

One more macropackage for typesetting structural formulæ with \LaTeX

Igor Stokov*

Novosibirsk Institute of Organic Chemistry, Siberian Division of Russian
Academy of Sciences,
Lavrentiev avenue 9, Novosibirsk 90, Russia

A new macropackage for \LaTeX provides a high quality, easy and uniform typesetting of structural formulas of almost any complexity. The use of the new package called `streeTeX` implies a depth-in traversal of a structure and description of bonds and vertex labels being passed. Additional features of `streeTeX` include input of mass spectra and simple flow charts.

keywords: TeX, LaTeX, chemical structures

* Tel: (3832) 354745, E-mail: stokov@nioch.nsc.ru

1 Introduction

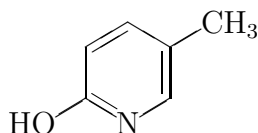
The choice of tools for scientific publishing is not very wide: most often it is MS WordTM or TeX — a typesetting language and system, developed in the beginning of 80–th by Knuth (Knuth, 1984). Briefly saying, TeX is eminent because of the detailed account of aspects affecting document appearance and, therefore, the perfect quality, practically unreachable by modern word processors. From the other side, TeX is not very comfortable for working with graphics, though it lets to import either various bitmaps (with sacrifice of device independence) or PostScript images. The last decision can lead to almost any visual effect but it requires special soft- and/or hardware understanding PostScript. At the same time, TeX itself is quite a powerful program language allowing to compose simple charts with the aid of pseudographical fonts. The most popular format of TeX, L^AT_EX (Lamport, 1984), can be said to establish a standard on the pseudographics use. Although most structural formulas of organic compounds can be fully typeset in this way, doing it by the direct use of L^AT_EX drawing commands is too laborious. There are at least three macros designed especially for chemical structures typesetting. Ramek's package (Ramek, 1990) is rather compact and easy in use, but it is distinctive by a formulas style which is rare in contemporary literature. More complex packages of Haas-O'Kane (Haas & O'Kane, 1987) and Fujita (Fujita, 1994) are free from this drawback. Both of them have similar design based on a set of different macrocommands, each for a specific chemical fragment. Usually a command defines some ring system, whose inner bond types and substituents can be altered by means of command parameters. For example, in Fujita's X²M_TE_X 5,5-dimethylcyclohexen-2-on-1 is entered as `\cyclohexanev[b]{1D==0;5Sb==;5Sa==}`, where `[b]` specifies the C=C double bond, `1D==0` — the carbonyl group, and `5Sb==;5Sa==` — two methyl substituents. Here, like in other cases, a command name correlates to a compound's systematic notation, which facilitates understanding of a command meaning. From the other side, description of structures on the level of characteristic fragments has some disadvantages. First of all, such fragments are that numerous, so any significant covering of their set is practically unreachible. This task is not fulfilled in both ChemTeX and X²M_TE_X, nevertheless the complete description of each one forms quite a volume book. Additionally, in cases when a formula consists of many fragments, their disposal on one figure becomes a tiresome task of coordinates calculation for nodes to be linked.

Therefore, making a description on a lower level of distinct bonds (after Ramek's steps) seems to be more appropriate. Bonds diversity compared to that of characteristic fragments is drastically lower, therefore formulas typesetting in terms of bonds promises to be more simple and universal. On the back side there is a possible loss of input speed and convenience. However, author's experience gave no such evidences. To support this statement let us examine how exactly structural formulas are entered in a new macropackage called *streeTeX*.

Following the TeX programmers' tradition, the new package name receives the common logo. The other part of the notation reflects the new approach essence: a chemical structure is regarded as a bond tree which is traversed in a depth-first order. The complete *streeTeX* constitutes single file `stree.sty`, whose usage is common to other style files in LaTeX.

2 Basics

All the input in *streeTeX* is held through single command `\stree{}`. A formula description containing inside brackets is composed in the following way: starting from any vertex one have to traverse all the structure moving along bonds and describing both the passing bonds and verteces labels met on the route. For example, the formula of 1-hydroxy-4-methylpyridine



is entered as `\stree{{HO} 2 0 >2 4 [2{CH$_3$}] >6 8N >10}`. Starting the traversal with vertex OH one have to first input its label: `{HO}`. A label is embraced in figure brackets if it contains more than one token. The next character 2 means: draw a bond in a direction at 2 o'clock on a 12-hours clockface. A traversal always has a current point (vertex). Setting a direction of a bond causes the current point to be shifted to the next vertex of this bond. E. g., after character 2 a ring vertex becomes current. Character 0 refers to the next bond at 0 o'clock (or upward). Then goes a double bond at 2 o'clock. If a bond is not single, then one or more tokens (so called *prefixes*) are put before its direction to specify bond features. A double bond may be displayed in several ways. In our case it is desirable to draw the second line

to the right from the main one. This very effect is achieved by prefix > (the right angle bracket).

