

X_YM_TE_X for Typesetting Chemical Structural Formulas. Enhanced Functions for Version 2.00

Shinsaku Fujita

Department of Chemistry and Materials Technology,
Kyoto Institute of Technology,
Matsugasaki, Sakyo, Kyoto, 606-8585 Japan

December 25, 1998 (Version 2.00)
(revised March 20, 1999 and June 20, 2001)
(typeset by T_EX/L^AT_EX Compatible Mode: December 20, 2004)

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 7 |
| 1.1 | History | 7 |
| 1.1.1 | Version 1.00 (1993) | 7 |
| 1.1.2 | Version 1.01 (1996) | 7 |
| 1.1.3 | Version 1.02 (1998, not released) | 8 |
| 1.2 | Version 2.00 (1998) | 10 |
| 2 | Bond Modifiers Added | 13 |
| 2.1 | Alternative Bond Modifiers for Up and Down Bonds | 13 |
| 2.2 | Bond Modifiers for Ring Fusion | 13 |
| 3 | Nested-Substituent Method | 15 |
| 3.1 | Introduction | 15 |
| 3.2 | “yl”-Functions | 15 |
| 3.3 | Nested “yl”-functions | 19 |
| 3.4 | Remarks | 24 |
| 3.4.1 | Drawing Domains | 24 |
| 3.4.2 | Reference Points | 26 |
| 4 | Linking Units | 27 |
| 4.1 | \ryl command | 27 |
| 4.2 | \lyl command | 31 |
| 4.3 | Nested \ryl and \lyl commands | 32 |
| 4.4 | \divalenth command | 35 |
| 4.5 | Remarks | 37 |
| 5 | Ring Fusion | 39 |
| 5.1 | Ring Fusion on Carbocyclic Compounds | 39 |
| 5.1.1 | Designation of Fused Bonds | 39 |
| 5.2 | Ring Fusion on Heterocyclic Compounds | 41 |
| 5.3 | Nested Ring Fusion | 43 |
| 5.4 | Remarks | 47 |
| 5.4.1 | OPT Arguments | 47 |
| 5.4.2 | X _Y MT _E X Warning | 47 |
| 6 | Fusing Units | 49 |
| 6.1 | Six-membered Fusing Units | 49 |
| 6.1.1 | Vertical Units of Normal and Inverse Types | 49 |
| 6.1.2 | Horizontal Units of Normal and Inverse Types | 51 |
| 6.2 | Five-membered Fusing Units | 52 |
| 6.2.1 | Vertical Units of Normal and Inverse Types | 52 |

| | | |
|-----------|---|------------|
| 6.2.2 | Horizontal Units of Normal and Inverse Types | 54 |
| 6.3 | Four-membered Fusing Units | 55 |
| 6.4 | Three-membered Fusing Units | 56 |
| 6.4.1 | Vertical Units of Normal and Inverse Types | 56 |
| 6.4.2 | Horizontal Units of Normal and Inverse Types | 57 |
| 7 | Spiro Rings | 59 |
| 7.1 | General Conventions for Spiro-Ring Attachment | 59 |
| 7.2 | Several Examples | 61 |
| 7.3 | Multi-Spiro Derivatives | 65 |
| 7.4 | Atom Replacement | 66 |
| 8 | Added Commands | 69 |
| 8.1 | Six-six Fused Carbocycles | 69 |
| 8.1.1 | Vertical-Bottom Forms of Decaline Derivatives | 69 |
| 8.1.2 | Vertical-Top Forms of Decaline Derivatives | 70 |
| 8.2 | Six-six Fused Heterocycles | 71 |
| 8.2.1 | Vertical-Bottom Forms | 71 |
| 8.2.2 | Vertical-Top Forms | 72 |
| 8.3 | Three-Membered Carbocycles | 73 |
| 8.4 | Three-Membered Heterocycles | 75 |
| 8.5 | Aliphatic Moieties | 78 |
| 8.5.1 | Trigonal Units | 78 |
| 8.5.2 | Ethylenes | 78 |
| 9 | Zigzag Polymethylene Skeletons | 81 |
| 9.1 | Dimethylenes | 81 |
| 9.2 | Trimethylenes | 82 |
| 9.3 | Tetramethylenes | 83 |
| 9.4 | Pentamethylenes | 84 |
| 9.5 | Hexamethylenes | 85 |
| 9.6 | Heptamethylenes | 86 |
| 9.7 | Octamethylenes | 87 |
| 9.8 | Nonamethylenes | 88 |
| 9.9 | Decamethylenes | 89 |
| 9.10 | Longer Polymethylene Chains | 90 |
| 9.11 | Cisoid Tetramethylenes | 90 |
| 9.12 | Ring Fusion to Polymethylenes | 91 |
| 9.13 | Ring Replacement to Polymethylenes | 92 |
| 9.14 | Branched Chains | 93 |
| 10 | Enhanced Functions of Commands for General Use | 95 |
| 10.1 | Expanded Format | 95 |
| 10.2 | Boldfaced and Dotted Bonds | 95 |
| 10.3 | Bond Deletion | 97 |
| 11 | Enhanced Functions of Commands for Ring Fusion | 101 |
| 11.1 | Expanded Format | 101 |
| 11.2 | Boldfaced and Dotted Bonds | 101 |
| 11.3 | Bond Deletion | 102 |
| 11.3.1 | Larger Rings from Two or More Three-Membered Rings | 102 |
| 11.3.2 | Further Rings | 103 |

| | |
|---|------------|
| 12 Reaction Schemes | 105 |
| 12.1 Compound Numbers | 105 |
| 12.2 Reaction Arrows | 107 |
| 12.3 Display Formulas and Tabular Schemes | 108 |

Chapter 1

Introduction

1.1 History

1.1.1 Version 1.00 (1993)

The first version of the $\hat{\text{X}}\hat{\text{M}}\hat{\text{T}}\hat{\text{E}}\hat{\text{X}}$ system (version 1.00, 1993) with a detailed on-line manual has been deposited to NIFTY-Serve archives (FPRINT library No. 7) by the author[1] and to the CTAN by volunteers[2]. The articles on the construction and usage of $\hat{\text{X}}\hat{\text{M}}\hat{\text{T}}\hat{\text{E}}\hat{\text{X}}$ have appeared in Ref. [3, 4]. Although the packages (style files) of the $\hat{\text{X}}\hat{\text{M}}\hat{\text{T}}\hat{\text{E}}\hat{\text{X}}$ system have originally aimed at using under the $\text{\LaTeX}2.09$ system, they also work effectively under the $\text{\LaTeX}2_{\epsilon}$ system [5, 6] without any changes. Thus, what you have to do is to rewrite a top statement for $\text{\LaTeX}2.09$ such as

```
\documentstyle[epic,carom,hetarom]{article}
```

into the counterpart for $\text{\LaTeX}2_{\epsilon}$, *e.g.*,

```
\documentclass{article}
\usepackage{epic,carom,hetarom}
```

1.1.2 Version 1.01 (1996)

The Version 1.01 of the $\hat{\text{X}}\hat{\text{M}}\hat{\text{T}}\hat{\text{E}}\hat{\text{X}}$ system has been released in 1996, when the system with a detailed on-line manual was deposited to NIFTY-Serve archives (FPRINT library No. 7) by the author [7]. The system is now available from Fujita's homepage [8] via internet or from a CD-ROM that is attached to the reference manual published in 1997 [9].¹

The purpose of version 1.01 is the updating of $\hat{\text{X}}\hat{\text{M}}\hat{\text{T}}\hat{\text{E}}\hat{\text{X}}$ to meet the $\text{\LaTeX}2_{\epsilon}$ way of preparing packages (option style files). The following items have been revised or added for encouraging the $\hat{\text{X}}\hat{\text{M}}\hat{\text{T}}\hat{\text{E}}\hat{\text{X}}$ users to write articles of chemical fields.

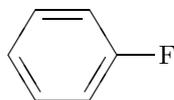
1. Each of the old sty files of $\hat{\text{X}}\hat{\text{M}}\hat{\text{T}}\hat{\text{E}}\hat{\text{X}}$ has been rewritten into a dtx file, from which we have prepared a new sty file by using the `docstrip` utility of $\text{\LaTeX}2_{\epsilon}$. If you want to obtain the document of each source file, you may apply $\text{\LaTeX}2_{\epsilon}$ to the corresponding drv file, which has also been prepared from the dtx file by using the `docstrip` utility.
2. Macros for drawing chair-form cyclohexanes and for drawing adamantanes of an alternative type have been added.
3. Macros for drawing polymers have been added.

¹The basic items described in the $\hat{\text{X}}\hat{\text{M}}\hat{\text{T}}\hat{\text{E}}\hat{\text{X}}\text{book}$ are common and applied also in Version 2.00. Please refer to the $\hat{\text{X}}\hat{\text{M}}\hat{\text{T}}\hat{\text{E}}\hat{\text{X}}\text{book}$, when they are used without explanations in this manual.

4. The package `chemist.sty`, which was originally prepared for [13], has been rewritten into a `dtx` file and added to $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ as a new component. This package enables us to use various functions such as
- the numbering and cross-reference of chemical compounds and derivatives,
 - various arrows of fixed and flexible length for chemical equations,
 - 'chem' version and chemical environments for describing chemical equations, and
 - various box-preparing macros for chemical or general use.

1.1.3 Version 1.02 (1998, not released)

The Version 1.02 of $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ has been devoted to the development of the nested-substitution method, which simplifies the coding of $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ commands. In $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ version 1.01, each substituent is assumed to be rather small so that it can be specified by means of a substitution list "SUBSLIST". For example, 1-fluorobenzene,



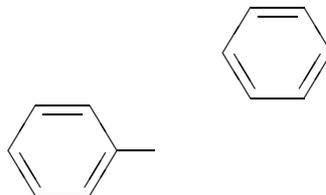
is drawn by the following code:

```
\bzdrh{4==F}
```

To draw a substituent with a complicated structure, a designation of the same line produces an insufficient result. Thus, if we simply write the code

```
\bzdrh{4==\bzdrh{}}
```

to draw a biphenyl structure, we have a separate structure as follows:



Within the scope of $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ version 1.01, such a substituent with a complicated structure can be treated by three distinct methods (see Chapters 14 and 15 of $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ book).

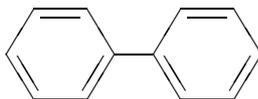
- (Method I) When we write a code `\bzdrh{4==}\bzdrh{}` to draw a biphenyl structure, we obtain an insufficient result such as



since each command has an area to draw its target structure. To remedy this situation, we can write

```
\bzdrrh{4==}\kern-33pt\bzdrrh{}
```

Then, we obtain the following structure:

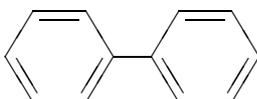


However, a more complicated adjustment is necessary to apply this method to a case in which the components of a structural formula are not linearly aligned.

2. (Method II) We can carry out the same task by using the \LaTeX `picture` environment. The code

```
\begin{picture}(1400,700)(0,0)
\put(0,0){\bzdrrh{4==}}
\put(546,0){\bzdrrh{}}
\end{picture}
```

produces the following structure:

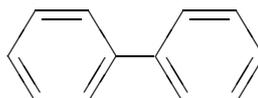


This method realizes such a complicated adjustment as mentioned above, since the `\put` is capable of putting components at arbitrary positions.

3. (Method III) In a further method of drawing the biphenyl structure, one phenyl group is regarded as a substituent of the other phenyl. These two parts can be combined by writing a code,

```
\bzdrrh{4==\kern-25pt\lower37pt\hbox to0pt{\bzdrrh{}}\hss}}
```

in which the commands `\kern` (for horizontal adjustment) and `\lower` (for vertical adjustment) are used to adjust the substitution site. Thereby, we have

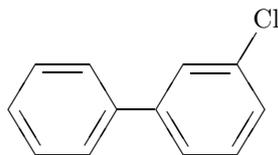


This method has a disadvantage of calculating adjustment values manually for every formula to be drawn.

These three methods are useful for drawing complicated structure. However, they have an essential disadvantage: their codes give no, or at most partial, connectivity data between parts to be combined, though such parts appear to be combined as a picture. For example, the code

```
\bzdrh{4==\kern-25pt\lower37pt\hbox to0pt{\bzdrh{3==Cl}\hss}}
```

producing

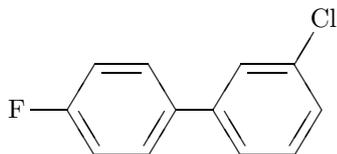


has no connectivity data at the meta position to the chlorine atom of the second benzene ring.

As clarified by the discussion in the preceding paragraphs, the $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ system should have a function to place substituents at appropriate sites without complex designation, where connectivity data are maintained during the process of drawing. The target of $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ Version 1.02 is to treat nested substitution with the automatic adjustment of substitution sites (named as the nested-substitution method). Concretely speaking, for example, such a code as

```
\bzdrh{1==F;4==\bzdrh{1==(y1);3==Cl}}
```

directly produces



where the code shows that the second benzene ring is linked to the para position of the first benzene ring at the meta position to the chlorine atom. Thus the target accomplished by the “y1”-function, as shown in this code.

1.2 Version 2.00 (1998)

The “y1”-function developed in $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ Version 1.02 is regarded as a modification of SUSBLISTs. As an extension of this methodology, BONDLISTs can be modified to treat ring fusion, since each ring fusion is considered to be a kind of substitution on a bond. In addition, ATOMLIST can also be used to treat spiro rings, since each spiro ring is a kind of atom replacement at an appropriate vertex.

To expand the scope of the $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ system, we introduce several new functions as follows.

1. Several bond modifiers are added to draw alternative up- and down-bonds as well as to treat ring fusion.
2. The “y1”-function for SUBSLISTs is further improved. The commands `\ry1` and `\ly1` are prepared to typeset intervening moieties.

3. Ring fusion is treated by adding a fusing unit to the BONDLIST of each command.
4. Several fusing units (three- to six-membered units) are developed (fusering.sty).
5. A new function for typesetting a spiro ring is introduced in each command for general use. A spiro ring is treated by ring-replacement technique, where the corresponding code is written in the ATOMLIST of each command.
6. Commands for typesetting zigzag polymethylenes are developed (methylen.sty).
7. Commands for drawing six-six fused carbocycles and heterocycles are added.
8. An optional argument SKBONDLIST is added to each command of general use for drawing bold-faced and dotted skeletal bonds.
9. An optional argument OMIT is added to each command of general use for drawing related skeletons by bond deletion.

The $\text{\X}\text{\M}\text{\T}\text{\E}\text{\X}$ system (version 2.00) consists of package files listed in Table 1.1. The package file ‘chemstr.sty’ is the basic file that is automatically read within any other package file of $\text{\X}\text{\M}\text{\T}\text{\E}\text{\X}$. It contains macros for internal use, *e.g.*, common commands for bond-setting and atom-setting. The other package files contain macros for users. These files are designed to work not only as packages for $\text{\L}\text{\A}\text{\T}\text{\E}\text{\X} 2_{\epsilon}$ but also as option style files for $\text{\L}\text{\A}\text{\T}\text{\E}\text{\X} 2.09$ (native mode).

Table 1.1: Package Files of $\text{\X}\text{\M}\text{\T}\text{\E}\text{\X}$

| package name | included functions |
|--------------|---|
| aliphat.sty | macros for drawing aliphatic compounds |
| carom.sty | macros for drawing vertical and horizontal types of carbocyclic compounds |
| lowcycle.sty | macros for drawing five-or-less-membered carbocycles. |
| ccycle.sty | macros for drawing bicyclic compounds etc. |
| hetarom.sty | macros for drawing vertical types of heterocyclic compounds |
| hetaromh.sty | macros for drawing horizontal types of heterocyclic compounds |
| hcycle.sty | macros for drawing pyranose and furanose derivatives |
| chemstr.sty | basic commands for atom- and bond-typesetting |
| locant.sty | commands for printing locant numbers |
| polymers.sty | commands for drawing polymers |
| fusering.sty | commands for drawing units for ring fusion |
| methylen.sty | commands for drawing zigzag polymethylene chains |
| xymtex.sty | a package for calling all package files |
| chemist.sty | commands for using ‘chem’ version and chemical environments |

The use of xymtex.sty calling all package files may sometimes cause the “ $\text{\T}\text{\E}\text{\X}$ capacity exceeded” error. In this case, you should call necessary packages distinctly by using the \usepackage command.

Chapter 2

Bond Modifiers Added

2.1 Alternative Bond Modifiers for Up and Down Bonds

In addition to the original bond modifiers (see the \LaTeX book), the present version of \LaTeX provides us with several bond modifiers that can be used in the argument `SUBSLIST` of each \LaTeX command. These modifiers are listed in Table 2.1 along with the original bond modifiers.

The added bond modifiers, ‘Sd’ (d for down) and ‘Su’ (u for up), designate α - and β -bonds in such an exchanged manner as the original bond modifiers, ‘SA’ and ‘SB’ designate. Figure 2.1 shows the comparison between the added bond modifiers and the original ones by using a cyclohexane skeleton (`\cyclohexanev`).

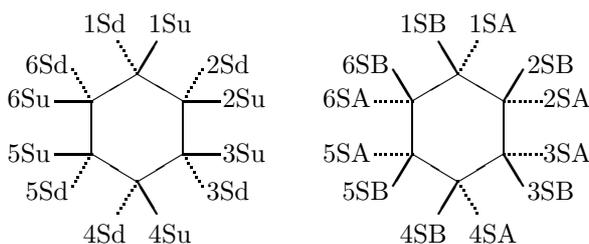


Figure 2.1: Bond Modifiers for α - and β -Bonds

2.2 Bond Modifiers for Ring Fusion

In the present version (2.00), we have added a new function for ring fusion. Since the function requires bond modifiers for designating substitution at such fused positions, we have added the modifiers, ‘FA’, ‘FB’, ‘GA’, and ‘GB’. These modifiers are illustrated in Figure 2.2

Table 2.1: Locant numbering and bond modifiers for SUBSLIST

| Bond Modifiers | Printed structures |
|--------------------------------|---|
| Original Bond Modifiers | |
| n or nS | exocyclic single bond at n -atom |
| nD | exocyclic double bond at n -atom |
| nA | alpha single bond at n -atom |
| nB | beta single bond at n -atom |
| nSa | alpha (not specified) single bond at n -atom |
| nSb | beta (not specified) single bond at n -atom |
| nSA | alpha single bond at n -atom (dotted line) |
| nSB | beta single bond at n -atom (boldface) |
| Bond Modifiers Added | |
| nSd | alpha single bond at n -atom (dotted line) with an alternative direction to nSA |
| nSu | beta single bond at n -atom (boldface) with an alternative direction to nSB |
| nFA | alpha single bond at n -atom (dotted line) for ring fusion |
| nFB | beta single bond at n -atom (boldface) for ring fusion |
| nGA | alpha single bond at n -atom (dotted line) for the other ring fusion |
| nGB | beta single bond at n -atom (boldface) for the other ring fusion |

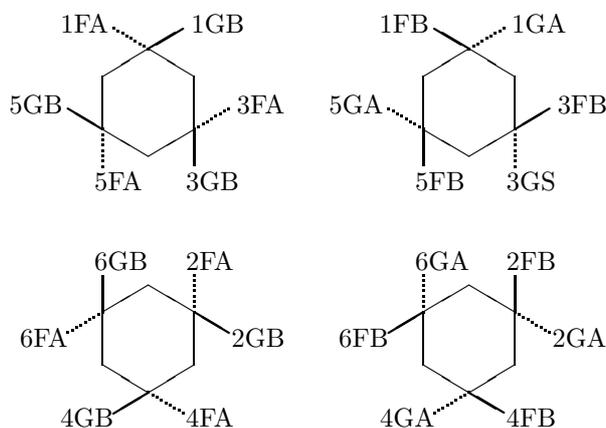


Figure 2.2: Bond Modifiers for Ring Fusion

Chapter 3

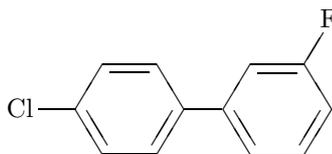
Nested-Substituent Method

3.1 Introduction

Chapter 14 (Combining Structures) and Chapter 15 (Large Substituents) of the $\text{\X}\text{\M}\text{\T}\text{\E}\text{\X}$ book have described several techniques to draw complicated formulas. Among them, the nested-substituent method is most promising. For example, the code

```
\bzdrh{1==Cl;4==\kern-25pt\lower37pt\hbox to0pt{\bzdrh{3==F}\hss}}
```

gives a combined structure,



Although the code shows the connectivity between the two phenyl groups, the following disadvantages remain:

1. The code contains no data indicating that the connection site is the meta-position concerning the fluorine atom.
2. The commands `\kern` (for horizontal adjustment) and `\lower` (for vertical adjustment) are necessary to adjust the substitution site.

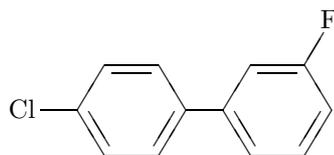
As clarified by the above examples, the main target of $\text{\X}\text{\M}\text{\T}\text{\E}\text{\X}$ Version 2.00 is to extend the nested-substituent method so that it provides a function of indicating full connectivity data as well as a function of automatic adjustment without using such commands as `\kern` and `\lower`.

3.2 “yl”-Functions

In $\text{\X}\text{\M}\text{\T}\text{\E}\text{\X}$ Version 2.00, the “yl”-function is added so as to improve the nested-substituent method. Thereby, any structure drawn by a $\text{\X}\text{\M}\text{\T}\text{\E}\text{\X}$ command (except a few special commands) can be converted into the corresponding substituent by adding the code (yl) with a locant number. The resulting code for the substituent can be added to the SUBSLIST of any other command for drawing a mother skeleton, where the final code contains the full connectivity data of the combined structure. For example, the code

```
\bzdrh{1==Cl;4==\bzdrh{1==(yl);3==F}}
```

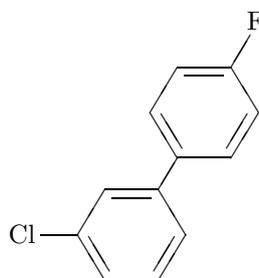
typesets the following structure,



Thus, fluorobenzene produced by the command `\bzdrh{3==F}` is converted into a substituent, i.e. 3-fluorophenyl, by adding the code `(y1)`, as shown in the code, `\bzdrh{1==(y1);3==F}`. Then, the resulting code is added to the SUBSLIST of another command `\bzdrh`.

