Recognition of Condensed Structural Formulas of Chemical Compounds Using a Formal Grammar

Fumiyasu Sato[1] and Akio Fujiyoshi[2]

[1] Graduate School of Science and Engineering, Ibaraki University
14nm710n@vc.ibaraki.ac.jp
[2] Faculty of Engineering, Ibaraki University
akio.fujiyoshi.cs@vc.ibaraki.ac.jp

Abstract

In this paper, we propose a method to recognize condensed structural formulas of chemical compounds using a formal grammar. A condensed structural formula is a type of a structural formula that consists of only characters. It appears as a part of a structural formula of a chemical compound. In order to develop optical chemical structure recognition, we need to analyze the information of bonds between atoms in condensed structural formulas. We define the syntax and semantics of condensed structural formulas using a formal grammar and, thus, a recognition of condensed structural formulas can be done as a parsing process of the grammar.

1 Introduction

In the field of chemistry and pharmacy, many chemical compounds are found, and papers and patent applications are published every day. Many structural formulas exist in those publications. As can be seen in Figure 1, a structural formula represents the molecular structure of a chemical compound by using atomic symbols and lines meaning the bonds between atoms. In order to digitalize structural formulas correctly, optical chemical structure recognition has been developed.

In the development of optical chemical structure recognition, condensed structural formulas have to be considered because they appear as a part of structural formulas. A condensed structural formula is a type of a structural formula that consists of only characters. For example, "N", "OH", "COOH" and "SO_3Na" in Figure 1 are condensed structural formulas. Condensed structural formulas can save drawing space. Though lines showing the bonds between atoms are usually omitted, we can identify the structure uniquely from a condensed structural formula. However, some knowledge of chemistry is required to understand the structures of them.

Existing optical chemical structure recognition application, such as MolRec [2, 3, 4], OSRA [1] and ChemInfty [5, 6], use a predefined dictionary of condensed structural formulas, which stores the structure of condensed structural formulas. However, condensed structural formulas not stored in the dictionary cannot be processed. Therefore, a more general method is desired.

We define the syntax and semantics of condensed structural formulas using a formal grammar and propose a method to recognize condensed structural formulas using the grammar. The proposed method consists of two stages. In the first stage, we expand a condensed structural formula by adding characters for omitted bonds and by spreading iterations. We call those expanded condensed structural formulas "unfolded string." In the second stage, an unfolded string is parsed by the grammar and the structural information is obtained. The grammar is in the form of a context-free grammar because it is easy to handle.

We tested the method using 1427 structural formulas in JAPIC's collection of structural formulas of Japanese medicine [7]. As a result, the condensed structural formulas appearing in 94%(1346/1427) of the structural formulas in [7] can be recognized correctly.

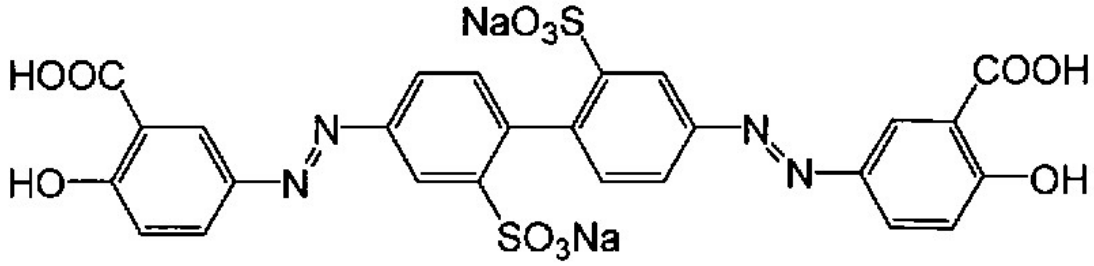


Figure 1: Structural formula including condensed structural formulas

2 Condensed Structural Formulas

Condensed structural formulas (CSF) are a type of structural formulas that consist of only characters. They can be defined recursively as follows:

