Syntax To Facilitate the Word Processing of Chemical Formulas

G. H. KIRBY* and SUSAN MILWARD

Department of Computer Studies, University of Hull, Hull, England

Received November 11, 1982

A syntax is presented for the unambiguous representation of chemical formulas as a string of characters on a single line without super- or subscripts. Software is described which analyzes formulas with this syntax and prints them with super- and subscripts as in normal printed chemical text. Rules are given for the recognition of single-line formulas from other text and demonstrate that chemical text could be handled by word-processing systems enhanced by software to recognize and analyze formulas. A technique is also described for indicating and printing Greek letters, Roman numerals, and other special characters all of which may occur in chemical formulas. Examples are given of input to and output from the software as implemented for a CBM microcomputer system.

INTRODUCTION

Chemical formulas embedded in text pose various problems for input and display on computer systems. The inclusion of chemical formulas in text for word processing requires careful consideration regarding the handling of super- and subscripts and the possible appearance of special characters such as Greek letters or Roman numerals.

Various systems for the unambiguous representation of chemical formulas on a single line and for conversion to the standard notation, with super- and subscripts, have been described.[1] Parsers have been developed for chemical formulas[2,3] although these tend to be more concerned with checking the syntax mainly for interactive computer-based learning or data-base query systems. These approaches are, however, somewhat limited in application, since they cannot distinguish between numbers denoting mass numbers, atomic numbers, simple atom or group repetition factors, nor repetition factors preceding charges.

Where the handling of chemical formulas has been incorporated into word-processing systems, the usual method adopted is to add special characters to the text to identify super- and subscripts. These characters explicitly indicate that the printer needs to move the paper up or down half a line or to backspace so that, for example, the charge may appear directly above the repetition factor. During printing, these characters are translated into the appropriate control codes.[4] Facilities are not provided for viewing the desired output on the screen prior to printing.

The identification of super- and subscripts is a problem present in the input of scientific text generally. The use of special characters introduces difficulties in the correct preparation and entering of material, a task that may well be performed by scientifically unskilled personnel. Furthermore, the available character set is restricted; where this has been overcome by the use of escape sequences, the input is made even more cumbersome. Finally, the meaning of the formula is obscured by these additional characters. For example, in the Wordcraft word-processing system on Commodore computers[5] $[O_2^+][PtF_6^-]$ would be input as

```
[OESC–2ESC+ESC+ESC!+ESC–]
                [PtFESC–6ESC+ESC+ESC!–ESC–]
```

where ESC+ gives positive half line-feed, ESC– gives negative half-line feed, and ESC! gives a backspace.

We have developed a simple syntax for chemical formulas presented on a single line which aids the recognition of formulas from ordinary text and allows the determination, by context, of those characters which are super- or subscripts. Only one reserved character is essential, acting as a separator between possible adjacent numbers such as the repetition factor of an atom and that of its charge, as in SO_4^{2-}, or between mass and atomic numbers, as in $^{13}_{6}C$.

This syntax applies to formulas as commonly used in inorganic chemistry and to empirical formulas of organic compounds. It is not intended to handle the many ways in common use for indicating both the organization of atoms and groups and the types of bonding within organic compounds.

We also show how characters encountered in chemical formulas that are not available in normal character sets, such as Greek letters and Roman numerals, can be indicated on input and displayed when the printing peripheral can produce user-defined characters.

THE SYNTAX

In general, chemical formulas consist of specific characters appearing in restricted positions as follows:

+ – ⟹ superscript

{} [] () A...Z a...Z ⟹ natural (i.e., middle or normal) line

0...9 ⟹ any of the three levels, including subscript

The three dots denotes a subrange of the characters, so A...Z means any of the upper case letters from A to Z.

Inputting a chemical formula on a single line means that in our scheme, for example