The next single bond at 4 hours leads to a substituted vertex of the ring. Let us describe the substituent first. Construction `[2{CH$_3$}]` fulfills this goal as follows: opening bracket `[` marks a current vertex, 2-hours bond goes to vertex CH_3 (`{CH$_3$}` is the label description), finally closing bracket `]` retrieves the marked vertex. A methyl group is quite common although sequence `{CH$_3$}` is not very easy to input. A shorter command `\Me` may be used instead, so that `2\Me` is equivalent to `[2{CH$_3$}]` (any other number designating a bond direction may take place of digit 2 here).

After setting a substituent the traversal may be continued. Three bonds remain: at 6, 8, and 10 hours. The 6 and 10-hours bonds are double, drawn inside a cycle, which is indicated by prefix >. The 8-hours bond leads to a vertex labelled “N”, therefore `N` follows immediately after 8. Figure brackets are needless here because the label is composed of a single token.

In this example token groups related to different bonds are separated by spaces. Though spaces are not obligatory (`\stree{{H0}20>242\Me>68N>10}` is also correct), one may use them to make an input more readable. However, a space is illicit before a label or a command. In our example writing `8 N` (also `2 {CH$_3$}` or `2 \Me`) would cause an error. In one position, however, a *binding* space is required: between digit 1 designating a 1-hour direction and following another digit (it prevents missing, say, two ones with eleven).

3 Formal description

Now let us try to state more rigorously the basic rules just introduced. An argument of `\stree` is composed of descriptions of distinct bonds. To describe a bond one have to specify at least its direction — most often it is a number of hours on a clockface. One or more prefix(es) modifying a bond length, degree, appearance, etc., may precede a direction. If a bond leads to a labelled vertex, then the label must follow the direction. A command (in T_EX’s terms — something starting with `\`) may take place of a label, allowing to shorter set the same structural fragment. Here a label or a command is regarded as a bond attribute called *a suffix* due to its terminal position. In contrast to prefixes, a bond may have only one suffix. Besides, suffix is never preceded by a space. The form `<possible prefix(es)><direction><possible suffix>` may resemble a common verbal explanation of a traversal: *such-and-such* bond

at *that* o'clock leads to *so-and-so* label (only words like *such-and-such* are substituted by conditional tokens).

A question may arise: how to distinguish the current bond suffix from a prefix (or a direction) of the next bond? Spaces generally affect only a visual perception, while a computer most often ignores them obeying the following syntax rules given below. (The Backus–Naur form, where an arrow \longrightarrow means “is defined as” and token $|$ means “or”, is used.) Let us start from definition of $\langle\text{structure}\rangle$, i. e., of an argument of command `\stree`.

$$\langle\text{structure}\rangle \longrightarrow \langle\text{none}\rangle | \langle\text{bond}\rangle\langle\text{structure}\rangle | \\ \langle\text{structure}\rangle[\langle\text{structure}\rangle]\langle\text{structure}\rangle$$

It is just a formal way to say an argument to be composed from distinct bonds descriptions and to contain coupled and possibly nested square brackets. Then a definition of $\langle\text{bond}\rangle$ itself follows:

$$\langle\text{bond}\rangle \longrightarrow \langle\text{possible prefix(es)}\rangle\langle\text{direction}\rangle\langle\text{possible suffix}\rangle\langle\text{possible space(s)}\rangle$$

$$\langle\text{possible prefix(es)}\rangle \longrightarrow \langle\text{prefix}\rangle\langle\text{possible prefix(es)}\rangle | \langle\text{none}\rangle$$

$$\langle\text{prefix}\rangle \longrightarrow / | , | = | > | < | - | ' | ' | \sim | " | . | : | * | ^$$

$$\langle\text{possible suffix}\rangle \longrightarrow \langle\text{command}\rangle | \langle\text{label}\rangle | \langle\text{none}\rangle$$

$$\langle\text{label}\rangle \longrightarrow \{\langle\text{any sequence of tokens}\rangle\} | \langle\text{any character other than digit or prefix}\rangle$$

$$\langle\text{direction}\rangle \longrightarrow \langle\text{hours direction}\rangle | \langle\text{offsets direction}\rangle$$

$$\langle\text{hours direction}\rangle \longrightarrow 0 | 1\langle\text{delimiter}\rangle | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | \\ 11 | 12$$

$$\langle\text{delimiter}\rangle \longrightarrow \langle\text{space}\rangle, \text{ if the next token is a digit, otherwise } \langle\text{none}\rangle.$$
















$$\langle\text{offsets direction}\rangle \longrightarrow \langle\text{sign}\rangle\langle\text{cardinal number}\rangle\langle\text{sign}\rangle\langle\text{cardinal number}\rangle$$


$$\langle\text{plus or minus}\rangle \longrightarrow + | -$$

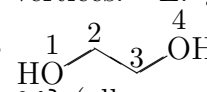
These rules eventually expand syntax elements in angular brackets either to definite tokens or something seemingly requiring no further explanation. Nothing is said, however, on meaning of prefixes or what $\langle\text{offsets direction}\rangle$ means. Semantics of distinct syntax elements is systematically explored in the following three sections.