The connectivity at the meta-position is represented by the statement `1==(y1)` of the inner code `\bzdrh{1==(y1);3==F}`. Note that the inner code `\bzdrh{1==(y1);3==F}` produces a substituent with no height and no width and that the reference point of the substituent is shifted to the point no. 1 by the `(y1)`-statement in order to link to the mother structure (the phenyl group produced by the code `\bzdrh{1==Cl;4=={\dots}}`).

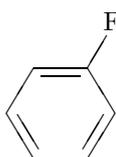
The shift of a reference point becomes clear when we examine a formula,



generated by the code,

```
\bzdrh{1==Cl;3==\bzdrh{6==(y1);3==F}}
```

The original structure of the substituent with no “y!” function is found to be



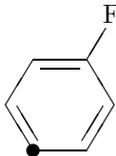
as generated by the code

```
\begin{picture}(700,800)(0,0)
\put(0,0){\bzdrh{3==F}}
\put(0,0){\circle*{50}}
\end{picture}
```

where the solid circle is the reference point. The picture shown above indicates that the reference point is different from any vertices of the benzene ring. On the other hand, the code with a “y!”-function,

```
\begin{picture}(700,800)(0,-200)
\put(0,0){\bzdrh{6==(y1);3==F}}
\put(0,0){\circle*{50}}
\end{picture}
```

typesets the following structure,

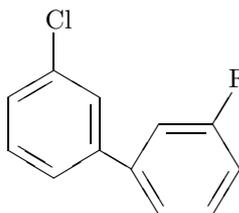


The picture shown above indicates that the reference point is shifted to the position no. 6 of the benzene ring.

The code `\bzdrh{1==(y1);3==F}` producing the substituent can be used in the argument of any structure-drawing command of \LaTeX . The following example is the one in which it is placed in the argument of a command `\bzdrv`. Thus, the code

```
\bzdrv{1==Cl;3==\bzdrh{1==(y1);3==F}}
```

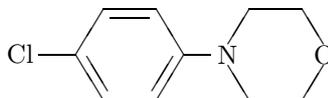
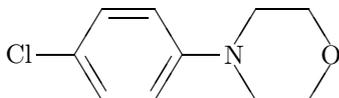
typesets the following structure,



The structural formula of 1-chloro-4-morpholinobenzene can be drawn in two different ways. The codes,

```
\bzdrh{1==Cl;4==\sixheteroh[] {1==N;4==O}{1==(y1)}}
\hskip 6cm
\sixheteroh[] {1==N;4==O}{1==\bzdrh{1==Cl;4==(y1)}}
```

produce the following formulas:

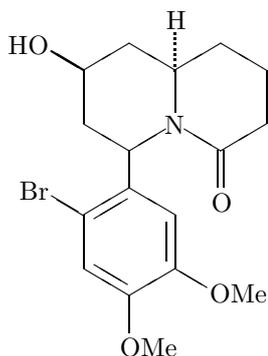


In the former code, the morpholino group is regarded as a substituent, as the name “1-chloro-4-morpholinobenzene” indicates. On the other hand, the chlorophenyl group is considered to be a substituent in the latter code so as to correspond to the name “N-(4-chlorophenyl)morpholine”.

The “yl”-function is quite versatile, as indicated by the code,

```
\decaheterov[] {4a==N}{4D==O;7B==HO;{{10}A}==H;%
5==\bzdrv{3==OMe;4==OMe;6==Br;1==(y1)}}
```

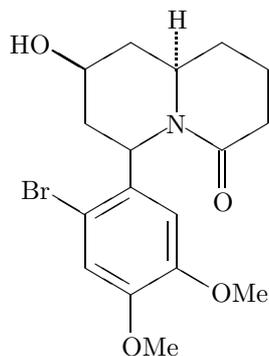
producing the following structure:



where the substituted phenyl group is regarded as a substituent. An opposite view can be realized by the code

```
\bzdrv{3==OMe;4==OMe;6==Br;%
1==\decaheterov[] {4a==N}{4D==O;7B==HO;{{10}A}==H;5==(y1)}}
```

which typesets the same structure:

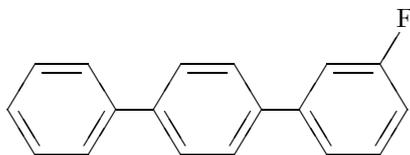


where the moiety drawn by the command `\decaheterov` is regarded as a substituent.

Two or more substituents generated by the “y1”-function can be introduced into an ATOMLIST. For example,

```
\bzdrh{1==\bzdrh{4==(y1)};4==\bzdrh{1==(y1)};3==F}}
```

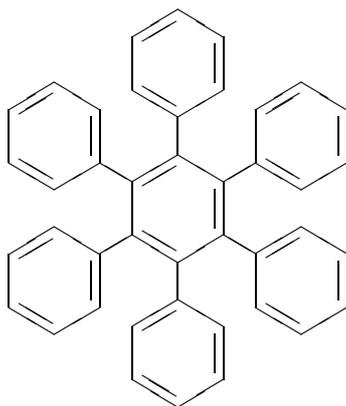
typesets the following structure,



The structural formula of hexaphenylbenzene can be drawn by this technique. Thus the code,

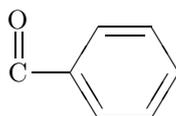
```
\bzdrv{1==\bzdrv{4==(y1)};%
2==\bzdrv{5==(y1)};3==\bzdrv{6==(y1)};%
4==\bzdrv{1==(y1)};5==\bzdrv{2==(y1)};%
6==\bzdrv{3==(y1)}}
```

generates the following formula:



3.3 Nested “yl”-functions

Two or more “yl”-functions can be nested. For example, a structure



depicted by the code,

```
\tetrahedral{0==C;1D==0;4==\bzdrh{1==(y1)}}
```

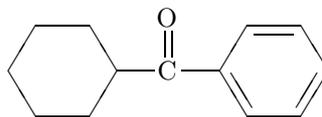
can be converted into a substituent by adding “yl”-function, as shown in the following code:

```
\tetrahedral{2==(y1);0==C;1D==0;4==\bzdrh{1==(y1)}}
```

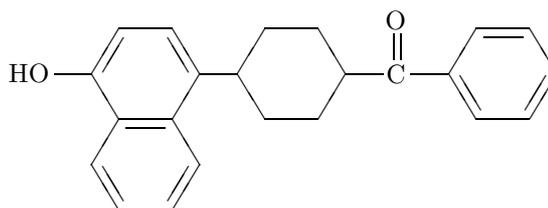
Then this substituent is nested in the SUBSLIST of the command `\cyclohexaneh` to give a code,

```
\cyclohexaneh[] {4==%
\tetrahedral{2==(y1);0==C;1D==0;4==\bzdrh{1==(y1)}}
```

Thereby we have the structural formula of benzoylcyclohexane:



The resulting structure can be further converted into a substituent by adding “yl”-function. The following example shows that the substituent is linked to the 4-position of a naphthol ring:



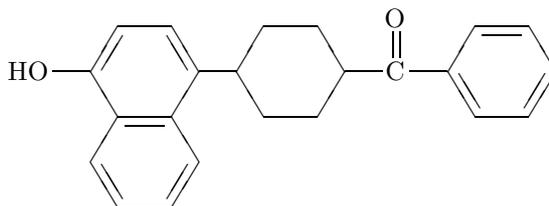
which is typeset by the triply nested code:

```
\naphdrh{1==HO;4==%
\cyclohexaneh[] {1==(y1);4==%
\tetrahedral{2==(y1);0==C;1D==O;4==\bzdrh{1==(y1)}}}}
```

The same structural formula can be drawn by regarding the 1-naphthol-4-yl group and the benzoyl group as substituents, as shown in the following code:

```
\cyclohexaneh[] {%
1==\naphdrh{1==HO;4==(y1)};%
4==\tetrahedral{2==(y1);0==C;1D==O;4==\bzdrh{1==(y1)}}}
```

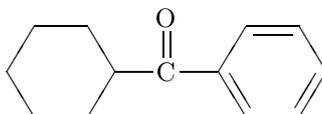
Accordingly, we have



The structure of benzoylcyclohexane can also be drawn by considering the `\tetrahedral` moiety as a mother skeleton, as shown in the code:

```
\tetrahedral{0==C;1D==O;4==\bzdrh{1==(y1)};2==\cyclohexaneh[] {4==(y1)}}
```

Thereby, we have the formula,

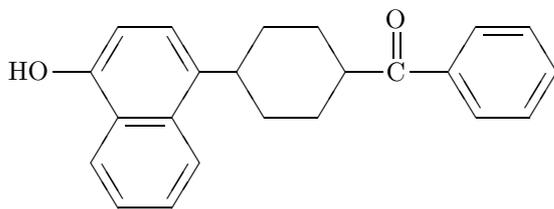


which shows that two or more substituents produced by the “yl”-function can be written in a SUBSLIST. This treatment corresponds to the alternative name of benzoylcyclohexane, i.e., cyclohexyl phenyl ketone, since the codes `\cyclohexaneh{4==(y1)}` and `\bzdrh{1==(y1)}` represent a cyclohexyl and a phenyl group, respectively.

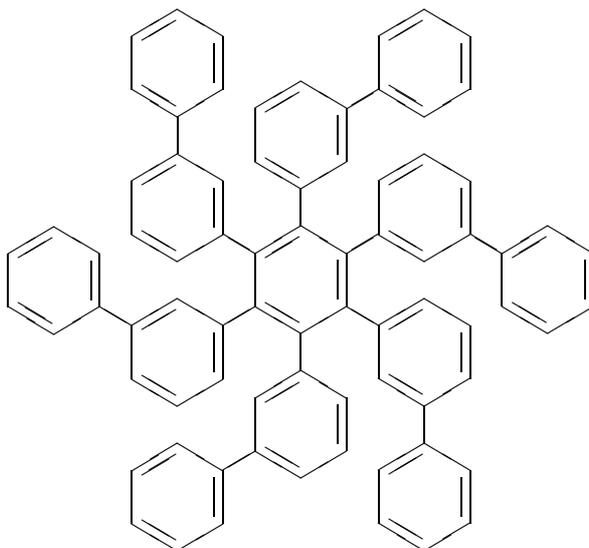
Although the structure drawn by `\cyclohexaneh` cannot be used as a substituent concerning the cyclohexane ring, the SUBSLIST of the command `\cyclohexaneh` is capable of accommodating the substituent `\naphdrh{1==HO;4==(y1)}` to give

```
\tetrahedral{0==C;1D==0;4==\bzdrh{1==(y1)}};%
2==\cyclohexaneh[] {4==(y1);1==\naphdrh{1==HO;4==(y1)}}}
```

which typesets the same structural formula:



The formula,



illustrates the more complicated structure of a code with nested “yl”-functions:

```
\bzdrv{%
1==\bzdrv{4==(y1);2==\bzdrv{5==(y1)}};%
2==\bzdrv{5==(y1);3==\bzdrv{6==(y1)}};%
3==\bzdrv{6==(y1);4==\bzdrv{1==(y1)}};%
4==\bzdrv{1==(y1);5==\bzdrv{2==(y1)}};%
5==\bzdrv{2==(y1);6==\bzdrv{3==(y1)}};%
6==\bzdrv{3==(y1);1==\bzdrv{4==(y1)}}}
```

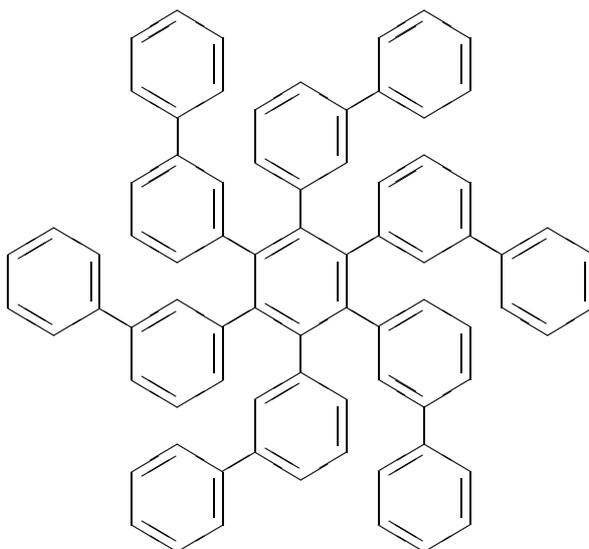
To simplify the coding, we define a macro drawing a biphenyl unit as follows:

```
\def\biph#1#2#3{\bzdrv{#1==(y1);#2==\bzdrv{#3==(y1)}}}
```

Then, this macro is used in the SUBSLIST of `\bzdrv` to give the code,

```
\bzdrv{%
1==\biph{4}{2}{5};%
2==\biph{5}{3}{6};%
3==\biph{6}{4}{1};%
4==\biph{1}{5}{2};%
5==\biph{2}{6}{3};%
6==\biph{3}{1}{4}}
```

Thereby, we have

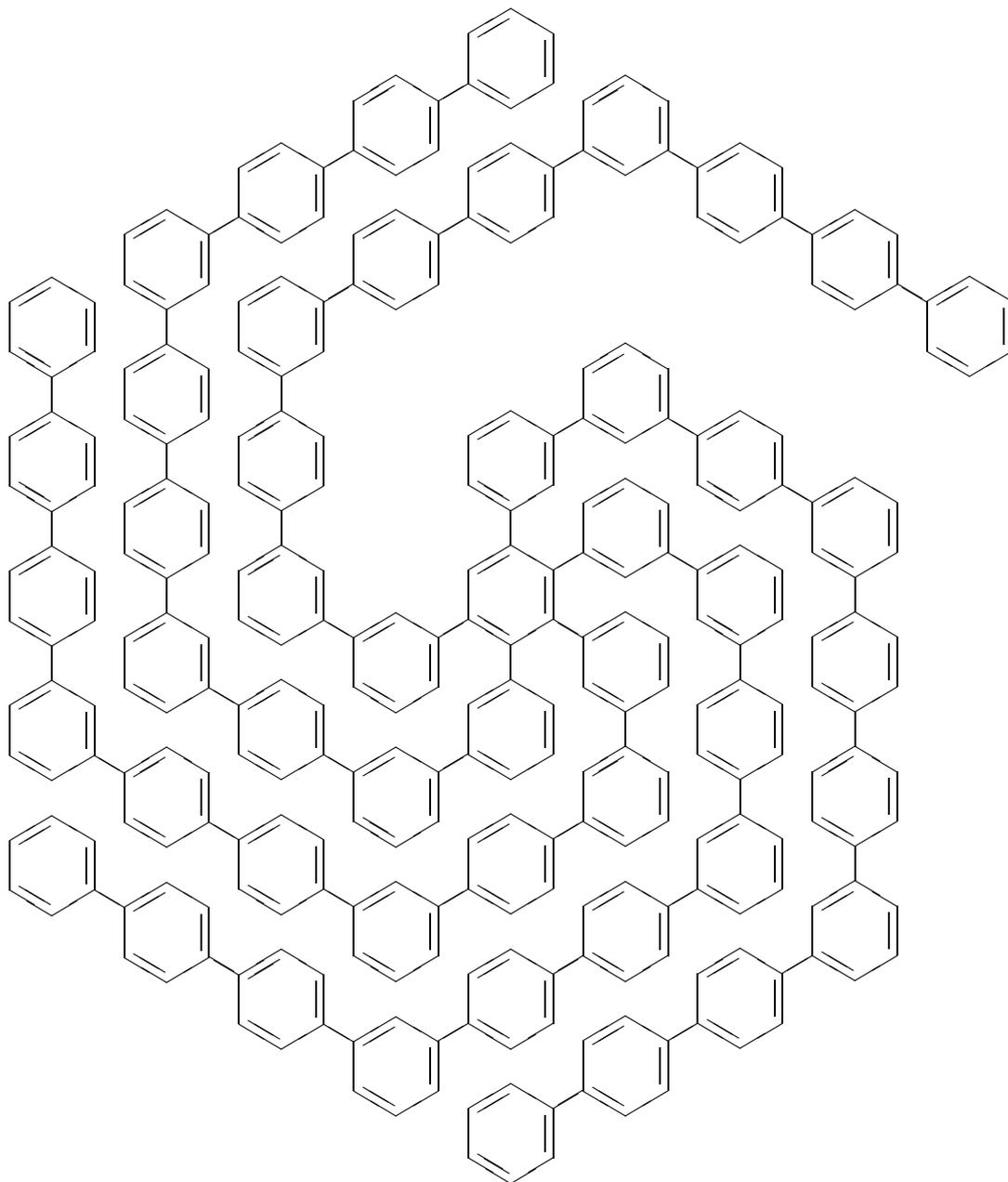


A more complex nested code,

```
\vspace*{8cm}
\bzdrv{%
1==\bzdrv{4==(y1);2==\bzdrv{5==(y1);3==\bzdrv{6==(y1);%
3==\bzdrv{6==(y1);4==\bzdrv{1==(y1);4==\bzdrv{1==(y1);%
4==\bzdrv{1==(y1);5==\bzdrv{2==(y1);5==\bzdrv{2==(y1);%
5==\bzdrv{2==(y1)}}}}}}}}};%
2==\bzdrv{5==(y1);3==\bzdrv{6==(y1);4==\bzdrv{1==(y1);%
4==\bzdrv{1==(y1);5==\bzdrv{2==(y1);5==\bzdrv{2==(y1);%
5==\bzdrv{2==(y1);6==\bzdrv{3==(y1);6==\bzdrv{3==(y1);%
6==\bzdrv{3==(y1)}}}}}}}}};%
3==\bzdrv{6==(y1);4==\bzdrv{1==(y1);5==\bzdrv{2==(y1);%
5==\bzdrv{2==(y1);6==\bzdrv{3==(y1);6==\bzdrv{3==(y1);%
6==\bzdrv{3==(y1);1==\bzdrv{4==(y1);1==\bzdrv{4==(y1);%
1==\bzdrv{4==(y1)}}}}}}}}};%
4==\bzdrv{1==(y1);5==\bzdrv{2==(y1);6==\bzdrv{3==(y1);%
6==\bzdrv{3==(y1);1==\bzdrv{4==(y1);1==\bzdrv{4==(y1);%
1==\bzdrv{4==(y1);2==\bzdrv{5==(y1);2==\bzdrv{5==(y1);%
2==\bzdrv{5==(y1)}}}}}}}}};%
5==\bzdrv{2==(y1);6==\bzdrv{3==(y1);1==\bzdrv{4==(y1);%
1==\bzdrv{4==(y1);2==\bzdrv{5==(y1);2==\bzdrv{5==(y1);%
2==\bzdrv{5==(y1);3==\bzdrv{6==(y1);3==\bzdrv{6==(y1);%
3==\bzdrv{6==(y1)}}}}}}}}};%
%6==\bzdrv{3==(y1);1==\bzdrv{4==(y1);2==\bzdrv{5==(y1);%
%2==\bzdrv{5==(y1);3==\bzdrv{6==(y1);3==\bzdrv{6==(y1);%
%3==\bzdrv{6==(y1);4==\bzdrv{1==(y1);4==\bzdrv{1==(y1);%
%4==\bzdrv{1==(y1)}}}}}}}}}}
}
```

produces the following formula:¹

¹To avoid “TeX capacity exceeded”, the last substituents are commented out. If your computer environment has a sufficient capacity, you can process the full list of commands.

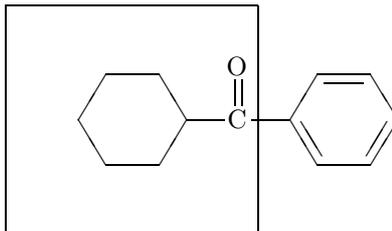


The code to draw this structural formula is too complicated to cause the “ \TeX capacity exceeded” error. To avoid the error, we use `\clearpage` commands before and after the output of the formula. In addition, we call only necessary packages to treat this document without the use of `xymtex.sty` calling all package files.

3.4 Remarks

3.4.1 Drawing Domains

Substituents produced by the “`y1`”-function have no dimensions. For example, benzoylcyclohexane



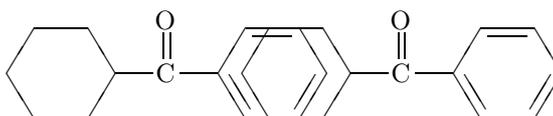
produced by the code

```
\cyclohexaneh[] {4==%
\tetrahedral{2==(y1);0==C;1D==0;4==\bzdrh{1==(y1)}}}
```

has a drawing domain around the cyclohexane mother skeleton, as encircled by a frame. Since the bezoyl moiety occupies no area, it may be superimposed on other contexts so as to require some space adjustments. For example, the above code duplicated without any space adjustment,

```
\cyclohexaneh[] {4==%
\tetrahedral{2==(y1);0==C;1D==0;4==\bzdrh{1==(y1)}}}
\cyclohexaneh[] {4==%
\tetrahedral{2==(y1);0==C;1D==0;4==\bzdrh{1==(y1)}}}%
```

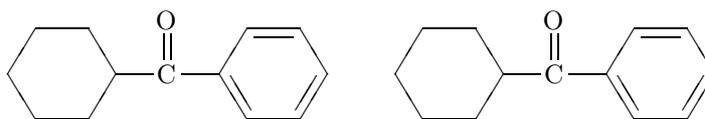
gives an insufficient result:



This superposition can be avoided by a horizontal spacing. Thus the code

```
\cyclohexaneh[] {4==%
\tetrahedral{2==(y1);0==C;1D==0;4==\bzdrh{1==(y1)}}}
\hskip2cm
\cyclohexaneh[] {4==%
\tetrahedral{2==(y1);0==C;1D==0;4==\bzdrh{1==(y1)}}}%
```

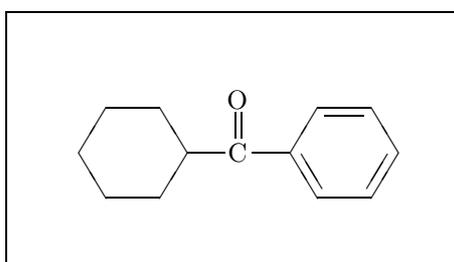
typesets improved formulas:



If a more thorough adjustment is required, a formula should be placed in a L^AT_EX picture environment as follows.

```
\begin{picture}(1600,900)(0,0)
\cyclohexaneh[] {4==%
\tetrahedral{2==(y1);0==C;1D==0;4==\bzdrh{1==(y1)}}}
\end{picture}
```

This code produces

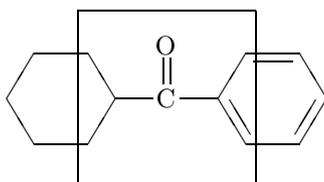


where a frame is added by means of a `\fbox` command.