$CuSO_4$ would be input as CuSO4

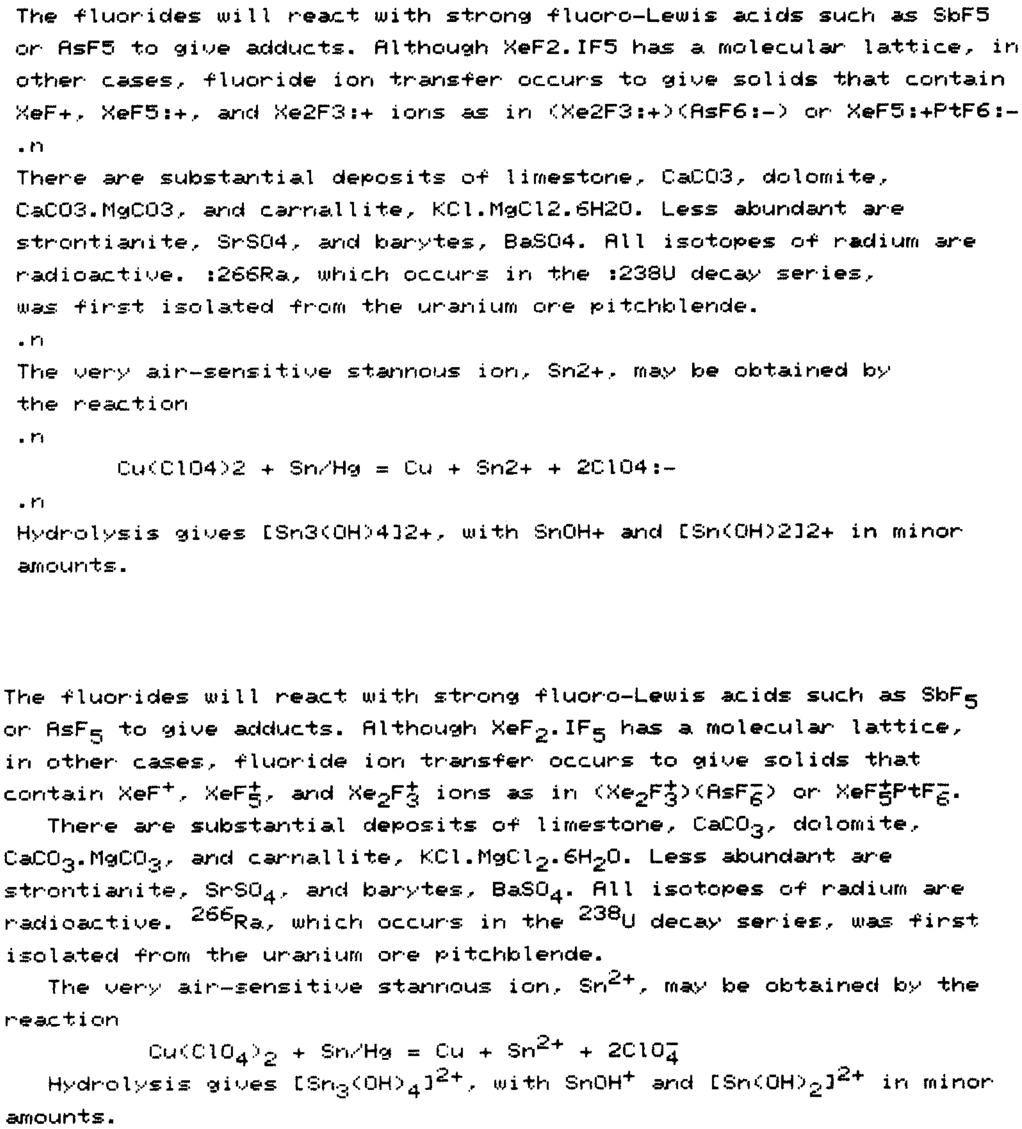
SO_4^{2-} would be input as SO4:2–

$[O_2^+][PtF_6^-]$ would be input as [O2:+][PtF6:–]

The colon has been chosen as the reserved character on the basis of previous discussion of the alternatives.[1] It acts as a separator in two situations: (i) to separate numbers; e.g., in the above examples we want SO_4^{2-} and not SO_{42}^- or SO^{42-}; (ii) to separate the repetition factor from a single charge; e.g., we want O_2^+ and not O^{2+}. While it should be possible to find which of these alternatives is correct through semantic checking, it would require sizeable tables of information and more sophisticated software, particularly to allow complete generality. In microcomputer applications, the size and speed of code are crucial; it is far better to impose a standard input format which is, in this case, very simple.