3.1 Prefixes

Table 1. Prefixes.

prefix	feature	view	category
/	skew		direction prefixes
//	very skew		
,	at 45°		
+ -	⟨sign⟩ in offsets directions		
=	double centered		bond degrees
>	double right		
<	double left		
==	triple		
">	right delocalized		
"<	left delocalized		
-	long (5/3 normal)		bond length
'	short (1/3 normal)		
~	invisible		visual effects
~~	invisible label		
:	dotted		
*	bold		
^	arrow		
^*	sphenoid		
.	short invisible	C [*]	used for marks

Previously we already used prefix > to specify double bonds. The full list of prefixes is given in Table 1. All the prefixes are divided into four categories due to their effect on a bond direction, type, length, or appearance. Prefixes of different categories can be combined with each other. For example, a result of `\stree{_/:=4}` is a long skew dotted double bond at 4 o'clock:  Let us consider, however, more helpful cases of prefixes use.

Prefix . (point) is especially designed to set numbers or similar symbols of vertices. E. g., to obtain the following formula with numbered vertices  one have to enter `\stree{{H0} .01 2 .02 4 .03 2{OH} .04}` (all spaces are dispensable here). In the full agreement with the form ⟨prefix⟩⟨direction⟩⟨suffix⟩ a construction like .01 means the following: put a *short invisible* bond at 0 hours and set label “1”. Meantime, use of this

prefix has two features: 1) a current vertex is not changed, and 2) a direction preceded by a point *must* be followed by a label. Moreover, *any* character at this place is necessarily regarded as a label. It allows not to take, say, digit “1” in figure brackets without risk of missing it with a 1-hour direction.

Symbols of vertices are usually smaller than other characters. A corresponding font definition is hold in macro `\numfnt` which normally equals to `\small`. By changing it to, say, `\mit` one can further mark vertices with math symbols just writing something like `.0a` (evidently, a font for every label can be specified also explicitly, e. g., `.0{\bf 9}`).

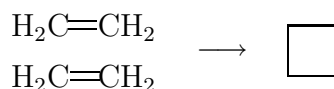
Three other prefixes were used to typeset the dummy stereoformula in this paragraph: `\stree{:1 :3 :5 *^7 **9 *^11}` (a result of a clockwise traversal starting from the left vertex). Here the use of colon `:` to draw dotted bonds is quite transparent. It is worth to note the same combination `*^` to produce first time a widening sphenoid bond and second time — a narrowing one according to a previous bond thickness. Such a program acumen is welcome until one need to explicitly specify a bond shape. The simplest way to get control over it is to insert a void bond of a complementary thickness. Thus, the isolated narrowing bond — is input as `\stree{**17 *^3}`, where 17 means an “unexistent” direction corresponding to a void bond of a null length.



Please note also the lower bond to be very bold (input with two `*`). Merely bold bonds (introduced with one `*`) are twice as thin, and doubly bolder than common ones. The normal thickness (0.4 pt) is enough for paper documents. However, such output might look too pale on a transparent film. To obtain a more bold formula one may either use `*` before every bond or say `\defwmode=1` in the beginning of a document (`\defwmode=0` returns the normal thickness).

Prefix `'` (apostrophe) allows to input short “dashes” usually denoting free valences in fragments. For example, `\stree{'8'0 4'6 2N '0'4}` gives formula . A short bond, like short invisible one, do not alter a current vertex.

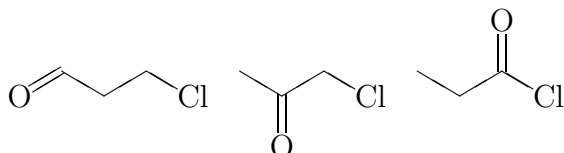
A pure invisibility can be achieved by prefix `~` (a tilde). Invisible bonds well conform to unconnected formulas. E. g., reaction scheme



can be input as

```
\stree{{H$_2$C}=3{CH$_2$} ~6{CH$_2$}=9{H$_2$C}}
\quad\longrightarrow\quad \stree{3690}
```

The invisibility is also helpful for alignment purposes. The main idea is to poise beetling substituents with their invisible counterparts. Let the following three isomers be placed in a row:



Since formulas base lines pass through their geometric centers, the right decision is

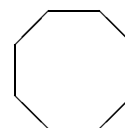
```
\stree{0 =2 [~0] 4 [~6] 2 4{Cl}} \quad
\stree{4 6\0 2 ~~0\0 4{Cl}} \quad
\stree{4 ~~6\0 2 0\0 4{Cl}}"
```

A small explanation is required here. First, the new command `\0` is intended for a carbonyl group. Second, two tildes at once in clauses `~~0\0` and `~~6\0` extend invisibility to both a bond and a label which ends a bond (one tilde would leave hanging “O”).

