# Identification of Chemical Compounds

**Cristina Alexandra Ropot** (c.a.ropot@student.rug.nl)
**Isabelle Ameena Kampono** (i.a.kampono@gmail.com)
**Amanda Komulainen** (a.p.komulainen@student.rug.nl)

January 20, 2023

# Contents

# 1 Introduction

Chemistry is a broad and convoluted field of science with many elements, compounds, and reactions. It deals with the substances of which matter is constructed and their properties. Each substance or compound has its own individual, specific properties; which makes the process of identifying these compounds cumbersome. This being said, such a task would pose no problem for a computer, especially for an artificially intelligent system. The domain of Artificial Intelligence deals with creating systems that use reasoning in ways that are similar to human reasoning. A subset of artificial intelligence is knowledge technology, which aims to integrate an expert's knowledge, of a rather small or specific field, into a computer program. In this report, we will describe a knowledge system that makes inferences about the identification of chemical compounds, more specifically, halogen compounds. This knowledge system is aimed for laboratory use. For this purpose, we used a user interface built using *Python* to extract facts from the user, which is used to infer which compound the user is dealing with. We will also explain how we constructed our system, its inner workings, the problem it faces, and how it could be improved in the future.

# 2 Problem

As mentioned before, the task of classifying compounds can be difficult and inefficient if done manually. For example, some of the properties one would have to determine is the nature of the compound, its saturation, reactivity, and number of atoms in the compound to begin the identification. This task would be significantly easier with the use of a computer program, namely our knowledge system. As there are various types of existing chemical compounds, we decided to focus on halogen based compounds to narrow down the focus of our knowledge system. Our expert also recommended this decision. Therefore, *the problem our expert system can solve is the problem of identifying chemical compounds that are halogen based.*

Narrowing the focus of our knowledge system to halogen compounds is also due to the real-life broad usage of halogen derivatives and the large number of organic synthesis reactions they can be a part of, which allows them to result in compounds belonging to new classes such as alcohols, carboxylic compounds, carbonylic compounds, hydrocarbons, and so on.

The system is intended to be used when the user is already dealing with an unknown compound, and can perform tests or measurements with it. The system, along with the tests done by the user, will help classify the compound. This type of information provides a base for everyday calculations in the field of chemistry as it is indispensable for any sort of project that uses halogens. Due to the necessary quality of this knowledge, a wide range of users that deal with chemistry would benefit from it, from high school students to more advanced university students, all the way to researchers. In the case of high school or university students, our system would be a useful teaching tool that guides the students through experimental practice to learn more about halogen compounds. Researchers can also use the system as a tool to increase researching efficiency, by saving time in compound classification. This being said, we believe that the system we created has broad usage capabilities.

# 3 Expert

The expert we chose to interview to extract knowledge for our system is Mihaela Ropot, a chemistry professor at the University of Bucharest. We contacted the expert based on the family relationship they have with one of the members of our group; the mother of Cristina Ropot. We considered her a good expert, as she has been working in the field of chemistry for more than 25 years.

# 4 Role of Knowledge Technology

The problem of identification and classification in the field of chemistry requires an abundance of detailed observations of chemical properties. Our knowledge system aims to structure these observations and present results to the user in a timely and efficient manner, whilst considering the various interacting factors that play a role in the identification of halogens. Efficiency can be extremely significant in terms of chemical

compound classification and determination, as their reactions can be hazardous and in worst-case scenarios, life-threatening. Besides this, the identification of compounds is generally useful for research in the field of chemistry. With a knowledge system as a supporting tool in research, researchers can devote more time to testing their hypotheses and let the expert system perform classifications that need to be made throughout the research. Hence, a knowledge technology identification undoubtedly trumps manual identification.

As we interviewed and elicited knowledge from our expert, we noticed the identification procedure uses a forward chaining method, which is suitable to be done by a knowledge system. We use forward chaining because we do not know what compound we are dealing with at the start, and hence need to determine it with input data from the user. Based on the data, we eliminate the least likely compounds until we find the actual compound the user is looking for. Considering that chemistry is a field that involves lots of explicit and technical rules, we think that a knowledge system is an appropriate tool to use as it can encapsulate all aspects of the problem accurately and deductively output results to the user. One issue with using a knowledge system in the chemistry domain also arises from the same point as our reasoning for using it: the vast amount of information present in the field of chemistry can be a double edged sword, as it makes a knowledge system optimal for it, but also means that in order to build a robust system, the data manipulation and entry would be a long and costly process.