In order to process formulas in this way, the syntax given in Table I was derived. Organic representations such as

0095-2338/83/1623-0057$01.50/0 © 1983 American Chemical Society



```
The fluorides will react with strong fluoro-Lewis acids such as SbF5
or AsF5 to give adducts. Although XeF2.IF5 has a molecular lattice, in
other cases, fluoride ion transfer occurs to give solids that contain
XeF+, XeF5:+, and Xe2F3:+ ions as in (Xe2F3:+)(AsF6:-) or XeF5:+PtF6:-.
.n
There are substantial deposits of limestone, CaCO3, dolomite,
CaCO3.MgCO3, and carnallite, KCl.MgCl2.6H2O. Less abundant are
strontianite, SrSO4, and barytes, BaSO4. All isotopes of radium are
radioactive. :266Ra, which occurs in the :238U decay series,
was first isolated from the uranium ore pitchblende.
.n
The very air-sensitive stannous ion, Sn2+, may be obtained by
the reaction
.n
        Cu(ClO4)2 + Sn/Hg = Cu + Sn2+ + 2ClO4:-
.n
Hydrolysis gives [Sn3(OH)4]2+, with SnOH+ and [Sn(OH)2]2+ in minor
amounts.
```

The fluorides will react with strong fluoro-Lewis acids such as SbF_5 or AsF_5 to give adducts. Although $XeF_2.IF_5$ has a molecular lattice, in other cases, fluoride ion transfer occurs to give solids that contain XeF^+, XeF_5^+, and $Xe_2F_3^+$ ions as in $(Xe_2F_3^+)(AsF_6^-)$ or $XeF_5^+PtF_6^-$.

There are substantial deposits of limestone, $CaCO_3$, dolomite, $CaCO_3.MgCO_3$, and carnallite, $KCl.MgCl_2.6H_2O$. Less abundant are strontianite, $SrSO_4$, and barytes, $BaSO_4$. All isotopes of radium are radioactive. ^{266}Ra, which occurs in the ^{238}U decay series, was first isolated from the uranium ore pitchblende.

The very air-sensitive stannous ion, Sn^{2+}, may be obtained by the reaction

$$Cu(ClO_4)_2 + Sn/Hg = Cu + Sn^{2+} + 2ClO_4^-$$

Hydrolysis gives $[Sn_3(OH)_4]^{2+}$, with $SnOH^+$ and $[Sn(OH)_2]^{2+}$ in minor amounts.

Figure 1. Chemical text before and after processing by the formula recognizer and analyzer software. The input in the upper half uses the directive .n to indicate new paragraphs.

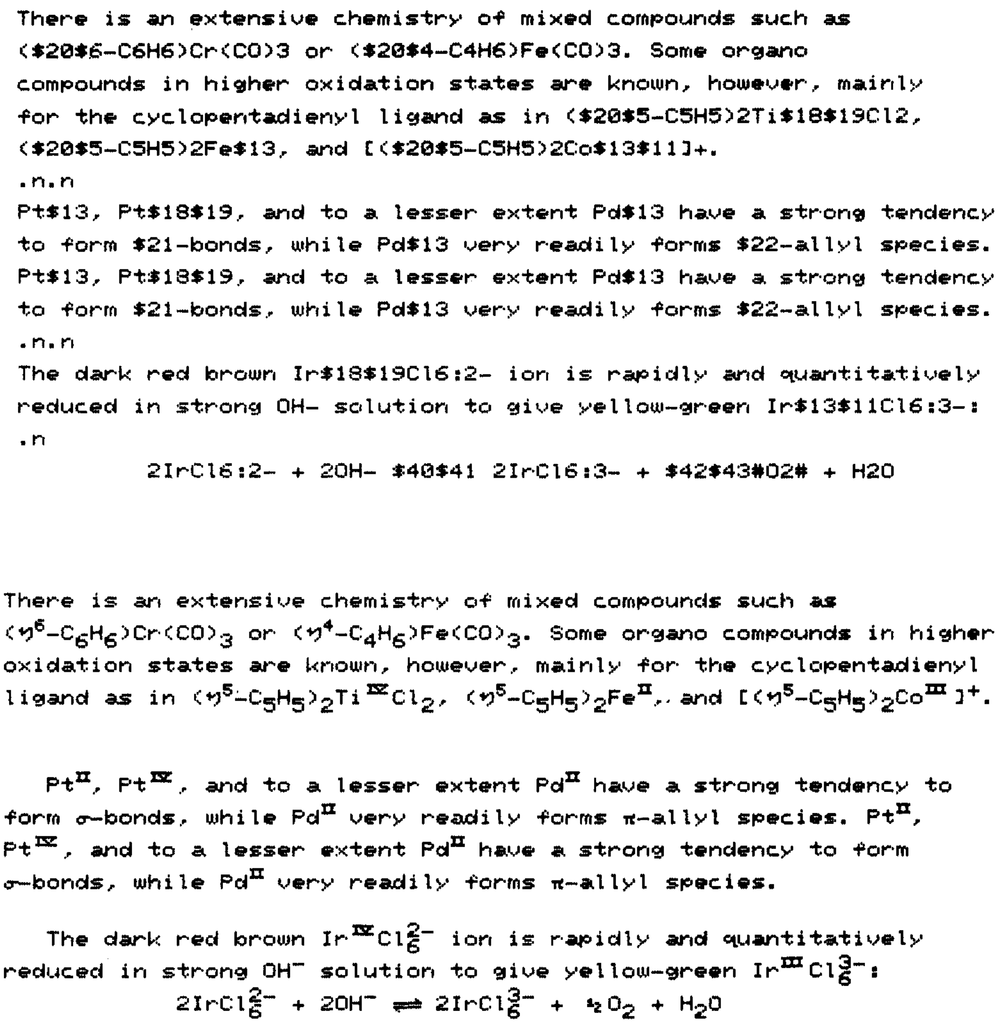
$CH_3.CH_2.CH_2.CH_3$ are covered, as well as the more usual inorganic formulas such as $K_2Cr_2O_7$ or ions like $[Fe(H_2O)_5(OH)]^{2+}$. Each must be terminated by a space so that a formula is treated like a word of text. These syntax rules are expressed in the standard syntactic meta-language defined by the British Standards Institution.[6] Note that symbols enclosed by an apostrophe are represented by themselves, optional symbols are enclosed in square brackets, i.e., [], and symbols that can be repeated any number of times, including zero, are enclosed in braces, i.e., {}.