- A string representing an atom is a CSF, e.g., "C", "H", "Cl" and "Na";
- A string representing a functional group is a CSF, e.g., "Me" and "Ph";
- A string of an atom with a subscript of a number is a CSF, e.g., "H_2" and "Cl_4";
- A string of an atom with a superscript of "+" or "-" (sometimes with a number) is a CSF, e.g., "H^+", "O^-" and "Ca^{2+}";
- A CSF enclosed by parentheses is a CSF, e.g., "(OH)" and "(CH_3)";
- A CSF enclosed by parentheses with a subscripts of a number is a CSF, e.g., "$(OH)_2$" and "$(CH_2)_{10}$"; and
- For CSF's x and y, xy, x-y, x=y and x≡y are CSF, e.g., "CH_3CH_3", "CH_3-CH_3", "CH_3=CH_3" and "CH_3≡CH_3".

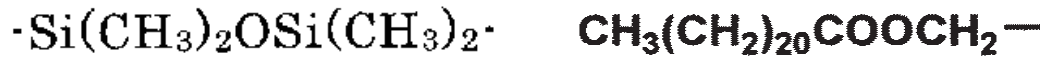
In Figure 1, "N", "OH", "HO", "COOH", "HOOC", "SO_3Na" and "NaO_3S" are condensed structural formulas. In general, hydrogen atom "H" is not omitted. Bonds between atoms are usually omitted, but the bond symbols are occasionally used. "-", "=" and "≡" are single bond, double bond and triple bond, respectively. Parentheses are used to represent branches and repeats of structures. A subscript of a number after atoms and parentheses indicates the number of iterations. A superscript of "+" or "-" (sometimes with a number) after atoms shows a charge.

Although condensed structural formulas can save the drawing space, some knowledge of chemistry is required to understand the structures of them. The understanding of condensed structural formulas has the following difficulty:

- Parentheses used to represent a branch or a repeat of a structure are ambiguous
- A valence of some atoms is ambiguous
- A direction of notation is changed by the connection to the main part of a structural formula

An example of the first difficulty is shown in Figure 2. It is impossible to determine whether parentheses mean branch or repeat of structure by only appearance. We need to check the valences of atoms in the parentheses.

Multiple valence of some atoms causes the second difficulty. For instance, sulfur has a valence of 2, 4, or 6. Neighboring atoms need to be considered to identify the valence of the sulfur as shown in Figure 3.



(a) (b)

Figure 2: (a) Brunch of structure and (b) repeat of structure

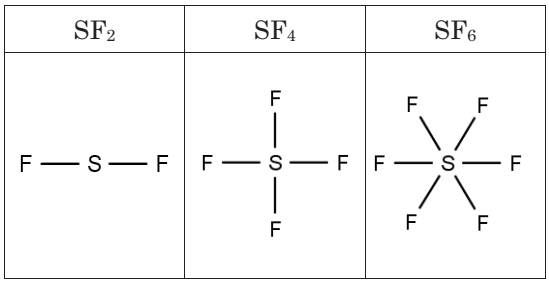


Figure 3: Sulfur has a valence of 2, 4 or 6

For an example of the third difficulty, the different directions of notation of a carboxy group is shown in Figure 4. The order becomes reverse when the main part of the structural formula comes to the right.

Figure 4: Carboxy group in different directions of notation

3 Formal Grammar

A formal grammar is a mechanism to describe a set of strings using production rules. A *formal grammar* is a four-tuple $G = (V_N, V_T, P, S)$, where

- V_N is a finite set of *nonterminal symbols*,
- V_T is a finite set of *terminal symbols*,
- $S \in V_N$ is the *start symbol*, and
- P is a finite set of *production rules* in the form of $\alpha \to \beta$, where α and β are a string of terminal and nonterminal symbols, the length of α is one or more, and the length of β is zero or more.

We define the *language* generated by G. To generate a string in the language, we begin with a string consisting only of the single start symbol S, and then successively applies the production rules in P to rewrite the string. At last, we obtain a string containing only terminal symbols in V_T. The language generated by G consists of all the strings that can be generated in this manner.

