is the last entry which specifies the version of the connection table used. In this example V2000 is used. Another option would be V3000 which allows entry numbers greater than 999. In this work only the V2000 standard will be discussed. For more information on V3000 please obtain the file specification from Accelrys[13]. Other options include a flag for chirality, the amount of reactants, products and intermediates in a reaction. The number 999 before the version declaration is a depreciated feature concerning the properties block and should be set to 999.

```
4 6 6 0 0 0 0 0 0 0 0999 V2000
```

Snippet 3: "Atom and Bond Counts"

2.4.3 Atoms Block

This block defines the positions of each atom in a xyz format. The position is followed by a specification of the atom type. Following the atom type the following attributes may be defined in this order: mass difference, charge, stereo configuration, hydrogen count, flag for ignoring the stereo configuration, valence, flag to specify no hydrogen atoms (redundant with hydrogen count), reaction type, reaction component number, atom mapping number, flag for retaining or inverting

the configuration and a flag for search queries that defines whether the atom must be exactly as shown or not.

```

5      -0.8660   -0.5000   0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
6      -1.7321   -0.0000   0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0  0  0
7      -1.7321    1.0000   0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0  0  0
8      -0.8660    1.5000   0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0  0  0
9      -0.0000    1.0000   0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0  0  0
10     0.0000    0.0000   0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0  0  0

```

Snippet 4: "Atom positions and types"

2.4.4 Bonds Block

The bond block defines a connection between two atoms in a molecule. The first number is the number of one bonding atom, the second the number of the other bonding partner. The atoms are ordered according to their number in the atoms block. Following the definition of the bond is the bond type, the stereo definition, an unused value, an indicator of the bond topology (ie. ring/chain/other) and the reaction center status.

```

11     1  6  1  0  0  0  0
12     1  2  2  0  0  0  0
13     2  3  1  0  0  0  0
14     3  4  2  0  0  0  0
15     4  5  1  0  0  0  0
16     5  6  2  0  0  0  0

```

Snippet 5: "Bond types and connections"

2.4.5 Properties Block

The properties block allows for additional properties - some only readable by proprietary software - to be added to the file. The file ends with the line "M END".^[14]

```

17 M  END

```

Snippet 6: "Properties Block - empty in this example"

2.5 JME

This is the file format used internally by the JavaScript Molecular Editor (JSME)^[15]. While working on implementing the editor in the webservice described later no documentation on the file format could be found but reverse engineering proved to be very easy.

In essence the .jme file is a single string encoding all the information needed to display a molecule in a 2 dimensional space. This

includes the coordinates of each atom as well as the atom type and the connectivity information as well as the bond types.

The string can be divided into two parts, the atoms and the bond set. The file only utilizes blank spaces to separate values. In the first part of the string each atom is listed as a triplet of two coordinates (x and y axis) as well as the abbreviated atom type as seen in the periodic table. The second part consists of triplets of bonds made up of two numbers indicating the atoms that form the bond and a third number indicating the bond type.

A simple conversion tool was written as part of the master thesis project that converts SMILES encoded molecules to JME strings, readable by the molecular editor.

WALK COUNTS

Walk counts are a way to calculate the complexity of a molecule. In chemistry it is generally agreed upon that complexity increases with size, branching, cyclicity, unsaturation and amount of heteroatoms. [16] To convert these arbitrary values to comparable numbers measures for complexity are needed. Several of those measures were compared by Bonchev in 1997[17]. In this work the Walk Counts were chosen and more specifically the total walk counts as well as the molecular walk counts of different lengths.

Atomic Walk Counts were introduced by Gerta and Christoph Rücker in 1993 [18]. These are just descriptors for each atom in a molecule how many walks of length k are possible, starting from this atom. At this point an important difference between walks and trails should be mentioned. Walks are a sequence of vertices and edges, with each edge's endpoint being the starting point of the next edge. In this set edges may appear multiple times. A trail is like the walk a sequence of edges connected by common vertices but with each edge being unique.[19] It should be noted that in chemical literature the term path is also often used wrongly as a synonyme for trail[18]. In graph theory a path is like walks and trails a set of connected edges but with each edge and vertex being unique.[20]

3.1 ATOMIC WALK COUNT

In its most basic form the atomic walk count is calculated as shown in equation (1) on the following page using a connection table of the molecule (example in table 6 on the next page). This connectivity table only shows which vertices are connected via edges. In this simple form no chemical information like bondtype or atom type is stored. This assumes the molecule as a graph without any weights connected with the vertices and edges.

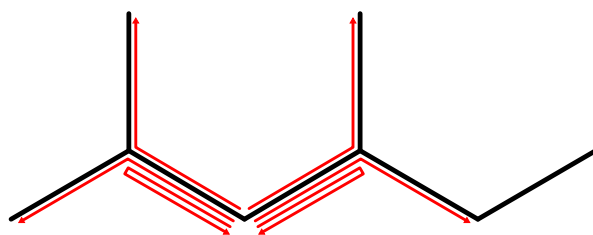
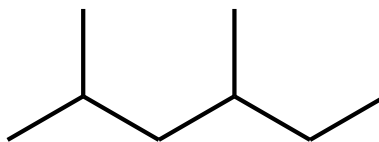


Figure 12: Atomic walks of length 2 starting from the same C atom



atom #	1	2	3	4	5	6	7	8
1	0	1	0	0	0	0	0	0
2	1	0	1	0	0	0	1	0
3	0	1	0	1	0	0	0	0
4	0	0	1	0	1	0	0	1
5	0	0	0	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	1	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0

Table 6: Connectivity table for 2,4-Dimethylhexane

To calculate the atomic walk count (awc) the connection matrix is simply raised to the power of the desired walk length and all the entries in the line corresponding to the atom chosen are summed up as see in equation (1)

$$awc_k(i) = \sum_j a_{ij}^{(k)} \quad (1)$$

where i, j = atom number

k = walk length

a = entry in the connectivity matrix A

All the atomic walks of length 1 to k can be summed up as shown in equation (2). This represents all the possible atomic walks in a molecule starting from a certain atom.

$$awcs(i) = \sum_k \sum_j a_{ij}^{(k)} \quad (2)$$

3.2 TOTAL WALK COUNT

The total walk count of all the walks in a molecule is just the sum of all the possible atomic walks in a molecule starting from each atom in the molecule. Since walks are symmetrical the sum has to be divided by two to result in the number of unique walks in a molecule as seen in equation (3)

$$twc = 1/2 \sum_i \sum_k \sum_j a_{ij}^{(k)} \quad (3)$$

This short example should make the development of the total walk count clear. In table 7 the connection table of propane is listed on the left side. Each entry in the matrix represents one possible connection. A 1 in an entry represents a connection a 0 indicates no connection. At the same time this also enumerates all the possible walks of length one starting from these atoms. The corresponding amount are listed as the vector awc_1 .

On the right side the connection matrix got squared. The numbers lost their topological connection to the original molecule but still the sum of all entries for one atom results in the amount of walks with length two possible starting from this atom as listed in awc_2 .

#	A^1			awc_1	A^2			awc_2
1	0	1	0	1	1	0	1	2
2	1	0	1	2	0	2	0	2
3	0	1	0	1	1	0	1	2

Table 7: Development of the atomic walk counts for propane. A^1 is identical to the connection table of the molecule.

In table 8 the total walk count is finally calculated. All the atomic walk count vectors are listed and summed up to result in the vector $awcs$. Each entry in this vector indicated the amount of walks with length one to two possible starting from this atom. The sum of all these entries would result in the amount of walks possible with mirrored walks counted as unique walks. To remove symmetrically mirrored walks the result is divided by two. As seen in the table the amount of unique walks of length one or two in propane is only five.

#	awc_1	awc_2	$awcs$
1	1	2	3
2	2	2	4
3	1	2	3
mwc	4	6	twc: 5

Table 8: Atomic Walk Counts, Sums of Atomic Walk Counts, Molecular Walk Counts and the Total Walk Count for propane

GREEN CHEMISTRY

The goals of green chemistry have been compiled by Paul Anastas and John Warner in a nice way in their book *Green Chemistry: Theory and Practice* [21], [22]:

1. **Prevention** of waste
2. **Atom Economy** should be maximized
3. **Less Hazardous Chemical Synthesis** should be designed to generate little or no toxic substances
4. **Designing Safer Chemicals** to reduce toxicity without affecting functionality
5. **Safer Solvents and Auxiliaries** need to be developed and amount of solvents and auxiliaries should be reduced
6. **Design for Energy Efficiency** and recognize energy consumption as an environmental impact. Reactions should be performed at ambient temperature and atmospheric pressure.
7. **Use of Renewable Feedstocks** instead of depleting raw materials
8. **Reduce Derivatives** such as protected/blocked versions to reduce waste.
9. **Catalysis** is superior to stoichiometric reagents.
10. **Design for Degradation** so products degrade at the end of their function and don't persist in the environment.
11. **Real-time analysis for Pollution Prevention** allow for in-process monitoring and control of potential hazards.
12. **Inherently Safer Chemistry for Accident Prevention** should be chosen to prevent accidents.

4.1 GREEN METRICS

To implement the above goals in an industrial environment - but also in low scale synthetic labs - synthesis plans have to be comparable

in regards to these properties. Removing environmentally hazardous steps is difficult without knowing which steps are problematic. Also comparing different plans for the same compound needs to be based on hard facts instead of acquired chemical knowledge. Comparing large numbers of plans is only possible if plans are stored in a machine readable format that allows for computational analysis.

Several analysis methods have been developed over the years to provide metrics for the greenness of reactions or whole synthesis plans. Several of the most important metrics are listed below including instructions how to calculate them and explanations on how to use them.

4.1.1 Yield ϵ

A basic chemistry metric. The yield defines the ratio between expected output under optimal conditions and the obtained output. Optimal conditions are usually assumed as stoichiometric conversion, therefore limited by the compound with the lowest availability as seen in table 9.

The yield is simply calculated as the fraction of obtained product by the expected product as shown in equation (4).

$$\text{reaction } \epsilon [\%] = \frac{\text{real } \epsilon [\text{g or mol}]}{\text{optimal } \epsilon [\text{g or mol}]} 100 [\%] \quad (4)$$

A simple example should make this concept easy to understand. Let us assume a simple equation (5) to work with:



In this example X will be the wanted product. This table 9 shows the amount of all the educts and measured amounts of products.

EDUCT [MOL]		PRODUCT [MOL]	
A	1.20	X	0.73
B	2.04	Y	not measured
C	0.94	Rest	not measured

Table 9: Amounts of products and educts

When comparing the stoichiometric factors and the amounts of product used it is obvious that C is the limiting factor in this reaction, therefore the *optimal* ϵ is assumed to be 0.94 mol. In this example the *real* ϵ is 0.73 mol or the amount of X produced. The reaction yield can now be easily calculated as seen in equation (6).

$$\text{reaction } \epsilon [\%] = \frac{0.73 \text{ mol}}{0.94 \text{ mol}} 100\% \approx 78\% \quad (6)$$

4.1.2 Environmental Factor

First introduced by R.A. Sheldon in 1992 [23] the environmental factor (*E* factor) as a very early descriptor in the field of green chemistry. To calculate *E* one simply divides the mass of waste by the mass of the final product as seen in equation (7).

$$E \text{ factor} = \frac{\text{mass of total waste}}{\text{mass of final product}} \quad (7)$$

The ideal value for the *E* factor is 0. Depending on the industry this may be very close to the current state of technology as seen in table 10.

INDUSTRY	ANNUAL PRODUCTION [TONNES]	E FACTOR
Oil refining	10 ⁶ -10 ⁸	<0.1
Bulk chemicals	10 ⁴ -10 ⁶	1-5
Fine chemicals	10 ² -10 ⁴	5-50
Pharmaceuticals	10 ¹ -10 ³	25-100

Table 10: Several branches of chemical industry compared by their *E* factor - according to Sheldon[23]

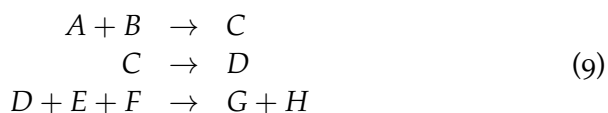
Environmental Factor based on molecular weight

4.1.3 Atom economy AE

The concept of Atom economy introduced by Trost in 1991 [24] answers the question, how many atoms from the educts can be found in the final product. *AE* is usually expressed as a fraction or percentage. For simple one step reactions the *AE* can be expressed as seen in equation (8).

$$AE = \frac{M_w(\text{product})}{\sum M_w(\text{educts})} \quad (8)$$

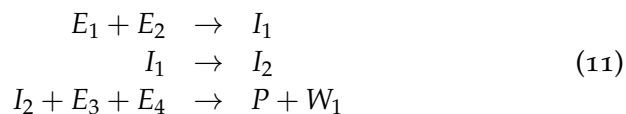
Multistep reactions or even whole synthesis plans may be calculated as easily as described by [25]. For this a dummy synthesis, with *G* being the wanted product, may be considered:



This leads to the following expression for the *AE*:

$$AE = \frac{M_w(G)}{\sum M_w(A, B, E, F)} \quad (10)$$

In a more abstract way with E_i denoting educts, I_i intermediates, P the product and W_i waste products:



The equation (10) on the previous page can be written as such:

$$AE = \frac{M_w(P)}{\sum_{i=1}^N M_w(E_i)} \quad (12)$$

Even more complex multistep reactions and convergent synthesis plans including reactions with different stoichiometric factors may be evaluated as proposed in [26]. In essence stoichiometric factors are introduced to the calculation of AE. Components that are introduced at multiple steps in the synthesis have their stoichiometric coefficients added up to represent the total amount of mass added to the system.

Actual atom economy

As described before the AE only takes the molecular weight of the substances in the reaction into account. For a realistic evaluation of the atom economy the yield of each reaction needs also to be considered. In [27] this is simply done by multiplying AE by ε as seen in equation (13).

$$AAE = \varepsilon AE \quad (13)$$

4.1.4 *Carbon efficiency*

The carbon efficiency CE is a rather simple measure that could be seen as a simplification of the AE. First proposed by [28] the CE can be calculated as shown in equation (14).

$$CE = \frac{\text{total carbon in product}}{\text{total carbon in educts}} \times 100 \quad (14)$$

4.1.5 *Effective mass yield EMY*

The effective mass yield compares the product mass produced in a reaction or synthesis with the amount of toxic or otherwise dangerous materials used in the process of generating the desired product. In the original publication[29] this value is defined as a percentage as seen in equation (15). Since the value could get higher than 100% it should be pointed out that defining this value as a simple ratio may be the more sensible way.

$$EMY = \frac{\text{mass of product}}{\text{mass of dangerous reagents}} \times 100 \quad (15)$$

4.1.6 *Strategy efficiency metrics*

In addition to these parameters Andraos defined some metrics that are strictly intended for synthesis plans[30]. A few of those shall be listed here:

1. **Type of Plan** either linear, convergent or divergent[31] [32]
2. **Number of reaction steps M** one step is defined as the interval between two isolated substances in a synthesis plan
3. **Number of bonds made per step B/M** ratio of the sum of target bonds made and the amount of reaction steps M
4. **Number of nonbonding steps per reaction stage** the fraction of the number of reaction stages that do not form target bonds and the amount of reaction stages

Part III

Methods

SYNTHESIS PLANNING

5.1 DEVELOPMENT OF CHEMICAL SYNTHESIS

Although nearly all processes in nature can be traced back to an underlying chemical process, the art of using these processes to our will is rather young. Arguably the turning point in this development towards a modern understanding of chemistry with the synthesis at it's core, was in 1828 when Wöhler synthesized urea. Although the founding myth of this discovery disproving the vitalism theory may be wrong after all[33], it still could be seen as a turning point in the development of modern chemistry.

It took nearly another 100 years for another turning point in chemistry that changed the way scientists thought in this area. Although the atom in it's modern form has already been described by Dalton in 1803 [34], the discovery of the electron by Thomson in 1897[35], atom nucleus by Rutherford in 1909[36], development of an atomic model by Bohr [37] and finally the orbital model by Schrödinger [38] really changed the way chemists could assess their problems and develop ways to synthesize new molecules.

These advances lead to some very notable achievements in chemical totalsynthesis: vitamin A by Isler (1949)[39], morphine by Gates (1956)[40] and chlorophyll by Woodward (1960)[41], just to name a few.

The last major breakthrough in chemical synthesis was the idea of retrosynthesis by Corey[42]. Which was also honored by awarding him the Nobel Prize for this achievement. Since the discovery of the retrosynthetic analysis the way we published and discovered new synthesis has stayed the same. One way this could change in the future would be with the help of computers and treating synthesis as an optimization problem that computers could solve.

Although chemistry was rather fast to adopt computational approaches (total)synthesis is still seen as a form of art instead of a computable problem. Today it becomes more and more apparent that chemist need tools to support or guide them to come up with more efficient solutions.[43]

5.2 PROBLEMS WITH CLASSICAL SYNTHESIS PLANS

Synthesis plans are usually published in the form of peer reviewed papers in scientific journals. This process works fine if the only target is to make information public and let the scientific community know

of the news. Most of these journals are searchable which makes it easy for a synthetic chemist to retrieve information. Tools like Reaxys [44] and the Scifinder[45] from CAS have made these information easily accessible.

While this form of publication seems to be the optimal way in the view of an experimental scientist, a computational scientist finds plenty of problems with this method.