The rules prevent any ambiguity when an atom X is possibly followed by super- and subscripts (i.e., charge and repetition factor) and the next atom Y is possibly preceded by super- and subscripts (i.e., mass and atomic numbers), as in $X_{i}^{j}{}_{l}^{k}Y$. Thus, in the single-line format, the repetition factor *i* must precede any charge *j*, a reasonable rule since this is the order in which chemists refer to ions. Note that where the repetition factor and charge are both present, the colon separator must be used, as previously mentioned, to avoid the possible ambiguity in, for example, SO_4^{2-} and O_2^+. Similarly, mass and atomic numbers, which can precede an atomic symbol, must both be preceded by a colon to separate them from a preceding repetition factor or to identify them. The exception to this is the case of a mass number following a charge, which itself delimits the previous atom. While a mass number may appear with or without an atomic number, an atomic number cannot appear on its own, a situation reflecting general usage of mass and atomic numbers. Hence :13C means ^{13}C and is distinct from 13C; $^{13}_{6}C$ is represented as :13:6C.

Table I. Syntax for Chemical Formulas Presented on One Line

```
fullformula = formula, {'.', formula}, ' ';
formula = [integer], body;
body = atom, {body} | bracketsequence, {body};
bracketsequence = bra, body, ket, [repfactcharge];
atom = [massatno], atsymbol, [repfactcharge];
massatno = [':'], massno, [':', atno];
repfactcharge = repfactor | [repfactor, ':'], [digit], sign |
  [repfactor, ':'], sign, {sign};
bra = '(' | '[' | '{';
ket = ')' | ']' | '}';
sign = '+' | '–';
atsymbol = capital, [lowercase];
capital = 'A' | 'B' | . . . . . | 'Z';
lowercase = 'a' | 'b' | . . . . . | 'z';
massno = integer;
atno = integer;
repfactor = integer;
integer = digit, {digit};
digit = '0' | '1' | . . . . . | '9';
```

Software was written to process formulas according to this syntax. This software first looks for any prefix and then takes atoms in sequence. When an open bracket is encountered, a bracket sequence is started, and the process of searching, either



```
There is an extensive chemistry of mixed compounds such as
($20$6-C6H6)Cr(CO)3 or ($20$4-C4H6)Fe(CO)3. Some organo
compounds in higher oxidation states are known, however, mainly
for the cyclopentadienyl ligand as in ($20$5-C5H5)2Ti$18$19Cl2,
($20$5-C5H5)2Fe$13, and [($20$5-C5H5)2Co$13$11]+.
.n.n
Pt$13, Pt$18$19, and to a lesser extent Pd$13 have a strong tendency
to form $21-bonds, while Pd$13 very readily forms $22-allyl species.
Pt$13, Pt$18$19, and to a lesser extent Pd$13 have a strong tendency
to form $21-bonds, while Pd$13 very readily forms $22-allyl species.
.n.n
The dark red brown Ir$18$19Cl6:2- ion is rapidly and quantitatively
reduced in strong OH- solution to give yellow-green Ir$13$11Cl6:3-:
.n
          2IrCl6:2- + 2OH- $40$41 2IrCl6:3- + $42$43#O2# + H2O
```