# 5 The Knowledge Models

## 5.1 Problem Solving Models

### 5.1.1 Monitoring Model

In order to make a complex model that can accurately identify halogen compounds, we incorporated two problem solving models, one for the monitoring process and another for the diagnostic process. The first problem solving model concerns itself with immediate measurements and observations that a scientist can obtain from a compound. The system will first ask whether or not the user has carried out certain laboratory tests on the compound, then use the obtained facts to infer values of the compound's main classes.

The rules for the problem solving model are all stored in a *XML* file, along with the questions that are linked to collect the necessary facts needed. In order to have a coherent flow of questions, we used states to group each question. The name of the states correspond to the different classes needed to classify the compound. By doing this, we are able to isolate the chain of questions for each compound attribute, meaning that if a user does not carry out a specific test, the system will still be able to determine other chemical attributes and output a list of possible halogen compounds based on that.

One special case we had to deal with was determining the value for the "number of halogen atoms"-class . In order to determine the number of atoms in a halogen, we have to plug the user input into a mathematical formula. Because of this, we decided to exclude the rules for this state from the *XML* file, and instead create a local function in *Python* that calculates this value and manually adds it to the *XML FactBase*.

The problem solving model uses a forward chaining inference engine written in *Python* wherein each fact that is obtained from the user input will be stored in a *FactBase*. This system will then go through the rules of the state and check if any of the antecedents are partially fulfilled by the facts present in the *FactBase*. If yes, the system will then ask a question that corresponds to the required antecedent needed to fire the rule. In the case that multiple rules have antecedents present in the *FactBase*, the system will check the recency of the fulfilled antecedent in the *FactBase* and prioritize asking questions to fulfil the rule that has the most recent fact.

If the system cannot fire anymore rules or ask anymore questions in a given state, it will proceed to the next state and begin a new line of questioning. Once the system has collected all the facts from the user and inferred values for the classes, we then go into the second problem solving model.

### 5.1.2 Diagnostic Model

The second problem solving model uses *Python* and *JSON* to identify the compound based on the inferred values of the classes. In order to do this, the system accesses the compound specifications stored in the *JSON* file and cross checks each chemical property with the ones inferred from the user input. When there is a

mismatch, the system will eliminate the compound from the list of possible answers. We decided to run the monitoring and diagnosing models separately as it would be redundant to run it in between each question when we do not have values for all of the classes.

### 5.2 Domain Model

The domain model holds all the facts that the problem-solving model and the rule model need to use to determine the chemical compound in question. Since we are only attempting to classify halogen compounds, the classes of the model will revolve around the important chemical properties of Halogens.

Our knowledge base will implement five classes :

- Nature of the halogen : fluorate, brominated, iodinated or chlorinated
- Number of halogen atoms in the molecule : monohalogenated or polyhalogenated
- Nature of hydro-carbonate radical : saturated, not saturated, aromatic
- Reactivity of halogen in the substitution reaction : low, normal, or high
- Aggregation state : solid, liquid or gas
- Solubility : in water , in organic compound
- Position : true or false

### 5.3 Rule Model

The system uses the rule model to infer facts from the user's input. For example, if the compounds burns in oxygen and it produces Carbon dioxide, then the compound is an organic halogen. Most of the rules are fired immediately after an input is received. The inferred facts are then used to determine the values of the main classes of the compound. However, some rules also make use of inferred facts from a different state, or even values from a specific class to fire. For example, if a compound has a "monohalogenated" value for its "number of atoms"-class, then it does not have a "position in chain" attribute. Because of this, we do not call the second problem solving model immediately. Moreover, most of the rules are only concerned with the monitoring stage of the problem solving model as the diagnostic model does not make use of rules to identify the compound.