Thus the combined use of prefixes gives a variety of useful effects within a limited set of tokens. Absence of digits and letters in this set allows to easily distinguish prefixes from other syntax elements. Only tokens `+` and `-` do not suit this picture: they are used to set `<offset directions>` whose meaning will be discussed in the following section.

3.2 Directions

Let us systematically consider all different ways to specify bond directions. In addition to twelve *straight* directions (agreeing with numbers of hours) one can derive about twice as more of them with the aid of prefixes `/` or `,` (see fig. 1). A comma applied to one of the four numbers (2, 5, 8, or 11) produces four *median* directions at 45° to the horizon (they usually appear in 8-member cycles like one in this paragraph).



The other prefix `/` (a slash) can precede all non-orthogonal directions making them to decline toward the nearest coordinate axis by $\simeq 10^\circ$. These

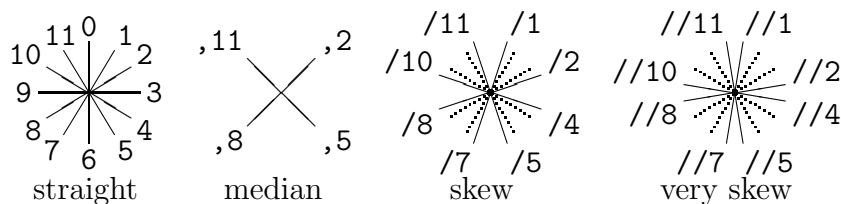
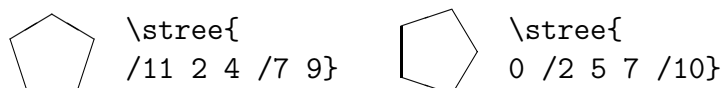



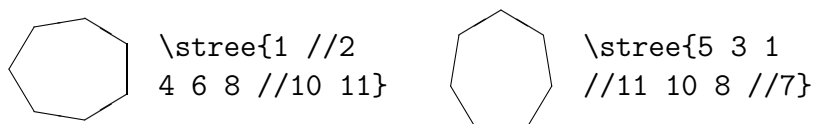
Figure 1. Bond directions (dotted lines show straight directions for a comparison with skew ones).

directions (let them be called *skew*) well conform to 5-membered cycles. A pentagon is usually drawn with one strictly vertical or horizontal side. If two bonds adjacent to that side are made skew, then the figure becomes almost right:

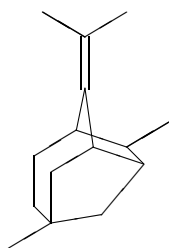


An irregular 5-membered ring  may also appear in formulas. Here are no skew directions, but one bond is 2/3 longer than others. The extra length is set by prefix `_` (an underscore), and the whole formula is input as `{\bondlen3.5mm\stree{0 _3 6 8 10}}` (here assignment `\bondlen3.5mm` answers for the reduced size of the formula, while the figure brackets bound the diminution).

To look decent, 7-membered rings require another eight directions which lie even closer than skew ones to coordinate axes. These directions hold notation of *very skew* and require one more slash. Their use is illustrated below:



Thus twelve numbers 0–11 optionally prefixed with `,` or `/` can compose $12 + 4 + 8 + 8 = 32$ different directions. Although it seems to cover a majority of formulas, in some cases a more general device is required. Due to it a direction is set by means of *two* numbers denoting horizontal and vertical bond offsets. The offset unit (called *a quad*) equals 1/6 of the default bond length. Each offset is a cardinal number starting with a *binding* sign `+` or `-` (the sign allows not to miss offsets from numbers of hours) and finishing by some non-digit. Let us comment the following example of offset directions use:



```
\stree{202/4-5-6//8-2+2
02[+1+5[=01042]+1-7]/428+2-2}
```

Having a figure of a formula is always helpful. Moreover, if it contains bonds of nonstandard length and/or direction, one should prefer a grid paper, one quad per a grid cell. Meanwhile preparing of such a figure requires a bit of skill, the further description in `\stree` is rather a mechanical procedure. One should only keep in mind that bonds with the short to long side proportion 0 : 6 and 3 : 5 correspond to straight, 2 : 6 — to skew, and 1 : 6 — very skew directions and thus can be specified via numbers of hours.

Let us start from the left methyl group and traverse the adjacent 5-membered ring clockwise. The first three bonds on this way are straight (at 2, 0, and 2 hours), the next one is skew at 4 hours. The next bond (5 quads left and 6 down) does not conform to any “clock” proportion and thus is wrote as -5-6. For the last bond (6 left and 1 down) we use notation //8, though -6-1 is also acceptable. The next object is the distorted hexagon around the just completed ring. Here sequence -2+2 02 leads to the bridge across the two rings (the bridge description is given in square brackets). The remained path to the first ring is described by /428+2-2, where pair 28 corresponds to the methyl group (a more general way [2] fits too).