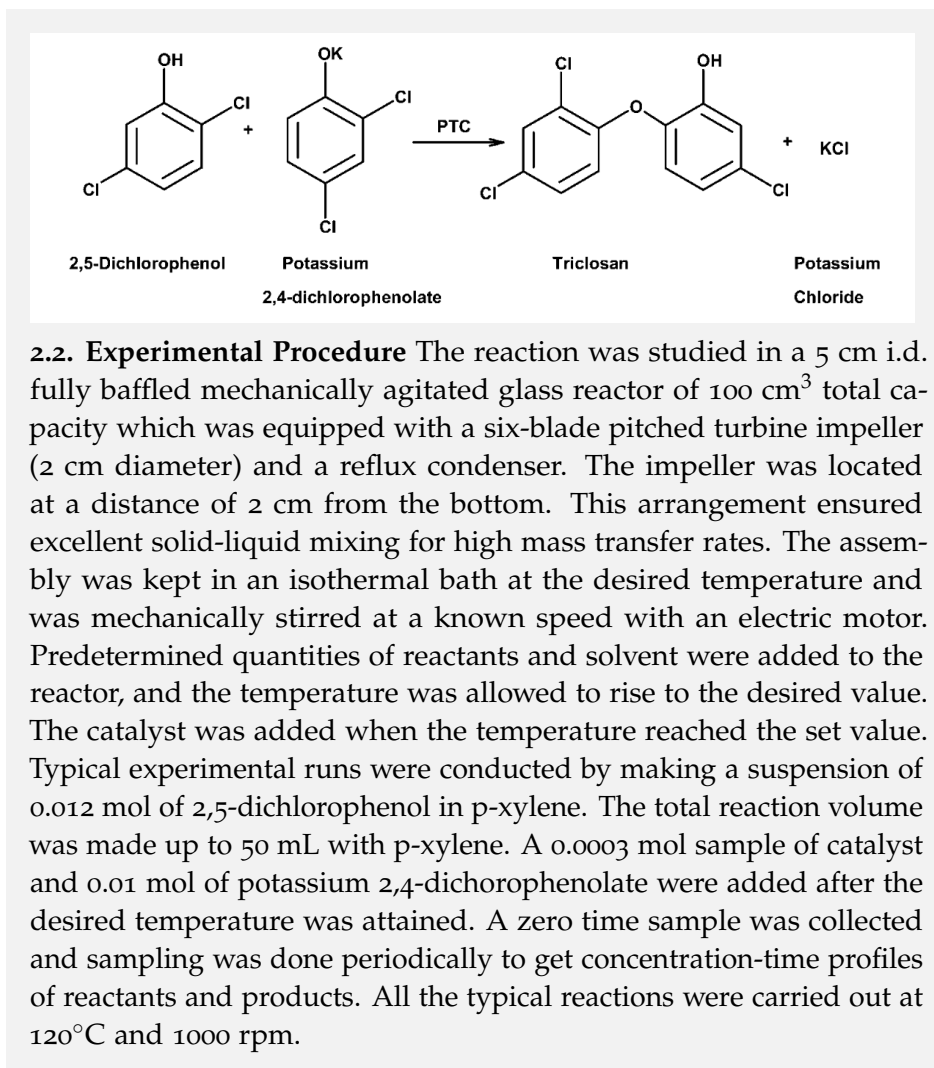


Figure 13: Published instructions how to synthesize the compound "Triclosan" - taken from [46]

An example for this way of publishing can be seen in figure 13. This kind of visualization combined with the detailed instructions is fine for a chemist if the goal is to reproduce the reaction. On the contrary it is even extremely important for a scientist trying to synthesize the compound to know all the specifics about the reaction. As can be easily seen this instruction calls for very specific reaction

vessels and quantities. But what if one wants to make calculations or simulations based on this data? What if the metrics presented in Chapter 3 - Walk Counts or Chapter 4 - Green Chemistry should be calculated?

It should be mentioned that there are already different approaches to the way we communicate information regarding synthesis plans. One such way would be the website synarchive.com[47]:

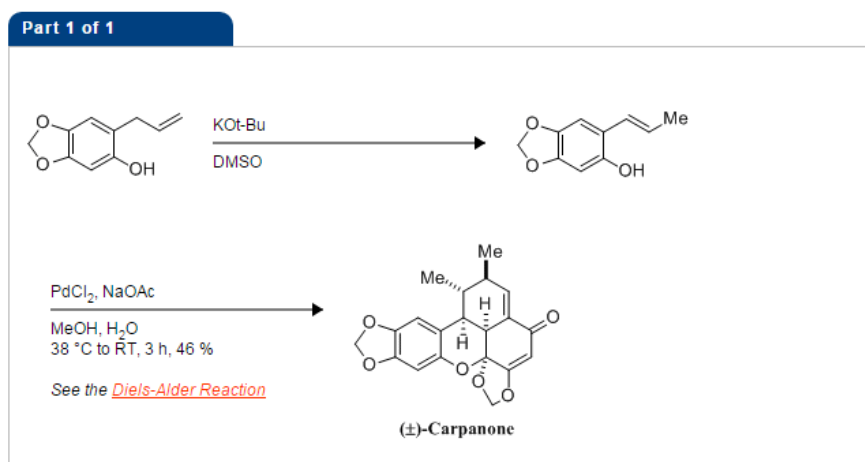


Figure 14: Synthesis plan for Carpanone - taken from[47]

On this site only the basic information is stored and shown to the user, so anyone trying to process this data can skip any filtering or preprocessing step necessary with the classical way of presenting the information. Still this approach lacks some very important aspects when considering to calculate any of the presented metrics. The molecules are represented as graphics - the least machine readable form possible and not all reactions have yields assigned to them. When a user is facing the task of comparing two or even more synthesis plans for the same molecule a lot of chemical knowledge is needed and this solution doesn't try to take this away from the user.

When comparing all the already existing solutions it becomes obvious that a new custom solution for storing synthesis information is needed if one wants to compare several synthetic approaches or simply wants to calculate how efficient a newly produced plan is. To tackle this problem a new data format was developed as presented in chapter 6 to include all the needed information in a sensible, machine readable format.

INPUT FILE SPECIFICATIONS

To input data in the SynthDB database an input file was developed that covers all the information needed to define a plan in a human readable but still compact way. Because of its uniqueness (see section 2.2) the InChI encoding was chosen to represent all compounds needed in the synthesis plan. The most current version of the input file specification specifies four main sections, some of which with subsections, that make up the file. All these sections will be covered in the next few pages.

Examples given here are from a real synthesis plan. This plan is available fully in the appendix section of this work A.

6.1 INFORMATION HEADER

This first section covers all the meta data that needs to be associated with the synthesis plan to be able to validate the data entered by the user - this includes a way to attribute a plan to a specific user, a declaration where the plan has been published and a timestamp.

6.1.1 *Timestamp*

This nonmandatory subsection covers the time the file was edited the last time. This should make it possible to differentiate between two versions with one being an improvement over the other. The time format is YYYY-mm-dd HH:ii:ssⁱ.

```
1  # Last changed timestamp: 2014-10-04 12:12:00
```

Snippet 7: Example of a timestamp

6.1.2 *Last Author*

The person who made the last changes to the file. Should be a username from the SynthDB. Entry is not essential.

ⁱ This format is specified in the text according to the PHP standard for date formatting: <http://php.net/manual/en/function.date.php> (visited on 11/06/2014)

```
1 # Last changed author: lb
```

Snippet 8: Example last editing author

6.1.3 *Literature*

Lit is a key that includes three lines separated by a newline character. The first line specifies the author(s) of the originally published article. The second line includes the caption of the article and the last line the journal including all the relevant information like, page number, issue and year of publication. This key is not essential but should be filled out whenever possible.

```
1 # Lit: Moldvai, I.; Temesvari-Major, E.; Incze,
    M.; Szentirmay, E.; Szanty, C.
2 # Enantioefficient Synthesis of r-Ergocryptine:
    First Direct Synthesis of (+)-Lysergic Acid
3 # J. Org. Chem . 2004
```

Snippet 9: Example literature listing

6.1.4 *DOI*

Specifies the Digital Object Identifier of the literature used. This key is not mandatory.

```
1 # DOI :
```

Snippet 10: Example DOI

6.1.5 *Author*

The last subsection of the header specifies the author who initially submitted the synthesis plan. This mandatory key needs to be written as FirstName LastName <email>.

```
1 # Aut: Lukas Bartonek <a0906858@unet.univie.ac.
    at>
```

Snippet 11: Example first author

6.2 SKELETON

The second section comprises of only one mandatory entry that defines the target molecule of the synthesis. As already mentioned this needs to be encoded in the standard InChI.

```
1  skeleton: InChI=1S/C16H16N2O2/c1-18-8-10(16(19)
      20)5-12-11-3-2-4-13-15(11)9(7-17-13)6-14(12)
      18/h2-5,7,10,14,17H,6,8H2,1H3,(H,19,20)/t10
      -,14-/m1/s1
```

Snippet 12: Example of a skeleton

6.3 COMPOUNDS

The third section lists all the compounds used in the course of the synthesis. This includes every molecule that is part of any reaction, while solvents and catalysts that are present but not part of the reaction don't have to be specified. Each compound is defined by an identifier and a molecule encoded in InChI. Any arbitrary combination of alphanumeric symbols may be chosen as an identifier. In this example the file was generated by a program therefore an ascending numbering was chosen as identifier. Identical molecules may be defined in multiple places. Only one compound is allowed for each line. This section is essential for the plan.

```
1  compounds:
2  comp1 InChI=1S/C11H11NO2/c13-11(14)
      6-5-8-7-12-10-4-2-1-3-9(8)10/h1-4,7,12H,5-6H2
      ,(H,13,14)
3  comp2 InChI=1S/C5H9ClO/c1-5(2,3)4(6)7/h1-3H3
4  comp3 InChI=1S/K.H2O/h;1H2/q+1;/p-1
5  [...]
6  comp104 InChI=1S/CH5N.BrH/c1-2;/h2H2,1H3;1H
```

Snippet 13: Example compounds

6.4 REACTIONS

The last section lists all the reactions in the synthesis plan. After starting the section with the key reactions: each reaction is specified by a unique identifier starting with R followed by an ascending numbering: R01, R02, ... Following the identifier a name may be specified. The name should be descriptive of the reaction but doesn't need to

be unique. Generic names like "Protection", "Functionalization", "Reduction", ... are widely used.

In the next line the main product of the reaction needs to be defined by the key `skeletonOut` followed by the molecule encoded in InChI.

The third line finally specifies the reaction using the compounds defined above including stoichiometric factors. The target molecule and its precursor need to be defined by adding vertical bars (|) left and right of the identifier. The product and educt side of the reaction are separated by an arrow (->) and stoichiometric factors are written as factor followed by an asterisk and the compound (ex: 2 * comp12).

```
1 reactions:
2 R01 "Protection"
3 skeletonOut InChI =1S/C16H19NO3/c1-16(2,3)15(20)
   17-10-11(8-9-14(18)19)12-6-4-5-7-13(12)17/h4
   -7,10H,8-9H2,1-3H3,(H,18,19)
4 |comp1| + comp2 + 2 * comp3 + comp4 -> |comp5| +
   2 * comp6 + 2 * comp7
```

Snippet 14: Example reaction

6.4.1 Reaction Meta Data

The last line of each reaction entry consists of optional meta data tags. Meta data is specified by a line starting with an at-sign followed by key value pairs. Any key may be used but typical keys include yield, solvent or catalyst. Key - value pairs are separated by blank spaces. Any amount of meta data may be added in one line.

```
1 @yield 97% cat HCl
```

Snippet 15: Example Meta Data

DATABASE

The database was developed by Carsten Grønbjerg Lützen and Daniel Fentz Johansen from SDU in Odense, Denmark. [48]. The database was created as part of the cooperation between our two groups. It was used throughout the work on synthesis plans. The most important features shall be explained here. Sadly the database is not publicly[49] available at the moment and the original authors shall be contacted for access to the code.

7.1 DATABASE STRUCTURE

The most important aspect of the database - storage - is best explained by the database schema depicted in figure 15 on the following page. A detailed explanation on specific tables shall only be done here for the most important tables.

synthesis_plan lists all the plans in the database, even those deleted using the webinterface. Entries are defined by unique integer keys. All the meta information needed for a plan is stored in this table. ID and name of the user who uploaded the plan are stored as well as the literature information concerning the plan. For a more detailed description on the literature information please refer to section 6.1 on page 49. In addition to these entries timestamps are saved for when the plan was created, uploaded and in case of deletion, deleted.

The *reaction* table lists all the reactions ever entered into the database. Each reaction is identified by an integer id and can be referenced by the skeletons inchi.

In the *reactand* table all reactands are listed. This table is connected to the molecule table but adds additional information, like if the molecule is on the input or output side and the multiplicity in the reaction.

molecule lists all the molecules ever used in synthesis plans. A molecule is identified by an integer id. The molecule type is saved in inchi and smiles encoding.

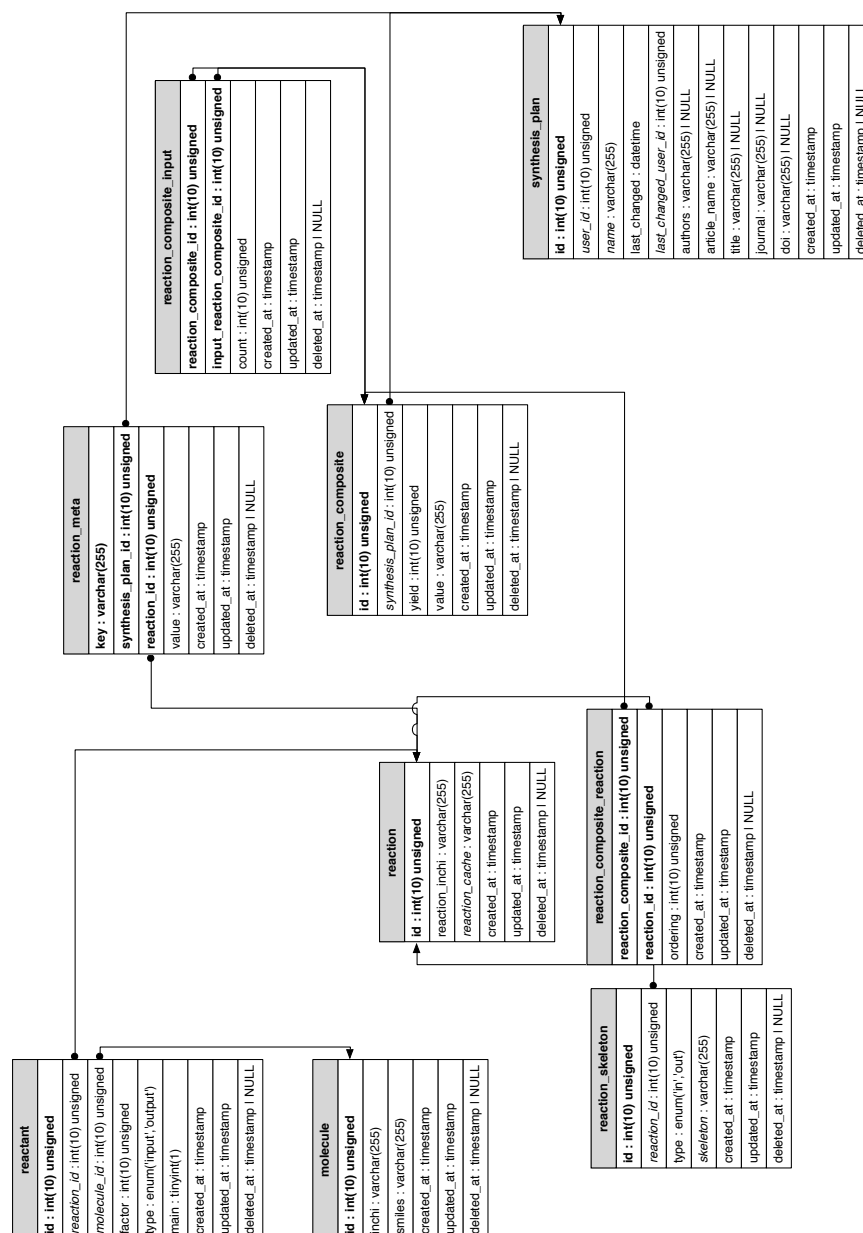


Figure 15: UML representation of the SynthDB database. Only tables related to synthesis plans are depicted. Other parts of the database like the user tables are omitted in this picture.

7.2 FRONTEND

In this section the frontend of the database will be discussed in a short manner and certain views shown.