The rule model is constructed using *XML*, in which the conditional rules and questions are defined. An example of the definition of conditional rules can be observed in the figure below. Note that these rules are only a portion of the entire rule model. The graphical representation of the complete rule can be seen in Appendix 11.

| **Condition (if)** | **Conclusion (then)** |
|---|---|
| Burns in oxygen | is organic |
| Produces CO2 | is organic |
| React to silver nitrate + is organic | contains halogen |
| Precipitate color is yellow + contains halogen | nature iodinated |
| Precipitate color is white + contains halogen | nature chlorinated |

## 6 User interface, functionality, tools used

The back-end of the user interface is created with *Python*, *XML*, and *JSON*. The Graphical User Interface (GUI) is implemented using the *PyQt5* library in *Python*. The main program is written in *Python* code, which interacts with the *XML* file that contains all the rules, as well as the *JSON* file that contains the knowledge base. The model is separated from the GUI so that the overall knowledge system remains modular and organized. The interaction between the model and the GUI occurs in the GUI module, where the GUI

creates a model object. The GUI uses functions from the model to manipulate the view and to inform the model of changes in the *FactBase* (stored in the *XML* file) based on the user's input.

We aimed to keep the GUI as simple, user-friendly, and intuitive as possible. This is because we do not know what level of computer expertise the users of our system will possess and we wanted to ensure that there would be no user incapable of using the identification system due to complexity reasons. We established the simplicity by presenting the questions one by one, with interactive buttons on the screen for user input under each question.

# 7 Walkthrough of a Session

The system starts with a start window with a single *START* button as seen in Figure 1. As the user presses the button, a question appears in the window along with buttons containing answer options, or a text box that requires a written answer as seen in Figures 2, 3, and 4. When an answer is provided by clicking or writing the answer, the window shows the next question. Several questions are asked in the same fashion until a conclusion is reached, seen in Figure 6. The user has the option to skip a written input by clicking a provided *NEXT* button that skips to the next question (Figure 4). However, this will result in an incomplete classification and the system cannot reach a singular conclusion. When this happens, the screen will show multiple possible compounds like in Figure 6. The user can either start over by clicking *RESET* or continue with the remaining questions as seen in Figure 5.

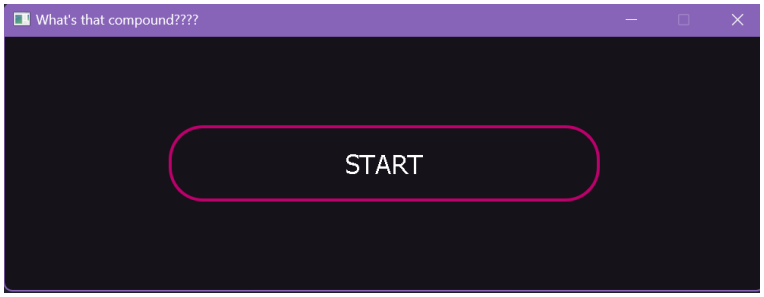


Figure 1: Start screen

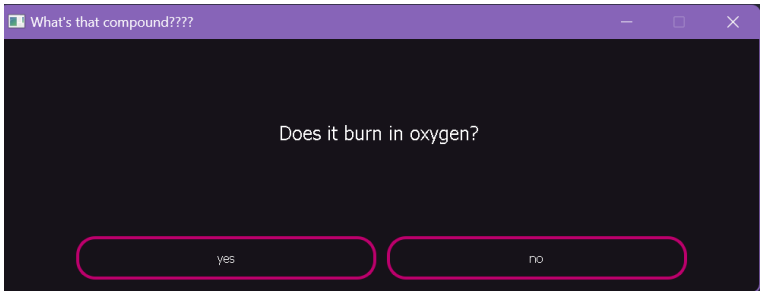


Figure 2: Two answer option screen

The complete walk-through is the following; The first state aims to determine whether the system can classify the compound at all, as it can only classify organic, halogen compounds. The first question inquires if the compound burns in an oxygen atmosphere with *YES* or *NO* answer options. If the user answer no, the compound is not organic and hence the system cannot classify it. Otherwise, it continues to the next question, which inquires if the compound produces Carbon dioxide when reacting with oxygen. The answer also determines whether the compound is organic or not. To determine if it is a halogen compound or not, the system inquires whether the compound reacts to silver nitrate. If the system is able to classify the compound based on the previous answer, the second state begins.