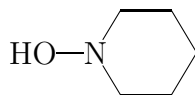
In this example all optional spaces were omitted. The single space is required only between -2+2 and 02[. Indeed, with no space -2+202 would mean 2 quads down and 202 right — a very long bond! The necessity to bind numbers is not the only specialty of offset directions. They also can not be used before commands which draw more than one bond (an explanation of this feature is leaved on section “commands”).

3.3 Labels

However, before coming to commands one should stay on simpler suffixes, i. e., labels. Let us repeat the most significant rules. If a bond leads to a labelled vertex, then this label must be wrote immediately after the bond direction.

No spaces are allowed before a label or a command¹. If a label consists of many characters, or is a digit, or any character reserved for prefixes, then it must be taken in figure brackets. If a label is just a letter (e. g., C, N, O, H) then brackets are needless (however it would not be an error to put them).

A vertex label requires all adjacent bonds to become shorter to avoid the mutual superposition. The main duty of `stree`_{TeX} just lies in proper calculation of label dimensions to update coordinates of lines which depict bonds. Because `TeX` has no arrays, only one current label (its metrics) is remembered each moment. A complete traversal of a structure may require some vertices to be met twice. Each time a vertex is met, its label has to be specified again. E. g., if formula



is traversed from vertex OH to N and along the ring back to N, then the nitrogen label should be set twice: `\stree{{HO}3N 1 3 5 7 9 11N}`. Absence of last N would result in crossing the first label N by the last 11-hours bond (since the fact of a label presence here is already forgotten). Meantime, square brackets allow almost always to avoid plural coming at a label. In our case, for example, the following construction will do: `\stree{{HO}3N [5] 1 3 5 7 9}`. Everything inside square brackets passes as if unnoticed: the closing bracket makes vertex N to be reminded so that the next 1-hour bond is drawn correctly.

A careful reader may note many formulas (e. g., the last one) to begin just from a label instead of a prefix or a direction as the formal syntax implies. Here a tiny trick works: number 17 (i. e., a void direction) is always inserted at the very beginning of an argument of `\stree` before its processing. If an argument begins with a prefix or a direction, then 17 does nothing, otherwise `17<label>` will set this `<label>` in a current point.

Another formal syntax violation belongs to justification of long labels. Evidently a label with more than one character allows diverse dispositions relative a vertex center. The default rule states: if a bond goes left (i. e., has negative horizontal offset) then the vertex center coincides with the center of the most right character, otherwise (a bond is vertical or goes right) — the

¹Since neither label nor command can start with a digit, this rule could not contradict with the necessity to separate numbers of quads or digit 1 denoting a 1-hour direction.

left one. Although it works perfectly with most terminal moieties, “internal” labels may require additional devices.

For example, the habitual clockwise traversal of the pyrrole structure would imply label NH to be justified by H instead of N as it should be. The default orientation may be altered by prefix ‘ (back apostrophe): `\stree{>1 3/>5 8‘{NH} 10}`. Since ‘ controls a behavior of a label (on contrary with the others answering for bonds), this prefix is the only exception allowed to stay before a label (of course, construction `8‘{NH}` is also true though not as logical as `8‘{NH}`).

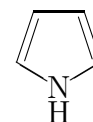


pyrrole

But let us return to the pyrrole. One may find the centered label (in this paragraph) to be more appropriate. Here the vertex center coincides with the center of the whole label, that is achieved by doubling the same prefix: `8‘‘{NH}`. However, one can get more fine effects by means of \TeX boxes, kerns, and glue (the next paragraph implies a reader to be familiar with these things).



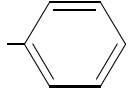


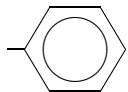


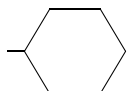

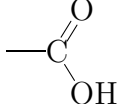

Indeed, both pyrrole formulas are not perfect, and one may try the vertical form of label NH by means of the following long construction: `\stree{>1 3/>5 8‘‘{\vtop{\baselineskip=0pt \hbox{N}\hbox{H}}}} 10}`. Let us comment it. Two ‘‘ before the label prevent an “automatic” alignment by some marginal token — evidently it is undesirable here. \TeX primitive `\vtop` makes a vertical box aligned by the topmost of contained boxes. In our case base lines of `\hbox{N}` and whole `\vtop` coincide, or, in other words, the height of `\vtop` equals to the height of `\hbox{N}`. As `stree\TeX` ignores a box depth while setting a label, the vertex center will just coincide with the center of N. Without assignment `\baselineskip=0pt` N and H would stand from each other as far as lines in a common paragraph. Funny, but almost the same result is achievable via a short invisible bond at 3 quads down: `\stree{>1 3/>5 8N [~-0-3H] 10}` (however, in this case a gap between N and H is dependable on fonts used for labels).