In figure 16 on the facing page the molecule list is shown. This view is filled with all the molecules encountered in the synthesis plans. For each molecule only one entry is found in here - although a molecule may appear many times throughout all the plans. If the InChI of an

Molecule

Molecules					
10 records per page		Search:			
ID	Preview	Name	Number of Reactions	Updated At	
1		InChI=1S(Cl)(Cl)Cl	1	2015-01-22 20:37:55	
2		InChI=1S(C12H10O)c1-3-7-11(8-4-1)13-12-9-5-2-6-10-12h1-10H	1	2015-01-22 20:37:55	
3		InChI=1S(C12H7O3)c13-8-1-4-10(5-2-8)16-12-6-3-9(14)7-11(12)15h1-7H	2	2015-01-22 20:37:55	
4		InChI=1S(Cl)H	9	2015-01-22 20:37:55	
5		InChI=1S(C2H3O)c1-2(3)4h1H3	1	2015-01-22 20:37:55	
6		InChI=1S(Al3Cl9h3)1Hq+3...p-3	2	2015-01-22 20:37:55	
7		InChI=1S(H2O)h1H2	11	2015-01-22 20:37:55	
8		InChI=1S(C14H9O3)c1-8(16)11-6-9(15)2-4-13(11)19-14-5-3-10(16)7-12(14)17h2-7h,1H3	2	2015-01-22 20:37:55	
9		InChI=1S(Al3H2O)h3)1H2q+3...p-3	2	2015-01-22 20:37:55	
10		InChI=1S(C7H5O3)c8-6-3-1-2-5(4-6)7(9)11-10h1-4,10H	1	2015-01-22 20:37:55	

Figure 16: Screen capture of the molecule list in the database. The names are clickable and lead to a page that shows all the reactions the molecule is used in.

molecule is clicked a detailed view is opened in which all the reactions are listed, that use this molecule. A rendered image of the molecule - in this case created using indigo-depict[50] - is displayed to allow the user to quickly identify specific molecules.

Figure 17 shows the reaction view as seen in the database interface. Reactions are identified by an ID number that is listed in the overview table. Once an entry is selected a more detailed view of the reaction is generated, that shows a rendered image of the whole reaction on top. A short and descriptive classification of the reaction is seen as part of the table below the image. In this table also all the synthesis plans are listed that use this certain reaction.

The last view that is discussed in detail is also the most important one of the database frontend. In the synthesis plan interface seen in figure 18 all the plans saved in the database are listed. For each entry a rendering of the target molecule is shown, as well as the user who uploaded the plan to the database. A searchbox in the top right allows for a quick selection of plans.

Once a plan is selected a very detailed representation of the whole synthesis is rendered. On top of the page the publication information on the plan is shown. Below a tree is displayed with rendered representations of the product for each reaction as leaves. Other visualization options are a view that displays each molecule used in the reaction as a single node and connects these molecules with reactions. This allows to quickly find molecules that get reused very often in several reactions. A second view is another tree view that displays

View Reaction #1

Reaction Tree

Synthesis Plans Using This Reaction

10 records per page

ID	Name	Reaction Name	Last Updated
4	4 imported from commandline	Selective Chlorination	2015-01-22 20:40:37
4	4 imported from commandline	Selective Chlorination	2015-01-22 20:40:37
5	test	Selective Chlorination	2015-01-23 10:58:44
5	test	Selective Chlorination	2015-01-23 10:58:44
6	test2	Selective Chlorination	2015-01-23 11:01:15
6	test2	Selective Chlorination	2015-01-23 11:01:15
7	testtest	Selective Chlorination	2015-01-23 11:02:46
7	testtest	Selective Chlorination	2015-01-23 11:02:46

Showing 1 to 4 of 4 entries

Previous 1 Next

Figure 17: Detailed view of a reaction. The synthesis plans that use this specific reaction are listed below.

the reactions in a more detailed way, showing not only the reaction products but every educt and product.

Synthesis Plans

Upload New Synthesis Plan

Create new synthesis plan

Reactions

10 records per page

ID	Target	Name	Download	User	Updated At	Delete plan
4		4 imported from commandline	Download	Cersten Grendjeng Lützen	2015-01-22 20:40:37	Remove plan
5		test	Download	Cersten Grendjeng Lützen	2015-01-23 10:58:44	Remove plan
6		test2	Download	Lukas Bartonek	2015-01-23 11:01:15	Remove plan
7		testtest	Download	Lukas Bartonek	2015-01-23 11:02:46	Remove plan
11		LED test	Download	Lukas Bartonek	2015-01-23 14:50:58	Remove plan

Showing 1 to 5 of 5 entries

Previous 1 Next

Figure 18: A list of all the synthesis plans in the database. A search option is in the top right corner.

PERL DANCER

A Server environment developed to be used with the Perl scripting language. The server allows for building a RESTfull server [51] using only the Perl language and comes with PSGI [52] support, therefore allowing quick development of portable applications.

The server described in this work was developed using only the Perl Dancer framework as a server side application and in a completely RESTfull way.

Another nice feature the Perl Dancer server comes with, just out of the box, is the included JSON serializer. Not having to deal with encoding and decoding JSON objects proved to make development of the server much faster.

8.1 REST

Short for Representational State Transfer is a paradigm for programming web applications. This concept was developed by Roy Fielding in his doctoral thesis in 2000[53].

The main feature of this protocol design is the idea, that the server never saves a state of a client. All the information needed to perform a request has to be delivered by the client to the server. This leads to a server where a call to a certain URL always returns the same response if the call is the same.

In terms of the server developed for this work this allows for writing the webapplication in a way that request - for instance a conversion between two file formats - can always be sent to a certain URL and will always return the expected result, regardless of the state of the user.

Not saving the user state on the server side also allows for better scalability of the service since no additional data storage is needed for serving additional clients.

8.2 CALLS SUPPORTED BY THE SERVER

The server supports several REST calls. All of them are utilized by the interface but could be in theory also used by directly accessing them using curl[54] or similar programs.

InChI

1 POST `./get_inchi/`

This routine accepts a json encoded SMILES string. If a SMILES encoded molecule is detected, it gets converted to an InChI encoding, otherwise an error will be returned.

1 { "smiles" : "c1ccccc1" }

Snippet 16: Example json for `./get_inchi/`

SVG

1 POST `./get_svg/`

As the name suggests this routine generates an .svg encoded vector graphic that is optimized for depiction in the webtools out of a json encoded SMILES string. Processing of the data includes cropping the bounding box of the image and adjusting the size of the molecule.

1 { "smiles" : "c1ccccc1" }

Snippet 17: Example json for `./get_svg/`

JME

1 POST `./get_jme/`

This method expects a molecule encoded in SMILES as a json string and returns the corresponding JME string. This is needed as the molecular editor works best with JME encoded strings.

1 { "smiles" : "c1ccccc1" }

Snippet 18: Example json for `./get_jme/`

Balance

1 POST `./is_balanced/`

This function accepts a whole reaction object encoded in json, including SMILES codes for all molecules, amount of molecules and the side (input or output) the molecule belongs to. Return is either true if the reaction is balanced or false if not.

```

1 { "reactions" : [
2   {
3     "balanced" : 0,
4     "title" : "Decomposition",
5     "kat" : "",
6     "solvent" : "",
7     "temp" : "100",
8     "reactionyield" : "78"
9     "molecules" : [
10    {
11      "smiles" : "[N+](=O) ([N+](=O) [O-]) [O-]",
12      "side" : 0,
13      "amount" : "1",
14      "skeleton" : true,
15      "inchi" : "InChI=1S/N2O4/c3-1(4)2(5)6\n" }
16    {
17      "smiles" : "[N](=O)=O",
18      "side" : 1,
19      "amount" : "2",
20      "skeleton" : true,
21      "inchi" : "InChI=1S/NO2/c2-1-3\n" }
22    ]
23  }
24 ],
25 "Author" : "",
26 "email" : "",
27 "LitAuthor" : "",
28 "LitName" : "",
29 "LitJournal" : "",
30 "LitDOI" : "" }

```

Snippet 19: Example json for ./is_balanced/

Submit

1 POST ./submit/

This URL corresponds to a routine that accepts a whole synthesis plan encoded as json object and generates a synthesis plan encoded in the input file format described in chapter 6. This function only returns

true for success or false if any problem is encountered. The plan gets saved on the server for review and submission to the database.

Part IV

Results and Discussion

INPUTSERVER

The centerpiece of this masterthesis was the creation of a server that allows users to enter synthesis plans in the format discussed in chapter 6 easily without having to go through the hassle of entering data manually. Development of this system consisted of several steps, each requiring special considerations.

In the first step a fitting server environment had to be determined. In the begining a Perl CGI based solution was considered but later on exchanged to the Perl Dancer environment described in chapter 8.

The second consideration - which turned out to be the most important one - was to make sure that users of all levels of computer and programming knowledge could use this system. To make sure that this tool is as easy to use as possible a very responsive interface was designed that includes a molecular editor and a modern interface as pictured later in this chapter.

In a third step this whole system was tested thoroughly and possible problems fixed as well as additional improvements and features added. In this step most of the features that stand out to the user were added. Examples for this are the balancing tests performed on each step and the connection to the cactus[55] webservice the NCI provides to convert names for molecules into string representations easily.

9.1 COMPATIBILITY

The front-end of the inputserver was developed to be as compatible as possible with as many systems and resolutions as possible. Several different configurations were tested and are shown here. Tested configurations include the browsers Firefox (version 34.0) and Chrome (version 39.0). As well as widescreen resolutions (1920px width) and half widescreen (960px width) for PCs. For mobile devices only Android was tested. The device used had a display width of 768px. The minimum window width that allows working with this interface in a convenient way was set to be 460px. This could pose a problem with some mobile devices, although it seems rather improbable that someone would use this interface with a mobile phone current statistics[56] suggest that virtually all devices could display this interface if mobile devices' landscape mode is included.

Synthesis:

User @ e-mail

Literature:

Title Author Journal DOI DOI

Add Reaction

Done

SynthDB Input Form

Synthesis: -

User

@ e-mail

Literature:

Title

Author

Journal

DOI DOI

Add Reaction

Done

☺

□

✗

✗^R

⚡

↶

↷

✕

⏏

◀

—

=

≡

~

△

□

⬡

⬢

⬣

⬤

FG

C

N

O

S

F

Cl

Br

I

P

X

Close

Done

© TBI, University of Vienna

◀

○

□

Figure 21: Interface on a Google Nexus 4 with a display width of 768px.
This image is a stitched version of 3 separate images.

66

9.2 USAGE

This section describes the process of entering a synthesis plan to the synthesis plan creation server in detail. For this example a synthesis of Triclosan (5-chloro-2-(2,4-dichlorophenoxy)phenol) is shown. This specific synthesis was published in the european patent number EP0887333[57] and a processed version was provided by Andraos[30].

Once started the site, the user is greeted by a simple form (see figure 22 to enter the basic information about the synthesis plan they are going to enter. Name and email of the user are critical to request further information if the need arises. Adding the literature information to the plan allows to have a certain quality level to strive for when thinking about adding a plan.

Making a publication of the synthesis a requirement assures a certain quality level of the data generated. This allows other users to treat the data in the database as 'correct'.

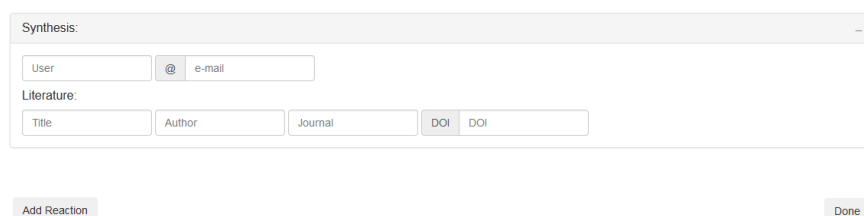


Figure 22: The first step when entering a synthesis plan. User details as well as literature sources have to be entered.

Once all the data has been entered the user may start adding data by clicking on the Add Reaction button on the bottom left side.

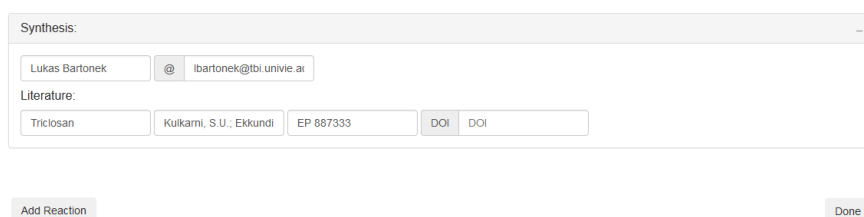


Figure 23: Once the users finished entering all the details a reaction may be added by clicking the 'Add Reaction' button.

This leads to an empty reaction object being created as seen in figure 24. The reaction consists of three different parts. At first the header which contains all the information besides the reactands, like the reaction yield, solvents used and catalysts needed. The header is here shown in red to signify that the reaction has not been balanced yet. The reason for this is that the reaction gets initialized in the unbalanced state as explained in subsection 9.3.2. Below the header

the reaction view is split into two parts - the educt and the product side.

Figure 24: Once a reaction is added a new table pops up below the header. The red boarder surrounding the title indicates that the reaction hasn't been tested for balancing yet.

After entering the essential header information the user may add molecules to the reaction by clicking the Add Input and Add Output buttons. This will create an empty molecule object in the corresponding reaction side and the user may start to draw (or use any other method listed in 9.3.1) the desired molecule in the editor below. Once the molecule has been specified a graphic representing the molecule is shown in the corresponding side of the reaction.

After defining the molecule the number of molecules may be selected by using the textbox in the top left of the molecule object or by clicking the up or down buttons next to the textbox. By clicking the checkbox below the molecule graphic a synthetic skeleton may be selected. This indicates which of the molecules added should be treated as the carbon backbone the synthesis is building up onto. This is essential for connecting the singular reactions together. Two reactions are treated as connected if the output skeleton of one reaction is equal to the input skeleton of another reaction.

In figure 25 a simple chlorination reaction is shown. The header information has been filled out, including a name for the reaction, catalysts needed and the reaction yield. Molecules have been added on each side, stoichiometric factors were chosen accordingly and a skeleton was defined on each side. To continue to the next reaction the user click on the Add Reaction button.

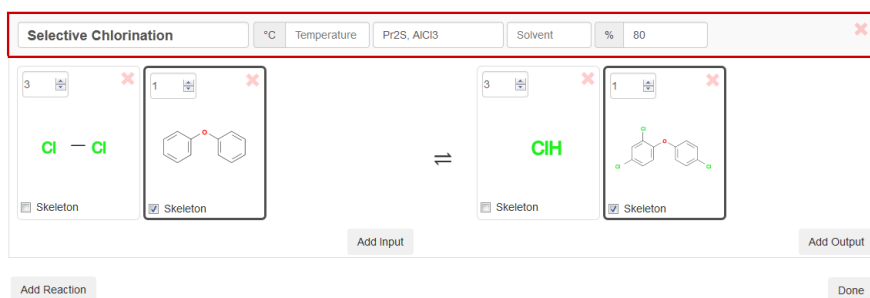


Figure 25: After the reaction information has been filled out and all the components have been added to the reaction another reaction may be added by clicking on the 'Add Reaction' button.

In figure 26 a second reaction has already been added. Note that the header of the first reaction has lost its red outline. This shows that the reaction has been tested for balancing and got approved by the system.

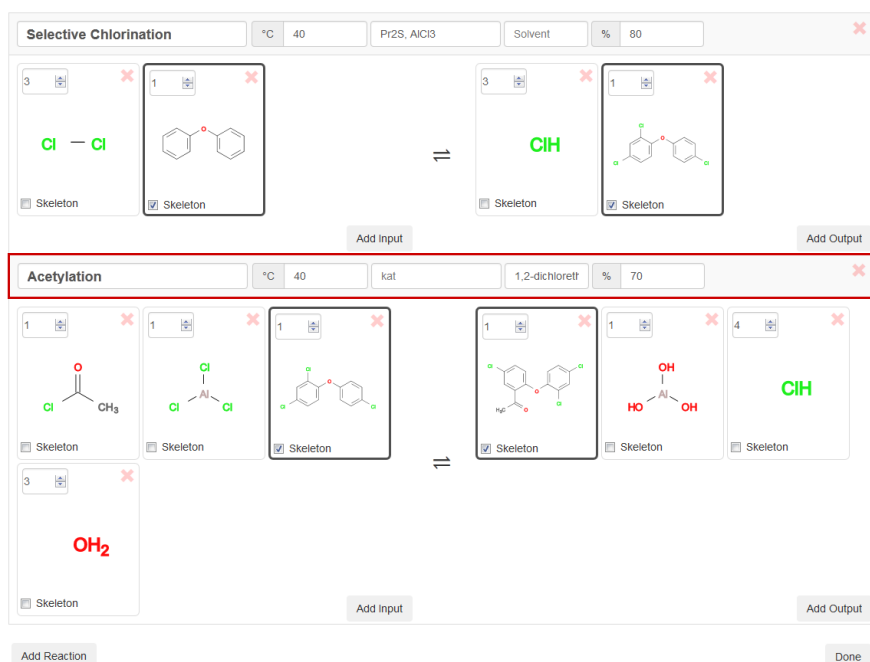


Figure 26: Whenever a reaction is added the previous reaction is checked for balancing. Only after this check is passed a new reaction is added. The removed red border indicates that the reaction is properly balanced.

The user may repeat these steps until the synthesis is entered completely. In figure 27 the whole synthesis, consisting of four reaction has been entered. All reactions have been tested for balancing. The user may now click on the Done button in the lower right corner.



Figure 27: Once all the steps have been added to the synthesis plan the plan may be submitted by clicking the 'Done' button.

Once the button is hit the server will test if each reaction had a skeleton assigned on each side. If the server approves the plan the user will be notified as seen in figure 28. The plan now gets stored on the server and can't be accessed by the user until it has been uploaded to the database. This step has to be done manually at this point to have a better control over the platform. Direct uploads to the database, as well as an optional download of the plan for the user are easy to add.