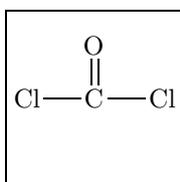
A drawing domain around a formula depends upon a mother skeleton selected. For example, the formula of benzoylcyclohexane at the top of this section has a drawing domain shown by the frame, since a `\cyclohexaneh` is selected as a mother skeleton. On the other hand, the alternative formula of benzoylcyclohexane depicted by the code,

```
\tetrahedral{0==C;1D==0;4==\bzdrh{1==(y1)};2==\cyclohexaneh[] {4==(y1)}}
```

has a drawing domain due to the `\tetrahedral` skeleton. Thus, the code gives the following output:



where the frame indicates such a drawing domain, when an `\fbox` command is used around the command `\tetrahedral`. The domain shown by the frame (due to `\fbox`) is equal to any domain based on the simple use of the `\tetrahedral` command (without using the “y1”-function). For example, compare the above frame with the one appearing in the formula,



depicted by the code,

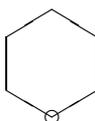
```
\fbox{\tetrahedral{0==C;1D==0;4==Cl;2==Cl}}
```

3.4.2 Reference Points

Each $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ command for drawing a mother skeleton has its reference point and its inner reference point. These points can be printed out by switching `\origpt` on. For example, the code

```
{  
\origpttrue  
\cyclohexanev{  
}
```

generates the diagram:



where the solid circle indicates the reference point (0,0) and the open circle indicates the inner reference point (400,240). The values of coordinates are output on a display and in a log file:

```
command 'sixheterov' origin: (0,0) ----> (400,240)
```

since `\cyclohexanev` is based on `\sixheterov`.

Chapter 4

Linking Units

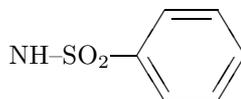
The commands `\ryl` and `\lyl` described in this chapter are added to the `chemstr` package (file name: `chemstr.sty`). The `\divalenth` command is added to the `aliphath` package (file name: `aliphath.sty`).

4.1 `\ryl` command

The “yl”-function provides us with a tool to generate a substituent that is linked *directly* to a substitution site of a mother skeleton. There are, however, many cases in which a substituent is linked to a substitution site by an intervening unit (e.g., O, SO₂ and NH). The command `\ryl` is used to generate a right-hand substituent with a linking unit. For example, the code

```
\ryl(5==NH--SO$_{2}$){4==\bzdrh{1==(y1)}}}
```

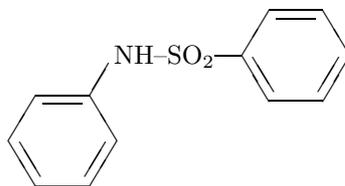
produces a benzenesulfonamido substituent,



The resulting unit is added to the SUBSLIST of a command for drawing a skeletal command. For example, the code

```
\bzdrh{3==\ryl(5==NH--SO$_{2}$){4==\bzdrh{1==(y1)}}}
```

generates the following formula:



The `\ryl` command takes two arguments.

```
\ryl(LINK){GROUP}
```

The first argument LINK in the parentheses indicates an intervening unit with an integer showing the slope of a left incidental bond. For example, the number 5 of the code `5==NH--SO$_{2}$` shown above represents that the left terminal is to be linked through $(-5, -3)$ bond, though the linking bond is not typeset by the `\ryl` command only. The slopes of the linking bonds are designated by integers between 0 and 8:

| | | | | | |
|---|----------|---|----------|---|----------|
| 0 | (0, 1) | 1 | (-3, 5) | 2 | (-1, 1) |
| 3 | (-5, 3) | 4 | (-1, 0) | 5 | (-5, -3) |
| 6 | (-1, -1) | 7 | (-3, -5) | 8 | (0, -1) |

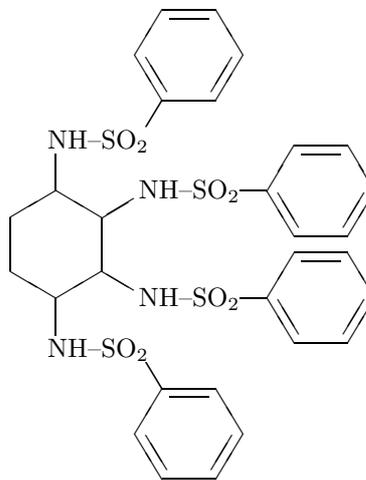
The second argument GROUP of `\ryl` is a substituent produced by a “yl”-function, where a number before a delimiter (`==`) indicates the slope of a right incidental bond. For example, the number 4 of the code `4==\bzdrh{1==(y1)}` shown above represents that the right terminal is to be linked through $(1, 0)$ bond to the benzene ring generated by the `\bzdrh` command. The slopes of the linking bonds are designated by integers between 0 and 8:

| | | | | | |
|---|---------|---|---------|---|---------|
| 0 | (0, 1) | 1 | (3, 5) | 2 | (1, 1) |
| 3 | (5, 3) | 4 | (1, 0) | 5 | (5, -3) |
| 6 | (1, -1) | 7 | (3, -5) | 8 | (0, -1) |

To illustrate linking bonds with various slopes, the code

```
\cyclohexanev[] {%
1==\ryl(8==NH--SO$_{2}$){1==\bzdrh{6==(y1)}};
2==\ryl(5==NH--SO$_{2}$){4==\bzdrh{1==(y1)}};
3==\ryl(3==NH--SO$_{2}$){4==\bzdrh{1==(y1)}};%
4==\ryl(0==NH--SO$_{2}$){7==\bzdrh{2==(y1)}}}
```

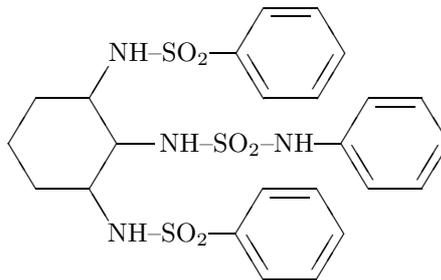
is written to give



Other examples are drawn by the code

```
\cyclohexaneh[] {%
3==\ryl(7==NH--SO$_{2}$){4==\bzdrh{1==(y1)}};
5==\ryl(1==NH--SO$_{2}$){4==\bzdrh{1==(y1)}};%
4==\ryl(4==NH--SO$_{2}$--NH){4==\bzdrh{1==(y1)}}}
```

giving



The first argument in the parentheses of the command `\ryl` contains a string of letters after an intermediate delimiter `==`, where a left linking site is shifted according to the length of the letter string. The above formula shows such an example as having $\text{NH-SO}_2\text{-NH}$.

The following examples compare the “yl”-function with the `\ryl` command.

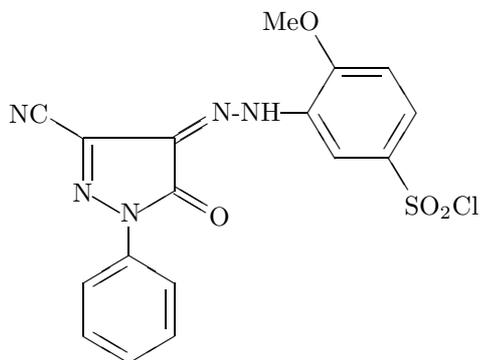
```
\cyclohexaneh{4==\bzdrh{1==(y1)}}
\hskip2cm
\cyclohexaneh{4==\ryl(4==0){4==\bzdrh{1==(y1)}}}
```



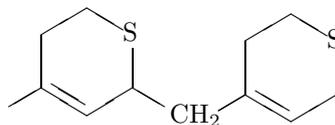
The compound **21** on page 299 of the \LaTeX book can be alternatively drawn by using the `\ryl` command, as shown in the code:

```
\fiveheterov[d]{1==N;5==N}{4==NC;1==\bzdrv{1==(y1)};2D==0;%
3D==\ryl(5==N-NH){4==\bzdrh{1==(y1)};2==\lmoiety{MeO};5==SO$_{2}$Cl}}
```

which typeset the following formula:



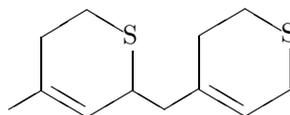
The first argument of the `\ryl` is optional; i.e., it can be omitted. Such an omitted case is useful to draw a methylene as a vertex. For example, a methylene is represented as a character string “CH₂”, as shown in the formula,



This formula is generated by the code,

```
\sixheterov[d]{2==S}{5==\null;%
3==\ryl(3==CH$_{2}$){3==\sixheterov[d]{2==S}{5==(y1)}}
```

where the `\ryl` command takes an optional argument in parentheses to draw CH₂ explicitly. Such a methylene can alternatively be represented as a simple vertex, as shown in the formula,



This formula is generated by the code,

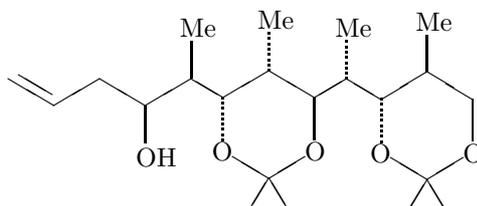
```
\sixheterov[d]{2==S}{5==\null;%
3==\ryl{3==\sixheterov[d]{2==S}{5==(y1)}}
```

where the `\ryl` command takes no optional argument.

The second argument of the `\ryl` command can accommodate substituents other than a substituent generated by the “y1” function. For example, the inner code `\ryl{OA==Me;...}` in the code,

```
\sixheterov({bB}{eA}){3==0;5==0}{1A==Me;4Sa==\null;4Sb==\null;%
6==\pentamethylenei[a]{}{4B==OH;5B==Me;5==(y1)};%
2==\ryl{OA==Me;5==\sixheterov({eA}){3==0;5==0}{6==(y1);1B==Me;%
4Sa==\null;4Sb==\null}}}
```

represents a methyl group on a vertex due to the command `\ryl`. Thereby, we have



4.2 \lyl command

The command `\lyl` is the left-hand counterpart of the command `\ryl`.

`\lyl(LINK){GROUP}`

The slopes of the linking bonds concerning the right terminal are designated by integers between 0 and 8:

| | | | | | |
|---|---------|---|---------|---|---------|
| 0 | (0, 1) | 1 | (3, 5) | 2 | (1, 1) |
| 3 | (5, 3) | 4 | (1, 0) | 5 | (5, -3) |
| 6 | (1, -1) | 7 | (3, -5) | 8 | (0, -1) |

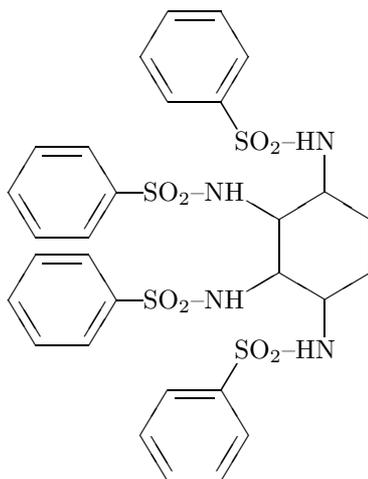
The slopes of the linking bonds concerning the left terminal are designated by integers between 0 and 8:

| | | | | | |
|---|----------|---|----------|---|----------|
| 0 | (0, 1) | 1 | (-3, 5) | 2 | (-1, 1) |
| 3 | (-5, 3) | 4 | (-1, 0) | 5 | (-5, -3) |
| 6 | (-1, -1) | 7 | (-3, -5) | 8 | (0, -1) |

To illustrate linking bonds with various slopes, the code

```
\cyclohexanev[] {%
1==\lyl(8==SO$_{2}$--HN){1==\bzdrh{5==(y1)}};%
6==\lyl(5==SO$_{2}$--NH){4==\bzdrh{4==(y1)}};%
5==\lyl(3==SO$_{2}$--NH){4==\bzdrh{4==(y1)}};%
4==\lyl(0==SO$_{2}$--HN){7==\bzdrh{3==(y1)}}}
```

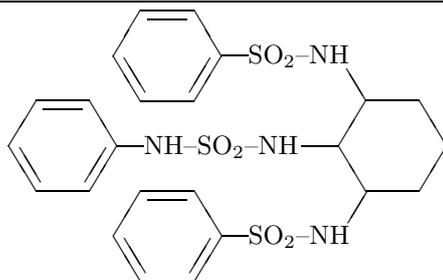
is written to give



Other examples are drawn by the code

```
\cyclohexaneh[] {%
2==\lyl(7==SO$_{2}$--NH){4==\bzdrh{4==(y1)}};%
6==\lyl(1==SO$_{2}$--NH){4==\bzdrh{4==(y1)}};%
1==\lyl(4==NH--SO$_{2}$--HN){4==\bzdrh{4==(y1)}}}
```

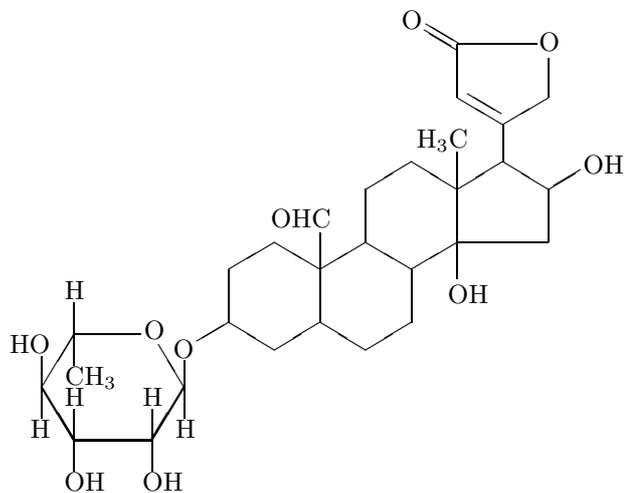
giving



The first argument in the parentheses of the command `\lyl` contains a string of letters after an intermediate delimiter `==`, where a left linking site is shifted according to the length of the letter string. The above formula shows such an example as having `NH-SO2-NH`.

The structural formula of adonitoxin, which has once been depicted in a different way in Chapter 15 of the $\text{\X}^{\text{M}}\text{T}_{\text{E}}\text{X}$ book can be obtained by the code,

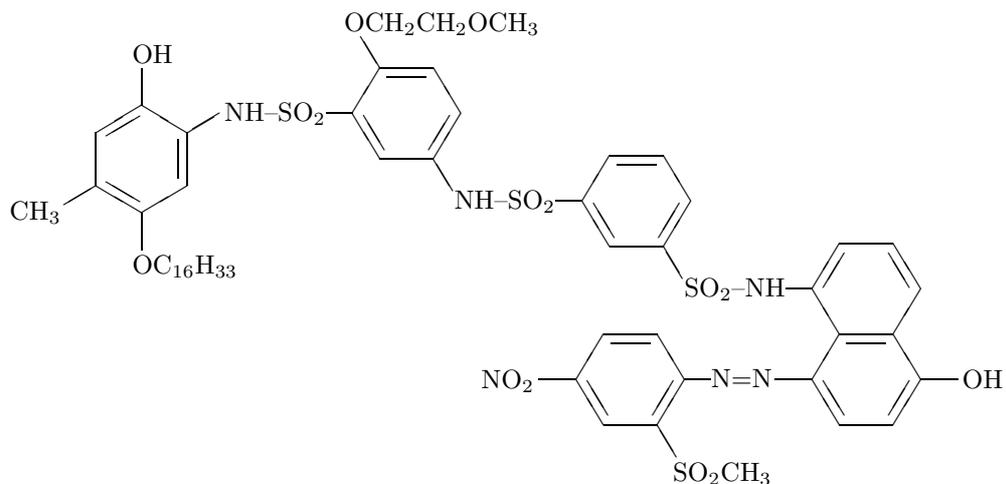
```
\steroid{{10}}==\lmoiety{OHC};{{14}}==OH;%
{{13}}==\lmoiety{H$_{3}$C};{{16}}==OH;%
{{17}}==\fiveheterov[e]{3==0}{4D==0;1==(y1)};%
3==\lyl(3==0){8==%
\pyranose{1Sb==(y1);1Sa==H;2Sb==H;2Sa==OH;3Sb==H;3Sa==OH;4Sb==HO;%
4Sa==H;5Sb==H;5Sa==CH$_{3}$}}
```



4.3 Nested `\ryl` and `\lyl` commands

Two or more `\ryl` and `\lyl` commands can be nested. Let us illustrate nesting processes by drawing a cyan dye releaser, which has once been depicted in different ways (see Chapters 14 and 15 of the

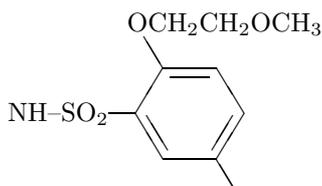
X_YMT_EXbook).



First, the code

```
\ryl(4==NH--SO$_{2}$){4==\bzdrh{1==(y1);2==OCH$_{2}$CH$_{2}$OCH$_{3}$;%  
5==\null}}
```

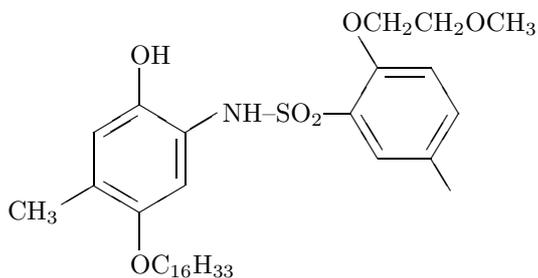
generates a substituent:



in which the command `\null` is used to show a further substitution site. The resulting substituent is nested in the `SUBSLIT` of another `\bzdrv` command as shown in the code:

```
\bzdrv{1==OH;5==CH$_{3}$;4==OC$_{16}$H$_{33}$;%  
2==\ryl(4==NH--SO$_{2}$){4==\bzdrh{1==(y1);2==OCH$_{2}$CH$_{2}$OCH$_{3}$;%  
5==\null}}}
```

Thereby we have



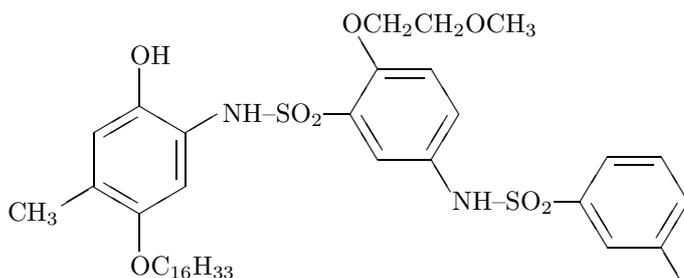
The inner code `5==\null` is replaced by a further code of substitution:

```
5==\ryl(2==NH--SO$_{2}$){4==\bzdrh{1==(y1);5==\null}}%
```

to give a code,

```
\bzdrv{1==OH;5==CH$_{3}$;4==OC$_{16}$H$_{33}$};%
2==\ryl(4==NH--SO$_{2}$){4==\bzdrh{1==(y1);2==OCH$_{2}$CH$_{2}$OCH$_{3}$};%
5==\ryl(2==NH--SO$_{2}$){4==\bzdrh{1==(y1);5==\null}}%
}}}
```

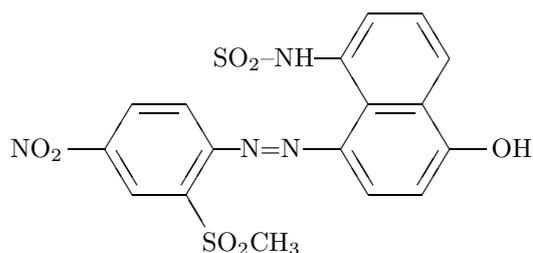
This code generates the following structure (Formula A):



Another substituent is typeset by the code,

```
\ryl(2==SO$_{2}$--NH){4==\naphdrh{1==(y1);5==OH;%
8==\lyl(4==N=N){4==\bzdrh{4==(y1);1==NO$_{2}$;5==SO$_{2}$CH$_{3}$}}}}
```

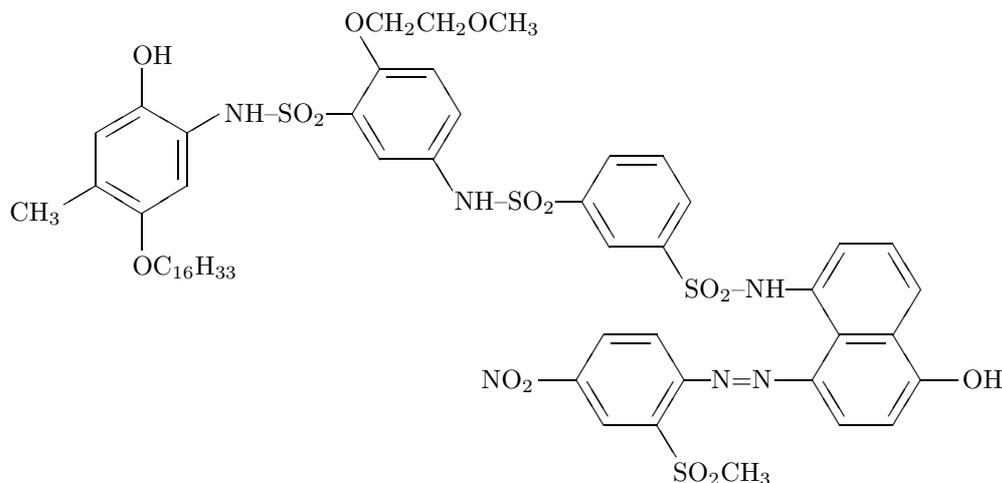
Then, we have a substituent (Formula B):