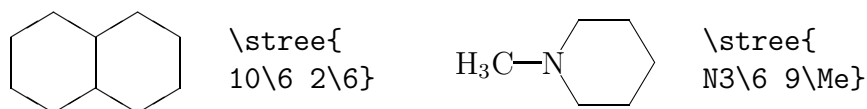
3.4 Commands

A command may substitute a label in the role of a suffix. We mean a regular \TeX command composed from signal character `\` and either one non-letter or a sequence of several letters. Specific commands may be used *inside* command `\stree` to facilitate input of some fragments (see Table 3). All the

Table 3. The available commands.

<code>\ph</code>		<code>\tbu</code>		<code>\O</code> <code>\CO</code>	
<code>\pho</code>		<code>\tbx</code>		<code>\OH</code>	
<code>\six</code> <code>\6</code>		<code>\ip</code>		<code>\COOH</code>	
		<code>\Me</code>			

available commands do not change a current vertex (though it is not a rule to obey). The reason of such behavior is evident for one-valence moieties, constituting columns 2 and 3 in Table 3. Cyclic fragments allow more flexible use. Terminal cycles require an additional bond before a command in the same direction (e. g., biphenyl is input as `\stree{9\ph 3 3\ph}`). It might seem not very comfortable unless it allowed to easy input condensed cycles and simple heterocycles:



A fragment is drawn in a direction which precedes a command (all figures in table 3 correspond to 3 o'clock). Sometimes a direction affects a label view or a whole fragment "chirality". An example is command `\COOH` which always places the carbonyl oxygen on top (try to ascertain in it). The definition of `\COOH` worth to be considered fully:

```
\def\COOH#1{{ % argument #1 = direction in hours
  \ang=#1      % remember the initial direction
  \b\ang[C]    % draw a bond ending with label C
  \ifnum\ang>6 % if more than 6 hours then
```

```

    \rot2          % turn 2 hours right,
    {\bt2\b\ang[0]} % put the carbonyl =0,
    \rot8          % turn 8 hours right (= 4 left),
    \OH\ang        % draw -OH
\else            % otherwise
    \rot2          % turn 2 hours right,
    \OH\ang        % draw -OH,
    \rot8          % turn 8 hours right,
    {\bt2\b\ang[0]} % draw =0
\fi}}
\def\OH#1{{\ifnum#1>6\b{#1}[HO]\else\b{#1}[OH]\fi}}

```

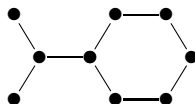
Let us explain three new commands used in this listing. The most important one is `\b` which does draw a bond in a given direction (the first binding parameter) and set an optional label taken in square brackets. Use of variable `\ang` instead of an absolute value makes a description dependable on an initial direction handed via the first parameter. After assignment `\ang=#1` this variable undergoes only relative increment corresponding to a rotation on a cardinal number of hours. Use of special command `\rot` for this purpose guarantees the resulting value to be kept within 11 hours. Evidently, that simple arithmetic is applicable only to directions expressed in numbers of hours. It explains why offset directions are not allowed before commands drawing more than one bond.

Command `\bt2` changes the character of a bond to double centered (equivalent to the action of prefix `=`). Other bond attributes (visibility, thickness, etc.) can be altered too by means of some control variables which however are not listed here. These specific data, if required, can be obtained from the source file `stree.sty`.

Additional figure brackets in definitions of `\COOH` and `\OH` involve the common grouping mechanism of T_EX to restore a current vertex after the command action. Let us also note a simpler command `\OH` to be used inside `\COOH`. The logic of `\OH` itself is limited to the choice of either label OH or HO depending on a bond direction.

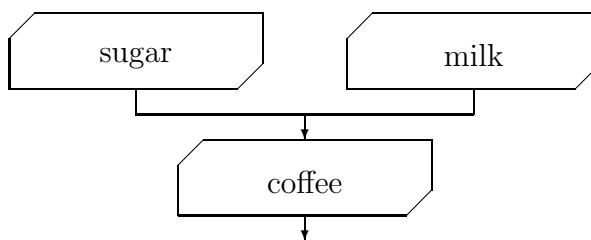
The set of commands to draw fragments may seem poor and inrepresentative. Its further extension, however, is in question, since the description on the level of bonds and labels is laconic enough to expect big advantages from broad use of internal commands. The optimal case assumes a few simple commands, easy to be used and kept in mind. It does not mean, however,

that defining of new commands is always senseless. So, authors of math papers often draw graphs with “shot” vertexes, e. g.



Bearing in mind TeX notation \bullet , one might honestly input the graph as \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet ... but there is no better way: to define command \bullet for setting a bullet in a given direction ($\def\bullet\{#\1\}\{\b{#\1}\{\bullet\}$) and then describe everything much shorter as $\stree{\bullet\bullet\bullet\bullet\bullet\bullet\bullet\bullet}$.