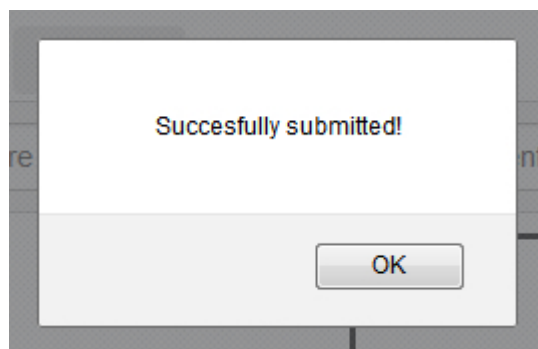


Figure 28: Successfully posted the plan

9.3 FEATURES

9.3.1 Editor Features

For an input tool ease of use is one of the most important aspects a programmer has to keep in mind. Synthesis Plans are rather long instructions consisting of a multitude of singular steps (see figure 27 and A). During testing drawing molecules was found to be the most time consuming process. For this reason an editor was chosen that had several API features implemented, that could help a user with his task.

The JME Editor [15] allows for web developers to set up a query to the cactus service of the National Cancer Institute at the National Institute of Health [55]. This feature was implemented in the input site and is described introduced here.

Any molecule could simply be added to the editor entering it's SMILES code. This could be done by right clicking the editor canvas and choosing the option Paste any chemical text repr.

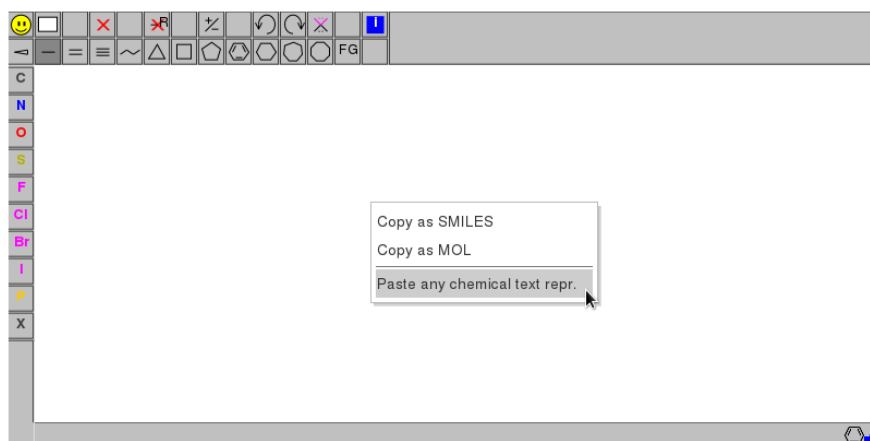


Figure 29: After right clicking the canvas clicking on 'Paste any chemical text repr.' allows to enter molecules by name.

In the following pop-up any SMILES code may be entered. To the knowledge and the testing of the author there are no restrictions concerning neither the features of the SMILES string used nor the length of the string entered.

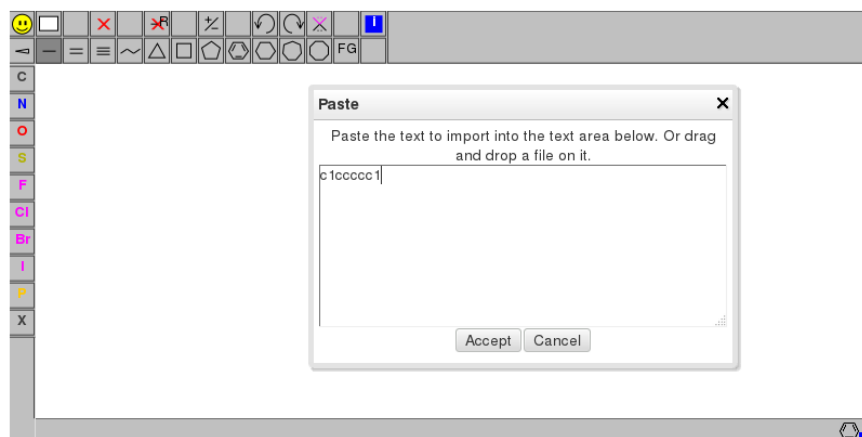


Figure 30: One of the accepted input formats is a SMILES string encoding the target molecule.

The string gets evaluated by the editor once the user hits Accept. Any correct SMILES string will be turned into an editable object inside the editor and could either be directly used as a molecule object in the synthesis plan or added onto.

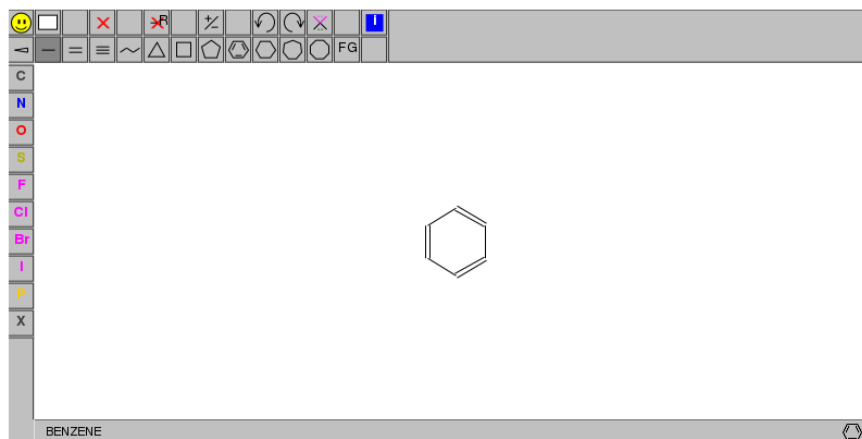


Figure 31: The SMILES string entered is resolved into a two-dimensional representation of the molecule.

SMILES strings for most common molecules are readily available. Not only does the English version of the Wikipedia list SMILES codes on most molecule entries but also the ChemSpider service of the Royal Society of Chemistry [4] lists SMILES codes, only to list a few. But testing has shown that most chemists are not or only slightly aware of SMILES encoding and prefer to draw molecules by hand.

To facilitate these users, instead of a SMILES code a name of a chemical compound may be entered into the textbox. The molecule names are restricted by the database the cactus service provides but testing showed that the range provided is rather vast and only a few unusual molecules had to be drawn by hand.

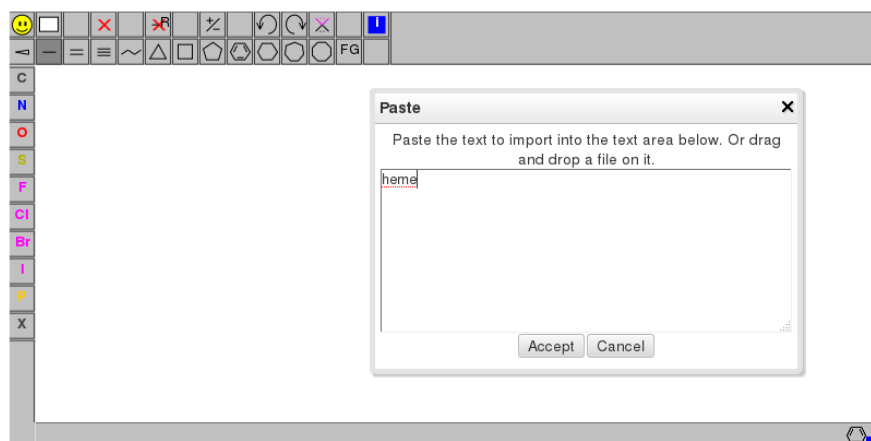


Figure 32: Not only ecodings for molecules may be added - because of the connection to the NCI Cactus database trivial names as well as canonical names for molecules may be addd and resolved.

As shown in figure 33 even very large molecules like the Heme molecule may be entered using this method, reducing the time needed to enter reactions drastically. This feature turned out to be a very popular way of entering information to the form because of the ease of usage.

It should be noted though that with the reliance on a third party web-service (cactus.nci.nih.gov) this feature is dependent on their servers to be online and available. In case this service is discontinued in the future these queries could be ported to other services like the ChemSpider API, at this point the cactus service proved to be reliable though.

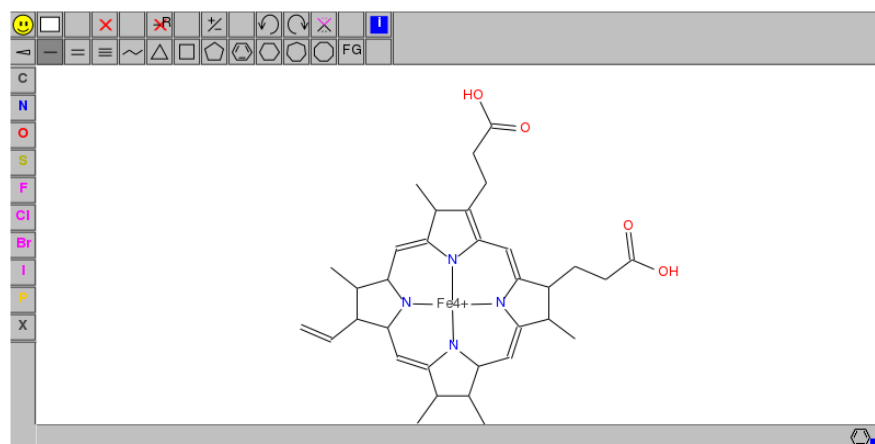


Figure 33: Entering massive molecules like the Heme compound is a task done in seconds using the database connection.

Because of the connection with the cactus database, the editor is also able to display the name of the molecule drawn in the bottom left corner - see figure 34.

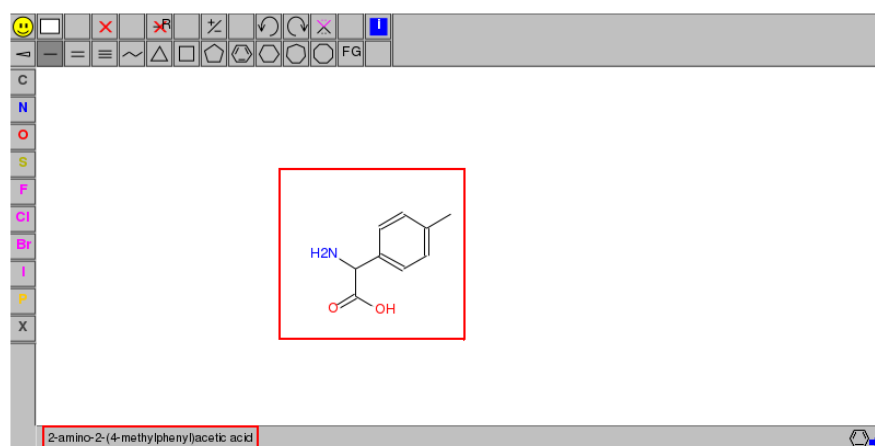


Figure 34: The name displayed in the bottom left corner is also due to the connection to the cactus service provided by the NCI

9.3.2 Balance Testing

During testing the tool the most common problem turned out to be unbalanced reactions. The SynthDB database needs reactions to be balanced, as well as any other applications which may make use of the data. For a user it is easy to miss some "minor" molecules a chemist probably wouldn't explicitly mention in a reaction. Most commonly this happened with condensation reactions where water was omitted.

Additionally to these purposely omitted molecules balance checking is also a good tool to make sure accidental errors are found by the user during entering the data and corrected immediately.

From a technical point of view the balance attribute is a boolean variable in the reaction class. Reactions are always initialized in the unbalanced state, although this is strictly speaking wrong, since the reaction is empty on both sides - therefore balanced - this shows the user, that the reaction has not been tested yet.

Balance Testing Interface

Once molecules have been added to the reaction the user may click on the Add Reaction button. This starts a routine that tries to add another reaction to the plan. Part of this routine is a balance check for the last entered reaction. Only if this check is positive another reaction may be added. This assures that a user only continues with the next reaction if the last one has been properly balanced. Once a reaction has been balanced and tested positively the red border around the header disappears to indicate this fact to the user (see figure 35 to figure 36).

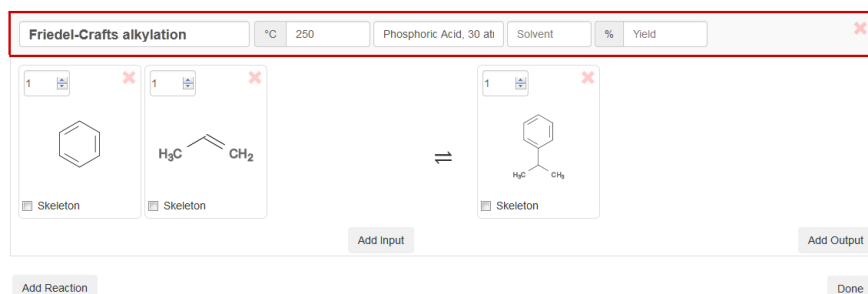


Figure 35: Red border indicating that the reactions has not been positively tested to be balanced

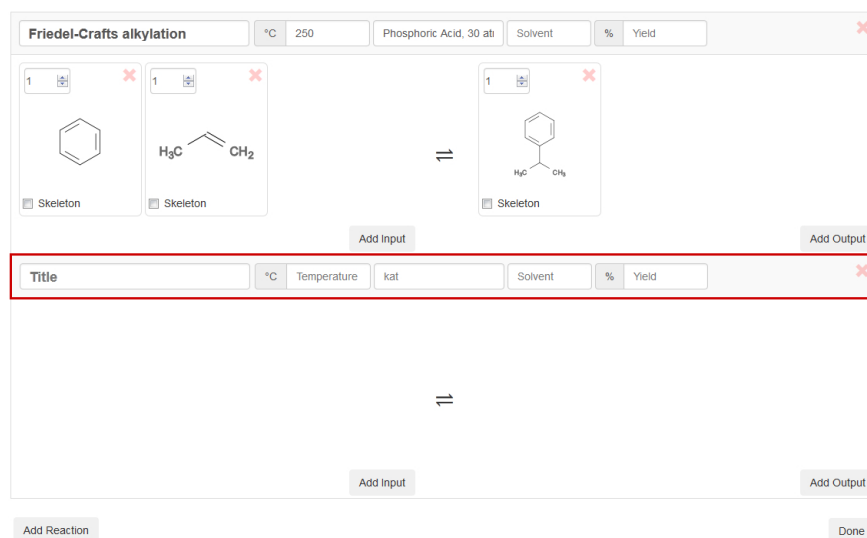


Figure 36: The red border gone indicates that this reaction is now balanced

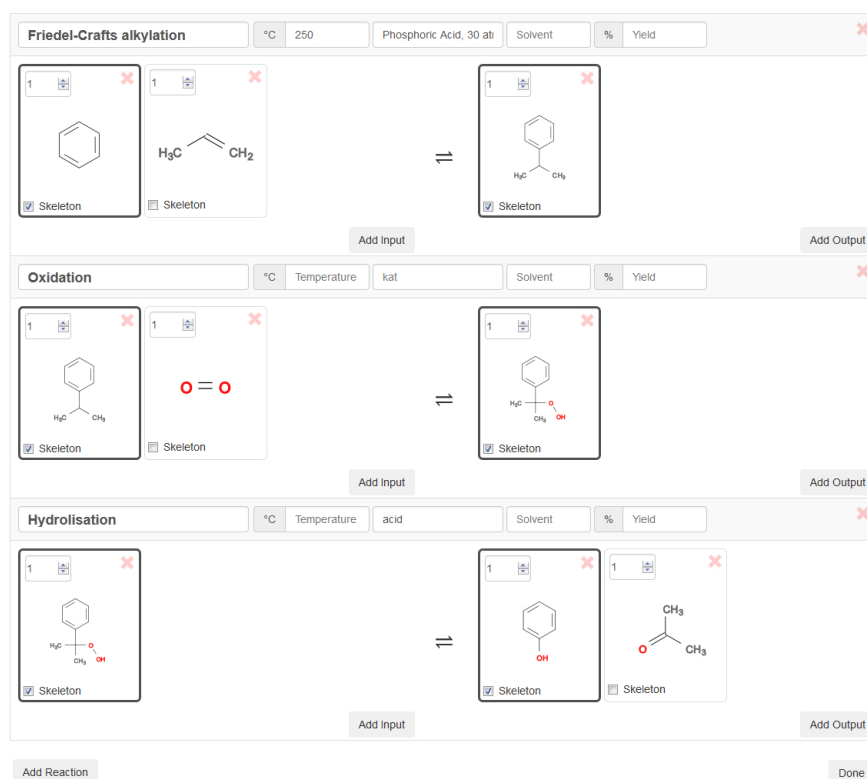


Figure 37: The complete synthesis plan has been entered and each reaction has been tested to be balanced - the plan could now be submitted.

The implementation of the balance test expects a JSON object as an input for the balance testing. In snippet 20 the loop that decodes the JSON object can be seen. The check tests whether a molecule is on the input or output side of a reaction, determines the number of this molecule that is used or produced and identifies the molecule via its SMILES code. Molecules on the input and output side as well as the amount of these are stored in separate arrays that are analyzed in a later step.

```
1 foreach my $molecule ( @molecules ) {
2   if ( $molecule->{'side'} == 0 ) {
3     push( @molCountLeft, "$molecule->{'amount'}");
4     push( @molLeft, "$molecule->{'smiles'}");
5   } elsif ( $molecule->{'side'} == 1 ) {
6     push( @molCountRight, "$molecule->{'amount'}");
7     push( @molRight, "$molecule->{'smiles'}");
8   }
9 }
```

Snippet 20: "Balance check implemented in Perl 5 using a JSON object as an input"

To get the molecular formula needed out of the SMILES encoded molecules, the SMILES code could easily be parsed. In the case of this piece of software an even simple route was taken by using openbabel's molecular report. Since the complete interface already relies heavily on the openbabel software, using it again at this point added no additional external software and made this balance check a bit more easy to implement.