Finally, the inner code `5==\null` for Formula A is replaced by the code for Formula B in order to combine Formula A with Formula B. Then we obtain a code represented by

```
\bzdrv{1==OH;5==CH$_{3}$;4==OC$_{16}$H$_{33}$};%
2==\ryl(4==NH--SO$_{2}$){4==\bzdrh{1==(y1);2==OCH$_{2}$CH$_{2}$OCH$_{3}$};%
5==\ryl(2==NH--SO$_{2}$){4==\bzdrh{1==(y1);%
5==\ryl(2==SO$_{2}$--NH){4==\naphdrh{1==(y1);5==OH;%
8==\lyl(4==N=N){4==\bzdrh{4==(y1);1==NO$_{2}$;5==SO$_{2}$CH$_{3}$}}}}}}}
```

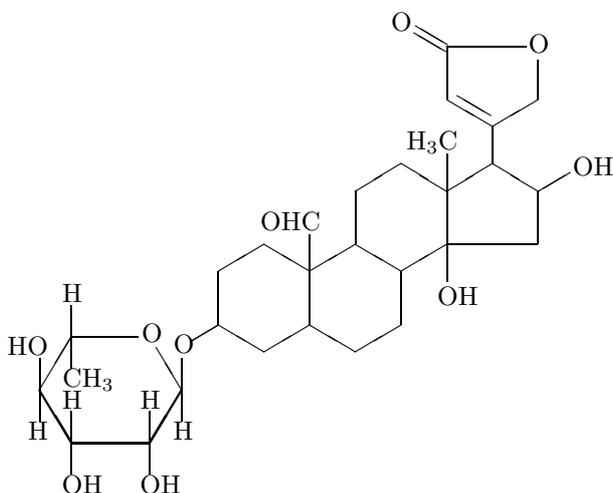
Thereby, we have a target formula:



The structural formula of adonitoxin, which has been drawn by considering the steroid nucleus to be a mother skeleton in the preceding subsection, can be alternatively drawn by nesting a “yl”-function and a `\ryl` command. In this case, the pyranose ring is regarded as a mother skeleton. Thus, the code

```
\pyranose{1Sa==H;2Sb==H;2Sa==OH;3Sb==H;3Sa==OH;4Sb==HO;%
4Sa==H;5Sb==H;5Sa==CH$_{3}$;%
1Sb==\ryl(8==0){3==%
\steroid{3==(y1);{10}==\lmoiety{OHC};{14}==OH;%
{13}==\lmoiety{H$_{3}$C};{16}==OH;%
{17}==\fiveheterov[e]{3==0}{4D==0;1==(y1)}}}
```

typesets the following formula:



4.4 `\divalenth` command

The command `\divalenth` generates a divalent skeleton with variable length.

```
\divalenth{GROUP}{SUBSLIST}
```

The divalent skeleton is given by a string of alphabets in the GROUP argument. The locant number in the GROUP argument is fixed to be zero. For example, the code

```
\divalenth{0==NHCONH}{1==CH$_{3}$;2==CH$_{3}$}
```

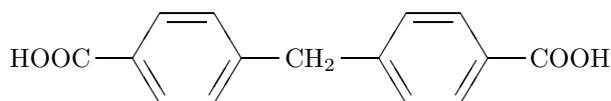
generates a linear formula:



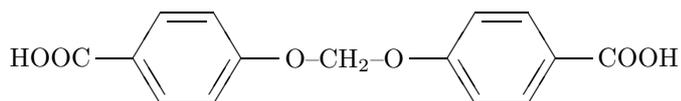
4,4'-Methylenedibenzoic acid can be drawn in the same line. The code

```
\divalenth{0==CH$_{2}$}{1==\bzdrh{4==(y1);1==HOOC};2==\bzdrh{1==(y1);4==COOH}}
```

generates



In place of the CH₂ unit described in the preceding example, we introduce the O-CH₂-O unit so as to give 4,4'-methylenedioxydibenzoic acid. The structural formula can be drawn to be

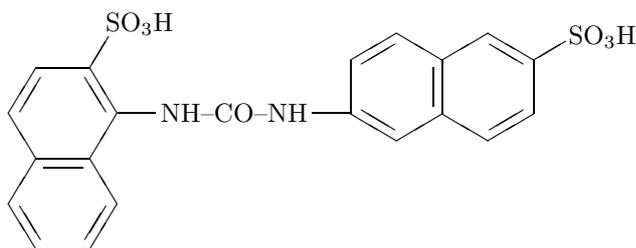


by means of the code:

```
\divalenth{0==O--CH$_{2}$--O}%
{1==\bzdrh{4==(y1);1==HOOC};2==\bzdrh{1==(y1);4==COOH}}
```

Note that the starting point of the moiety generated by the code `2==\bzdrh{1==(y1);4==COOH}` is automatically shifted so as to accommodate the O-CH₂-O unit.

An additional example of the use of the `\divalenth` command is the drawing of 1,6'-ureylenedi-2-naphthalenesulfonic acid



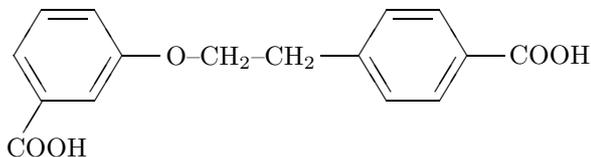
by means of the code

```
\divalenth{0==NH--CO--NH}%
{1==\naphdrh{4==(y1);3==SO$_{3}$H};2==\naphdrv{6==(y1);2==SO$_{3}$H}}
```

p-[2-(*m*-Carboxyphenoxy)ethyl]benzoic acid is drawn by the code

```
\divalenth{0==O--CH$_{2}$--CH$_{2}$}%
{1==\bzdrh{4==(y1);6==COOH};2==\bzdrh{1==(y1);4==COOH}}
```

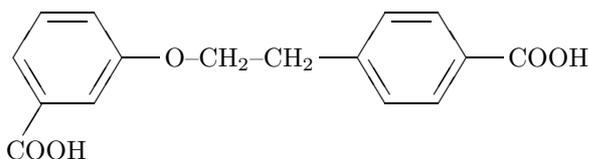
which generates a formula:



The same structure can be depicted by applying the “yl”-function to the `\divalenth` command. The code

```
\bzdrh{6==COOH;4==%
\divalenth{0==O--CH$_{2}$--CH$_{2}$}{1==(y1);2==\bzdrh{1==(y1);4==COOH}}}
```

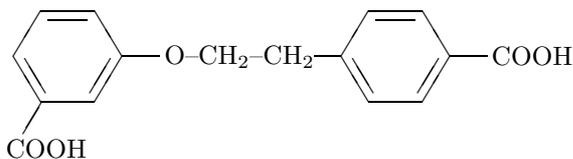
generates the same formula:



This type of usage gives an equivalent function of the command `\ryl` or `\lyl`. Compare this with an example using the `\ryl` command:

```
\bzdrh{6==COOH;4==%
\ryl(4==O--CH$_{2}$--CH$_{2}$){4==\bzdrh{1==(y1);4==COOH}}}
```

This code gives the same formula:

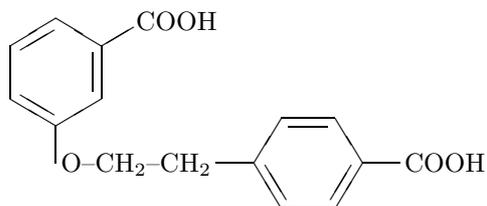


4.5 Remarks

The use of `\divalenth` with a “yl”-function has no means of adjusting the left-hand point of linking. For example, the code,

```
\bzdrv{2==COOH;4==%
\divalenth{0==O--CH$_{2}$--CH$_{2}$}{1==(y1);2==\bzdrh{1==(y1);4==COOH}}}
```

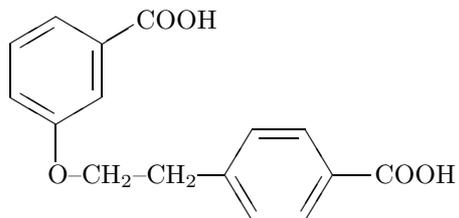
give an insufficient formula:



where the left-hand point of linking should be shifted to a more appropriate direction. On the other hand, the `\ryl` (or `\lyl`) command can correctly specify the left-hand point of linking. Thus the code,

```
\bzdrv{2==COOH;4==%
\ryl(0==0--CH$_{2}$--CH$_{2}$){4==\bzdrh{1==(y1);4==COOH}}
```

typesets a formula:



where the code `0==0--CH$_{2}$--CH$_{2}$` specifies the left-hand terminal of the unit $\text{O-CH}_2\text{-CH}_2$ is linked at the upper point of the oxygen atom.

Chapter 5

Ring Fusion

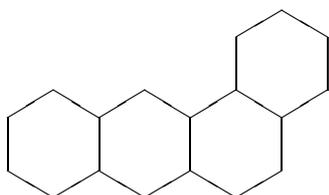
5.1 Ring Fusion on Carbocyclic Compounds

5.1.1 Designation of Fused Bonds

A unit to be fused is written in the BONDLIST of a command with a bond specifier (a lowercase or uppercase alphabet). For example, the code

```
\hanthracenev[{\A\sixfusev{}{}{d}}]{}
```

gives a perhydroanthracene with a fused six-membered ring at the bond ‘a’ of the perhydroanthracene nucleus:



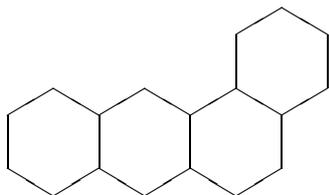
The letter ‘A’ of the code `{\A\sixfusev{}{}{d}}` is a bond specifier that represents the older terminal of the bond ‘a’ of the perhydroanthracene nucleus (For the designation of the bonds of perhydroanthracene, see Chapter 5 of the *X_YTEXbook*.¹ Note that the younger terminal of the bond ‘a’ is designated by the letter ‘a’. On the other hand, the code `\sixfusev{}{}{d}` of `{\A\sixfusev{}{}{d}}` in the BONDLIST represents the fused six-membered ring with the bond ‘d’ omitted. The letter ‘d’ indicates that the fusing point of the unit is the younger terminal of the omitted bond ‘d’. If the the fusing point of the unit is the other (older) terminal, the corresponding uppercase letter ‘D’ should be used.

Accordingly, the same formula can be drawn by the code exchanging uppercase and lowercase letters,

```
\hanthracenev[{\a\sixfusev{}{}{D}}]{}
```

Thereby, we have

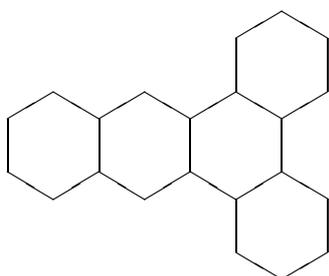
¹The word ‘older’ or ‘younger’ is concerned with the order of numbering of vertices. For a six-membered ring, the numbering 1—2—3—4—5—6—1 shows that the terminal 1 of the bond ‘a’ (1—2) is younger, while the terminal 2 of the bond ‘a’ is older. It should be noted that the terminal 6 of the bond ‘f’ (6—1) is younger, while the terminal 1 of the bond ‘f’ is older.



Two or more rings can be fused. For example, the code

```
\hanthracenev[ $\{A\}$ sixfusev $\{\}$ ]{ $\{d\}$ }{ $C\}$ sixfusev $\{\}$ ]{ $\{f\}$ }] $\{\}$ 
```

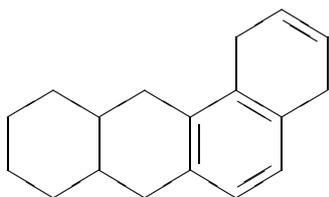
generates a formula with two fused rings at the bonds ‘a’ and ‘c’ of a perhydroanthracene nucleus.



The BONDLIST can accommodate usual bond specifiers without a fusing unit in order to designate inner double bonds. For example, the code

```
\hanthracenev[aco{ $A\}$ sixfusev[ $a\}$ ]{ $\{d\}$ }] $\{\}$ 
```

gives a hydroanthracene that has inner double bonds as well as a fused six-membered ring:

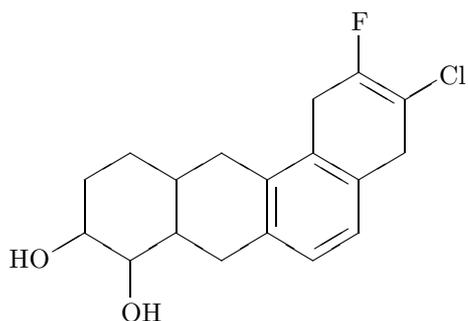


Note that the command `\sixfusev` can take an optional argument to designate inner double bonds, as shown by the code `\sixfusev[a]{d}`.

In order to specify substituents in addition, we can use the SUBSLIST of the command `\hanthracenev` as well as the one of the command `\sixfusev`. For example, the code

```
\hanthracenev[aco{ $A\}$ sixfusev[ $a\}$ ]{ $1==F;2==Cl\}$ ]{ $d\}$ ]{ $5==OH;6==HO\}$ 
```

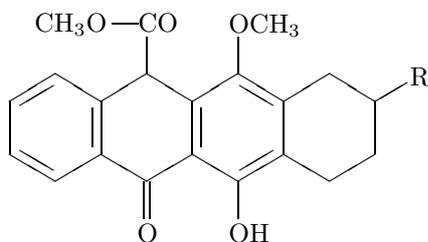
gives a hydroanthracene having additional substituents:



The compound **13** on page 294 (Chapter IV-4) of the \LaTeX book can alternatively be drawn by applying the present technique. Thus, the code

```
\hanthracenev[achjop{b\sixfusev}{-}{-}{E}]{%
1==OCH$_{3}$;4==OH;{10}D==O;%
9==\lyl(8==C\rlap{0}){4==CH$_{3}$0}}
```

gives the following formula:

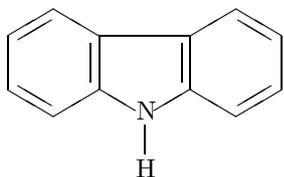


5.2 Ring Fusion on Heterocyclic Compounds

The methodology of ring fusion for heterocyclic compounds is the same as described for carbocyclic compounds. Thus, a unit to be fused is written in the BONDLIST of a command with a bond specifier (a lowercase or uppercase alphabet). For example, the code

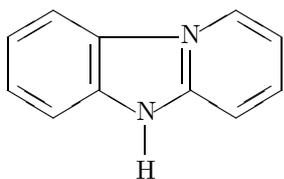
```
\nonaheterov[begj{b\sixfusev[ac]{}{}{e}}]{1==N}{1==H}
```

gives the structural formula of carbazole:



which is depicted by attaching a six-membered ring ($\text{\sixfusev[ac]{}{}{e}}$) to the bond 'b' of an indole nucleus.

Let us consider the substitution of a carbon atom with a nitrogen atom at one of the fused positions in the above compound, as shown by the following formula:



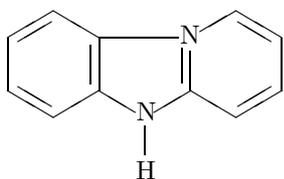
This formula is obtained by writing the code:

```
\nonaheterov[begj{b\sixfusev[ac]{6==\null}{e}}]{1==N;3==N}{1==H}
```

where the code `6==\null` in the ATOMLIST of `\sixfusev` (for the fused six-membered ring) and the code `3==N` in the ATOMLIST of `\nonaheterov` produces the nitrogen atom at the fused position. The specification of the nitrogen atom is also available by exchanging `\null` and `N`. Thus the code

```
\nonaheterov[begj{b\sixfusev[ac]{6==N}{e}}]{1==N;3==\null}{1==H}
```

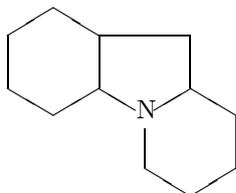
gives the same structural formula:



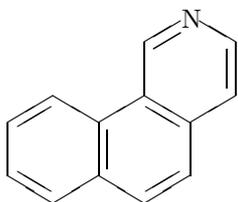
The ring fusion at the bond 'a' of perhydroindole is represented by the code

```
\nonaheterov[{a\sixfusev{6==\null}{f}}]{1==N}{}
```

which gives a heterocycle:



Benz[*h*]isoquinoline,



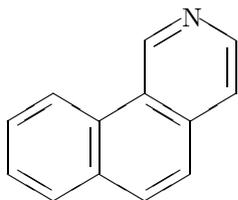
can be typeset by the code,

```
\decaheterovt[acfhk{h\sixfusev[df]{}{}{B}}]{2==N}{}
```

in which the bond specifier ‘h’ corresponds to the *h* of the IUPAC name. Note that the IUPAC name regards the structure as an isoquinoline (drawn by `\decaheterovt`) fused by a benzo moiety. The same structure can be drawn by the alternative code:

```
\decaheterov[acfhk{a\sixfusev[bf]{1==N}{D}}]{}{}{}
```

which regards the structure as a naphthalene nucleus (drawn by `\decaheterov`) with a fused heterocycle. Thereby, we have

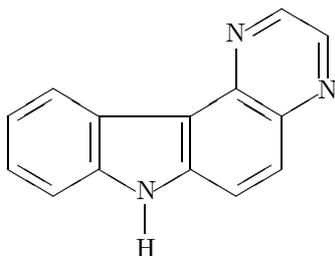


5.3 Nested Ring Fusion

The `\sixfusev` command is capable of accommodating another `\sixfusev` command in a nested fashion. By this technique, the carbazole structure can take a further fused ring so as to produce the structural formula of 7*H*-pyrazino[2,3-*c*]carbazole. Thus, the code,

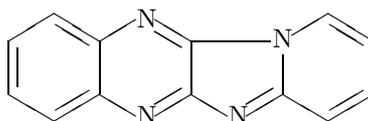
```
\nonaheterov[begj{b\sixfusev[%
ac{a\sixfusev[bf]{6==N;3==N}{D}}]{}{}{e}}]{1==N}{1==H}
```

gives the structural formula of the fused heterocycle:



which is depicted by attaching a six-membered ring (`\sixfusev[ac]{}{}{e}`) to the bond ‘b’ of an indole nucleus.

The structural formula of pyrido[1',2':1,2]imidazo[4,5-*b*]quinoxaline,



is generated by the code,

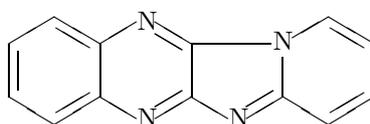
```
\nonaheterov[adh%
{b\sixfusev[ac]{6==\null}{e}}%
{f\sixfusev[ace]{b}}{1==N;3==N;4==N;7==N}{}
```

Since this code is intended to contain no nested ring fusion, the order of structure construction is different from that of the IUPAC name.

The IUPAC name, pyrido[1',2':1,2]imidazo[4,5-*b*]quinoxaline, corresponds to a quinoxaline with a fused five-membered ring (an imidazo moiety) which is in turn fused by a six-membered ring (a pyrido moiety). The order of constructing the IUPAC name is realized in the code with nested ring fusion,

```
\decaheterov[acegi%
{b\fivefusev[a{b\sixfusev[ac]{6==\null}{e}}{1==N;3==N}{d}}]
{1==N;4==N}{}
```

which produces the same structure,

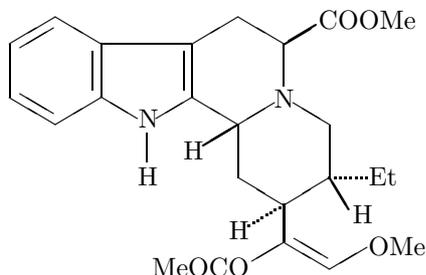


Note that the indicators '1',2' and '1,2' of the locant [1',2':1,2] in the IUPAC name correspond respectively to the bond specifiers, 'E' and 'b', appeared in the code, `{b\sixfusev[ac]{6==\null}{E}}`. On the other hand, the indicators, '4,5' and 'b' of the locant [4,5-*b*] are respectively associated with the specifiers, 'd' and 'b', appeared in the code, `{b\fivefusev[...]{1==N;3==N}{d}}`.

An alkaloid with a corynanthe skeleton (R. T. Brown and C. L. Chapple, *Chem. Commun.*, 1973, 887) can be typeset by the code with nested fusion,

```
\nonaheterov[bejj{b\sixfusev[%
{c\sixfusev{1==\null}{3SB==H;3SA==Et;%
4GA==H;%
4B==\dimethylenei[a]{1==(y1);2W==OMe;1W==MeOCO}{F}}}%
{3==N}{4GB==H;2B==COOMe}{e}}{1==N}{1==H}
```

where a six-five ring drawn by the command `\nonaheterov` is regarded as a mother skeleton. Thus, we have

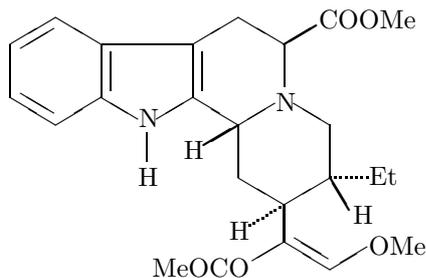


For the command `\dimethylenei`, see the chapter at issue.