Indeed, suffix commands may have an additional parameter which must immediately follow a command and be in figure brackets. Accidentally this feature can be applied for input of flow charts composed from standard polygons, e. g.,



After defining a polygon by means of command \com

```
\def\(#1,#2){\sx#1\sy#2\ab30[]} % draw a bond offset (#1,#2)
\def\com#1[#2]{\((15,0)\(0,-6)\(-3,-3)\(-27,0)\(0,6)\(3,3)
\((12,0)\bmode-1 \dmode2 \sy-5 \ab30[#2]\bmode-1 \((0,-4)}
```

one can input the whole scheme as kind of a chemical formula:

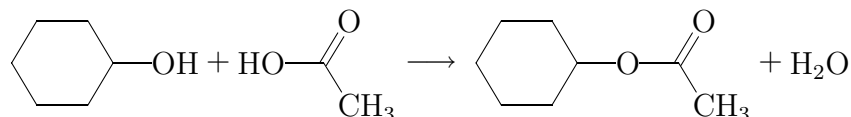
```
\stree{0\com{sugar}-0-3+20-0
[+20-0-0+3~-0+9 0\com{milk}]
::-0-3 0\com{coffee}::-0-3}
```

Let us make necessary comments. Variables \sx and \sy contain offsets used for a bond if its direction equals a conditional value of 30 hours. Assignment $\bmode-1$ makes a bond invisible while $\dmode2$ answers for a centered label (equivalent to ‘‘).

This example completes the systematic consideration of all the components in a bond description. Now a bit else remains: how to use \stree inside other L^AT_EX constructions, i. e., alignment modes.

4 Alignment

By default, `\stree` with the aid of \LaTeX environment `picture` produces `\hbox` whose height slightly (by `1ex`) exceeds its depth. This small difference makes structures to conform with math symbols in displayed formulas. E. g., the following equation

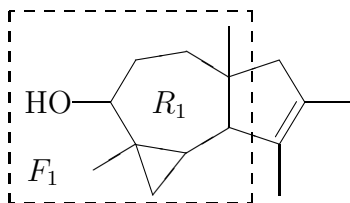


is input just as

```
\stree{9\63\OH}+\stree{9\OH1\05\Me}\longrightarrow
\stree{9\63031\05\Me}+\stree{\H$_2$O}
```

Vertical alignment may also be specified explicitly with the aid of a token in square brackets put immediately after `\stree`. In agreement with the \LaTeX conventions `[t]` means top alignment and `[b]` — bottom (a centered box is obtained by default, making use of `[c]` obsolete). Parameter `[u]` (no alignment) leads to a box with null dimensions at the place of a starting vertex. This is useful if `\stree` itself is a part of another picture whose elements are bind to vertices. Since all vertices are situated in nodes of a one quad sized grid, knowing of exact coordinates of one vertex allows all the others to be easy calculated. Use of `[u]` guarantees this knowledge for a starting vertex, whereas other parameters — do not, since a box produced by `\stree` in these cases accurately bounds everything found in a formula: labels, marks, etc.

On the other hand, variant `[u]` makes a user himself to monitor sizes of a formula. There is still a way deprived of this drawback: instead of `\stree{ <description> }` one can write `\begstr \tree{ <description> } \endstr`. Before `\endstr` or after `\begstr` any \LaTeX commands for `\picture` environment are allowed as if coordinates (0,0) belong to a starting vertex and unit of measure `\unitlength` equals the standard bond length `\bondlen`. In the following example this method is used to mark a fragment with a dashed box:

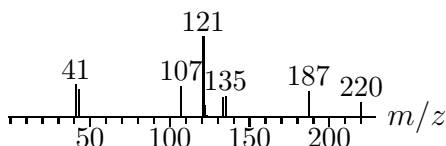


This figure is input as

```
\begstr
\tree{9\OH1//2[~/7{\$R_1\$}]4[0~2] [6]/2539>760/108
 [-4-5/118~9{\$F_1\$}]/1011}
\put(-2,-2){\dashbox{0.2}(4.8,3.8){}}
\endstr
```

5 Additional features

Chemical character of this work allows us to note also the ability to input mass and ^{13}C NMR spectra with `streeTeX`. For example, the spectrum



is obtained in result of

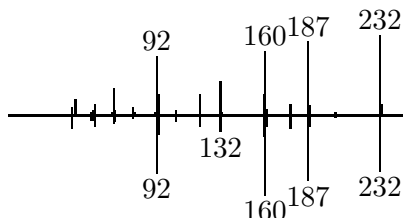
```
\massp{220-18!187-33!135-26!133-25,121-100!14,2,
 107-38!43-35,41-41!}
```

Here is just a sequence of numbers divided by punctuation marks: a mass number is ended by a dash, a relative intensity (in percents) — by either comma or exclamation. The last makes an m/z value to be set over a peak. Several intensities following one mass number (e. g., 121-100!14,2,) correspond to neighbor peaks with increasing m/z . A comma may be omitted if it is the last token in a `\massp` argument.