In snippet 21 the process of retrieving the formula for a molecule can be seen. The SMILES code is sent to openbabel and only the string following the pattern defined in line 4 is saved in the \$formula variable.

```
1 foreach my $molecule ( @molLeft ) {
2   $formula = qx( obabel -:"$molecule" -omolreport );
3   $formula =~ /FORMULA: (.*)\n/;
4   push ( @sumLeft, $1 );
5 }
```

Snippet 21: "Balance check implemented in Perl 5 using a JSON object as an input"

The next step in testing the balance of a reaction is shown in snippet 22. It starts with summing up all the atoms on both sides as well as

the charges, since a reaction can be assumed to be balanced if both of these attributes are equal on each side.

The formula retrieved from the molecular report in the last step is matched using a regular expression to retrieve at first each atom type (line 7), then the amount of this atom type (line 9). If only one atom of a certain type is present in the molecular formula an empty string would be returned, therefore this case is tested for and the amount is set to 1.

The sum of each atom type multiplied with the stoichiometric coefficient of the molecule in the reaction and added to the hash that keeps count of the amount of atoms in the reaction. The hash is built up in a way that the key is the element symbol taken from the periodic table of elements and the value is the total number of this atom type present in the respective side of the reaction.

Dealing with the charge of the molecules is even more easy as the molecular report adds charges to the formula by adding the respective amount of charge signs to the formula. Therefore a Ca^{2+} would be returned as Ca^{++} . This allows for simply counting the '+' and '-' signs and multiplying the amount found with the stoichiometric coefficients.

```
1 $count = 0;
2
3 foreach my $molecule (@sumLeft) {
4     while ($molecule =~ m/([A-Z][a-z]{0,2}\d*)/g) {
5         $temp = $1;
6         $temp =~ m/([A-Z][a-z]{0,2})(\d*)/;
7         $amount = $2;
8         if ($amount eq "") {
9             $amount = 1;
10        }
11        $massLeft{$1} += ($amount * $molCountLeft[
12            $count]);
13    }
14    while ($molecule =~ m/([+-])/g) {
15        if ($1 eq '+') {
16            $chargeLeft += $molCountLeft[$count];
17        } elsif ($1 eq '-') {
18            $chargeLeft -= $molCountLeft[$count];
19        }
20    }
21    $count++;
```

Snippet 22: "Balance check implemented in Perl 5 using a JSON object as an input"

In the last step the hashes for the left and right side of the reaction are finally compared. This is shown in snippet 23.

In the first step the hash for the left and right side is compared if both sides contain the same keys. If both hashes have the same keys saved in them the testing continues. Otherwise the routine immediately returns false.

In a next step the values of the charges are compared. If these are equal all the values in the hashes (therefore the masses) are compared with each other. Once all these tests are passed without an error the value true is returned by the subroutine and the reaction is accepted as balanced.

```
1 if ( %massLeft ~~ %massRight ) {
2   # hash keys are the same
3   if ( $chargeLeft == $chargeRight ) {
4     # charges are equal
5     for ( keys %massLeft ) {
6       unless ( exists $massRight{$_} ) {
7         # key not found in the second hash
8         next;
9       }
10      if ( $massLeft{$_} eq $massRight{$_} ) {
11        # equal masses
12      }
13      else {
14        return false;
15      }
16    }
17  }
18  else {
19    return false;
20  }
21 } else {
22   return false;
23 }
24 return true;
```

Snippet 23: "Balance check implemented in Perl 5 using a JSON object as an input"

9.4 SYNTHESIS PLAN ANALYSIS

This section presents an application for the metrics described in the chapters 3 and 4. For this five different approaches for the synthesis of phenol were chosen and compared using the total walk counts as a

complexity measure and the atom economy as a measure for how the plan performs in regards of environmental impact.

9.4.1 *Phenol synthesis approaches*

Each of the five plans compared in this analysis was taken from the book Industrial Chemicals[58]. The five different processes analyzed are shown in table 11. Detailed synthesis plans for the five processes can be found in the Appendix B:

NAME	PROCESS
FaithG1	Benzenesulfonate process
FaithG2	Chlorination-caustic process
FaithG3	Raschig process
FaithG4	Cumene peroxidation process
FaithG5	Toluene oxidation process

Table 11: All the processes used in this analysis listed with the short names used in the figures associated with them.

All of the processes analysed in detail are linear synthetic approaches. Four of the five approaches start with benzene as educt, only the Toluene oxidation process (FaithG5) uses toluene as starting material. The cumene peroxidation process (FaithG4) is the only process listed here that produces not only phenol but also acetone as a product.

In a first step the total walk counts were calculated for each of the skeleton molecules in each step. As a skeleton molecule the molecule on which was built upon was considered. As mentioned in chapter 3 weights for different atom types in twc analysis are not published. Therefore a simple approach with carbon equal to a weight of 0 and oxygen having a weight of 1 was chosen. For non-connected molecules like salts the matrix was still sized according to the amount of atoms present but no ionic bonds were defined in the connection table. The result of this is shown in figure 38.

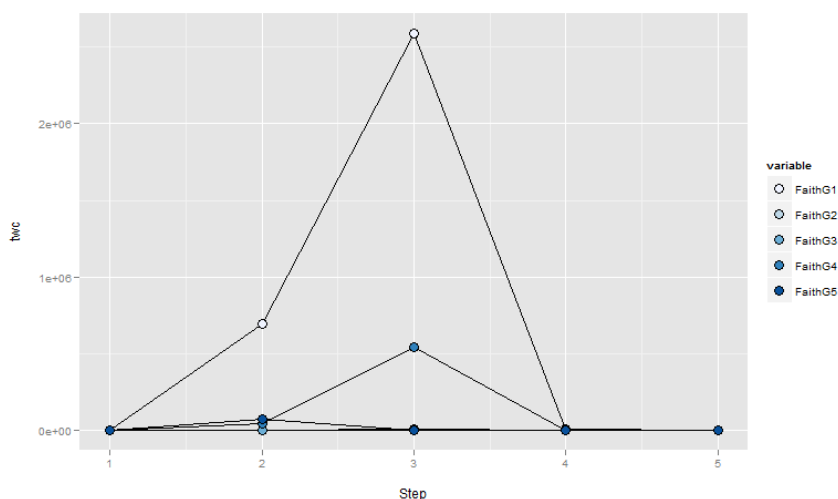


Figure 38: Complexity development for each of the five phenol synthesis analyzed. The obvious domination by FaithG1 and FaithG4 is due to the addition of large functional groups in the synthesis.

In a theoretical efficient synthesis the complexity would increase linearly with each step in the synthesis until the target complexity is reached. In this example it is very apparent that two plans (FaithG1 and FaithG4) dominate this figure. Adding large functional groups like sulfonic acids (FaithG1) or acetone as a secondary synthetic product (FaithG4) lead to very large values for twc. For the second figure these two dominant plans were excluded to analyze the - in terms of complexity development - more efficient plans.

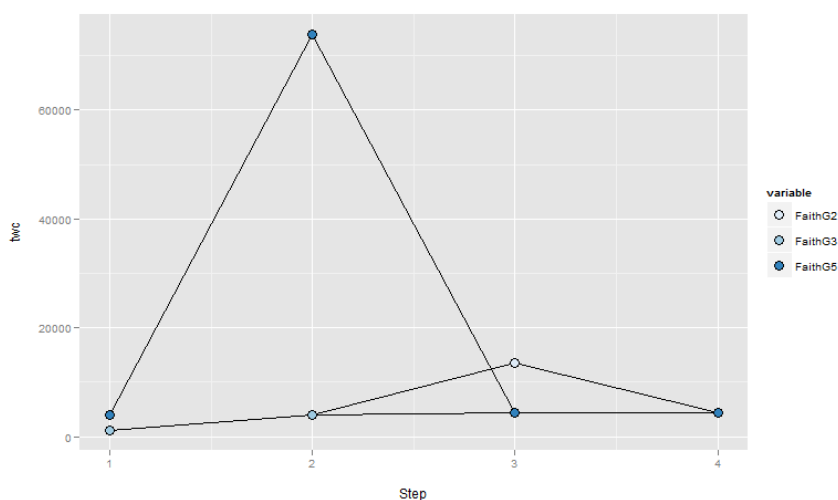


Figure 39: Complexity development for the three plans with the lowest values for twc. FaithG3 is the only plan that never shows a twc higher than the target product.

Here the Raschig process (FaithG3) fulfills the criteria of the theoretical optimal synthesis. By chlorinating the benzene ring and hydrolysis of the chlorine in a second step the complexity is always below the complexity of the target molecule.

Obviously complexity is not the only interesting property to analyze therefore another metric was utilized. For this second analysis the atom economy (AE) (see 4.1.3 for details) of each synthesis was investigated. The results are compiled in figure 40:

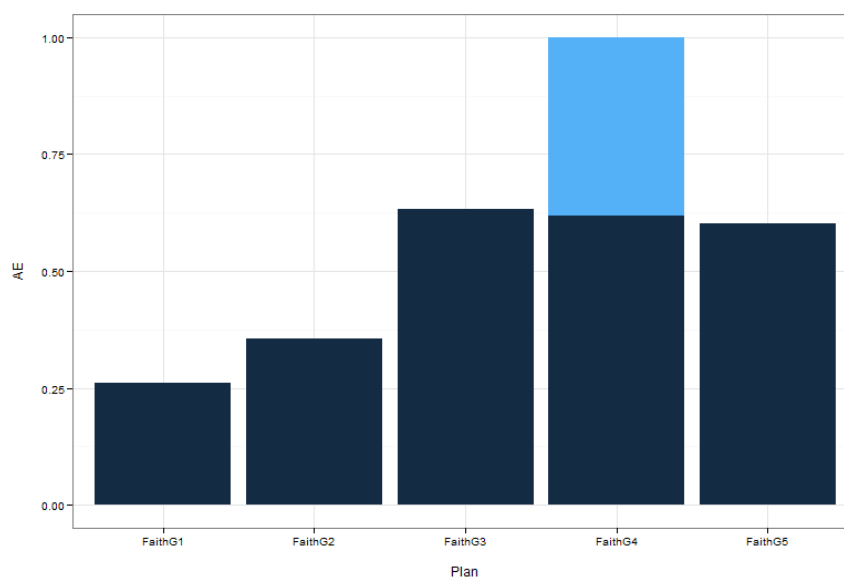


Figure 40: Atom economy (AE) for each of the synthesis. Dark blue indicates the AE for the creation of phenol, light blue shows the AE including acetone as a secondary product in addition to phenol.

The AE scales from 0 for no atoms from the starting material ending up in the final product up to 1 which means that every atom was utilized. From figure 40 it becomes apparent that the plan FaithG1 which already turned out to be very inefficient also performs poorly in terms of AE. The plan FaithG3 which would be considered the best in terms of complexity development also performs well when AE is considered. Special interest should be drawn to FaithG4 which performed rather poor in the first analysis but is comparable to FaithG3 in this second analysis. The reason for this is that this plan also produces acetone as a secondary product which leads to a higher complexity than other approaches but when considering acetone as a wanted product along with phenol this plans leads to a phenomenal AE of 1.00.

As this short analysis shows the best plan to use is dependent on the situation. If both acetone and phenol are of interest FaithG4 should be

used - if only phenol is wanted FaithG3 provides the most efficient way.

Such analysis would not be possible without utilizing modern computational methods as well as a database filled with synthesis plans to compare. As seen here several published plans perform very differently - differences that are easy to define with a database based system.

Part V

Future

SUMMARY

10.1 JUSTIFICATION

Chemical synthesis is a discipline that needs many years of training and experience. Single reactions are rather easy for an experienced chemist to figure out and plan, but when it comes to creating synthesis plans for large molecules - especially in the field of total synthesis - it can take years which the multitude of PhD theses on this subject proves easily.

At this point there is no computer based tool available that allows to support chemists with their work in this field. For single reactions there are many ways to find information available to the organic chemist - SciFinder by CAS[45] and Reaxys by Elsevier[44] just to name a few. The Reaxys page even features a synthesis planer that allows to create plans by connecting known reactions but the fact that this service doesn't serve as a library for already performed syntheses makes this tool somewhat unreliable. There are no services out there that allow a chemist to compare different approaches to a synthesis and while the tools listed above help a chemist to create plans they don't support calculating descriptors for these synthesis plans.

In industry - but nowadays also in the chemical laboratories - there is a great interest in making different approaches to create certain compounds comparable in regards to different factors. To do this a format that serves as a machine readable transmissions protocol is needed. On the other hand chemists are usually not experienced with more complicated computer based tools and command line applications. To make this new approach interesting to chemists in every position and specialization the tools need to be easily understandable and to perform checks on the data provided by the users.

10.2 PRODUCT

In this work a novel data format for saving synthesis plans is presented. This format allows saving every information needed to completely describe a synthesis in a chemically correct way. In addition also information about the creator of the synthesis plan, the original author of the synthesis and the journal it was published in, is saved.

To create these files in a simple and comfortable way an responsive online tool was created that allows to enter all the information needed to describe the plan via a simple form. This tool was thoroughly tested and additional features implemented as needed.

10.3 USABILITY

To ensure the tool created in this work is actually used by chemists the focus has to be put on usability. In cheminformatics as well as most aspects of scientific computing command line tools are seen as sufficient, if not even superior to programmes that come with an graphical interface - since these not only limit the interaction available to the user by the way the interface is designed but also add an overhead to the whole software that needs to be computed in addition to the values of interest.

For people usually working in labs this has to be done differently. The experience has to be as comfortable and intuitive as possible. For this a web based approach has been chosen that includes modern html5 features as well as a server backend that provides internal computations, but also graphical representations for the user and checks performed on the input.

The interface consists of a form for the header information like the username, journal the plan was published, the original author and so on. Below this buttons dynamically add new reactions and new compounds to the plan. Each reaction gets tested for balancing and indicates negative balance tests by a colored border to the user. Adding new molecules is done using a molecular editor which turned out to be easily understandable by users. In addition to the possibility of drawing molecules a database could be queried to retrieve SMILES encoded representations (used only internally) of molecules by stating trivial names of chemical compounds. Each molecule added gets rendered in the user interface, so a user can always see what has been added already. Addition of new content to a plan doesn't have to be performed in a linear way but could be added at any point in the workflow - although the balance check would most likely indicate problems with the plan to the user.

Additional attributes like solvents used, yield of a reaction or catalyst present can be added via simple form fields present above each reaction. The amount of molecules present in a reaction can be changed using up and down arrows.

FUTURE

11.1 CONSIDERATIONS ABOUT THE FUTURE OF THE PROJECT

Probably the most important aspect of chemical synthesis plans is their function as a recipe for chemists to produce chemical compounds. The biggest problem with this is that until now synthesis plans were not comparable to each other. Searching for the best fitting plan was impossible in an automated fashion.

With a system in place that allows for synthesis plans to be machine readable, this is now possible but also raises new questions especially in terms of framework design. These possibilities but also the new questions shall be discussed in this final part of this work. The topics presented here vary in terms of actual time spent working on them from only thoughts about synthesis plans to being in a working prototype state. The current state will be mentioned in each section.

11.1.1 *Database interaction*

The most important goal for the input server was to be easily usable for a chemist with little to no computer experience. This requires a clever interface design which - hopefully - was achieved. This is not so important if only the data interpretation of these plans is considered. Analyzing the data obtained through this input format should be possible for computer scientists via an automated way. Whether this happens via the proposed document format or a more established format is open for discussion and needs to be evaluated.

One idea that could be easily implemented is the possibility of writing an API for the database that allows to query whole entries in the database and retrieve them in a yet to be specified format. In the current state the load on the server for this needs to be considered but with modern server architecture and a somewhat modern computer this should be easily doable.

For the data format the JSON format shall be proposed here. Since the file format described in this work is actually just a list of key-value pairs, converting this to a JSON format is easy and allows scientists to use the already established ways of parsing JSON objects.

11.1.2 *Total walk counts*

The total walk counts (TWC) presented in chapter 3 were already implemented in a simple script that could be applied to a synthesis

plan. The script simply computes the TWC for all the compounds in the synthesis. This should be seen as an example of what could be done with the information found in synthesis plans.

Calculating the TWC for a whole synthesis plan turned out to be rather easy since by using the machine readable format described in this work the work boiled down to writing a program that could compute the TWC for one molecule and a simple file parser that could send the molecules of interest to this program.

11.1.3 Further calculations

As already stated before, the calculation of the TWC should only be seen as an example of what could be done with this kind of system. There are multiple ways to evaluate synthesis plans - some of them were already presented in chapter 4 - that could be calculated.

This would be possible on the basis of everyone who is interested in a certain feature developing a script for exactly this or by adding modules to the database that would do exactly that. From a users' point of view the integration into the database would be the most favourable option.

Since the author of this work did not develop the database himself, this would either call for the authors of the database to continue developing the database or investing vast amounts of time to rework parts of the system already in place. In an optimal world this would allow for developers all around the world to write analysis scripts that could be incorporated into the database in an open source fashion and be available to every user.

In current modern synthesis there is no tool available to the scientific community that could calculate descriptors of synthesis plans and help not only scientists but also managers to decide which way would be the best in certain situations.

11.2 APPLICATIONS

For this work I am happy to add a section that focuses on stuff that has already been done with the help of the system described in this work. Carsten Grønbjerg Lützen and Daniel Fentz Johansen who developed the database described in chapter 7 investigated for their master thesis[48] synthesis plans in a more mathematical approach.