When a six-six ring drawn by the command `\decaheterovb` is regarded as a mother skeleton, as shown in the code with another nested ring fusion,

```
\decaheterovb[f{f\fivefusev[d{d\sixfusev[df]{}{}{b}}}%
{1==N}{1==H}{b}}]{8a==N}{9B==H;2SA==Et;2SB==H;8B==COOMe;%
3GA==H;%
3B==\dimethylenei[a]{}{1==(y1);2W==OMe;1W==MeOCO}}
```

we find another way of drawing the same structural formula,



The following example shows a code with complicated nested structure:

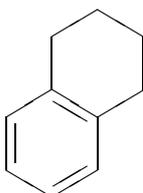
```
\cyclohexanev[%
{a\sixfusev[{b\sixfusev[{c\sixfusev[{c\sixfusev[%
{d\sixfusev[{d\sixfusev[{d\sixfusev[%
{e\sixfusev[{e\sixfusev[{e\sixfusev[{e\sixfusev[%
{f\sixfusev[{f\sixfusev[]{}{}{C}}]{}{}{C}}%
]{}{}{B}}]{}{}{B}}]{}{}{B}}]{}{}{B}}%
]{}{}{A}}]{}{}{A}}]{}{}{A}}]{}{}{F}}%
]{}{}{F}}]{}{}{E}}]{}{}{D}}%
{c\sixfusev[{d\sixfusev[{e\sixfusev[{e\sixfusev[%
{f\sixfusev[{f\sixfusev[{f\sixfusev[%
{a\sixfusev[{a\sixfusev[{a\sixfusev[{a\sixfusev[%
{b\sixfusev[{b\sixfusev[]{}{}{E}}]{}{}{E}}%
]{}{}{D}}]{}{}{D}}]{}{}{D}}]{}{}{D}}%
]{}{}{C}}]{}{}{C}}]{}{}{C}}]{}{}{B}}%
]{}{}{B}}]{}{}{A}}]{}{}{F}}%
{e\sixfusev[{f\sixfusev[{a\sixfusev[{a\sixfusev[%
{b\sixfusev[{b\sixfusev[{b\sixfusev[%
{c\sixfusev[{c\sixfusev[{c\sixfusev[{c\sixfusev[%
{d\sixfusev[{d\sixfusev[]{}{}{A}}]{}{}{A}}%
]{}{}{F}}]{}{}{F}}]{}{}{F}}]{}{}{F}}%
]{}{}{E}}]{}{}{E}}]{}{}{E}}]{}{}{D}}%
]{}{}{D}}]{}{}{C}}]{}{}{B}}%
]{}
]
```

This code generates a multiply fused formula:

5.4 Remarks

5.4.1 OPT Arguments

It should be noted that the OPT arguments of such commands as `\bzdrv`, `\naphdrv`, and `\anthracenev` cannot be used for the ring-fusion technique. In place of the OPT argument, the BONDLIST argument of the corresponding general command, e.g. `\cyclohexanev` or `\sixheterov` corresponding to `\bzdrv`, should be used for the purpose of ring fusion. . For example, a benzene ring of the formula,



should be drawn by using the `\cyclohexanev` command, as shown in the code:

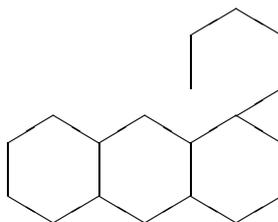
```
\cyclohexanev[ace{a\sixfusev{}{}{D}}]{}
```

5.4.2 X_yMT_EX Warning

An incorrect result due to a wrong specification of a fused bond is notified by a X_yMT_EX warning. For example, the code,

```
\hanthracenev[{a\sixfusev{}{}{d}}]{}
```

gives a formula of wrong fusion:



According to this wrong situation, a X_yMT_EX warning appears in a display or in a log file, e.g.,

```
XyMTeX Warning: Mismatched fusion at bond 'a, i, or other'
on input line 1904
```

There are two ways to correct the wrong fusion and, as a result, to avoid such a X_yMT_EX warning. First, the code

```
\hanthracenev[{A\sixfusev{}{}{d}}]{}
```

in which the acceptor bond specifier ‘a’ is changed into ‘A’, gives a correct result, as found in the top example of this chapter. Alternatively, the donor bond specifier ‘d’ can be changed into ‘D’. Thus, the code,

```
\hanthracenev[ $\{a\}$ sixfusev $\{\}$ ]{ $\{D\}$ }{ $\}$ }
```

also typesets the second formula with correct fusion.

Chapter 6

Fusing Units

The commands described in this chapter are stored in the `fusing` package (file name: `fusing.sty`).

6.1 Six-membered Fusing Units

6.1.1 Vertical Units of Normal and Inverse Types

In \LaTeX version 1.01, we can use `\sixunitv` and `\fiveunitv` as building blocks, where one or more bonds can be omitted. In the present version, we prepare such commands as `\sixfusev` and `\sixfusevi`, producing building blocks with only one deleted bond. These commands can be used in the `BONDLIST` of another command so as to give a fused structural formula, as described in the preceding chapter. The commands `\sixfusev` and `\sixfusevi` have formats represented by

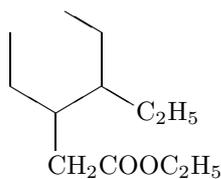
```
\sixfusev[BONDLIST]{ATOMLIST}{SUBSLIST}{FUSE}  
\sixfusevi[BONDLIST]{ATOMLIST}{SUBSLIST}{FUSE}
```

where the argument `FUSE` is an alphabetical character (a–f) or the uppercase counterpart (A–F), each of which is a bond specifier representing one bond to be omitted. A lowercase character (a–f) represents the younger terminal of the omitted bond. The corresponding uppercase character (A–F) designates the other terminal of the bond to be omitted. The other arguments have the same formats as described in the general conventions (see \LaTeX book). The locant numbers and the bond specifiers of the command `\sixfusev` correspond to those of the command `\sixheterov` (see \LaTeX book). The command `\sixfusevi` is the inverse counterpart of `\sixfusev` and corresponds to the command `\sixheterovi`. Moreover, the `BONDLIST` is capable of accommodating the ring-fusion function described in the preceding chapter, the `ATOMLIST` can accommodate the spiro-ring function described afterward, and the `SUBSLIST` serves a method producing substituents (“yl”-function) describe previously.

For example, the last argument ‘F’ of the `\sixfusev` appearing in the code,

```
\sixfusev[] {1==\null}%  
{3==C$_2$H$_5$;4==CH$_2$COOC$_2$H$_5$}{F}
```

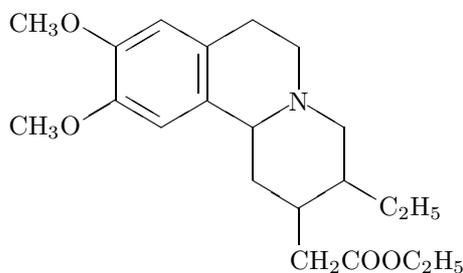
results in the deletion of the bond ‘f’ between atom no. 6 (younger terminal) and atom no. 1 (older terminal) from a hexagon, typesetting the following building block:



where the reference point for superposition is the older terminal (i.e. atom no. 1) of the bond 'f'. The code `1==\null` gives a vacancy at the position of atom no. 1. When the building block is used in the BONDLIST of the `\decaheterov`, as shown in the code,

```
\decaheterov[fhk%
{c\sixfusev[] {1==\null}}%
{3==C$_2$H$_5$;4==CH$_2$COOC$_2$H$_5$}{F}]{3==N}{6==CH$_3$0;7==CH$_3$0}
```

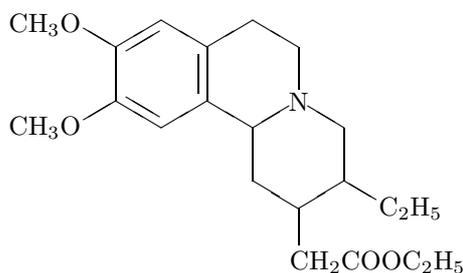
we have the following structure,



The last argument 'F' of the `\sixfusev` can be changed into 'f', as found in the code,

```
\decaheterovi[fhk%
{a\sixfusev[] {1==\null}}%
{3==C$_2$H$_5$;4==CH$_2$COOC$_2$H$_5$}{f}]{2==N}{6==CH$_3$0;7==CH$_3$0}
```

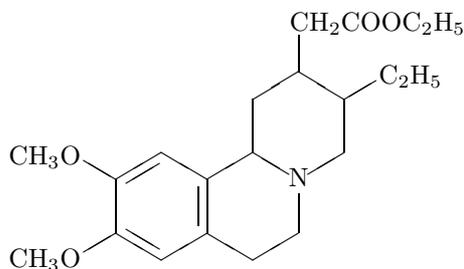
where we use `\decaheterovi` in place of `\decaheterov` for drawing the bicyclic mother skeleton. Thereby, we have the following structure,



The vertically opposite formula can be drawn by the combination of `\sixfusevi` and `\decaheterovi` with no other changes of designation (in comparison with the first code of this section), i.e.

```
\decaheterovi[fhk%
{c\sixfusevi[] {1==\null}%
{3==C$_2$H$_5$;4==CH$_2$COOC$_2$H$_5$}{F}}]{3==N}{6==CH$_3$;7==CH$_3$}
```

Thereby we have



6.1.2 Horizontal Units of Normal and Inverse Types

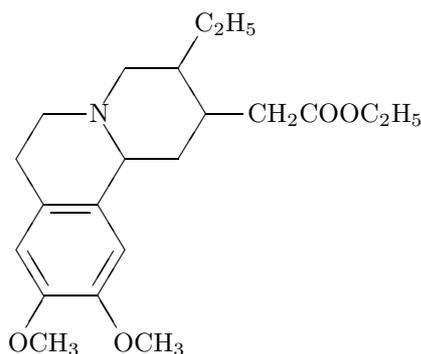
For drawing horizontal fusing units, we can use the commands `\sixfuseh` and `\sixfusehi`, which are represented by

```
\sixfuseh[BONDLIST]{ATOMLIST}{SUBSLIST}{FUSE}
\sixfusehi[BONDLIST]{ATOMLIST}{SUBSLIST}{FUSE}
```

The horizontal formula of normal type related to the tricyclic formulas described in the preceding subsection can be drawn by the combination of `\sixfuseh` and `\decaheteroh` with few changes of designation (CH_3O to OCH_3), i.e.

```
\decaheteroh[fhk%
{c\sixfuseh[] {1==\null}%
{3==C$_2$H$_5$;4==CH$_2$COOC$_2$H$_5$}{F}}]{3==N}{6==OCH$_3$;7==OCH$_3$}
```

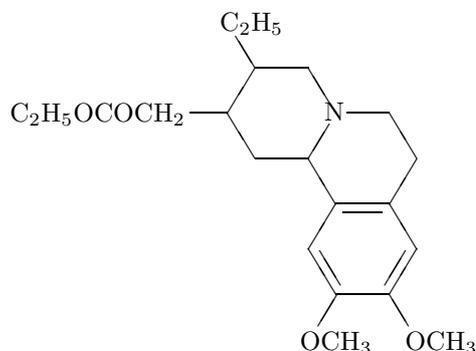
which typesets the following structure,



The horizontally opposite formula can be drawn by the combination of `\sixfusehi` (in place of `\sixfuseh`) and `\decaheterohi` (in place of `\decaheteroh`) with slight changes concerning the handedness of substituents, i.e.

```
\decaheterohi [fhk%
{c\sixfusehi [] {1==\null}%
{3==C$_2$H$_5$;4==C$_2$H$_5$OCOCH$_2$}{F}}]{3==N}{6==OCH$_3$;7==OCH$_3$}
```

Thereby we have



6.2 Five-membered Fusing Units

6.2.1 Vertical Units of Normal and Inverse Types

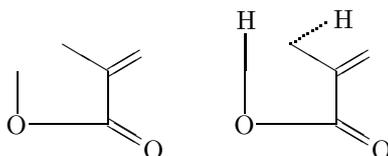
To obtain a vertical five-membered building block, we can use `\fivefusev` and `\fivefusevi`:

```
\fivefusev [BONDLIST] {ATOMLIST}{SUBSLIST}{FUSE}
\fivefusevi [BONDLIST] {ATOMLIST}{SUBSLIST}{FUSE}
```

where the argument FUSE is an alphabetical character (a–e) or the uppercase counterpart (A–E), each of which is a bond specifier representing one bond to be omitted. The other specifications have the same formats as found in the preceding section.

The following example (left) gives the use of the `\fivefusevi` command by itself, where its SUBSLIST contains some substituents:

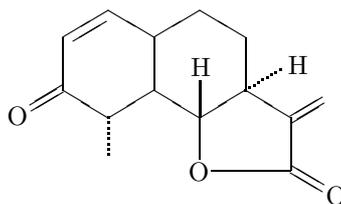
```
\fivefusevi{4==O}{2D==;3D==O}{E} \hspace 3cm
\fivefusevi{4==O}{1GA==H;5GB==H;2D==;3D==O}{E}
```



To show hydrogen substitution at the fused positions, we add the designation of `1GA==H;5GB==H` to the SUBSLIST of the `\fivefusevi` command (right above). Then, the latter code is written in the BONDLIST of a command `\decalinev`, as found in the code:

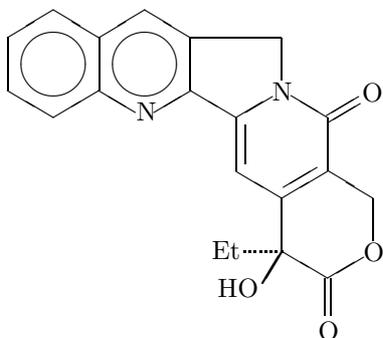
```
\decalinev[h{c\fivefusevi{4==O}{1GA==H;5GB==H;2D==;3D==O}{E}}]
{6D==O;5A==;0FB==;0GA==H}
```

Thereby, we obtain



Fusing units such as `\fivefusev` can be multiply nested in itself and in other types of fusing units. The following example shows such a triply-nested case.

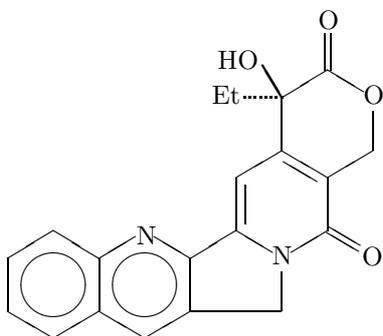
```
\decaheterovi[AB%
{b\fivefusev[{a\sixfusev[ce%
{c\sixfusev{3==0}{4D==0;5SB==HO;5SA==Et}{F}}]{1==\null}{2D==0}{f}}}%
{2==N}{D}}]{1==N}{}
```



When all of the commands in the above code are changed into the inverse counterparts with a suffix 'v' in place of 'vi' (i.e. `\decaheterovi` to `\decaheterov`; `\fivefusev` and `\fivefusevi`; and `\sixfusev` to `\sixfusevi`), the code is transformed into another code,

```
\decaheterov[AB%
{b\fivefusevi[{a\sixfusevi[ce%
{c\sixfusevi{3==0}{4D==0;5SB==HO;5SA==Et}{F}}]{1==\null}{2D==0}{f}}}%
{2==N}{D}}]{1==N}{}
```

Thereby, we can obtain the formula of vertically inverse type.



6.2.2 Horizontal Units of Normal and Inverse Types

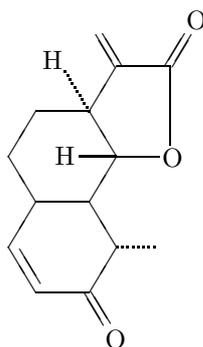
Horizontal five-membered building blocks are obtained by using `\fivefuseh` and `\fivefusehi`:

```
\fivefuseh[BONDLIST]{ATOMLIST}{SUBSLIST}{FUSE}
\fivefusehi[BONDLIST]{ATOMLIST}{SUBSLIST}{FUSE}
```

where the argument FUSE is an alphabetical character (a–e) or the uppercase counterpart (A–E), each of which is a bond specifier representing one bond to be omitted. The other specifications have the same formats as found in the preceding section.

The example given for a vertical command `\fivefusevi` is changed into the one using the horizontal counterpart `\fivefusehi`:

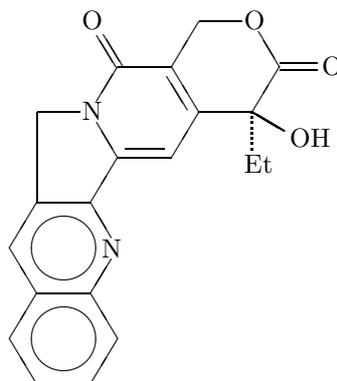
```
\decalineh[h{c\fivefusehi{4==0}{1GA==H;5GB==H;2D==;3D==0}{E}}]{5A==;6D==0}
```



Note that no changes of other designation are necessary except that `\decalineh` and `\fivefusehi` are used in place of the vertical counterpart described above.

The multiply nested example described above for drawing a structure of vertical type can be changed into the corresponding one of horizontal type, if all of the commands are changed into horizontal types (`\decaheterovi` to `\decaheterohi`; `\fivefusev` to `\fivefuseh`; and `\sixfusev` to `\sixfuseh`).

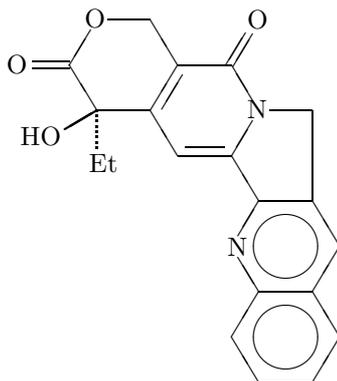
```
\decaheterohi[AB%
{b\fivefuseh[{a\sixfuseh[ce%
{c\sixfuseh{3==0}{4D==0;5SB==OH;5SA==Et}{F}}]{1==\null}{2D==0}{f}}]%
{2==N}{D}}]{1==N}{}
```



When all the commands in the above code are changed into the inverse counterparts (`\decaheterohi` to `\decaheteroh`; `\fivefuseh` and `\fivefusehi`; and `\sixfuseh` to `\sixfusehi`), the code is transformed into another code,

```
\decaheteroh[AB%
{b\fivefusehi[{{a\sixfusehi[ce%
{c\sixfusehi{3==0}{4D==0;5SB==HO;5SA==Et}{F}}]{1==\nu11}{2D==0}{f}}}%
{2==N}{D}}]{1==N}{}
```

Thereby, we can obtain the formula of horizontally inverse type.



6.3 Four-membered Fusing Units

To obtain a four-membered building block, we can use `\fourfuse`:

```
\fourfuse[BONDLIST]{ATOMLIST}{SUBSLIST}{FUSE}
```

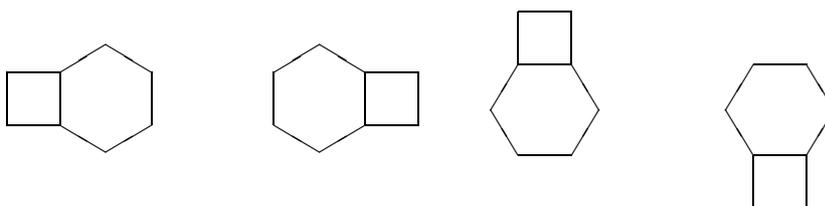
where the argument FUSE is an alphabetical character (a–d) or the uppercase counterpart (A–D), each of which is a bond specifier representing one bond to be omitted. The assignment of characters (a to d) and locants (1 to 4) for the command `\fourhetero` is applied in the same way to this case. The other specifications have the same formats as those of the command `\fourhetero`.

For example, the code,

```
\sixheterov[{{e\fourfuse}}]{{b}}]{}{}
\sixheterov[{{b\fourfuse}}]{{d}}]{}{}
\sixheteroh[{{b\fourfuse}}]{{a}}]{}{}
\sixheteroh[{{e\fourfuse}}]{{c}}]{}{}

```

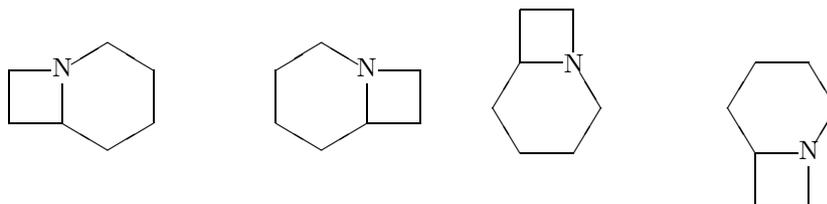
produces the following structural formulas.



A hetero atom at a fused position is designated in the ATOMLIST of `\fourfuse`, which is associated the code `\null` in the ATOMLIST of a command for drawing a mother skeleton. For example, the code

```
\sixheterov[ $\{e\backslashfourfuse\{3==N\}\}\{b\}\}\{6==\null\}\}$ 
\sixheterov[ $\{b\backslashfourfuse\{4==N\}\}\{d\}\}\{2==\null\}\}$ 
\sixheteroh[ $\{b\backslashfourfuse\{2==N\}\}\{a\}\}\{3==\null\}\}$ 
\sixheteroh[ $\{e\backslashfourfuse\{3==N\}\}\{c\}\}\{5==\null\}\}$ 
```

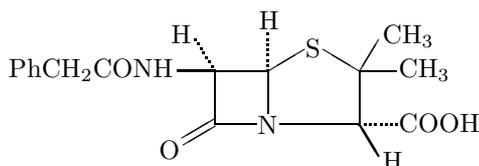
produces the following structural formulas.



Penicillin G can be drawn by using the `\fourfuse` command in the code,

```
\fiveheterovi[ $\{d\backslashfourfuse\{2==\null\}\{1D==O;4Su==PhCH\$_{2}\}\$CONH;4Sd==H\}\{b\}\}\%$ 
 $\{1==S;4==N\}\{2Sa==CH\$_{3}\}\};2Sb==CH\$_{3}\}\};3SA==COOH;3SB==H;5GA==H\}$ 
```

which typeset the following formula:



6.4 Three-membered Fusing Units

6.4.1 Vertical Units of Normal and Inverse Types

To obtain three-membered building blocks of vertical type, we can use `\threefusev` and `\threefusevi`:

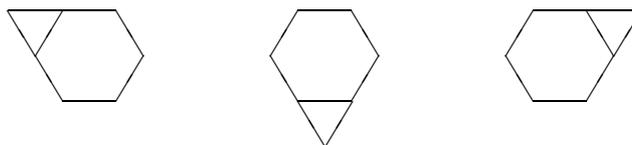
```
\threefusev[BONDLIST]{ATOMLIST}{SUBSLIST}{FUSE}
\threefusevi[BONDLIST]{ATOMLIST}{SUBSLIST}{FUSE}
```

where the argument FUSE is an alphabetical character (a–c) or the uppercase counterpart (A–C), each of which is a bond specifier representing one bond to be omitted. The assignment of characters (a to c) and locants (1 to 3) for the command `\threeheterov` or `\threeheterovi` is applied in the same way to this case. The other specifications have the same formats as those of the command `\threeheterov` or `\threeheterovi`.