The four parameters control a mass spectrum view:

- `\mzlen` — the horizontal distance corresponding to 1 m/z (default `\mzlen=0.6pt`),
- `\imax` — the height of the maximal peak in units of `\mzlen` (default `\imax=50`),
- `\numfnt` — the font used to set numbers on a figure (default `\def\numfnt{\small}`),
- `\msdir` — the direction of peaks.

The last parameter is rarely used. Default `\msdir=1` means the common direction — upward; any other value make peaks to be oriented downward. The last case may be useful for visual comparison of two spectra:

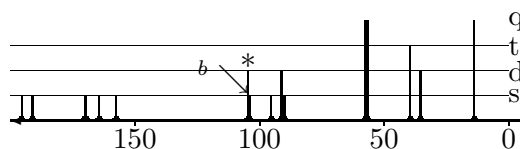


This figure is input in the following way:

```
\begms
\mass{39-11,0,21,51-5,7,15,64-5,35,7,77-10,5,91-4,74!27,104-6,
119-26,132-42,4,159-26,81!8,176-15,187-93!12,204-4,232-100!14}
\put(-1,0){\line(1,0){250}}\msdir=-1
\mass{39-16,51-7,6,15,65-11,8,77-5,92-73!24,104-9,119-15,
132-21!159-26,100!14,176-16,187-83!14,204-3,232-70!}\vpos=1
\endms
```

By analogy with `\stree`, instead of `\mass{ <argument> }` one can write `\begms \mass{ <argument> } \scale1{< > } \endms` and insert any commands from the \LaTeX `picture` environment after `\begms` or before `\endms`. In our example a common scale drawn by `\scale1{< > }` is substituted by the plain horizontal (its length is the rounded maximal mass number in the spectrum), and then another spectrum is drawn contrariwise since it is preceded by `\msdir=-1`. Finally, assignment `\vpos=1` switches on the bottom alignment (i. e., the box depth equals zero). Though command `\massp` itself presumes the common alignment control with the aid of parameters `[b]`, `[c]`, or `[t]`, on the lower level there acts variable `\vpos` (it is true also for the decomposition of `\stree` onto `\begstr`, `\tree`, and `\endstr`). Value 0 means center alignment (same as parameter `[c]`), 1 — bottom (`[b]`), 2 — top (`[t]`), and 3 — no alignment (valid only for structural formulas and corresponds to `[u]`).

`stree \LaTeX` also allows to input simple figures of ^{13}C NMR spectra like the following (a peak height reflects its multiplicity):



The corresponding text is:

```
\cnmrs{2000}{1912s1645s354d1041s{\llap{\b\searrow}}}%  
901s138q1576s1703s575q912d1047d{*$}}%  
566q1696s1956s571q953s395t}
```

Command `\cnmrs` has two binding parameters: first is the scale length (in p. p. m. * 10), and second — a spectrum description which contains for each signal its chemical shift value (multiplied by 10) and multiplicity, where **s** means singlet, **d** — doublet, **t** — triplet, and **q** — quartet. A peak label may be set in figure brackets after a multiplicity token (in our case construction `\llap{\b\searrow}` is used to shift the label left and prevent its interference with the neighbor doublet marked by a star).

Both kinds of spectra are input in quite a similar way: as a sequence of numbers with delimiters. However, there are small differences. So, long command `\massp` can be broken in strings at any places, though in `\cnmrs` end of line should be commented by `%`. Besides, in `\cnmrs` the scale (and so whole figure) length is set implicitly, while for a mass spectrum the total length depends on a largest m/z value (to obtain a mass spectrum with a given length one could just specify a void peak with null intensity and corresponding m/z).

6 Conclusion

An apparent question might arise during acquaintance with this work: does somebody need to study another “bird language” to input chemical formulas if a plenty of convenient programs for visual editing exist? Essentially it is an old dispute *pro et contra* a command interface vs. graphic one. Professional programmers know very well that a command language provides a faster and more flexible control though it requires more time for studying. Indeed there is no contradiction between these modes of interaction, and the optimal case should include both abilities. Anyhow, in preparation of scientific manuscripts words do play the main role, i. e., a writer basically remains in the verbal mode of thinking. If numerous chemical formulas in a text are input by a graphic editor then a writer will frequently switch from one program to another. Even if the switching is fast and elegant (a case rarely met in practice), the change of the interface mode will still require some psychological adaptation. In this case an ability to input a formula

just in text without tackling a mouse and becoming a designer could save you a lot of time and mind energy.

References

Knuth D. (1984) *The T_EXbook*. Addison-Wesley, Reading.

Lamport L. (1984) *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading.

Ramek M. (1990) *T_EX: Applications, Uses, Methods*. ed. Clark M., p. 227. Ellis Horwood, London.

Haas R. T. & O’Kane K. C. (1987) *Comput. Chem.*, **11**, 251

Fujita S. (1994) *Comput. Chem.*, **18**, 109