3.1 Context-Free Grammar

A context-free grammar is a formal grammar such that the left hand side of all production rules in P is a single nonterminal symbol. For example, we see the context-free grammar G that generates the language $\{a^n b^n | n \geq 0\}$:

$G = (N, \Sigma, P, S)$, where

- $N = \{S\}$,
- $\Sigma = \{a, b\}$, and
- $P = \{S \to ab, S \to aSb\}$.

The string *"aaabbb"* is generated by G as follows.

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$$

3.2 Parse Tree

A parse tree is a tree structure that represents how the context-free grammar generates a string. Figure 5 shows the parse tree representing how the string "*aaabbb*" is generated by the context-free grammar G. We can understand how a string is generated with the production rules of a grammar by tracing from the roots to the leaves in a parse tree. The task of creating a parse tree from a given string is called "parsing".

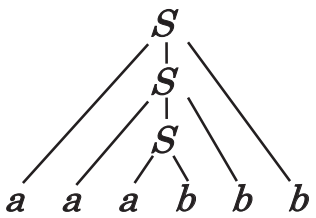


Figure 5: Parse tree

4 Proposed Method

We define the syntax and semantics of condensed structural formulas using a formal grammar and proposed a method to recognize condensed structural formulas using it. The proposed method consists of two stages. In the first stage, we expand a condensed structural formula by adding characters for omitted bonds and by spreading iterations. We call those expanded condensed structural formulas "unfolded string." In the second stage, an unfolded string is parsed by the grammar and the structural information of the condensed structural formula is obtained.

4.1 Preprocess

In the first stage, we convert a condensed structural formula to unfolded string as shown in the following.

```
* C H = C H 2    →    * ? C ? H = C ? H ? H
```

"`*`" represents the connection to the main part of a structural formula. We use "`?`" if bond type between atoms is unknown. Also, an abbreviation of a functional group is translated using a predefined table. For example, the abbreviation of methyl group "`Me`" is translated to "`-C-H-H-H`." The table consists of a collection of abbreviations of functional groups, and its meaning is different from the dictionary used in traditional optical chemical structure recognition application.

4.2 Formal Grammar to Generate Unfolded Stinrg

A grammar that defines the syntax of unfolded string is shown in Figure 6. The grammar is in the form of a context-free grammar because they are easy to handle. Figure 6 (1)"SS" is the start symbol. A_0, A_1, A_2, A_3, A_4, A_5 and A_6 are nonterminal symbols. The character strings enclosed by double quotations, such as "`H`", "`Tos`", "`?`", "`=`", "`(`" and "`)`", are terminal symbols.

Some grammar rules are defined based on a valence of each atom. The valence of an atom is the number of bonds the atom can share with other atoms, and it is often called "the number of hands" of an atom. A subscript of a nonterminal symbol represents the number of hands that is available to bind to other atoms. For example, the number of hands of hydrogen "`H`" is one, and the rule $A_1 \rightarrow$ "`H`" exists. Figure 6 (16) - (23) are rules to assign the number of hands to terminal symbols representing atoms or functional groups.

Figure 6 (2) - (10) are rules to form bonds. For example, A_1 and A_2 use one hand each other to form a bond, and the rule $A_1 \rightarrow A_1$"`?`"A_2 exists. Also, the rule $A_1 \rightarrow A_1$"`-`"A_2 exists because the explicit single bond symbol "`-`" is sometimes used.

Figure 6 (11) - (15) are rules to represent conversion of atoms enclosed in parentheses. In calculation of hands in parentheses, we exclude hands which is used between inside and outside of parentheses before we calculate the number of hands. For example, when an atom in parentheses form a single bond to an atom out of parentheses, we add a symbol representing an atom which has one hand before the first atom in parentheses. Then, if calculation of the hands in parentheses done correctly and A_0 is resulted, A_0 is converted to A_1 because the excluded one hand is restored. Therefore, the rule $A_1 \rightarrow$ "`(-`"A_0"`)`" exists.