For example, the code using `\threefusev`,

```
\sixheteroh[ $\{a\backslashthreefusev\}\}\{a\}\}\{\}$ 
\sixheteroh[ $\{e\backslashthreefusev\}\}\{b\}\}\{\}$ 
\sixheteroh[ $\{c\backslashthreefusev\}\}\{c\}\}\{\}$ 
```

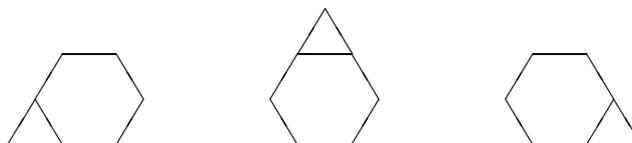
produces the following structural formulas.



The use of the inverse type is shown in the code,

```
\sixheteroh[ $\{F\}$ \threefusevi $\{\}$ \{ $\{a\}\}$ \{\}\{\}\}
\sixheteroh[ $\{B\}$ \threefusevi $\{\}$ \{ $\{b\}\}$ \{\}\{\}\}
\sixheteroh[ $\{D\}$ \threefusevi $\{\}$ \{ $\{c\}\}$ \{\}\{\}\}
```

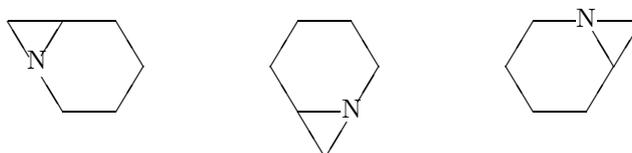
which produces the following structural formulas.



Hetero-atoms at fused positions can be typeset by designating ATOMLISTs. For example, the code,

```
\sixheteroh[ $\{a\}$ \threefusev $\{1==N\}$ \{ $\{a\}\}$ \{ $1==\null\}$ \{\}
\sixheteroh[ $\{e\}$ \threefusev $\{2==N\}$ \{ $\{b\}\}$ \{ $5==\null\}$ \{\}
\sixheteroh[ $\{c\}$ \threefusev $\{3==N\}$ \{ $\{c\}\}$ \{ $3==\null\}$ \{\}
```

produces the following structural formulas.



6.4.2 Horizontal Units of Normal and Inverse Types

Three-membered building blocks of horizontal type can be obtained by using commands with a suffix ‘h’ or ‘hi’, e.g. `\threefuseh` and `\threefusehi`:

```
\threefuseh[BONDLIST]{ATOMLIST}{SUBSLIST}{FUSE}

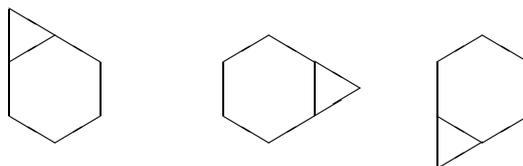

```

where the argument FUSE is an alphabetical character (a–c) or the uppercase counterpart (A–C), each of which is a bond specifier representing one bond to be omitted. The assignment of characters (a to c) and locants (1 to 3) for the command `\threeheteroh` or `\threeheterohi` is applied in the same way to this case. The other specifications have the same formats as those of the command `\threeheteroh` or `\threeheterohi`.

For example, the code using `\threefuseh`,

```
\sixheterov[ $\{F\}$ \threefuseh $\{\}$ \{ $\{a\}\}$ \{\}\{\}\}
\sixheterov[ $\{B\}$ \threefuseh $\{\}$ \{ $\{b\}\}$ \{\}\{\}\}
\sixheterov[ $\{D\}$ \threefuseh $\{\}$ \{ $\{c\}\}$ \{\}\{\}\}
```

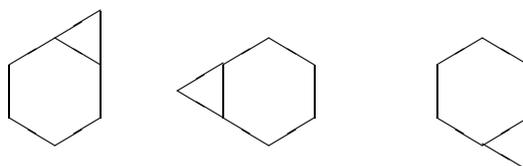
produces the following structural formulas.



The use of the inverse type is shown in the code,

```
\sixheterov[ $\{a\}$ \threefusehi $\{\}\{\}\{a\}\}\{\}\{\}$ 
\sixheterov[ $\{e\}$ \threefusehi $\{\}\{\}\{b\}\}\{\}\{\}$ 
\sixheterov[ $\{c\}$ \threefusehi $\{\}\{\}\{c\}\}\{\}\{\}$ 
```

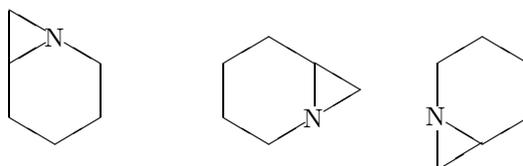
which produces the following structural formulas.



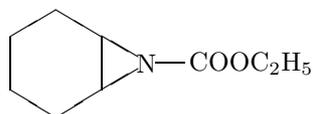
Hetero-atoms at fused positions can be typeset by designating ATOMLISTs. For example, the code,

```
\sixheterov[ $\{F\}$ \threefuseh $\{1==N\}\{\}\{a\}\}\{1==\nu\text{ull}\}\{\}$ 
\sixheterov[ $\{B\}$ \threefuseh $\{2==N\}\{\}\{b\}\}\{3==\nu\text{ull}\}\{\}$ 
\sixheterov[ $\{D\}$ \threefuseh $\{3==N\}\{\}\{c\}\}\{5==\nu\text{ull}\}\{\}$ 
```

produces the following structural formulas.



An aziridine derivative,



can be drawn by the code,

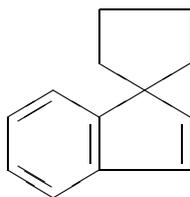
```
\sixheterov[ $\{B\}$ \threefuseh $\{1==N\}\{1==\text{COOC}\_{2}\$\text{H}\_{5}\}\{b\}\}\{\}\{\}$ 
```

Chapter 7

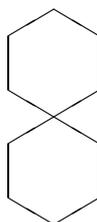
Spiro Rings

7.1 General Conventions for Spiro-Ring Attachment

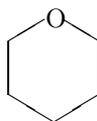
There are several ways for naming spiro compounds in the light of the IUPAC nomenclature. Rule A-41.4 allows us to use such a name as spiro[cyclopentane-1,1'-indene] for representing the following structure:



The same structure is named indene-1-spiro-1'-cyclohexane in terms of Rule A-42.1. Spiro[5.5]undecane, the name due to Rule A-41.1 and A-41.2, is alternatively referred to as cyclohexanespirocyclohexane in terms of Rule A-42.1:



where the 'cyclohexanespiro' shows the replacement of a carbon atom in a cyclohexane by another cyclohexane ring. These rules essentially have the same methodology as the IUPAC replacement nomenclature, e.g., oxacyclohexane (more formally, oxane or tetrahydropyran) for the formula



generated by the code,

```
\sixheterov[] {1==0} {}
```

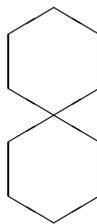
where the prefix ‘oxa’ shows the replacement of a carbon atom with an oxygen atom. Obviously, the prefix ‘cyclohexanespiro’ of the name ‘cyclohexanespirocyclohexane’ is akin to the prefix ‘oxa’ of the name ‘oxacyclohexane’ or ‘oxane’ from the viewpoint of the construction of names. Since the unit due to the latter prefix is designated by the `1==0` involved in the ATOMLIST, the former prefix can be treated in the same way. Hence, spiro compounds are drawn as follows:

1. $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ regards a spiro ring as a unit for the IUPAC replacement nomenclature, which is generated from an appropriate structure by “yl”-function.
2. the code of the unit due to the “yl”-function is added to the ATOMLIST of a mother skeleton.

Spiro[5.5]undecane is regarded as ‘cyclohexana’-cyclohexane (more formally, ‘cyclohexanespiro’-cyclohexane), as found in the code,

```
\sixheterov[] {1s==\sixheterov[] {} {4==(y1)}} {}
```

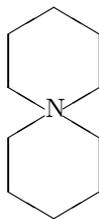
where the code `\sixheterov[] {} {4==(y1)}` produced by the “yl”-function corresponds to the suffix ‘cyclohexana’ and is written in the ATOMLIST of the outer `sixheterov` command. Thereby, we can obtain



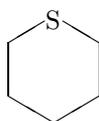
Note that the atom modifier ‘s’ in the code `1s==\sixheterov[] {} {4==(y1)}` represents no hetero-atom at the spiro position. When a hetero-atom is present at the spiro position, an atom modifier ‘h’ is used in place of ‘s’. For example, the code

```
\sixheterov[] {1h==\sixheterov[] {4==N} {4==(y1)}} {}
```

typeset the following formula:



It should be noted that the absence of such atom modifiers represents a usual replacement by a hetero atom, as found in the formula of oxane shown above or in the one of thiacyclohexane (tetrahydrothiane):

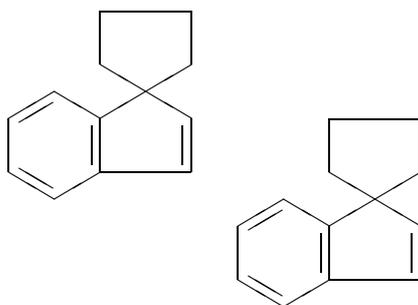


generated by the code,

```
\sixheterov[1==S]{}
```

7.2 Several Examples

Spiro[cyclopentane-1,1'-indene] described above can be drawn in two ways:

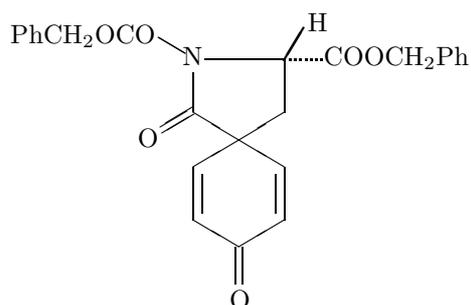


where we use two different codes:

```
\nonaheterovi[begj]{1s=\fiveheterov{}{1==(y1)}}{}
\fiveheterov{1s=\nonaheterovi[begj]{}{1==(y1)}}{}
```

which correspond to ‘cyclohexane-1-spiro-1'-indene’ and ‘indene-1-spiro-1'-cyclohexane’ (formal), respectively.

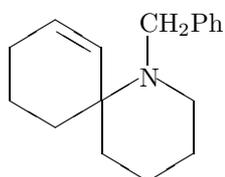
A spiro dienone



can be drawn by writing a code,

```
\sixheterov[be]{%
1s==\fiveheterov{4==N}{1==(y1);3SB==H;3SA==COOCH$_{2}$Ph;%
4==PhCH$_{2}$OCO;5D==0}}{4D==0}
```

1-Azaspiro[5.5]undecene which is the skeleton present in histrionicotoxin (Tetrahedron Lett., 1981, **22**, 2247)



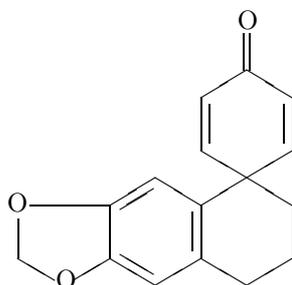
can be drawn by the code,

```
\sixheterov[] {1==N;6s==\cyclohexanev[a]{3==(y1)}}{1==CH$_{2}$Ph}
```

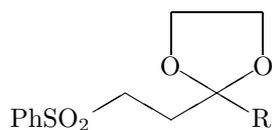
The following example shows a case to which both ring fusion and spiro attachment are applied. The code,

```
\decaheterov[fhk%
{g\fivefusev{1==0;4==0}}{b}}%
]{1s==\cyclohexanev[be]{1D==0;4==(y1)}}{}
```

gives the following formula:



A 1,3-dioxolane derivative

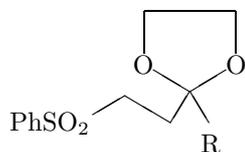


can be drawn by the code,

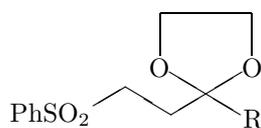
```
\fiveheterov{2==0;5==0;1s==\trimethylenei-}{3==(y1);1W==PhSO$_{2}$;3W==R}-%
```

The same compound is also drawn by usual techniques as follows:

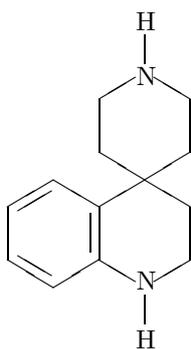
```
\fiveheterov{2==0;5==0}{1Sb==\dimethylenei-}{2==(y1);1W==PhSO$_{2}$};1Sa==R}
```



```
\fiveheterov{2==0;5==0}{1G==\dimethylenei-}{2==(y1);1W==PhSO$_{2}$};1F==R}
```



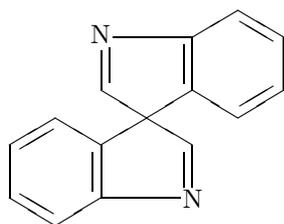
1,2,3,4-Tetrahydroquinoline-4-spiro-4'-piperidine,



can be drawn by writing a code,

```
\decaheterovi[fhk]{1==N;4s==\sixheterov[] {1==N}{4==(y1);1==H}}{1==H}
```

3,3'-Spirobi[3*H*-indole],



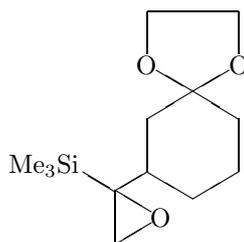
is typeset by the code,

```
\nonaheterovi[begj]{3==N;%
1s==\fiveheterov[bd{b\sixfusev[ac]{}{}{e}}]{4==N}{1==(y1)}}{}
```

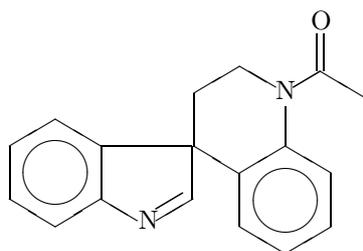
The code,

```
\sixheterov[] {1s==\fiveheterov{2==0;5==0}{1==(y1)}}{%
5==\threeheteroh{1==0}{3Sb==Me$_{3}$Si;3==(y1)}}}
```

typesets the following structure:



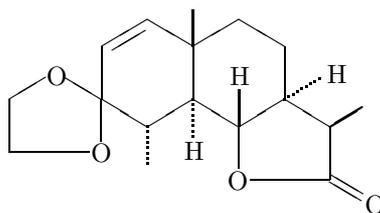
A spiro intermediate during spiro annelation (T. S. T. Wang, *Tetrahedron Lett.*, 1975, 1637),



can be drawn by the code,

```
\nonaheterov[aA]{1==N;%
3s==\decaheterovb[B]{8==N}{5==(y1);8==\dimethylenei{}{1D==0;1==(y1)}}{}}
```

A lactone intermediate containing a protected ketone (A. Grieco and M. Nishizawa, *Chem. Commun.*, 1976, 582),

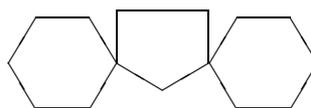


is drawn by the code,

```
\decaheterov[h{c\fivefusevi{4==0}{1GA==H;5GB==H;2B==;3D==0}{E}}]{%
6s==\fiveheterovi{1==0;3==0}{2==(y1)}}{5A==;{10}B==;9A==H}
```

7.3 Multi-Spiro Derivatives

Multi-spiro derivatives are drawn by nesting spiro function. For example, cyclohexanespirocyclopentane-3'-spirocyclohexane (Rule A-42.4),

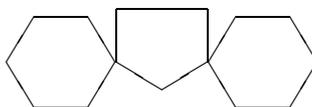


is typeset by the code,

```
\sixheteroh[] {4s==\fiveheterov{%
2s==\sixheteroh[] {}{1==(y1)}}{5==(y1)}}{}}
```

When `\fiveheterov` is a mother skeleton, such a nested command is unnecessary:

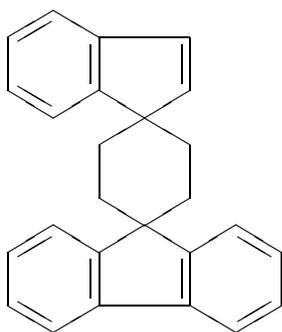
```
\fiveheterov{2s==\sixheteroh[] {}{1==(y1)}};%
5s==\sixheteroh[] {}{4==(y1)}}{}}
```



The name (Rule A-42.4), fluorene-9-spiro-1'-cyclohexane-4'-spiro-1'-indene, corresponds to the code,

```
\nonaheterovi[begj{b\sixfusev[ac] {}{}{E}}]{%
1s==\sixheterov[] {1s==\nonaheterov[begj] {}{1==(y1)}}{4==(y1)}}{}}
```

which gives



7.4 Atom Replacement

The ATOMLIST of each command is capable of accommodating a group if a sufficient space is available. For example, compare two codes,

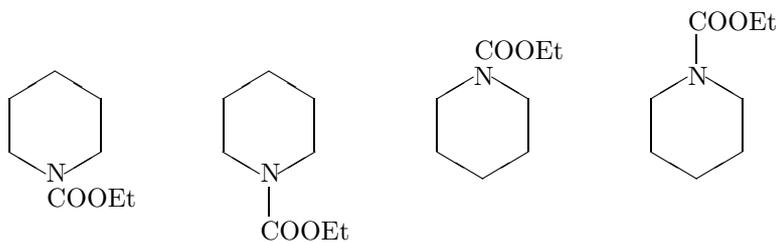
```
\sixheteroh{4==NCOOEt}{-}
\hskip 2cm
\sixheteroh{4==N}{4==COOEt}
```

generating formulas equivalent chemically to each other:



Note that the former example uses an ATOMLIST and the latter uses an SUBSLIST for describing substituents.

Even when no such space is available, the use of a command, `\upnobond` or `\downnobond`, give a solution (see $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ book pages 259–260). Compare the following formulas,

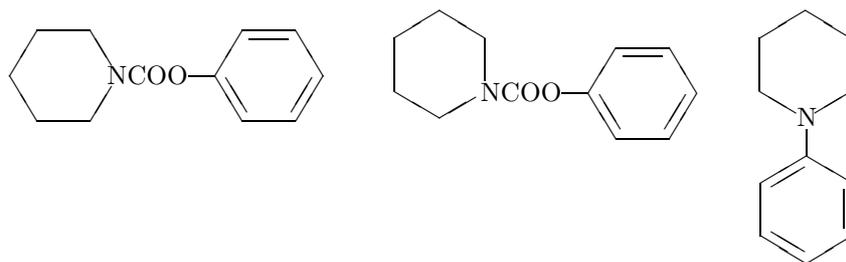


generated by the code,

```
\sixheterov{4==\downnobond{N}{COOEt}}{-}
\sixheterov{4==N}{4==COOEt}
\sixheterov{1==\upnobond{N}{COOEt}}{-}
\sixheterov{1==N}{1==COOEt}
```

These examples show that a substituent (e.g. NCOOEt) can be regarded as a component for atom replacement using a ATOMLIST. This methodology can be applied to a case in which such a substituent is generated by the “yl”-function or by such a linking command as \ryl or \lyl. The following example shows the use the \ryl command in the ATOMLIST of \sixheteroh.

```
\sixheteroh{4h==\ryl(4==NCOO){4==\bzdrh{1==(y1)}}}{-}
\hskip2cm
\sixheterov{3h==\ryl(4==NCOO){4==\bzdrh{1==(y1)}}}{-}
\hskip2cm
\sixheterov{4h==\ryl(0==N){8==\bzdrv{1==(y1)}}}{-}
```

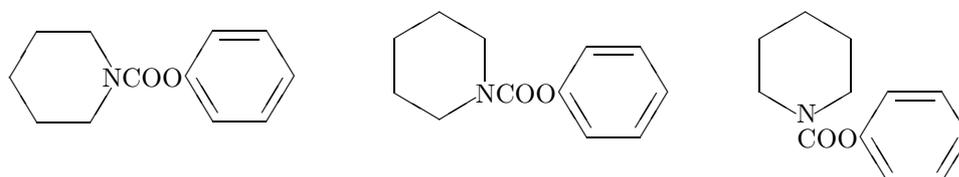


A bond between a COO unit and a phenyl group is frequently omitted. For this purpose, we use command \ayl defined as

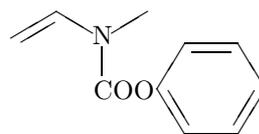
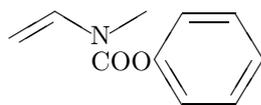
```
\makeatletter
\def\ayl{\@ifnextchar({\@ayl@}{\@ayl@(10,40)}}
\def\@ayl@(#1,#2)#3{%
\begingroup\yl@xdiff=0 \yl@ydiff=0%
\kern#1\unitlength\raise#2\unitlength\hbox to0pt{#3\hss}%
\endgroup}
\makeatother
```

Thereby, we have the following examples.

```
\sixheteroh{4==NCOO\ayl{\bzdrh{1==(y1)}}}{-}
\hskip2cm
\sixheterov{3==NCOO\ayl{\bzdrh{1==(y1)}}}{-}
\hskip2cm
\sixheterov{4==\downnobond{N}{COO\ayl{\bzdrh{1==(y1)}}}}{-}
```



```
\tetramethylene[a]{3==\downnobond{N}{COO\ayl{\bzdrh{1==(y1)}}}}{-}
\hskip2cm
\tetramethylene[a]{3==N}{3==COO\ayl{\bzdrh{1==(y1)}}}
```



Chapter 8

Added Commands

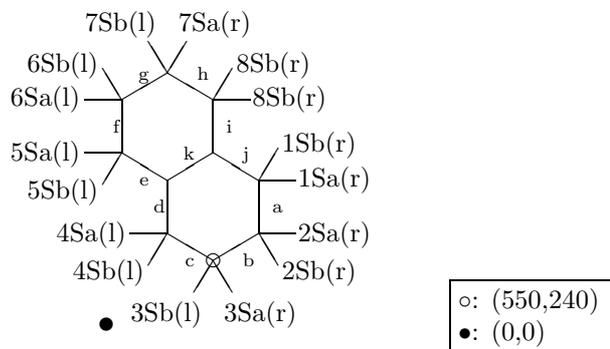
8.1 Six-six Fused Carbocycles

8.1.1 Vertical-Bottom Forms of Decaline Derivatives

The macro `\decalinevb` is used to draw decaline derivatives of vertical-bottom type (added to `carom.sty`), where the numbering of atoms is given from the bottom to the left-upper part. The word “vertical” means that each benzene ring is a vertical type. The word “bottom” means that the benzene ring with young locant numbers is located at the bottom. The format of this command is as follows:

```
\decalinevb[BONDLIST]{SUBSLIST}
```

Locant numbers for designating substitution positions and characters for showing bonds to be doubled are represented by the following diagram:

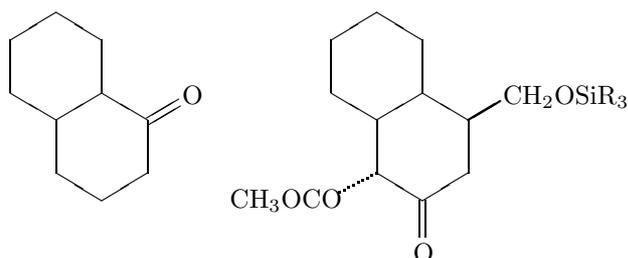


The handedness for each oriented or double-sided position is shown with a character set in parentheses. The option argument `BONDLIST` is based on the assignment of characters (a–k) to respective bonds as shown in the above diagram. A bond modifier in the argument `SUBSLIST` for $n = 1-8$ can be one of bond modifiers shown in Table 2.1. The substitution at the bridgehead positions is designated as shown in Table 4.3 of \LaTeX book.