In their work they compared real plans like the one listed in the appendix of this work against plans they created using only computational methods just by breaking or creating bonds without taking real reactions into account. This is a nice example how the system discussed in this work opens up new ways for different scientific areas to interact and improve onto each other.

EXAMPLE PLAN

A complete example how a real synthesis plan would be encoded in the file format described in this work. In this case the plan covers one of the many ways to synthesize (+)-Lysergic Acid in form of a total synthesis and was published by Szanty et al. in 2004. [59]

This plan features most (if not all) of the features this system is capable of managing. For purposes of software development or testing of parsers the plan is available in a digital form from the author.

```

1 # Last changed timestamp:
2 # Last changed author:
3 # Lit: Moldvai, I.; Temesvari-Major, E.; Incze, M.;
      Szentirmay, E.; Szanty, C.
4 #      Enantioefficient Synthesis of r-Ergocryptine: First
      Direct Synthesis of (+)-Lysergic Acid
5 #      J. Org. Chem. 2004
6 # DOI:
7 # Aut: Lukas Bartonek <a0906858@unet.univie.ac.at>
8
9 skeleton: InChI=1S/C16H16N2O2/c1-18-8-10(16(19)20)
      5-12-11-3-2-4-13-15(11)9(7-17-13)6-14(12)18/h2
      -5,7,10,14,17H,6,8H2,1H3,(H,19,20)/t10-,14-/m1/s1
10
11
12 compounds:
13 comp1 InChI=1S/C11H11N2O2/c13-11(14)
      6-5-8-7-12-10-4-2-1-3-9(8)10/h1-4,7,12H,5-6H2,(H,13,14)
14 comp2 InChI=1S/C5H9ClO/c1-5(2,3)4(6)7/h1-3H3
15 comp3 InChI=1S/K.H2O/h;1H2/q+1;/p-1
16 comp4 InChI=1S/ClH/h1H
17 comp5 InChI=1S/C16H19N2O3/c1-16(2,3)15(20)
      17-10-11(8-9-14(18)19)12-6-4-5-7-13(12)17/h4-7,10H,8-9
      H2,1-3H3,(H,18,19)
18 comp6 InChI=1S/H2O/h1H2
19 comp7 InChI=1S/ClH.K/h1H;/q;+1/p-1
20 comp8 InChI=1S/C16H19N2O3/c1-16(2,3)15(20)
      17-10-11(8-9-14(18)19)12-6-4-5-7-13(12)17/h4-7,10H,8-9
      H2,1-3H3,(H,18,19)
21 comp9 InChI=1S/Cl2OS/c1-4(2)3
22 comp10 InChI=1S/C16H18ClN2O2/c1-16(2,3)15(20)
      18-10-11(8-9-14(17)19)12-6-4-5-7-13(12)18/h4-7,10H,8-9
      H2,1-3H3
23 comp11 InChI=1S/O2S/c1-3-2
24 comp12 InChI=1S/ClH/h1H
25 comp13 InChI=1S/C16H18ClN2O2/c1-16(2,3)15(20)
      18-10-11(8-9-14(17)19)12-6-4-5-7-13(12)18/h4-7,10H,8-9
      H2,1-3H3
26 comp14 InChI=1S/C2H2Cl2O/c3-1-2(4)5/h1H2
27 comp15 InChI=1S/Al.3ClH/h;3*1H/q+3;;;/p-3

```

28 comp16 InChI=1S/H2O/h1H2
 29 comp17 InChI=1S/C16H17N02/c1-16(2,3)15(19)
 17-9-10-7-8-13(18)11-5-4-6-12(17)14(10)11/h4-6,9H,7-8H2
 ,1-3H3
 30 comp18 InChI=1S/Al.3H2O/h;3*1H2/q+3;;;/p-3
 31 comp19 InChI=1S/ClH/h1H
 32 comp20 InChI=1S/C2H3Cl02/c3-1-2(4)5/h1H2,(H,4,5)
 33 comp21 InChI=1S/C16H17N02/c1-16(2,3)15(19)
 17-9-10-7-8-13(18)11-5-4-6-12(17)14(10)11/h4-6,9H,7-8H2
 ,1-3H3
 34 comp22 InChI=1S/C5H5N.Br3/c1-2-4-6-5-3-1;1-3-2/h1-5H;/q;-1/
 p+1
 35 comp23 InChI=1S/C16H16BrN02/c1-16(2,3)15(20)18-8-9-7-11(17)
 14(19)10-5-4-6-12(18)13(9)10/h4-6,8,11H,7H2,1-3H3
 36 comp24 InChI=1S/C5H5N.BrH/c1-2-4-6-5-3-1;/h1-5H;1H
 37 comp25 InChI=1S/BrH/h1H
 38 comp26 InChI=1S/C16H16BrN02/c1-16(2,3)15(20)18-8-9-7-11(17)
 14(19)10-5-4-6-12(18)13(9)10/h4-6,8,11H,7H2,1-3H3
 39 comp27 InChI=1S/C2H6O2/c3-1-2-4/h3-4H,1-2H2
 40 comp28 InChI=1S/C18H20BrN03/c1-17(2,3)16(21)
 20-10-11-9-14(19)18(22-7-8-23-18)12-5-4-6-13(20)15(11)
 12/h4-6,10,14H,7-9H2,1-3H3
 41 comp29 InChI=1S/H2O/h1H2
 42 comp30 InChI=1S/C18H20BrN03/c1-17(2,3)16(21)
 20-10-11-9-14(19)18(22-7-8-23-18)12-5-4-6-13(20)15(11)
 12/h4-6,10,14H,7-9H2,1-3H3
 43 comp31 InChI=1S/CH5N/c1-2/h2H2,1H3
 44 comp32 InChI=1S/C13H12BrN02/c14
 -11-6-8-7-15-10-3-1-2-9(12(8)10)13(11)16-4-5-17-13/h1
 -3,7,11,15H,4-6H2
 45 comp33 InChI=1S/C6H13N0/c1-6(2,3)5(8)7-4/h1-4H3,(H,7,8)
 46 comp34 InChI=1S/C13H12BrN02/c14
 -11-6-8-7-15-10-3-1-2-9(12(8)10)13(11)16-4-5-17-13/h1
 -3,7,11,15H,4-6H2
 47 comp35 InChI=1S/H2O/h1H2
 48 comp36 InChI=1S/C11H8BrN0/c12-8-4-6-5-13-9-3-1-2-7(10(6)9)
 11(8)14/h1-3,5,8,13H,4H2
 49 comp37 InChI=1S/C2H6O2/c3-1-2-4/h3-4H,1-2H2
 50 comp38 InChI=1S/C11H8BrN0/c12-8-4-6-5-13-9-3-1-2-7(10(6)9)
 11(8)14/h1-3,5,8,13H,4H2
 51 comp39 InChI=1S/C6H13N02/c1-6(5-7-2)8-3-4-9-6/h7H,3-5H2,1-2
 H3
 52 comp40 InChI=1S/C17H20N2O3.BrH/c1-17(21-6-7-22-17)10-19(2)
 14-8-11-9-18-13-5-3-4-12(15(11)13)16(14)20;/h3
 -5,9,14,18H,6-8,10H2,1-2H3;1H
 53 comp41 InChI=1S/C17H20N2O3.BrH/c1-17(21-6-7-22-17)10-19(2)
 14-8-11-9-18-13-5-3-4-12(15(11)13)16(14)20;/h3
 -5,9,14,18H,6-8,10H2,1-2H3;1H
 54 comp42 InChI=1S/ClH/h1H
 55 comp43 InChI=1S/CH2O3.2Na/c2-1(3)4;;;/h(H2,2,3,4);;;/q;2*+1/p
 -2
 56 comp44 InChI=1S/C17H20N2O3/c1-17(21-6-7-22-17)10-19(2)
 14-8-11-9-18-13-5-3-4-12(15(11)13)16(14)20/h3-5,9,14,18
 H,6-8,10H2,1-2H3
 57 comp45 InChI=1S/BrH.Na/h1H;/q;+1/p-1
 58 comp46 InChI=1S/ClH.Na/h1H;/q;+1/p-1
 59 comp47 InChI=1S/H2O/h1H2

60 comp48 InChI=1S/C02/c2-1-3
 61 comp49 InChI=1S/C17H20N2O3/c1-17(21-6-7-22-17)10-19(2)
 14-8-11-9-18-13-5-3-4-12(15(11)13)16(14)20/h3-5,9,14,18
 H,6-8,10H2,1-2H3
 62 comp50 InChI=1S/H2O/h1H2
 63 comp51 InChI=1S/ClH/h1H
 64 comp52 InChI=1S/Na.H2O/h;1H2/q+1;/p-1
 65 comp53 InChI=1S/C15H16N2O2/c1-9(18)8-17(2)
 13-6-10-7-16-12-5-3-4-11(14(10)12)15(13)19/h3-5,7,13,16
 H,6,8H2,1-2H3
 66 comp54 InChI=1S/C2H6O2/c3-1-2-4/h3-4H,1-2H2
 67 comp55 InChI=1S/ClH.Na/h1H;/q;+1/p-1
 68 comp56 InChI=1S/H2O/h1H2
 69 comp57 InChI=1S/C15H16N2O2/c1-9(18)8-17(2)
 13-6-10-7-16-12-5-3-4-11(14(10)12)15(13)19/h3-5,7,13,16
 H,6,8H2,1-2H3
 70 comp58 InChI=1S/BrH.Li/h1H;/q;+1/p-1
 71 comp59 InChI=1S/C6H15N/c1-4-7(5-2)6-3/h4-6H2,1-3H3
 72 comp60 InChI=1S/C15H14N2O/c1-17-8-10(18)
 6-12-11-3-2-4-13-15(11)9(7-16-13)5-14(12)17/h2
 -4,6-7,14,16H,5,8H2,1H3
 73 comp61 InChI=1S/Li.H2O/h;1H2/q+1;/p-1
 74 comp62 InChI=1S/C6H15N.BrH/c1-4-7(5-2)6-3;/h4-6H2,1-3H3;1H
 75 comp63 InChI=1S/C15H14N2O/c1-17-8-10(18)
 6-12-11-3-2-4-13-15(11)9(7-16-13)5-14(12)17/h2
 -4,6-7,14,16H,5,8H2,1H3
 76 comp64 InChI=1S/C18H18O6/c19-17(20)
 15(23-11-13-7-3-1-4-8-13)16(18(21)22)
 24-12-14-9-5-2-6-10-14/h1-10,15-16H,11-12H2,(H,19,20)(H
 ,21,22)
 77 comp65 InChI=1S/C18H18O6.C15H14N2O/c19-17(20)
 15(23-11-13-7-3-1-4-8-13)16(18(21)22)
 24-12-14-9-5-2-6-10-14;1-17-8-10(18)
 6-12-11-3-2-4-13-15(11)9(7-16-13)5-14(12)17/h1-10,15-16
 H,11-12H2,(H,19,20)(H,21,22);2-4,6-7,14,16H,5,8H2,1H3/t
 ;14-/m.1/s1
 78 comp66 InChI=1S/C18H18O6.C15H14N2O/c19-17(20)
 15(23-11-13-7-3-1-4-8-13)16(18(21)22)
 24-12-14-9-5-2-6-10-14;1-17-8-10(18)
 6-12-11-3-2-4-13-15(11)9(7-16-13)5-14(12)17/h1-10,15-16
 H,11-12H2,(H,19,20)(H,21,22);2-4,6-7,14,16H,5,8H2,1H3/t
 ;14-/m.1/s1
 79 comp67 InChI=1S/Na.H2O/h;1H2/q+1;/p-1
 80 comp68 InChI=1S/C15H14N2O/c1-17-8-10(18)
 6-12-11-3-2-4-13-15(11)9(7-16-13)5-14(12)17/h2
 -4,6-7,14,16H,5,8H2,1H3/t14-/m1/s1
 81 comp69 InChI=1S/C18H18O6.Na/c19-17(20)
 15(23-11-13-7-3-1-4-8-13)16(18(21)22)
 24-12-14-9-5-2-6-10-14;/h1-10,15-16H,11-12H2,(H,19,20)(
 H,21,22);/q;+1/p-1
 82 comp70 InChI=1S/H2O/h1H2
 83 comp71 InChI=1S/C15H14N2O/c1-17-8-10(18)
 6-12-11-3-2-4-13-15(11)9(7-16-13)5-14(12)17/h2
 -4,6-7,14,16H,5,8H2,1H3/t14-/m1/s1
 84 comp72 InChI=1S/C9H9N02S/c1-8-3-5-9(6-4-8)13(11,12)7-10-2/
 h3-6H,7H2,1H3
 85 comp73 InChI=1S/C4H9O.K/c1-4(2,3)5;/h1-3H3;/q-1;+1

86 comp74 InChI=1S/H2O/h1H2
 87 comp75 InChI=1S/C24H23N3O3S/c1-15-6-8-18(9-7-15)31(29,30)
 24(26-14-28)17-10-20-19-4-3-5-21-23(19)16(12-25-21)
 11-22(20)27(2)13-17/h3-10,12,14,22,25H,11,13H2,1-2H3,(H
 ,26,28)/t22-/m1/s1
 88 comp76 InChI=1S/C4H10O/c1-4(2,3)5/h5H,1-3H3
 89 comp77 InChI=1S/K.H2O/h;1H2/q+1;/p-1
 90 comp78 InChI=1S/C24H23N3O3S/c1-15-6-8-18(9-7-15)31(29,30)
 24(26-14-28)17-10-20-19-4-3-5-21-23(19)16(12-25-21)
 11-22(20)27(2)13-17/h3-10,12,14,22,25H,11,13H2,1-2H3,(H
 ,26,28)/t22-/m1/s1
 91 comp79 InChI=1S/CH3O.Na/c1-2;/h1H3;/q-1;+1
 92 comp80 InChI=1S/C16H15N3/c1-19-9-10(7-17)
 5-13-12-3-2-4-14-16(12)11(8-18-14)6-15(13)19/h2
 -5,8,15,17-18H,6,9H2,1H3/t15-/m1/s1
 93 comp81 InChI=1S/C2H4O2/c1-4-2-3/h2H,1H3
 94 comp82 InChI=1S/C7H7O2S.Na/c1-6-2-4-7(5-3-6)10(8)9;/h2-5H,1
 H3;/q-1;+1
 95 comp83 InChI=1S/C16H15N3/c1-19-9-10(7-17)
 5-13-12-3-2-4-14-16(12)11(8-18-14)6-15(13)19/h2
 -5,8,15,17-18H,6,9H2,1H3/t15-/m1/s1
 96 comp84 InChI=1S/C16H15N3/c1-19-9-10(7-17)
 5-13-12-3-2-4-14-16(12)11(8-18-14)6-15(13)19/h2
 -5,8,10,15,18H,6,9H2,1H3/t10?,15-/m1/s1
 97 comp85 InChI=1S/C16H15N3/c1-19-9-10(7-17)
 5-13-12-3-2-4-14-16(12)11(8-18-14)6-15(13)19/h2
 -5,8,10,15,18H,6,9H2,1H3/t10?,15-/m1/s1
 98 comp86 InChI=1S/CH4O/c1-2/h2H,1H3
 99 comp87 InChI=1S/H2O/h1H2
 100 comp88 InChI=1S/ClH/h1H
 101 comp89 InChI=1S/C17H18N2O2/c1-19-9-11(17(20)21-2)
 6-13-12-4-3-5-14-16(12)10(8-18-14)7-15(13)19/h3
 -6,8,11,15,18H,7,9H2,1-2H3/t11?,15-/m1/s1
 102 comp90 InChI=1S/ClH.H3N/h1H;1H3
 103 comp91 InChI=1S/C17H18N2O2/c1-19-9-11(17(20)21-2)
 6-13-12-4-3-5-14-16(12)10(8-18-14)7-15(13)19/h3
 -6,8,11,15,18H,7,9H2,1-2H3/t11?,15-/m1/s1
 104 comp92 InChI=1S/Na.H2O/h;1H2/q+1;/p-1
 105 comp93 InChI=1S/ClH/h1H
 106 comp94 InChI=1S/C16H16N2O2/c1-18-8-10(16(19)20)
 5-12-11-3-2-4-13-15(11)9(7-17-13)6-14(12)18/h2
 -5,7,10,14,17H,6,8H2,1H3,(H,19,20)/t10-,14-/m1/s1
 107 comp95 InChI=1S/CH4O/c1-2/h2H,1H3
 108 comp96 InChI=1S/ClH.Na/h1H;/q;+1/p-1
 109 comp97 InChI=1S/C3H5BrO/c1-3(5)2-4/h2H2,1H3
 110 comp98 InChI=1S/C2H6O2/c3-1-2-4/h3-4H,1-2H2
 111 comp99 InChI=1S/C5H9BrO2/c1-5(4-6)7-2-3-8-5/h2-4H2,1H3
 112 comp100 InChI=1S/H2O/h1H2
 113 comp101 InChI=1S/C5H9BrO2/c1-5(4-6)7-2-3-8-5/h2-4H2,1H3
 114 comp102 InChI=1S/CH5N/c1-2/h2H2,1H3
 115 comp103 InChI=1S/C6H13N2O2/c1-6(5-7-2)8-3-4-9-6/h7H,3-5H2
 ,1-2H3
 116 comp104 InChI=1S/CH5N.BrH/c1-2;/h2H2,1H3;1H
 117
 118
 119 reactions:
 120 R01 "Protection"

```

121 skeletonOut InChI=1S/C16H19NO3/c1-16(2,3)15(20)
      17-10-11(8-9-14(18)19)12-6-4-5-7-13(12)17/h4-7,10H,8-9
      H2,1-3H3,(H,18,19)
122 |comp1| + comp2 + 2 * comp3 + comp4 -> |comp5| + 2 * comp6
      + 2 * comp7
123 @yield 78%
124
125 R02 "intermediary step"
126 skeletonOut InChI=1S/C16H18ClNO2/c1-16(2,3)15(20)
      18-10-11(8-9-14(17)19)12-6-4-5-7-13(12)18/h4-7,10H,8-9
      H2,1-3H3
127 |comp8| + comp9 -> |comp10| + comp11 + comp12
128 @yield R02-R03 55%
129
130 R03 "Friedel-Crafts Acylation"
131 skeletonOut InChI=1S/C16H17NO2/c1-16(2,3)15(19)
      17-9-10-7-8-13(18)11-5-4-6-12(17)14(10)11/h4-6,9H,7-8H2
      ,1-3H3
132 |comp13| + comp14 + comp15 + 4 * comp16 -> |comp17| +
      comp18 + 5 * comp19 + comp20
133 @yield R02-R03 55%
134
135 R04 "Addition of Bromine"
136 skeletonOut InChI=1S/C16H16BrNO2/c1-16(2,3)15(20)
      18-8-9-7-11(17)14(19)10-5-4-6-12(18)13(9)10/h4-6,8,11H
      ,7H2,1-3H3
137 |comp21| + comp22 -> |comp23| + comp24 + comp25
138 @yield 85% cat light
139
140 R05 "Acetal formation"
141 skeletonOut InChI=1S/C18H20BrNO3/c1-17(2,3)16(21)
      20-10-11-9-14(19)18(22-7-8-23-18)12-5-4-6-13(20)15(11)
      12/h4-6,10,14H,7-9H2,1-3H3
142 |comp26| + comp27 -> |comp28| + comp29
143 @yield 81% cat pTsOH
144
145 R06 "Deprotection"
146 skeletonOut InChI=1S/C13H12BrNO2/c14
      -11-6-8-7-15-10-3-1-2-9(12(8)10)13(11)16-4-5-17-13/h1
      -3,7,11,15H,4-6H2
147 |comp30| + comp31 -> |comp32| + comp33
148 @yield 88%
149
150 R07 "Deprotection"
151 skeletonOut InChI=1S/C11H8BrNO/c12
      -8-4-6-5-13-9-3-1-2-7(10(6)9)11(8)14/h1-3,5,8,13H,4H2
152 |comp34| + comp35 -> |comp36| + comp37
153 @yield 97% cat HCl
154
155 R08 "intermediary step"
156 skeletonOut InChI=1S/C17H20N2O3.BrH/c1-17(21-6-7-22-17)
      10-19(2)14-8-11-9-18-13-5-3-4-12(15(11)13)16(14)20;/h3
      -5,9,14,18H,6-8,10H2,1-2H3;1H
157 |comp38| + |comp39| -> |comp40|
158 @yield R08-R09 56%
159
160 R09 "Neutralization"