For example, we show how the unfolded string "`*?C?H=C?H?H`" is parsed by the grammar.

$$
\begin{aligned}
SS &\Rightarrow A_0 \\
&\Rightarrow A_1\texttt{?}A_1 \Rightarrow A_1\texttt{?H} \\
&\Rightarrow A_2\texttt{?}A_1\texttt{?H} \Rightarrow A_2\texttt{?H?H} \\
&\Rightarrow A_2\texttt{=}A_4\texttt{?H?H} \Rightarrow A_2\texttt{=C?H?H} \\
&\Rightarrow A_3\texttt{?}A_1\texttt{=C?H?H} \Rightarrow A_3\texttt{?H=C?H?H} \\
&\Rightarrow A_1\texttt{?}A_4\texttt{?H=C?H?H} \Rightarrow A_1\texttt{?C?H=C?H?H} \\
&\Rightarrow \texttt{*?C?H=C?H?H}
\end{aligned}
$$

Figure 7 (a) indicates the parse tree corresponding to the unfolded string. Figure 7 (b) shows the complete structural formula obtained from the parse tree using the semantics of a condensed structural formula.

(1)	$SS \rightarrow$	A_0
(2)	$A_0 \rightarrow$	A_1 "?" A_1 | A_2 "?" A_2 | A_3 "?" A_3 | A_1 "-" A_1 | A_2 "=" A_2 | A_3 "#" A_3
(3)	$A_1 \rightarrow$	A_1 "?" A_2 | A_2 "?" A_1 | A_2 "?" A_3 | A_3 "?" A_2 | A_3 "?" A_4 | A_4 "?" A_3
(4)		| A_1 "-" A_2 | A_2 "-" A_1 | A_2 "=" A_3 | A_3 "=" A_2 | A_3 "#" A_4 | A_4 "#" A_3
(5)	$A_2 \rightarrow$	A_1 "?" A_3 | A_3 "?" A_1 | A_2 "?" A_4 | A_4 "?" A_2 | A_3 "?" A_5 | A_5 "?" A_3
(6)		| A_1 "-" A_3 | A_3 "-" A_1 | A_2 "=" A_4 | A_4 "=" A_2 | A_3 "#" A_5 | A_5 "#" A_3
(7)	$A_3 \rightarrow$	A_1 "?" A_4 | A_4 "?" A_1 | A_2 "?" A_5 | A_5 "?" A_2 | A_3 "?" A_6 | A_6 "?" A_3
(8)		| A_1 "-" A_4 | A_4 "-" A_1 | A_2 "=" A_5 | A_5 "=" A_2 | A_3 "#" A_6 | A_6 "#" A_3
(9)	$A_4 \rightarrow$	A_1 "?" A_5 | A_5 "?" A_1 | A_2 "?" A_6 | A_6 "?" A_2 | A_1 "-" A_5 | A_5 "-" A_1 | A_2 "=" A_6 | A_6 "=" A_2
(10)	$A_5 \rightarrow$	A_1 "?" A_6 | A_6 "?" A_1 | A_1 "-" A_6 | A_6 "-" A_1
(11)	$A_1 \rightarrow$	"(-" A_0 ")"
(12)	$A_2 \rightarrow$	"(-" A_1 ")" | "(=" A_0 ")"
(13)	$A_3 \rightarrow$	"(#" A_0 ")"
(14)	$A_4 \rightarrow$	"(=" A_2 ")"
(15)	$A_6 \rightarrow$	"(#" A_3 ")"
(16)	$A_0 \rightarrow$	"H^{+}" | "Na^{+} | "K^{+}" | "Al^{3+}" | "Ca^{2+}" | "Zn^{2+}" | "Gd^{3+}"
(17)		| "Cl^{-}" | "Br^{-}" | "I^{-}"
(18)	$A_1 \rightarrow$	"H" | "F" | "Na" | "Cl" | "K" | "Br" | "Ag" | "I" | "O^{-}" | "R" | "Ph" | "*" | "."
(19)	$A_2 \rightarrow$	"O" | "Mg" | "S" | "Ca" | "Zn" | "Hg" | "Co^{+}" | "Pt^{2+}" | "N^{-}" | "Pph" | "$" | ":"
(20)	$A_3 \rightarrow$	"N" | "Al" | "P" | "Co" | "As" | "Gd" | "Bi" | "S^{+}" | "%" | ";"
(21)	$A_4 \rightarrow$	"C" | "Si" | "S" | "Ge" | "Pt" | "N^{+}"
(22)	$A_5 \rightarrow$	"P" | "Sb"
(23)	$A_6 \rightarrow$	"S" | "Tc"