Example:

```
\decalinevb{1D==0;0FB==H;0GA==H} \quad  
\decalinevb{1B==CH$_{2}$OSiR$_{3}$;3D==0;4A==CH$_{3}$OC0;%  
0FB==CH$_{3}$;0GA==H}
```

These commands produce:



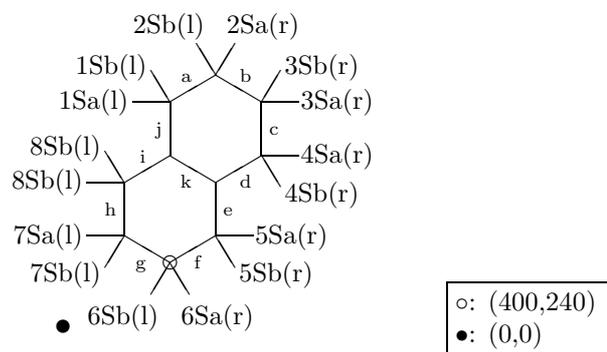
The related commands, `\naphdrv` and `\tetralinevb`, have been defined on the basis of the command `\decalinevb`.

8.1.2 Vertical-Top Forms of Decaline Derivatives

The macro `\decalinevt` (added to `carom.sty`) is used for drawing decaline derivatives of vertical-bottom type (numbering from the top to the left-down part). The word “vertical” means that each benzene ring is a vertical type. The word “top” means that the benzene ring with young locant numbers is located at the top. The format of this command is as follows:

`\decalinevt [BONDLIST] {SUBSLIST}`

Locant numbers for designating substitution positions and characters for showing bonds to be doubled are represented by the following diagram:

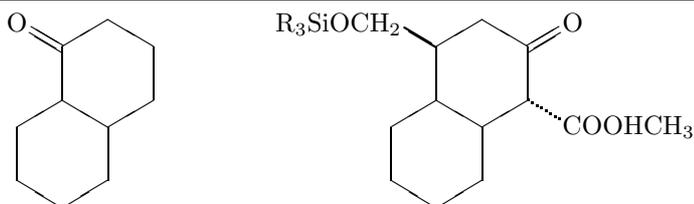


The handedness for each oriented or double-sided position is shown with a character set in parentheses. The option argument `BONDLIST` is based on the assignment of characters (a–k) to respective bonds as shown in the above diagram. A bond modifier in the argument `SUBSLIST` for $n = 1-8$ can be one of bond modifiers shown in Table 2.1. The substitution at the bridgehead positions is designated as shown in Table 4.3 of $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ book.

Example:

```
\decalinevt{1D==O;0FB==H;OGA==H} \quad
\decalinevt{1B==R$_{3}$SiOCH$_{2}$;3D==O;4A==COOHCH$_{3}$;%
0FB==CH$_{3}$;OGA==H}
```

These commands produce:



The related commands, `\naphdrv` and `\tetralinevt`, have been defined on the basis of the command `\decalinevt`.

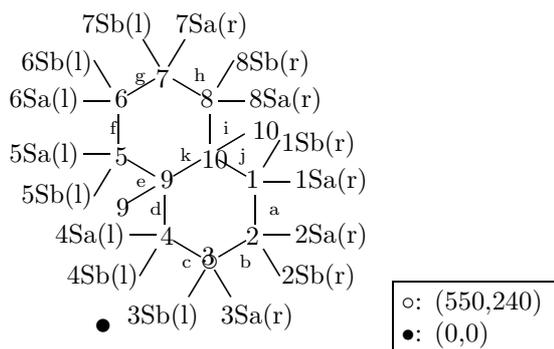
8.2 Six-six Fused Heterocycles

8.2.1 Vertical-Bottom Forms

The macro `\decaheterovb` is generally used to draw six-six-fused heterocycles of vertical-bottom type (`hetarom.sty`).

```
\decaheterovb[BONDLIST]{ATOMLIST}{SUBSLIST}
```

Locant numbers for designating substitution positions as well as characters for setting double bonds are shown in the following diagram:



The handedness for each oriented or double-sided position is shown with a character set in parentheses. The optional argument `BONDLIST` is used to specify a bond pattern.

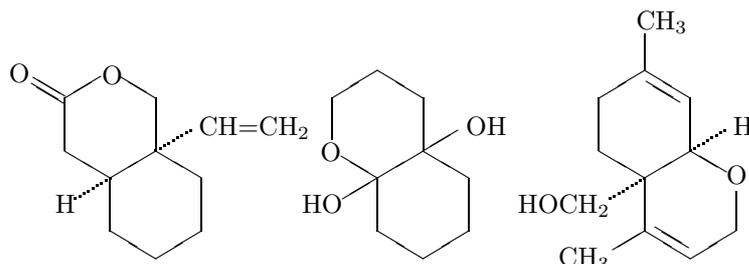
The argument `ATOMLIST` has a similar format concerning the positions of $n = 1$ to 8. A hetero-atom on the 4a-position is designated to be `4a==N` or `9==N`; and a hetero-atom on the 8a-position is given as to be `8a==N` or `{{10}}==N`.

The argument `SUBSLIST` for this macro takes a general format, in which the modifiers listed in Table 2.1 are used. Note that 9 and 10 should be used for designating 4a and 8a positions.

Example:

```
\decaheterovb[] {7==0}{6D==0;9A==H;{{10}A}==CH=CH$_{2}$}
\decaheterovb[] {5==0}{9==HO;{{10}}==OH}
\decaheterovb[ch] {1==0}{9A==HOCH$_{2}$;{{10}A}==H;%
4==CH$_{3}$;7==CH$_{3}$}
```

produce the following structures:



Macros for drawing related fused heterocycles are also defined. The formats of these commands are as follows:

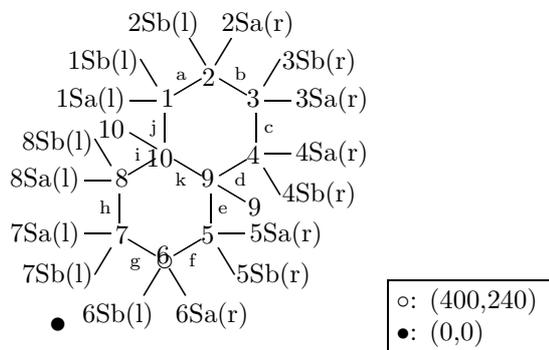
```
\quinolinevb[BONDLIST]{SUBSLIST}
\isoquinolinevb[BONDLIST]{SUBSLIST}
\quinoxalinevb[BONDLIST]{SUBSLIST}
\quinazolinevb[BONDLIST]{SUBSLIST}
\cinnoilinevb[BONDLIST]{SUBSLIST}
\pteridinevb[BONDLIST]{SUBSLIST}
```

8.2.2 Vertical-Top Forms

The macro `\decaheterovt` is generally used to draw six-six-fused heterocycles of vertical-top type (`het-arom.sty`).

```
\decaheterovt[BONDLIST]{ATOMLIST}{SUBSLIST}
```

Locant numbers for designating substitution positions as well as characters for setting double bonds are shown in the following diagram:



The handedness for each oriented or double-sided position is shown with a character set in parentheses. The optional argument `BONDLIST` is used to specify a bond pattern.

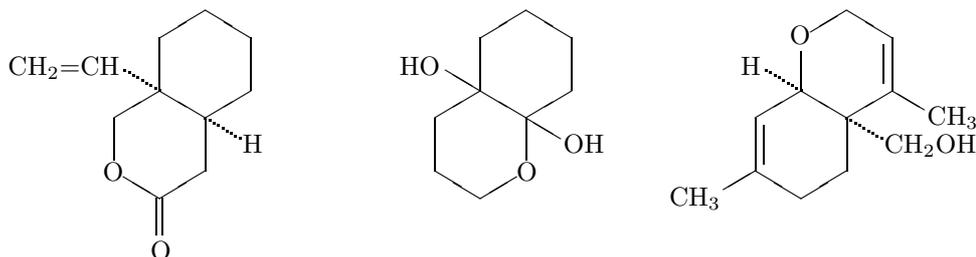
The argument `ATOMLIST` has a similar format concerning the positions of $n = 1$ to 8. A hetero-atom on the 4a-position is designated to be `4a==N` or `9==N`; and a hetero-atom on the 8a-position is given as to be `8a==N` or `{{10}}==N`.

The argument `SUBSLIST` for this macro takes a general format, in which the modifiers listed in Table 2.1 are used. Note that 9 and 10 should be used for designating 4a and 8a positions.

Example:

```
\decaheterovt[] {7==0}{6D==0;9A==H;{{10}A}==CH$_{2}$=CH}
\decaheterovt[] {5==0}{9==OH;{{10}}==HO}
\decaheterovt[ch] {1==0}{9A==CH$_{2}$OH;{{10}A}==H;%
4==CH$_{3}$;7==CH$_{3}$}
```

produce the following structures:



Macros for drawing related fused heterocycles are also defined. The formats of these commands are as follows:

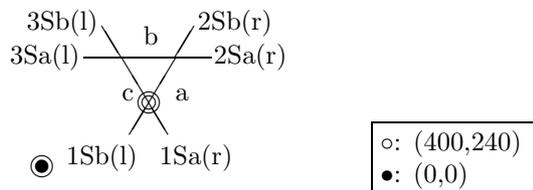
```
\quinolinevt[BONDLIST]{SUBSLIST}
\isoquinolinevt[BONDLIST]{SUBSLIST}
\quinoxalinevt[BONDLIST]{SUBSLIST}
\quinazolinevt[BONDLIST]{SUBSLIST}
\cinnoilinevt[BONDLIST]{SUBSLIST}
\pteridinevt[BONDLIST]{SUBSLIST}
```

8.3 Three-Membered Carbocycles

The macro `\cyclopropanevt` (the same command as `\cyclopropane`) for drawing three-membered carbocycles has the following format (`lowcycle.sty`)

```
\cyclopropanevt[BONDLIST]{SUBSLIST}
```

The locant numbering (1-3) and the bond description (a-c) are common as shown in the following diagram:



For the arguments, `BONDLIST` and `SUBSLIST`, see `XYMTEXbook`.

Example:

```
\cyclopropanevt{2Sa==COOCH$_{3}$;2Sb==COOCH$_{3}$}\quad
\cyclopropanevt{2Sa==COOH;2Sb==COOH}\quad\quad
\cyclopropanevt{3Sa==H$_{3}$C;3Sb==H$_{3}$C}
```

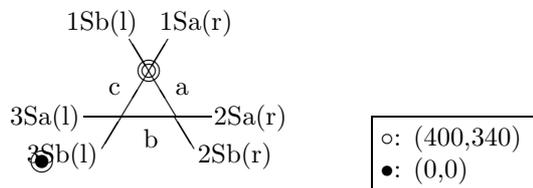
produce the following structures:



The macro `\cyclopropanevi` (the same command as `\cyclopropanei`) for drawing three-membered carbocycles of inverse type has the following format (`lowcycle.sty`)

`\cyclopropanevi [BONDLIST] {SUBSLIST}`

The following diagram shows The locant numbering (1–3) and the bond description (a–c):

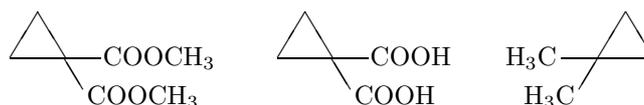


For the arguments, `BONDLIST` and `SUBSLIST`, see the counterparts of `\cyclopropane` described in $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ book.

Example:

```
\cyclopropanevi{2Sa==COOCH$_{3}$;2Sb==COOCH$_{3}$}\quad
\cyclopropanevi{2Sa==COOH;2Sb==COOH}\quad\quad
\cyclopropanevi{3Sa==H$_{3}$C;3Sb==H$_{3}$C}
```

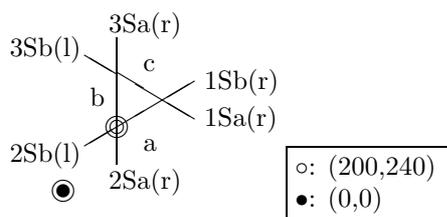
produce the following structures:



The macro `\cyclopropaneh` for drawing three-membered carbocycles of horizontal type has the following format (`lowcycle.sty`)

`\cyclopropaneh [BONDLIST] {SUBSLIST}`

The locant numbering (1–3) and the bond description (a–c) are common as shown in the following diagram:

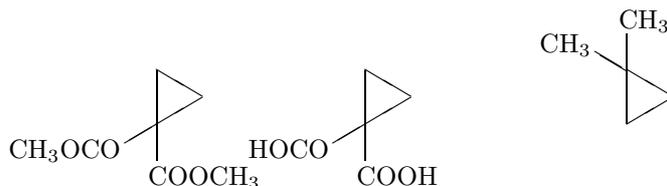


For the arguments, `BONDLIST` and `SUBSLIST`, see $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ book.

Example:

```
\cyclopropaneh{2Sa==COOCH$_{3}$;2Sb==CH$_{3}$OC}\quad
\cyclopropaneh{2Sa==COOH;2Sb==HOC}\quad\quad
\cyclopropaneh{3Sa==CH$_{3}$;3Sb==CH$_{3}$}
```

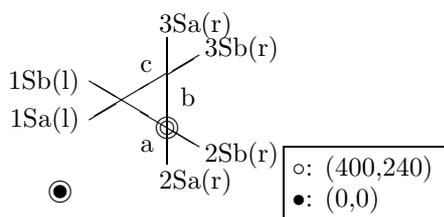
produce the following structures:



The macro `\cyclopropanehi` for drawing three-membered carbocycles of inverse horizontal type has the following format (`lowcycle.sty`)

```
\cyclopropanehi [BONDLIST] {SUBSLIST}
```

The locant numbering (1–3) and the bond description (a–c) are common as shown in the following diagram:



For the arguments, `BONDLIST` and `SUBSLIST`, see `XYTEXbook`.

Example:

```
\cyclopropanehi{2Sa==COOCH$_{3}$;2Sb==COOCH$_{3}$}\quad
\cyclopropanehi{2Sa==COOH;2Sb==COOH}\quad\quad
\cyclopropanehi{3Sa==CH$_{3}$;3Sb==CH$_{3}$}
```

produce the following structures:

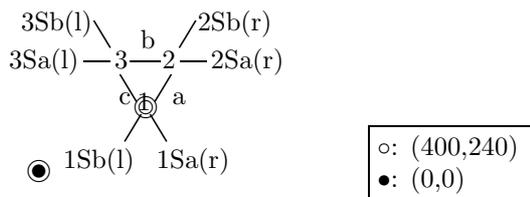


8.4 Three-Membered Heterocycles

The macro `\threeheterov` (the same command as `\threehetero`) for drawing three-membered heterocycles has the following format (`hetarom.sty`)

```
\threeheterov [BONDLIST] {ATOMLIST} {SUBSLIST}
```

The locant numbering (1–3) and the bond description (a–c) are common as shown in the following diagram:

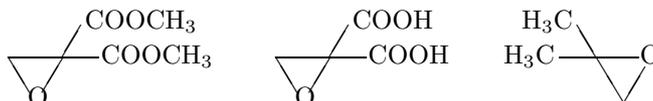


For the arguments, BONDLIST and SUBSLIST, see $\X\TeX$ book.

Example:

```
\threeheterov{1==0}{2Sa==COOCH$_{3}$;2Sb==COOCH$_{3}$}\quad
\threeheterov{1==0}{2Sa==COOH;2Sb==COOH}\quad\quad
\threeheterov{2==0}{3Sa==H$_{3}$C;3Sb==H$_{3}$C}
```

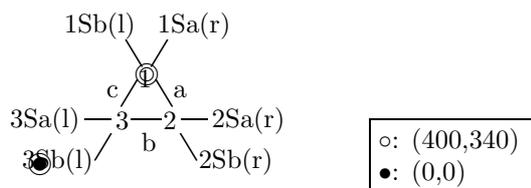
produce the following structures:



The macro `\threeheterovi` (the same command as `\threeheteroi`) for drawing three-membered heterocycles of inverse type has the following format (`hetarom.sty`)

```
\threeheterovi[BONDLIST]{ATOMLIST}{SUBSLIST}
```

The following diagram shows The locant numbering (1–3) and the bond description (a–c):

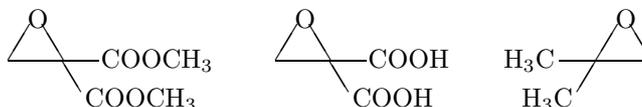


For the arguments, BONDLIST and SUBSLIST, see the counterparts of `\threehetero` described in $\X\TeX$ book.

Example:

```
\threeheterovi{1==0}{2Sa==COOCH$_{3}$;2Sb==COOCH$_{3}$}\quad
\threeheterovi{1==0}{2Sa==COOH;2Sb==COOH}\quad\quad
\threeheterovi{1==0}{3Sa==H$_{3}$C;3Sb==H$_{3}$C}
```

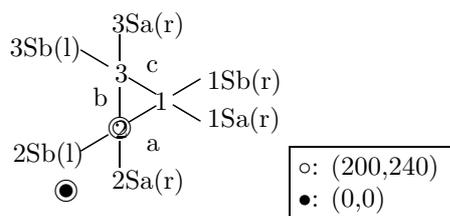
produce the following structures:



The macro `\threeheteroh` for drawing three-membered heterocycles of horizontal type has the following format (`hetaromh.sty`)

```
\threeheteroh[BONDLIST]{ATOMLIST}{SUBSLIST}
```

The locant numbering (1–3) and the bond description (a–c) are common as shown in the following diagram:

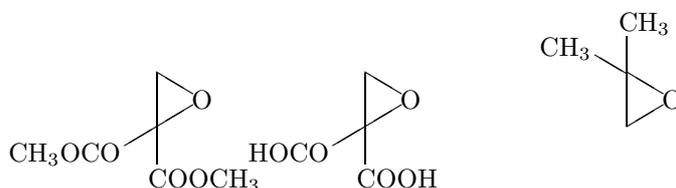


For the arguments, BONDLIST and SUBSLIST, see \XfMTExbook .

Example:

```
\threeheteroh{1==0}{2Sa==COOCH$_{3}$;2Sb==CH$_{3}$OC}\quad
\threeheteroh{1==0}{2Sa==COOH;2Sb==HOCO}\quad\quad
\threeheteroh{1==0}{3Sa==CH$_{3}$;3Sb==CH$_{3}$}
```

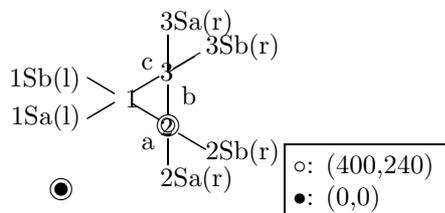
produce the following structures:



The macro `\threeheterohi` for drawing three-membered heterocycles of inverse horizontal type has the following format (`hetatomh.sty`)

```
\threeheterohi [BONDLIST]{ATOMLIST}{SUBSLIST}
```

The locant numbering (1–3) and the bond description (a–c) are common as shown in the following diagram:

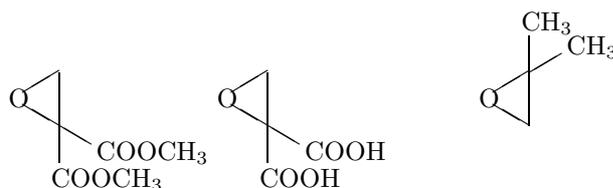


For the arguments, BONDLIST and SUBSLIST, see \XfMTExbook .