```

```

161 skeletonOut InChI=1S/C17H20N2O3/c1-17(21-6-7-22-17)10-19(2)
      14-8-11-9-18-13-5-3-4-12(15(11)13)16(14)20/h3-5,9,14,18
      H,6-8,10H2,1-2H3
162 |comp41| + comp42 + comp43 -> |comp44| + comp45 + comp46 +
      comp47 + comp48
163 @yield R08-R09 56%
164
165 R10 "Deprotection"
166 skeletonOut InChI=1S/C15H16N2O2/c1-9(18)8-17(2)
      13-6-10-7-16-12-5-3-4-11(14(10)12)15(13)19/h3-5,7,13,16
      H,6,8H2,1-2H3
167 |comp49| + comp50 + comp51 + comp52 -> |comp53| + comp54 +
      comp55 + comp56
168 @yield 100%
169
170 R11 "Ring formation"
171 skeletonOut InChI=1S/C15H14N2O/c1-17-8-10(18)
      6-12-11-3-2-4-13-15(11)9(7-16-13)5-14(12)17/h2
      -4,6-7,14,16H,5,8H2,1H3
172 |comp57| + comp58 + comp59 -> |comp60| + comp61 + comp62
173 @yield 60%
174
175 R12 "stereoselective reaction"
176 skeletonOut InChI=1S/C22H26O2.C15H14N2O/c1-17(2)
      21(23-15-19-11-7-5-8-12-19)22(18(3)4)
      24-16-20-13-9-6-10-14-20;1-17-8-10(18)
      6-12-11-3-2-4-13-15(11)9(7-16-13)5-14(12)17/h5-14,21-22
      H,1,3,15-16H2,2,4H3;2-4,6-7,14,16H,5,8H2,1H3/t;14-/m.1/
      s1
177 |comp63| + comp64 -> |comp65|
178 @yield 79%
179
180 R13 "Neutralization"
181 skeletonOut InChI=1S/C15H14N2O/c1-17-8-10(18)
      6-12-11-3-2-4-13-15(11)9(7-16-13)5-14(12)17/h2
      -4,6-7,14,16H,5,8H2,1H3/t14-/m1/s1
182 |comp66| + comp67 -> |comp68| + comp69 + comp70
183 @yield 38%
184
185 R14 "Refunctionalization"
186 skeletonOut InChI=1S/C24H23N3O3S/c1-15-6-8-18(9-7-15)
      31(29,30)24(26-14-28)17-10-20-19-4-3-5-21-23(19)
      16(12-25-21)11-22(20)27(2)13-17/h3-10,12,14,22,25H
      ,11,13H2,1-2H3,(H,26,28)/t22-/m1/s1
187 |comp71| + comp72 + comp73 + comp74 -> |comp75| + comp76 +
      comp77
188 @yield 77%
189
190 R15 "intermediary step"
191 skeletonOut InChI=1S/C16H15N3/c1-19-9-10(7-17)
      5-13-12-3-2-4-14-16(12)11(8-18-14)6-15(13)19/h2
      -5,8,15,17-18H,6,9H2,1H3/t15-/m1/s1
192 |comp78| + comp79 -> |comp80| + comp81 + comp82
193 @yield R15-R16 70%
194
195 R16 "Rearrangement"

```

```

196 skeletonOut InChI=1S/C16H15N3/c1-19-9-10(7-17)
      5-13-12-3-2-4-14-16(12)11(8-18-14)6-15(13)19/h2
      -5,8,10,15,18H,6,9H2,1H3/t10?,15-/m1/s1
197 |comp83| -> |comp84|
198 @yield R15-R16 70%
199
200 R17 "Refunctionalization"
201 skeletonOut InChI=1S/C17H18N2O2/c1-19-9-11(17(20)21-2)
      6-13-12-4-3-5-14-16(12)10(8-18-14)7-15(13)19/h3
      -6,8,11,15,18H,7,9H2,1-2H3/t11?,15-/m1/s1
202 |comp85| + comp86 + comp87 + comp88 -> |comp89| + comp90
203 @yield 72%
204
205 R18 "Esterhydrolysis"
206 skeletonOut InChI=1S/C16H16N2O2/c1-18-8-10(16(19)20)
      5-12-11-3-2-4-13-15(11)9(7-17-13)6-14(12)18/h2
      -5,7,10,14,17H,6,8H2,1H3,(H,19,20)/t10-,14-/m1/s1
207 |comp91| + comp92 + comp93 -> |comp94| + comp95 + comp96
208 @yield 54%
209
210 R19 "Acetal formation"
211 skeletonOut InChI=1S/C5H9BrO2/c1-5(4-6)7-2-3-8-5/h2-4H2,1H3
212 |comp97| + comp98 -> |comp99| + comp100
213 @yield 60%
214
215 R20 "Refunctionalization"
216 skeletonOut InChI=1S/C6H13NO2/c1-6(5-7-2)8-3-4-9-6/h7H,3-5
      H2,1-2H3
217 |comp101| + 2 * comp102 -> |comp103| + comp104
218 @yield 80%

```

PHENOL PLANS

The synthesis plans listed here were compiled by the books Industrial Chemicals [58] as well as Algebra of Organic Synthesis [30].

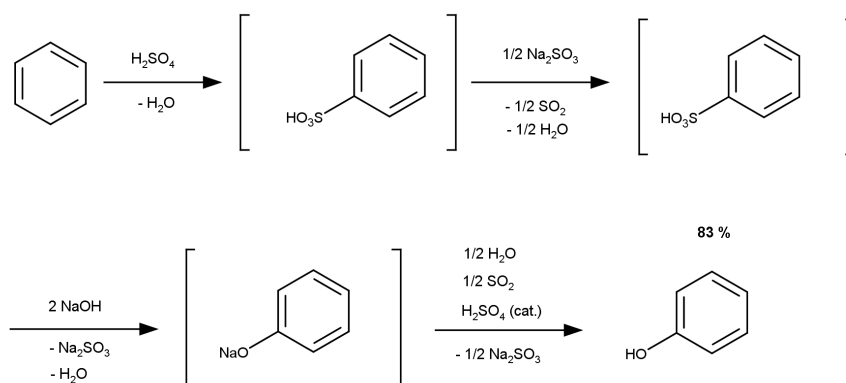


Figure 41: Benzenesulfonate process for the synthesis of phenol (FaithG1)

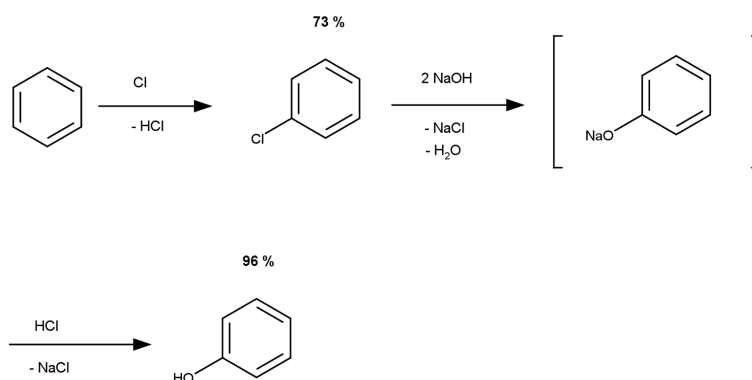


Figure 42: Chlorination-caustic process for the synthesis of phenol (FaithG2)

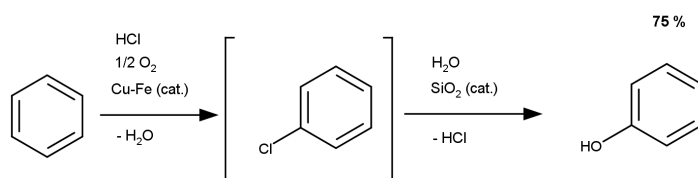


Figure 43: Raschig process for the synthesis of phenol (FaithG3)

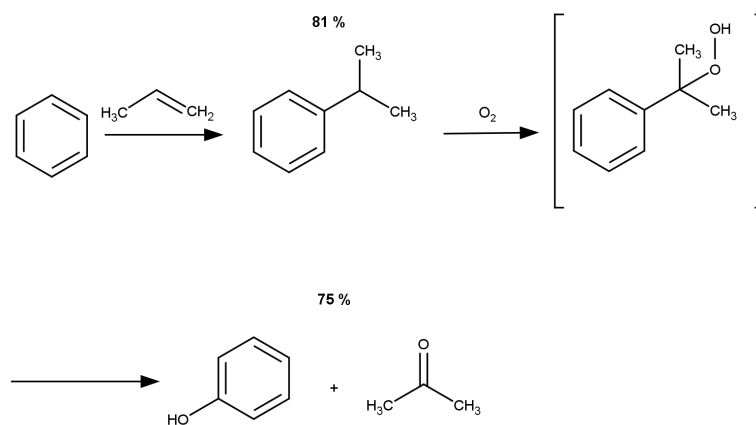


Figure 44: Cumene peroxidation process for the synthesis of phenol (FaithG4)

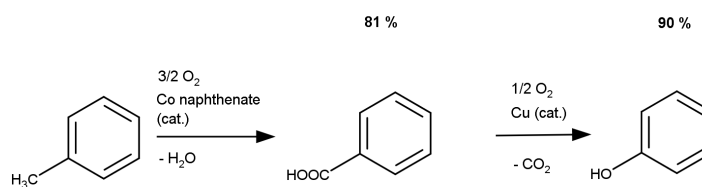


Figure 45: Toluene oxidation process for the synthesis of phenol (FaithG5)

BIBLIOGRAPHY

I tried to trace the copyright holder of the image material used in this work and get their permission. Should still an infringement of copyright emerge I ask kindly to contact me.

Ich habe mich bemüht, sämtliche Inhaber der Bildrechte ausfindig zu machen und ihre Zustimmung zur Verwendung der Bilder in dieser Arbeit eingeholt. Sollte dennoch eine Urheberrechtsverletzung bekannt werden, ersuche ich um Meldung bei mir.

- [1] F. Wöhler. "Ueber künstliche Bildung des Harnstoffs". In: *Annalen der Physik* 88.2 (Jan. 1, 1828), pp. 253–256. ISSN: 1521-3889. DOI: 10.1002/andp.18280880206. URL: <http://onlinelibrary.wiley.com/doi/10.1002/andp.18280880206/abstract> (visited on 09/03/2014).
- [2] David Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". In: *Journal of Chemical Information and Computer Sciences* 28.1 (Feb. 1, 1988), pp. 31–36. ISSN: 0095-2338. DOI: 10.1021/ci00057a005. URL: <http://dx.doi.org/10.1021/ci00057a005> (visited on 06/10/2014).
- [3] "The Wiswesser Line-Formula Chemical Notation. By ELBERT G. SMITH. McGraw-Hill Book Co., 330 West 42nd Street, New York, NY 10036, 1968. 15 Å 22.5 cm. xv + 302 pp. Price \$15.00". In: *Journal of Pharmaceutical Sciences* 58.4 (Apr. 1, 1969), pp. 510–510. ISSN: 1520-6017. DOI: 10.1002/jps.2600580439. URL: <http://onlinelibrary.wiley.com/doi/10.1002/jps.2600580439/abstract> (visited on 03/06/2015).
- [4] Harry E. Pence and Antony Williams. "ChemSpider: An Online Chemical Information Resource". In: *Journal of Chemical Education* 87.11 (2010), pp. 1123–1124. ISSN: 0021-9584. DOI: 10.1021/ed100697w. URL: <http://dx.doi.org/10.1021/ed100697w> (visited on 01/30/2015).
- [5] *OpenSMILES*. URL: <http://www.opensmiles.org/opensmiles.html> (visited on 04/02/2015).
- [6] David Weininger, Arthur Weininger, and Joseph L. Weininger. "SMILES. 2. Algorithm for generation of unique SMILES notation". In: *Journal of Chemical Information and Computer Sciences* 29.2 (May 1, 1989), pp. 97–101. ISSN: 0095-2338. DOI: 10.1021/ci00062a008. URL: <http://dx.doi.org/10.1021/ci00062a008> (visited on 06/10/2014).

- [7] Stephen E. Stein et al. *IUPAC International Chemical Identifier (InChI) InChI version 1, software version 1.04 (2011) Technical Manual*. URL: http://www.inchi-trust.org/wp/wp-content/uploads/2014/06/InChI_TechMan.pdf (visited on 10/14/2014).
- [8] *IUPAC - International Union of Pure and Applied Chemistry: Project Details*. URL: http://www.iupac.org/home/projects/project-db/project-details.html?tx_wfqbe_pi1%5bproject_nr%5d=2000-025-1-800 (visited on 10/14/2014).
- [9] Wouter Penard and Tim van Werkhoven. "On the Secure Hash Algorithm family". In: URL: http://www.staff.science.uu.nl/~werkh108/docs/study/Y5_07_08/infocry/project/Cryp08.pdf.
- [10] Christopher Southan. "InChI in the wild: an assessment of InChIKey searching in Google". In: *Journal of Cheminformatics* 5 (Feb. 11, 2013). DOI: 10.1186/1758-2946-5-10. URL: <http://www.jcheminf.com/content/5/1/10>.
- [11] Arthur Dalby et al. "Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited". In: *Journal of Chemical Information and Computer Sciences* 32.3 (1992), pp. 244–255. ISSN: 0095-2338. DOI: 10.1021/ci00007a012. URL: <http://dx.doi.org/10.1021/ci00007a012> (visited on 01/07/2015).
- [12] *Open Babel*. URL: http://openbabel.org/wiki/Main_Page.
- [13] *CTFile Formats*. URL: <http://accelrys.com/products/informatics/cheminformatics/ctfile-formats/no-fee.php>.
- [14] *CTFile Formats*. June 2005. URL: c4.cabrillo.edu/404/ctfile.pdf.
- [15] Bruno Bienfait and Peter Ertl. "JSME: a free molecule editor in JavaScript". In: *Journal of Cheminformatics* 5.1 (May 21, 2013), p. 24. ISSN: 1758-2946. DOI: 10.1186/1758-2946-5-24. URL: <http://www.jcheminf.com/content/5/1/24/abstract> (visited on 01/21/2015).
- [16] Gerta Rücker and Christoph Rücker. "Walk Counts, Labyrinthicity, and Complexity of Acyclic and Cyclic Graphs and Molecules". In: *Journal of Chemical Information and Computer Sciences* 40.1 (Jan. 1, 2000), pp. 99–106. ISSN: 0095-2338. DOI: 10.1021/ci9900579. URL: <http://dx.doi.org/10.1021/ci9900579> (visited on 11/08/2014).
- [17] D. Bonchev. "Novel Indices for the Topological Complexity of Molecules". In: *SAR and QSAR in Environmental Research* 7.1 (Dec. 1, 1997), pp. 23–43. ISSN: 1062-936X. DOI: 10.1080/10629369708039123. URL: <http://dx.doi.org/10.1080/10629369708039123> (visited on 12/01/2014).