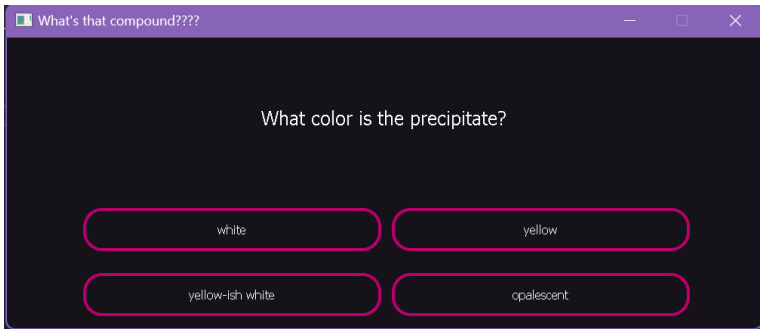


Figure 3: Four answer option screen

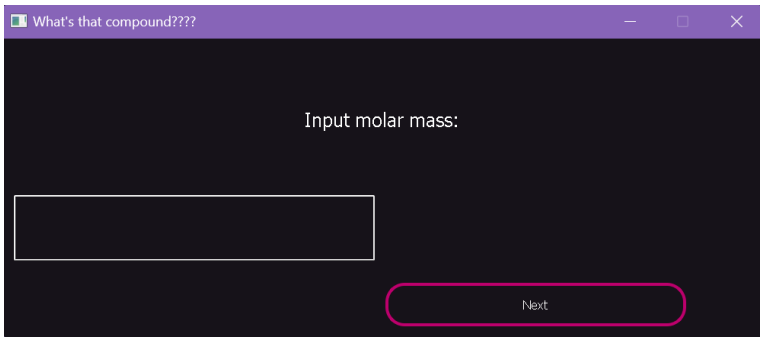


Figure 4: Text input screen

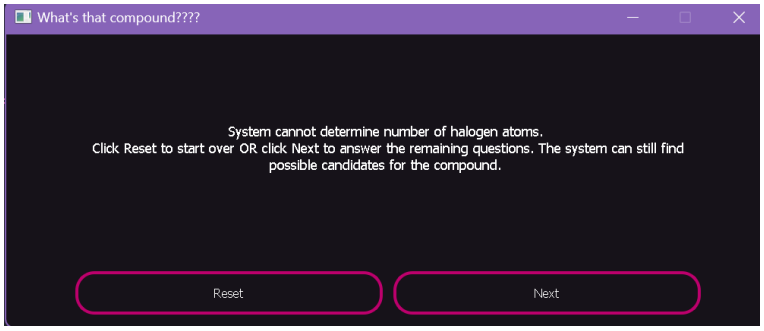


Figure 5: Early conclusion screen due to missing input

Then we move on to the second state, where the actual identification of the compound begins. In this state, we aim to determine the nature of the halogen compound. Following that, we move on to the third state, which determines the number of atoms in the compound. Here, the system asks the user questions that require text input. In order to achieve this we implemented a text box for the user to edit and a *NEXT* button that needs to be clicked in order to advance to the next question. The calculation of the number of atoms is done using a mathematical formula that we implemented using a function. This means that we need the user to be able to answer all the text input questions in order to calculate the number of atoms. If the user cannot answer one of the questions in this state then we immediately move on to the next state.

In the fourth state, we determine the aggregation state. This is done by inquiring whether the compound is in a liquid, gas, or solid-state. The fifth state determines the compound's solubility in water or organic compounds. The last state determines the reactivity of the compound. Here, the user is asked to determine the bond type and bond length of the compound by performing a Nuclear Magnetic Resonance test. It is the only way to determine the reactivity that we have implemented and therefore, the user is required to perform the test if they have not already. After the bond type is provided, the system also asks if the user has