Figure 6: The grammar rules

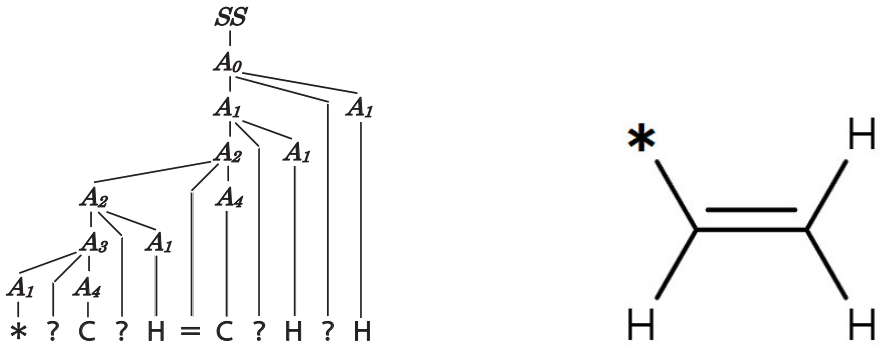
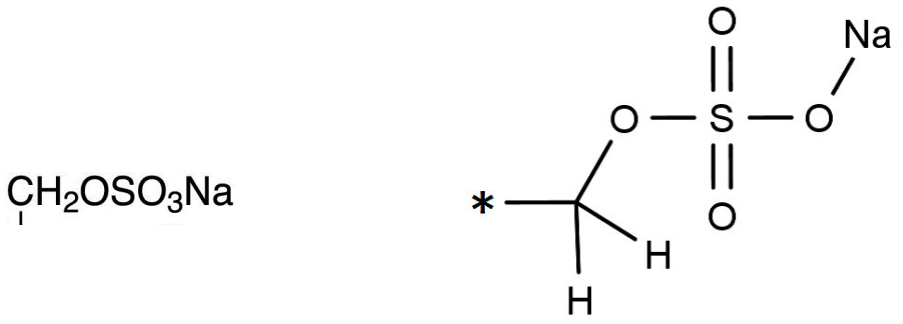


Figure 7: (a) Parse tree and (b) complete structural formula

5 Evaluation

For an evaluation of the method, we tested the method using 1427 structural formulas in JAPIC's collection of structural formulas of Japanese medicine [7]. As a result, the condensed structural formulas appearing in 94%(1346/1427) of the structural formulas in [7] can be recognized correctly. Figure 8 (a) is an example of a condensed structural formula recognized correctly. We add "∗" to connection part for the main structure and convert to unfolded string "*?C?H?H?O?S?O?O?O?Na". Figure 8 (b) shows a complete structural formula obtained by parsing the unfolded string.



(a) (b)

Figure 8: (a) Condensed structural formula recognized correctly and (b) complete structural formula obtained

81 condensed structural formulas which couldn't be recognized correctly are classified into the following three case:

(1) Chemical formula different from a condensed structural formula was used (71/81),

(2) An ionic bond was used (1/81), and

(3) The number representing atomic weight was attached to an atom (9/81).

Figure 9 shows an example of (1). The strings pointed by the arrows in Figure 9 are molecular formulas. Molecular formulas only indicate the simple number of each type of atom without information of structure. To convert a molecular formula to a complete structural formula, we require more knowledge of chemistry.