- [18] Gerta Ruecker and Christoph Ruecker. "Counts of all walks as atomic and molecular descriptors". In: *Journal of Chemical Information and Computer Sciences* 33.5 (Sept. 1, 1993), pp. 683–695. ISSN: 0095-2338. DOI: 10.1021/ci00015a005. URL: <http://dx.doi.org/10.1021/ci00015a005> (visited on 11/08/2014).
- [19] Eric W. Weisstein. *Walk*. URL: <http://mathworld.wolfram.com/Walk.html> (visited on 12/01/2014).
- [20] John Harris, Jeffry L. Hirst, and Michael Mossinghoff. *Combinatorics and Graph Theory*. Springer Science & Business Media, Apr. 3, 2009. 392 pp. ISBN: 9780387797113.
- [21] Paul T. Anastas and John Charles Warner. *Green Chemistry: Theory and Practice*. Oxford University Press, 1998. 135 pp. ISBN: 9780198502340.
- [22] *12 Principles of Green Chemistry - American Chemical Society*. URL: <http://www.acs.org/content/acs/en/greenchemistry/what-is-green-chemistry/principles/12-principles-of-green-chemistry.html> (visited on 01/30/2015).
- [23] R.A. Sheldon. "Organic Synthesis - Past, Present and Future". In: *Chem. & Ind.* (July 12, 1992), pp. 903–906.
- [24] B. M. Trost. "The atom economy—a search for synthetic efficiency". In: *Science (New York, N.Y.)* 254.5037 (Dec. 6, 1991), pp. 1471–1477. ISSN: 0036-8075.
- [25] David J. C. Constable, Alan D. Curzons, and Virginia L. Cunningham. "Metrics to 'green' chemistry - which are the best?" In: *Green Chemistry* 4.6 (Dec. 5, 2002), pp. 521–527. ISSN: 1463-9270. DOI: 10.1039/B206169B. URL: <http://pubs.rsc.org/en/content/articlelanding/2002/gc/b206169b> (visited on 01/30/2015).
- [26] Marco Eissen et al. "Atom Economy and Yield of Synthesis Sequences". In: *Helvetica Chimica Acta* 87.2 (2004), pp. 524–535. ISSN: 1522-2675. DOI: 10.1002/hlca.200490050. URL: <http://onlinelibrary.wiley.com/doi/10.1002/hlca.200490050/abstract> (visited on 01/30/2015).
- [27] Kenneth M. Doxsee and James E. Hutchison. *Green Organic Chemistry: Strategies, Tools, and Laboratory Experiments*. Thomson-Brooks/Cole, 2004. 260 pp. ISBN: 9780534388515.
- [28] David J. C. Constable et al. "Green chemistry measures for process research and development". In: *Green Chemistry* 3.1 (Jan. 1, 2001), pp. 7–9. ISSN: 1463-9270. DOI: 10.1039/B007875L. URL: <http://pubs.rsc.org/en/content/articlelanding/2001/gc/b007875l> (visited on 01/30/2015).

- [29] Tomas Hudlicky et al. "Toward a 'reagent-free' synthesis". In: *Green Chemistry* 1.2 (Jan. 1, 1999), pp. 57–59. ISSN: 1463-9270. DOI: 10.1039/A901397K. URL: <http://pubs.rsc.org/en/content/articlelanding/1999/gc/a901397k> (visited on 01/30/2015).
- [30] John Andraos. *The Algebra of Organic Synthesis: Green Metrics, Design Strategy, Route Selection, and Optimization*. CRC Press, Nov. 16, 2011. 1213 pp. ISBN: 9781420093292.
- [31] Martin D. Burke and Stuart L. Schreiber. "A Planning Strategy for Diversity-Oriented Synthesis". In: *Angewandte Chemie International Edition* 43.1 (2004), pp. 46–58. ISSN: 1521-3773. DOI: 10.1002/anie.200300626. URL: <http://onlinelibrary.wiley.com/doi/10.1002/anie.200300626/abstract> (visited on 01/30/2015).
- [32] Naoya Kumagai, Giovanni Muncipinto, and Stuart L. Schreiber. "Short Synthesis of Skeletally and Stereochemically Diverse Small Molecules by Coupling Petasis Condensation Reactions to Cyclization Reactions". In: *Angewandte Chemie International Edition* 45.22 (2006), pp. 3635–3638. ISSN: 1521-3773. DOI: 10.1002/anie.200600497. URL: <http://onlinelibrary.wiley.com/doi/10.1002/anie.200600497/abstract> (visited on 01/30/2015).
- [33] Peter J. Ramberg. "The Death of Vitalism and The Birth of Organic Chemistry: Wohler's Urea Synthesis and the Disciplinary Identity of Organic Chemistry". In: *Ambix* 47.3 (Nov. 1, 2000), pp. 170–195. ISSN: 0002-6980. DOI: 10.1179/amb.2000.47.3.170. URL: <http://www.maneyonline.com/doi/abs/10.1179/amb.2000.47.3.170> (visited on 01/09/2015).
- [34] Elizabeth C. Patterson. *John Dalton and the Atomic Theory: The Biography of a Natural Philosopher*. New York: Doubleday, 1970.
- [35] J. J. Thomson. "XL. Cathode Rays". In: *Philosophical Magazine Series* 5 44.269 (Oct. 1, 1897), pp. 293–316. ISSN: 1941-5982. DOI: 10.1080/14786449708621070. URL: <http://dx.doi.org/10.1080/14786449708621070> (visited on 01/09/2015).
- [36] E. Rutherford. "The scattering of α and β particles by matter and the structure of the atom". In: *Philosophical Magazine* 92.4 (Feb. 1, 2012), pp. 379–398. ISSN: 1478-6435. DOI: 10.1080/14786435.2011.617037. URL: <http://dx.doi.org/10.1080/14786435.2011.617037> (visited on 01/09/2015).
- [37] N. Bohr. "XXXVII. On the constitution of atoms and molecules". In: *Philosophical Magazine Series* 6 26.153 (Sept. 1, 1913), pp. 476–502. ISSN: 1941-5982. DOI: 10.1080/14786441308634993. URL: <http://dx.doi.org/10.1080/14786441308634993> (visited on 01/09/2015).

- [38] E. Schrödinger. "Quantisierung als Eigenwertproblem". In: *Annalen der Physik* 386.18 (1926), pp. 109–139. ISSN: 1521-3889. DOI: 10.1002/andp.19263861802. URL: <http://onlinelibrary.wiley.com/doi/10.1002/andp.19263861802/abstract> (visited on 01/09/2015).
- [39] O. Isler et al. "Synthese des Vitamin A". In: *Helvetica Chimica Acta* 30.6 (1947), pp. 1911–1927. ISSN: 1522-2675. DOI: 10.1002/hlca.19470300666. URL: <http://onlinelibrary.wiley.com/doi/10.1002/hlca.19470300666/abstract> (visited on 01/12/2015).
- [40] Marshall Gates and Gilg Tschudi. "The Synthesis of Morphine". In: *Journal of the American Chemical Society* 78.7 (1956), pp. 1380–1393. ISSN: 0002-7863. DOI: 10.1021/ja01588a033. URL: <http://dx.doi.org/10.1021/ja01588a033> (visited on 01/12/2015).
- [41] R. B. Woodward et al. "THE TOTAL SYNTHESIS OF CHLOROPHYLL". In: *Journal of the American Chemical Society* 82.14 (1960), pp. 3800–3802. ISSN: 0002-7863. DOI: 10.1021/ja01499a093. URL: <http://dx.doi.org/10.1021/ja01499a093> (visited on 01/12/2015).
- [42] E. J. Corey. "Robert Robinson Lecture. Retrosynthetic thinking-essentials and examples". In: *Chemical Society Reviews* 17.0 (Jan. 1, 1988), pp. 111–133. ISSN: 1460-4744. DOI: 10.1039/CS9881700111. URL: <http://pubs.rsc.org/en/content/articlelanding/1988/cs/cs9881700111> (visited on 02/05/2015).
- [43] Christoph Rücker, Gerta Rücker, and Steven H. Bertz. "Organic Synthesis – Art or Science?". In: *Journal of Chemical Information and Computer Sciences* 44.2 (Mar. 1, 2004), pp. 378–386. ISSN: 0095-2338. DOI: 10.1021/ci030415e. URL: <http://dx.doi.org/10.1021/ci030415e> (visited on 11/08/2014).
- [44] Reaxys. URL: <https://www.reaxys.com/reaxys/secured/search.do>.
- [45] SciFinder. URL: <http://www.cas.org/products/scifinder>.
- [46] Ganapati D. Yadav and Bhavana G. Motirale. "Novelties of Solid-Liquid Phase Transfer Catalyzed Synthesis of Triclosan from Potassium 2,4-Dichlorophenolate and 2,5-Dichlorophenol". In: *Industrial & Engineering Chemistry Research* 47.23 (2008), pp. 9055–9060. ISSN: 0888-5885. DOI: 10.1021/ie800191k. URL: <http://dx.doi.org/10.1021/ie800191k> (visited on 02/05/2015).
- [47] Daniel Beaudoin and Philippe Murphy. *SynArchive*. URL: <http://www.synarchive.com/>.
- [48] Carsten Grønberg Lützen and Daniel Fentz Johansen. "A Computational and Mathematical Approach to Synthesis Planning". Master Thesis. Odense: University of Southern Denmark.
- [49] Carsten Grønberg Lützen. *SynthDB repository*. URL: <https://bitbucket.org/Puppetmaster/synthdb>.

- [50] *Indigo-Depict*. URL: <http://scitouch.net/opensource/indigo/indigo-depict>.
- [51] *Perl Dancer REST Plugin*. URL: <https://metacpan.org/pod/Dancer::Plugin::REST>.
- [52] *PSGI*. URL: <http://search.cpan.org/~miyagawa/PSGI-1.102/>.
- [53] Roy Thomas Fielding. "Architectural Styles and the Design of Network-based Software Architectures". PhD thesis. Irvine: University of California, 2000. URL: http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf.
- [54] *Curl*. Version 7.29.0. URL: <http://curl.haxx.se/>.
- [55] *Cactus Service*. URL: <http://cactus.nci.nih.gov/>.
- [56] *W3Counter: Global Web Stats - December 2014*. URL: <http://www.w3counter.com/globalstats.php?year=2014&month=12> (visited on 01/13/2015).
- [57] "Process for the production of halogeno-o-hydroxydiphenyl compounds". Pat. EP0887333. Dec. 30, 1998. URL: http://ep.espacenet.com/publicationDetails/originalDocument?CC=EP&NR=0887333A1&KC=A1&FT=D&ND=3&date=19981230&DB=ep.espacenet.com&locale=en_EP.
- [58] W.L. Faith, D.B. Keyes, and R.L. Clark. *Industrial Chemicals*. 3rd edition. New York: Wiley, 1966.
- [59] István Moldvai et al. "Enantioefficient synthesis of alpha-ergocryptine: first direct synthesis of (+)-lysergic acid". In: *The Journal of Organic Chemistry* 69.18 (Sept. 3, 2004), pp. 5993–6000. ISSN: 0022-3263. DOI: 10.1021/jo049209b.

Zusammenfassung

Synthesepläne stellen das Grundgerüst der organischen Synthesechemie dar. Ähnlich der Rezepte eines Koches erlauben sie es einem erfahrenen Chemiker die Errungenschaften der Vorgänger zu reproduzieren. Klassischerweise werden diese Anleitungen in Journalen publiziert und seit einiger Zeit für die elektronische Suche indiziert. Bis dato gab es aber keine Versuche dieses essenzielle Werkzeug in ein maschinenlesbares Format zu übertragen. Ein solches Format erlaubt es Software zu schreiben, welche Synthespläne automatisch analysiert und in Hinblick auf verschiedene Zielgrößen reihen kann. In dieser Arbeit wird ein solches System vorgestellt, welches es erlaubt Synthesepläne über ein Webinterface einzugeben und diese in einem automatisch auswertbaren Format zu speichern.

Ein eigens für diesen Zweck erstelltes Dateiformat wird vorgestellt, wobei ein besonderes Augenmerk auf den Vergleich verschiedener - bekannter - Formate zur Codierung von chemischen Strukturen gelegt wird. Im Gegensatz zu den vorhandenen Möglichkeiten bietet das neue Format auch die Möglichkeit zusätzliche Informationen wie Katalysatoren und Ausbeute zu berücksichtigen.

Weiters wurde ein System entwickelt, das es einem Chemiker erlaubt erstellte Synthesepläne über ein interaktives Webinterface mit Hilfe eines Moleküleditors hochzuladen und in dem beschriebenen Format zu speichern. Um die Qualität dieser Daten zu gewährleisten wurden Systeme implementiert, welche die Eingaben des Users überprüfen und im Falle widersprüchlicher Angaben dies dem Benutzer signalisieren. Um die Eingaben der Daten zu erleichtern wurde eine Verknüpfung des Editors mit chemischen Datenbanken sichergestellt.

In der wissenschaftlichen Literatur sind zumeist mehrere Syntheseansätze für eine Substanz zu finden. Welcher dieser vielen Ansätze der optimale für eine Anwendung ist musste von einem erfahrenen Chemiker entschieden werden. Insbesondere im Hinblick auf Umweltverträglichkeit wurden in der Vergangenheit viele Metriken entwickelt welche aber nur bedingt eingesetzt werden konnten da dies ein automatisches System voraussetzt. Einige dieser beschriebenen Metriken wurden als Computerprogramm implementiert das auf das entwickelte Dateiformat angewendet werden kann. Am Beispiel einer Analyse diverser Phenolsynthesen wird der Umfang dieser Möglichkeiten hervorgehoben.

In einem letzten Abschnitt werden die Möglichkeiten für ein solches System in der Zukunft erläutert, sowie die Arbeit von Kollaborationspartnern vorgestellt. Im Besonderen wird auch auf technische Aspekte im Projekt eingegangen die angepasst werden können.

Abstract

Synthesis plans are the backbone of organic synthetic chemistry. Analogous to a recipe for a chef synthesis plans allow a trained chemist to reproduce the achievements of other scientists. In a classical approach these plans are published in journals where they are indexed and searchable using online search engines. Until now there were no approaches of making these published plans machine readable. Creating such a format allows scientists to write software that analyzes synthesis plans automatically and rank them according to specified metrics. This work presents such a system that allows a user to enter synthesis plans using a modern web interface and saving them in a novel machine readable format.

A data format developed specifically for this project is presented - with a special focus on comparison of several existing formats for encoding molecular information. In contrast to the presented encoding the format used to encode synthesis plans allows for additional information like catalysts and reaction yield to be added to the file.

Furthermore a system was developed that allows a chemist to upload created synthesis plans using an interactive and responsive web interface and save them in the described format. To ensure the quality of the data entered a system was implemented to check the user input and in case of inconsistent information to signal the user the detected problems. To allow for easy synthesis plan creation a molecular editor with connections to chemical databases was used.

In scientific literature usually multiple synthesis plans for a specific substance are found. Which of these approaches is the best for a certain application had to be decided by an experienced chemist. Especially in regards to environmental impact several metrics have been developed in the past but application of those without a computer based system proved to be difficult. Some of the described metrics were implemented as a computer program that can calculate these for a synthesis plan provided in the developed file format. With the example of an analysis of several phenol synthesis the scope of these is shown.

In a last part the future of such a system is discussed. In one part the work of collaboration partners is presented. In the discussion a special focus on technical aspects is made to show some areas that could get adjusted.

Lukas Bartonek

Curriculum Vitae

Persönliche Daten

Name Lukas Bartonek
Staatsbürgerschaft Österreich

Ausbildung

- 09/2012 – laufend **Masterstudium Chemie an der Universität Wien**
- Schwerpunkte: Computersimulation und Datenanalyse sowie Physikalische Chemie
 - Masterarbeit: „*Making chemical Synthesis Plans Machine Readable*“ in Zusammenarbeit mit IMADA SDU Denmark
- 09/2009 – 09/2012 **Bachelorstudium Chemie an der Universität Wien**
- Bachelorarbeit im Bereich der bioanorganischen Chemie: „*Synthese neuartiger Indolochinoline zur Herstellung von wasserlöslichen Ruthenium- und Osmiumkomplexen als potentielle Zytostatika*“
- 2000 – 2008 **Realgymnasium BGRG 8, Albertgasse 18-22, 1080, Wien**
- Fachbereichsarbeit im Zuge der Matura im Bereich Chemie „*Proteinfärbung in der Gelelektrophorese*“ – prämiert durch die GÖCH

Sprachkenntnisse

Deutsch Muttersprache
Englisch Fließend in Wort und Schrift
Latein vier Jahre Ausbildung im Gymnasium

Stipendien & Konferenzen

- 2015 TBI Winterseminar in Bled, Slowenien
2014 – 2015 Forschungsförderung für eine wissenschaftliche Arbeit – Universität Wien
- 2012/13 Leistungsstipendium der Universität Wien
2011/12 Leistungsstipendium der Universität Wien
2009/10 Leistungsstipendium der Universität Wien