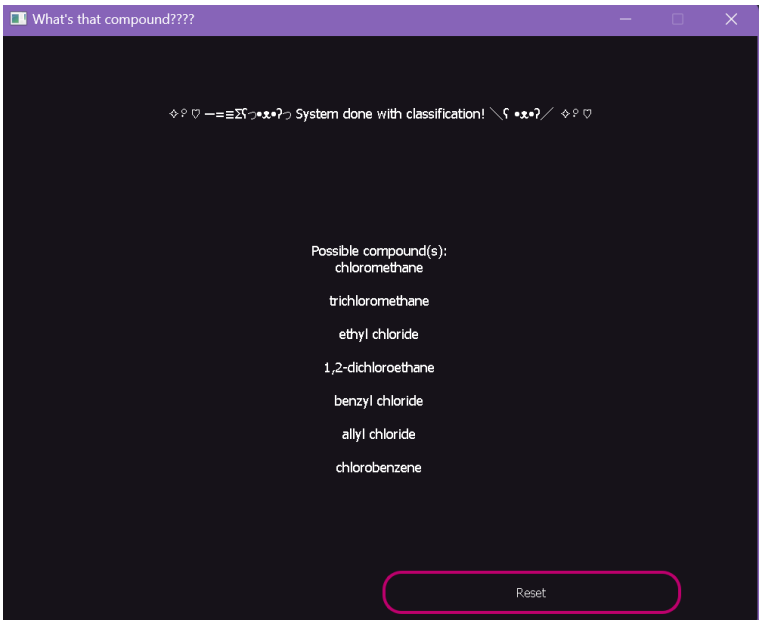


Figure 6: Final conclusion screen

performed a hydrogenation test to see if the hydrogen gets consumed. This test is also a strong requirement for successful classification. Based on these answers the reactivity type is concluded.

Lastly, after all the questions have been answered and all the facts have populated the fact base, the system outputs the answer.

# 8 Validation

After doing another walk-through of the knowledge system with our expert, a couple of improvements were suggested.

Firstly, when the system inquires about whether the user has performed mass spectrometry (which is used to determine the molar mass of the compound), the expert noted that as soon as the molar mass is known, the system should already be able to classify the compound. However, we chose not to implement this way of classification since the molar mass is too specific and too exact of a number to have in the compound's main class. Including the molar mass would make the knowledge system similar to an encyclopedia.

A second point of improvement was to add more physical attributes in the compound's main class for a better classification, such as solubility. We decided to implement this in the final version of the system. Other physical attributes the expert suggested were the density, and the boiling and melting points. However, for the same reason as the molar mass, we decided not to include them.

Overall feedback on the interface was that it is intuitive and user friendly. The expert liked the design choices and considered the system useful in classroom uses and research. She also confirmed that the rule model is accurate to the thought process of an actual chemist in identifying compounds.

# 9 Reflection

Overall, the experience of making this project has been challenging yet insightful. We thoroughly enjoyed the experience of incorporating knowledge from another field into one of our projects. It was also rewarding to learn how to make something that could be used to solve problems in real life. Moreover, we also faced lots of challenges in devising the model. First, we had to deal with settling on a topic and finding an expert to help us out with the project. This was challenging as it took us about two weeks to find an expert that would have enough time for us. Then, we had to decide which programming language would be most appropriate

to use with the knowledge base and rule base that we had in mind. After a lengthy process of trial and error, we were able to settle on a programming language that would support our design.

Designing the GUI posed a major challenge in this project as well. Connecting the problem-solving model along with the domain and rule model with the user interface was not always successful, and we had to change some of our approaches in the models. One of the causes for this problem was inconsistencies between the modules, such as variable names. Therefore, one point that we would improve is to fully plan out our modules and inference engine before we start implementing it.

Another point of improvement would be to make further questions that go deeper into how a scientist determines certain chemical properties. For example, we could make questions that asks about certain instrument readings in the lab that determine certain chemical properties.

Moreover, seeing that we did not implement all available chemical properties into the system, future developers should take into account that they could use more specific properties to classify a compound, i.e. by using its molar mass or boiling/melting points. It is also possible to widen the scope of the system to include non organic halogen compounds into the knowledge base.

# 10 Participation

Our teamwork was productive and well coordinated. We were able to establish good communication with each other through frequent meetings and group work. We all agree that the group work was divided equally and everyone contributed as much as necessary. The well-established communication enabled us to discuss most of the project together, even when working remotely. This good member relationship helped us be efficient and productive while working on this report.

The knowledge elicitation was done together, with Cristina acting as translator as the expert's native language is Romanian. The problem-solving model (back-end) was designed and implemented by Isabelle. The domain model and the rule model was defined together. The front-end of the GUI was mainly implemented by Cristina and Amanda, in collaboration with Isabelle to connect the back-end with the front-end. Overall, all of us worked with the code and report.

# 11 Appendix

Exemplification of questions, rules, states