There is an extensive chemistry of mixed compounds such as $(\eta^6\text{-}C_6H_6)Cr(CO)_3$ or $(\eta^4\text{-}C_4H_6)Fe(CO)_3$. Some organo compounds in higher oxidation states are known, however, mainly for the cyclopentadienyl ligand as in $(\eta^5\text{-}C_5H_5)_2Ti^{IV}Cl_2$, $(\eta^5\text{-}C_5H_5)_2Fe^{II}$,. and $[(\eta^5\text{-}C_5H_5)_2Co^{III}]^+$.

Pt^{II}, Pt^{IV}, and to a lesser extent Pd^{II} have a strong tendency to form σ-bonds, while Pd^{II} very readily forms π-allyl species. Pt^{II}, Pt^{IV}, and to a lesser extent Pd^{II} have a strong tendency to form σ-bonds, while Pd^{II} very readily forms π-allyl species.

The dark red brown $Ir^{IV}Cl_6^{2-}$ ion is rapidly and quantitatively reduced in strong OH^- solution to give yellow-green $Ir^{III}Cl_6^{3-}$:

$$2IrCl_6^{2-} + 2OH^- \rightleftharpoons 2IrCl_6^{3-} + \tfrac{1}{2}O_2 + H_2O$$

Figure 2. Chemical text containing special characters as printed on the CBM 4022P dot-matrix printer. Special characters are signified on input in the upper half by a $ followed by the key and embedded formulas by a #.

for an atom or an opening bracket, is started again. A closing bracket terminates the innermost level of bracket sequence nesting. The formula is either output while being processed or, alternatively, saved in a three-line buffer.

The syntax analyzer has the following structure:

```
repeat
   test for prefix (*i.e. 5 as in $5H_2O$*)
   call BODY
   test for dot (*as in $CuSO_4.5H_2O$*)
until end of formula
```

BODY is coded as a separate procedure to allow for the recursion necessary to deal with possible nesting of brackets, as follows:

```
procedure BODY
repeat
   if currentcharacter is open bracket
   then
      call BODY
      handle closing bracket
      test for any repetition factor and/or charge
   else (*character must belong to an atom*)
      repeat
         test for mass number and atomic number
         handle atomic symbol
         test for any repetition factor and/or charge
      until current character not in A..Z or 1..9
         (*because these imply a further atom*)
until end of formula or closing bracket or dot found
```

Syntax error reporting could very easily be added, for example, the detection of unmatched brackets.

RECOGNITION OF FORMULAS

The detection of formulas presented on a single line according to the syntax rules just described may be based solely on alphabet. A method was developed of distinguishing between formulas and ordinary text, so that the former are analyzed with the software just described and output with the correct alterations for super- and subscripts. All other text passes through unchanged, including formulas containing no super- or subscripts, e.g., 3HCl.

The rules that must be satisfied for the item to be analyzed as a formula for correct display are as follows: (i) The first character must be one of 1...9, A...Z, (, [, {, −, or :. This excludes immediately the majority of text words, which start with a lower case letter. (ii) After disregarding any initial digits, it must contain at least one of + − or 0...9, since only these characters may appear off the natural line. (iii) There must be at least one upper case letter. (iv) At least one of each pair of consecutive letters must be upper case.

We have written and run software in Pascal on a Commodore Business Machines CBM 8032 microcomputer to demonstrate the application of these rules coupled with the single-line syntax for presentation of chemical formulas.[7] Figure 1 gives examples of chemical text prepared for input in the upper half and in the lower half the resulting output after processing.

EXTENSIONS AND DISCUSSION

The syntax can be easily extended to allow for the inclusion of Greek letters, Roman numerals, and other special characters used in chemical notations, for example to denote locants, various types of bonding, and oxidation states. In these cases the problem posed is not one of recognition in the input but rather the output of such characters. We have accomplished this for the CBM dot-matrix 4022P printer and for daisy-wheel printers such as Diablo and Qume in two contrasting ways.

User-defined characters may be created for output on the CBM dot-matrix printer. The character is drawn in a 7 × 6 matrix. A dot in a particular position takes the value of 2 to the power of that row, and the pattern of dots required for a special character is sent to the printer as the sum of dot values for each column. Whenever the character code 254 is included in an output statement, the current special character is printed. Using this facility, it is possible for any Greek letters, Roman numerals, or other special character to be printed as they are encountered within the text.