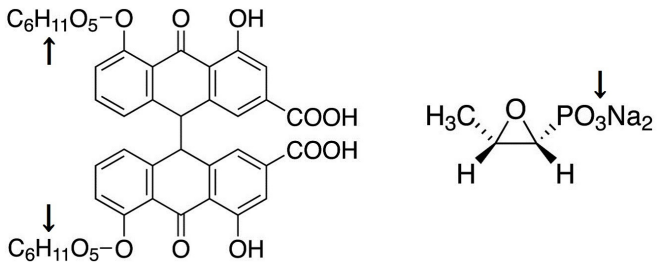


Figure 9: Molecular formulas (they are not a condensed structural formula)

Figure 10 shows an example of (2). The string pointed by the arrow in Figure 10 includes an ionic bond. The grammar used in our proposed method based on the calculation of valences for covalent bonds. Therefore, the condensed structural formula with ionic bonds cannot be processed.

Figure 11 shows an example of (3). The atomic symbols pointed by the arrows in Figure 11 have the number representing atomic weight. They are a notation to distinguish isotopes. The grammar in our method cannot deal with this notation.

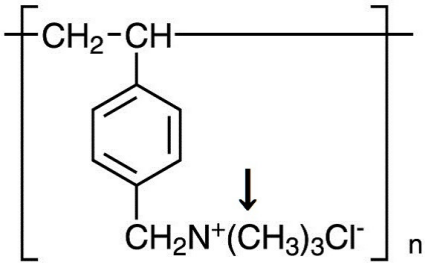


Figure 10: A condensed structural formula with an ionic bond

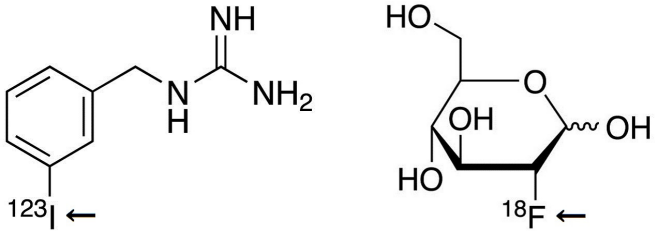


Figure 11: Atoms with atomic weight

6 Conclusion and Future Works

For the recognition of condensed structural formulas, a formal grammar defining the syntax of condensed structural formulas is introduced. The parse tree of a condensed structural formula can be obtained by a parsing process of the grammar. In a creation of a complete structural formula from the parse tree, the semantics of condensed structural formulas is used.

As a future work, we want to employ two-dimensional grammar to describe the total recognition process of a structural formula. It is difficult to recognize structural formulas correctly if characters and lines are touched each other. To recognize them correctly, we will divide them into line particles and consider the combinations of sets of particles using the two-dimensional grammar.

Acknowledgement

This work was partially supported by JSPS KAKENHI Grant Number 26282044.

References

[1] I. V. Filippov and M. C. Nicklaus, "Extracting chemical structure information: Optical structure recognition application," in *Pre-Proceedings of the 8th IAPR International Workshop on Graphics Recognition (GREC 2009)*, 2009, pp. 133–142.

[2] N. Sadawi, "A rule-based approach for recognition of chemical structure diagrams," PhD thesis, University of Birmingham, 2013, `http://etheses.bham.ac.uk/4325/`.

[3] N. Sadawi, A. Sexton, V. Sorge, "MolRec at CLEF 2012-Overview and Analysis of Results," in *the CLEF 2012 chemical structure recognition task*, 2012.

[4] N. Sadawi, A. Sexton, V. Sorge, "Performance of MolRec at TREC 2011 Overview and Analysis of Results," in *Text REtrieval Conference (TREC 2011)*, 2011.

[5] A. Fujiyoshi, K. Nakagawa and M. Suzuki, "Robust Method of Segmentation and Recognition of Chemical Structure Images in ChemInfty," in *Pre-Proceedings of the 9th IAPR International Workshop on Graphics Recognition (GREC 2011)*, 2011, pp. 121–125.

[6] D. Karzel, K. Nakagawa, A. Fujiyoshi and Masakazu Suzuki, "Inconsistency-Driven Chemical Graph Construction in ChemInfty," in *Proceedings of the 9th IAPR International Workshop on Graphics Recognition (GREC 2011)*, 2010, pp. 119–128.

[7] Japan Pharmaceutical Information Center (JAPIC), "Collection of structural formulas of Japanese medicine," 2010.