The method we have adopted is to set up a file containing a series of records, each one consisting of six integers, corresponding to the pattern for the six columns, preceded by an integer serving to identify that character so that each record has a unique key. The program initially reads in these patterns, storing them in an array. Wherever a special character is required within text, it is then indicated in the input by the $ character to signal the occurrence of a special character followed by the key to specify which one is required. The stored keys are then searched for a matching value, and the appropriate bit pattern is sent to the printer. Any number of special characters may be drawn on a given line with no extra effort on the part of the user. In Figure 2 both input and, in the lower half, the resulting output are shown for text containing Greek letters, Roman numerals, small digits and the ⇌ sign.

In comparison, for achievement of similar results on a daisy-wheel printer, two or more print wheels are required, with the bulk of the text being printed the first time through, leaving gaps for the special characters. The paper is then rewound, the daisy wheel changed so that some or all of the gaps can be filled in, and so on. Some look-up table similar to that described above is needed to identify which code to put in output statements in order to print the appropriately special characters on second and later print wheels.

Embedded formulas, as encountered, for example, in expressions in chemical kinetics, may or may not be recognized by our software, depending on whether they start with a valid character. For example, $K[I^-][H_2O_2]$ presented as K[I–][H202] would be recognized, whereas k[I–][H202] would not since it starts with a lower case letter.

There are two methods of indicating embedded formulas. A space could be left before each formula, hence breaking it into appropriate "words". For example, the above could be denoted by

$$k\ [I–][H202] \Rightarrow k\ [I^-][H_2O_2]$$

or by

$$k\ [\ I -][\ H202\ ] \Rightarrow k\ [\ I^-\ ][\ H_2O_2\ ]$$

Alternatively, formulas can be enclosed by a character reserved for the purposes of identifying embedded formulas, e.g., #, in which case the above example would become

k#[I–][H202]#

and would be printed as a single word without spaces.

On the other hand, atomic symbols combined with the minus or hyphen character to denote bonds, as in Al–Si, are valid formulas as defined by the syntax and would be recognized and output as Al^-Si. The minus character is used in the syntax to indicate a negative charge and, without enforcing the unnatural use of the digit 1 to indicate a single charge as in Cl^{1-}, its alternative use to denote a single bond, as in Cl–, cannot be allowed. If a different character were acceptable for indicating single bonds in the input of chemical text, such as a left arrow (←), then bonds could be recognized and output by using the minus character. We have implemented a simpler approach which uses a further reserved character to identify "words" not to be tested by the formula recognizer, namely, the asterisk character, so that *Al–Si gives Al–Si as output.

Two additional reserved characters therefore need to be introduced to permit embedded formulas and "words" which are also valid formulae to be correctly formatted. Such cases are, however, fairly rare, and so these reserved characters only occur infrequently.

The techniques described here for handling chemical formulas input on a single line with other text and for displaying such formulas with super- and subscripts provide a simple solution to the problem of applying word processing to chemical text. The software that has been developed could serve as a model either for inclusion within a word-processing package for chemical text or as a preprocessor to convert chemical text to a form more suitable for input to a word-processing system.

ACKNOWLEDGMENT

We are grateful to the Department of Education and Science for the award of a Research Studentship in Information Science to S.M. and to Mr. J. D. Rayner for suggestions and assistance.

REFERENCES AND NOTES

(1) Kirby, G. H.; Morgan, C. H. "Chemical Formulae for Computer Input and Output". *Comput. Chem.* **1978**, *2*, 95–98.

(2) Barker, P. G. "Syntactic Definition and Parsing of Molecular Formulae: Part 1. Initial Syntax Definition and Parser Implementation". *Comput. J.* **1975**, *18*, 355–359.

(3) Kirby, G. H.; Morgan, C. H.; Rayner, J. D. "Microcomputer Formulae". *Educ. Chem.* **1981**, *18*, 15–17.

(4) Demas, J. N.; Demas, S. E. "A Microcomputerized Text Editor for Chemical Manuscripts". *J. Chem. Educ.* **1980**, *57*, 252.

(5) Dowson, P. L. "Wordcraft 80 Reference Guide"; Dataview: Colchester, England, 1982.

(6) "Method of Defining Syntactic Metalanguage"; British Standards Institution: London, 1981; BS 6154.

(7) The software is available at reasonable cost from the authors.