Example:

```
\threeheterohi{1==0}{2Sa==COOCH$_{3}$;2Sb==COOCH$_{3}$}\quad
\threeheterohi{1==0}{2Sa==COOH;2Sb==COOH}\quad\quad
\threeheterohi{1==0}{3Sa==CH$_{3}$;3Sb==CH$_{3}$}
```

produce the following structures:



8.5 Aliphatic Moieties

8.5.1 Trigonal Units

In addition to the macros `\Rtrigonal` and `\Ltrigonal` (see $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ book), macros for broader bond angles, `\Rtrigonal` and `\Ltrigonal`, are added to the `aliph` package (`aliph.sty`). The formats of these commands are as follows:

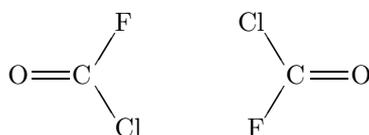
```
\Rtrigonal[AUXLIST]{SUBSLIST}
\Ltrigonal[AUXLIST]{SUBSLIST}
```

The bond angles of 2–0–3 are 120° in the trigonal units printed with these commands. The arguments `AUXLIST` and `SUBSLIST` are the same as those of `\tetrahedral`.

Example:

```
\Rtrigonal{0==C;1D==0;2==Cl;3==F}\quad
\Ltrigonal{0==C;1D==0;2==Cl;3==F}
```

produce the following structures:

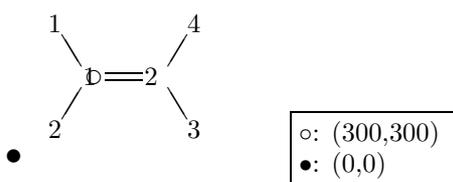


8.5.2 Ethylenes

The macro `\Ethyleneh` or `\Ethylen` is a broader-angled counterpart of the macro `\ethyleneh` or `\ethylene` (see $\hat{X}\hat{M}\hat{T}\hat{E}\hat{X}$ book), which is used to draw ethylene derivatives with angles 120° (`aliph.sty`). The format of this command is as follows:

```
\Ethyleneh[BONDLIST]{ATOMLIST}{SUBSLIST}
\Ethylen[BONDLIST]{ATOMLIST}{SUBSLIST}
```

The following diagram shows the numbering for designating substitution positions:

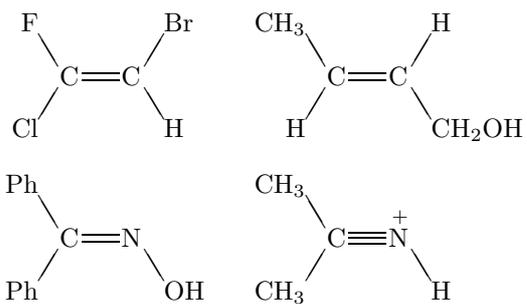


The argument `BONSLIST` is used for giving the C–C bond. The argument `ATOMLIST` is used for giving central atoms. The argument `SUBSLIST` is the same as that of `\tetrahedral`.

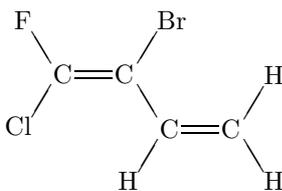
Example:

```
\Ethyleneh{1==C;2==C}{1==F;2==Cl;3==H;4==Br}\quad
\Ethyleneh{1==C;2==C}{1==CH$_{3}$;2==H;3==CH$_{2}$OH;4==H}\par
\Ethyleneh{1==C;2==N}{1==Ph;2==Ph;3==OH}\quad
\Ethyleneh[t{2+}]{1==C;2==N}{1==CH$_{3}$;2==CH$_{3}$;3==H}
```

produce the following structures:



A butadiene derivative,



can be drawn by the code,

```
\Ethyleneh{1==C;2==C}{1==F;2==Cl;4==Br;%
3==\Ethyleneh{1==C;2==C}{1==(y1);2==H;3==H;4==H}}
```


Chapter 9

Zigzag Polymethylene Skeletons

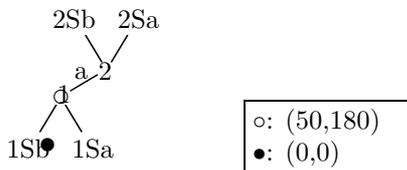
9.1 Dimethylenes

The macro `\dimethylene` has two arguments `ATOMLIST` and `SUBSLIST` as well as an optional argument `BONDLIST` (`methylen.sty`):

```
\dimethylene[BONDLIST]{ATOMLIST}{SUBSLIST}
```

The `BONDLIST` argument contains one character `a` or `A`, each of which indicates the presence of an inner (endo-chain) double bond on the corresponding position. A lowercase letter is used to typeset a double bond at a lower-side of an outer skeletal bond, while an uppercase letter typesets a double bond at an upper-side of an outer skeletal bond (Note that the option ‘`A`’ represents an aromatic circle in commands `\sixheterov` etc.). The `ATOMLIST` and `SUBSLIST` arguments follow the conventions of the \TeX system.

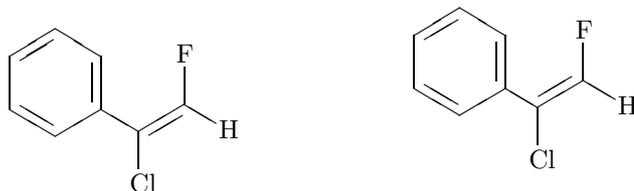
The following diagram shows the numbering for designating substitution positions:



Lowercase vs. uppercase letters (`‘a’` vs. `‘A’`) in the `BONDLIST` of the `\dimethylene` command designate the position of an bond added to the bond `‘a’`, as shown in the code,

```
\dimethylene[a]{1W==\bzdrv{3==(y1)};1==Cl;2W==H;2==F}  
\hskip2cm  
\bzdrv{3==\dimethylene[A]{1W==(y1)};1==Cl;2W==H;2==F}
```

which typesets the following formulas:



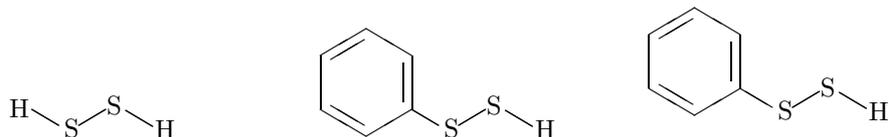
In addition to the standard bond modifiers listed in Table 2.1, the terminal positions of the command `\dimethylene` can take a bond modifier `‘W’`. For example, the code,

```

\dimethylene{1==S;2==S}{1W==H;2W==H}
\hspace{4cm}
\dimethylene{1==S;2==S}{1W==\bzdrv{3==(y1)};2W==H}
\hspace{1cm}
\bzdrv{3==\dimethylene{1==S;2==S}{1W==(y1);2W==H}}

```

generates the following formulas:

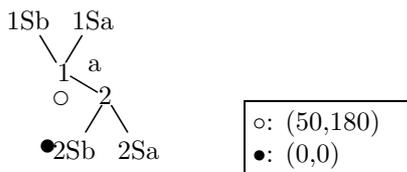


where the ATOMLIST is used to set two sulfur atoms in the dimethylene chain.

The macro `\dimethylenei` is the inverse counterpart of `\dimethylene`, where arguments ATOMLIST, SUBSLIST, and BONDLIST take such common formats as found in the definition of the latter (`methylen.sty`):

```
\dimethylenei[BONDLIST]{ATOMLIST}{SUBSLIST}
```

The following diagram shows the numbering for designating substitution positions:

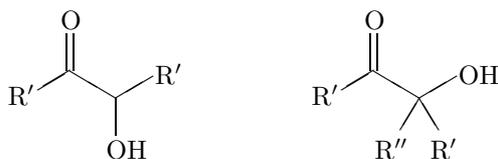


Note that the coordinate of position no. 1 is (50, 283), where $180 + 103 = 283$. The following example shows a specification of the SUBSLIST.

```

\dimethylenei{}{1W==R$^{\prime}$;2W==R$^{\prime}$;1D==O;2==OH}
\hspace{3cm}
\dimethylenei{}{1W==R$^{\prime}$;%
2Sa==R$^{\prime}$;2Sb==R$^{\prime}$;1D==O;2W==OH}

```



9.2 Trimethylenes

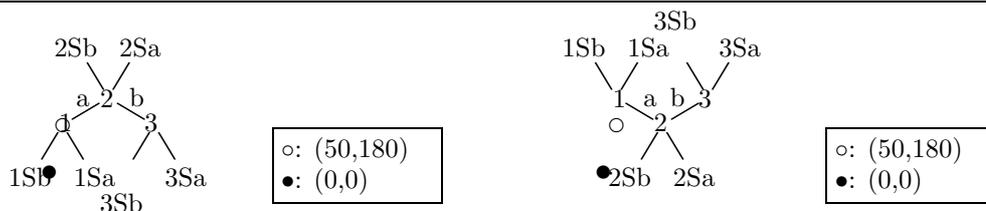
The macros `\trimethylene` and `\trimethylenei` and have two arguments ATOMLIST and SUBSLIST as well as an optional argument BONDLIST (`methylen.sty`).

```

\trimethylene[BONDLIST]{ATOMLIST}{SUBSLIST}
\trimethylenei[BONDLIST]{ATOMLIST}{SUBSLIST}

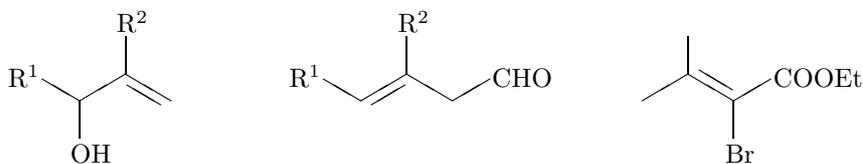
```

The following diagram shows the numbering for designating substitution positions:

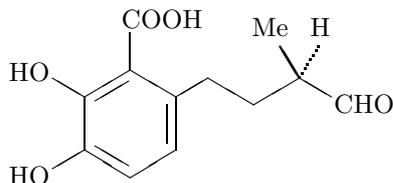


Examples:

```
\trimethylene[b]{1W==R$^{1}$;1==OH;2==R$^{2}$}
\hskip2cm
\trimethylene[a]{1W==R$^{1}$;2==R$^{2}$;3W==CHO}
\hskip2cm
\trimethylene[B]{2==\null;3W==COOEt;3==Br}
```



```
\trimethylenei{}{1W==\bzdrv{2==(y1);1==COOH;5==HO;6==HO};%
3W==CHO;3SA==H;3SB==Me}
```



9.3 Tetramethylenes

The macros `\tetramethylene` and `\tetramethylenei` have two arguments `ATOMLIST` and `SUBSLIST` as well as an optional argument `BONDLIST` (methylen.sty).

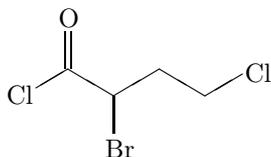
```
\tetramethylene[BONDLIST]{ATOMLIST}{SUBSLIST}
\tetramethylenei[BONDLIST]{ATOMLIST}{SUBSLIST}
```

The following diagram shows the numbering for designating substitution positions:

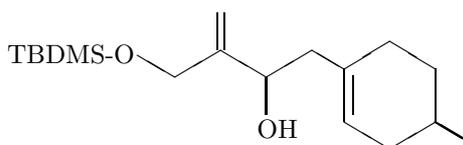


Examples:

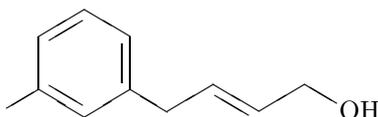
```
\tetramethylenei-{}{1W==Cl;1D==O;2B==Br;4W==Cl}
```



```
\tetramethylene-{}{1W==TBDMS-O;2D==\null;3B==OH;%
4W==\cyclohexanev[e]{6==(y1);3B==\null}}
```



```
\tetramethylene[b]-{}{1W==\bzdrv{5==\null;3==(y1)};4W==OH}
```

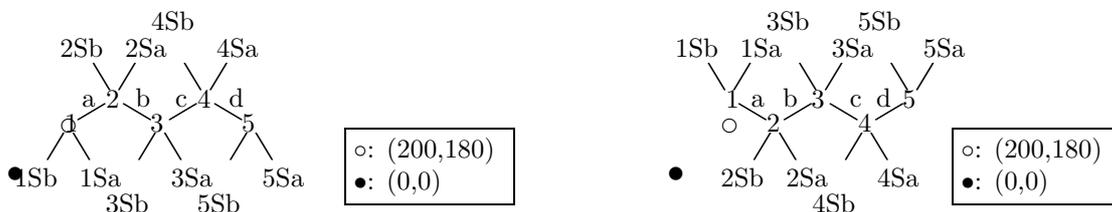


9.4 Pentamethylenes

The macros `\pentamethylene` and `\pentamethylenei` have two arguments `ATOMLIST` and `SUBSLIST` as well as an optional argument `BONDLIST` (`methylen.sty`).

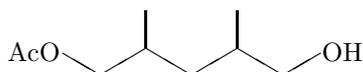
```
\pentamethylene[BONDLIST]{ATOMLIST}{SUBSLIST}
\pentamethylenei[BONDLIST]{ATOMLIST}{SUBSLIST}
```

The following diagram shows the numbering for designating substitution positions:

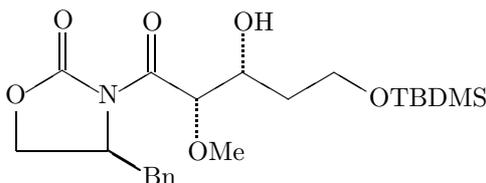


Examples:

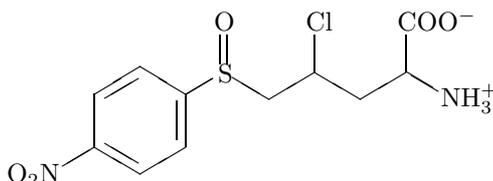
```
\pentamethylene}{1W==AcO;2B==\null;4B==\null;5W==OH}
```



```
\pentamethylenei}{1W==\fiveheterovi{2==N;5==O}{1D==0;2==(y1);3B==Bn};%
1D==0;2A==OMe;3A==OH;5W==OTBDMS}
```



```
\pentamethylenei{1==S}{1W==\bzdrv{2==(y1);5==O$_{2}$N};%
1D==0;3==Cl;5==COO$^{-}$;5W==NH$_{3}^{+}$}
```

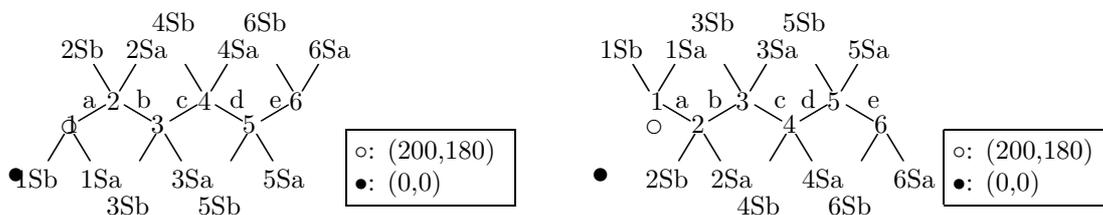


9.5 Hexamethylenes

The macros `\hexamethylene` and `\hexamethylenei` have two arguments `ATOMLIST` and `SUBSLIST` as well as an optional argument `BONDLIST` (`methylen.sty`).

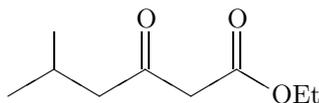
```
\hexamethylene [BONDLIST] {ATOMLIST} {SUBSLIST}
\hexamethylenei [BONDLIST] {ATOMLIST} {SUBSLIST}
```

The following diagram shows the numbering for designating substitution positions:

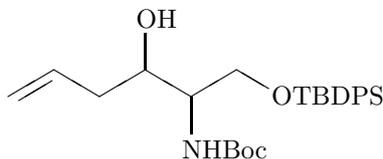


Examples:

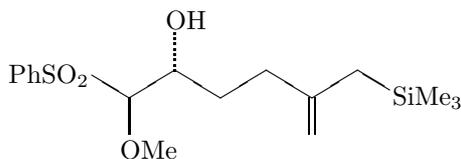
```
\hexamethylene}{2==\null;4D==0;6D==0;6W==0Et}
```



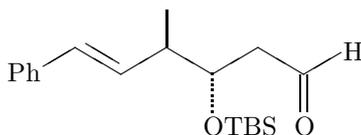
```
\hexamethylene[a]{4B==OH;5B==NHBoc;6W==OTBDPS}
```



```
\hexamethylene{1W==PhSO$_{2}$;1B==OMe;2A==OH;5D==\null;6W==SiMe$_{3}$}
```



```
\hexamethylenei[a]{1W==Ph;3B==\null;4A==OTBS;6D==0;6W==H}
```

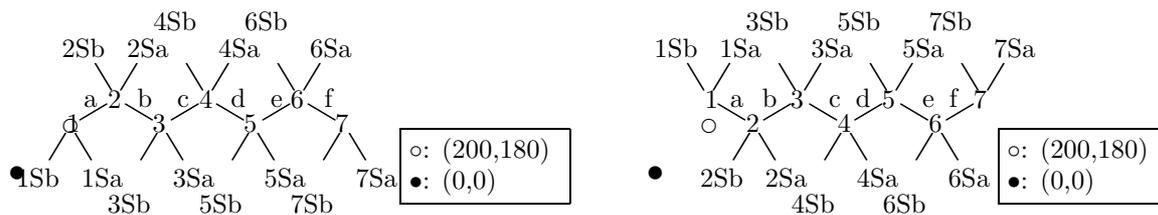


9.6 Heptamethylenes

The macros `\heptamethylene` and `\heptamethylenei` have two arguments `ATOMLIST` and `SUBSLIST` as well as an optional argument `BONDLIST` (`methylen.sty`).

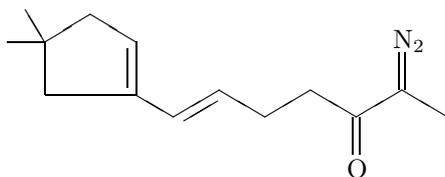
```
\heptamethylene[BONDLIST]{ATOMLIST}{SUBSLIST}
\heptamethylenei[BONDLIST]{ATOMLIST}{SUBSLIST}
```

The following diagram shows the numbering for designating substitution positions:

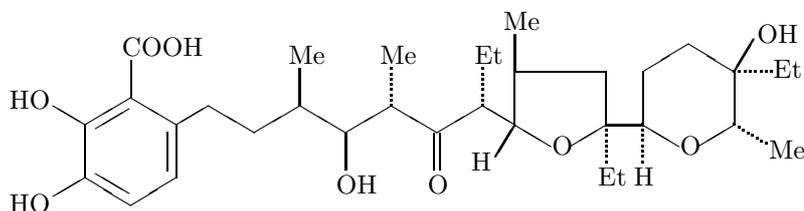


Examples:

```
\heptamethylene[a]{1W==\cyclopentanevi[b]{3==(y1);5Sa==\null;5Sb==\null};%
5D==0;6D==N$_{2}$}
```



```
\heptamethylenei{-}{1W==\bzdrv{1==COOH;2==(y1);5==HO;6==HO};%
3B==Me;4B==OH;5A==Me;6D==O;7A==Et;%
7W==\fiveheterov{1==O}{5==(y1);5SB==H;4GB==Me;2GA==Et;%
2Su==\sixheterovi{1==O}{6==(y1);6FA==H;3SB==OH;3SA==Et;2A==Me}}}
```

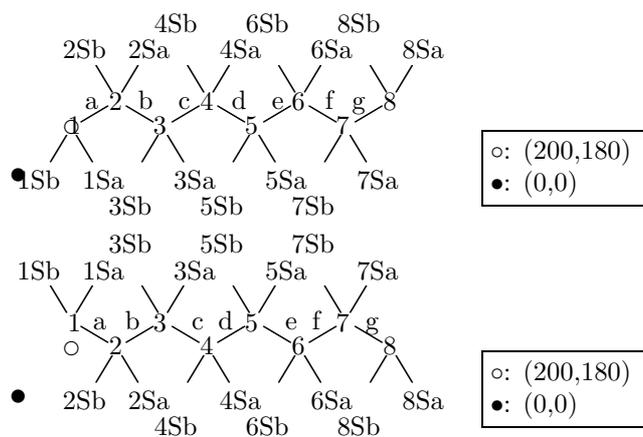


9.7 Octamethylenes

The macros `\octamethylene` and `\octamethylenei` have two arguments `ATOMLIST` and `SUBSLIST` as well as an optional argument `BONDLIST` (`methylen.sty`).

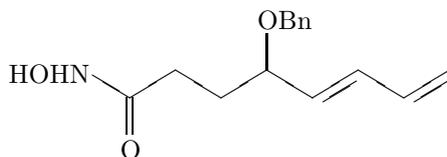
```
\octamethylene[BONDLIST]{ATOMLIST}{SUBSLIST}
\octamethylenei[BONDLIST]{ATOMLIST}{SUBSLIST}
```

The following diagram shows the numbering for designating substitution positions:

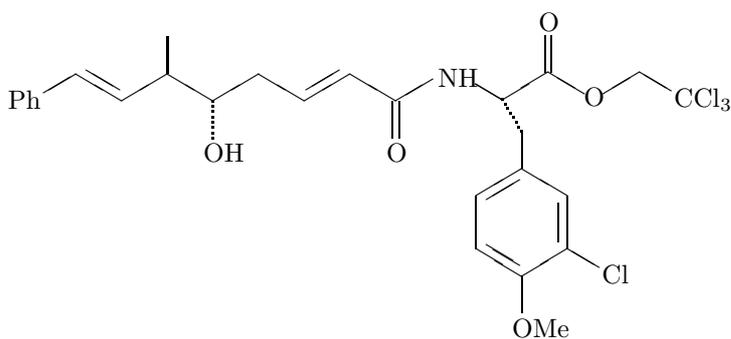


Examples:

```
\octamethylene[eg]{1W==HOHN;1D==0;4B==OBn}
```



```
\octamethylenei[af]{1W==Ph;3B==\null;4A==OH;8D==0;%
8W==\ryl(4==NH){%
5==\tetramethylene{3==0}{1==(y1);2D==0;4W==CC1$_{3}$;%
1SA==\ryl{8==\bzdrv{1==(y1);3==Cl;4==OMe}}}}
```

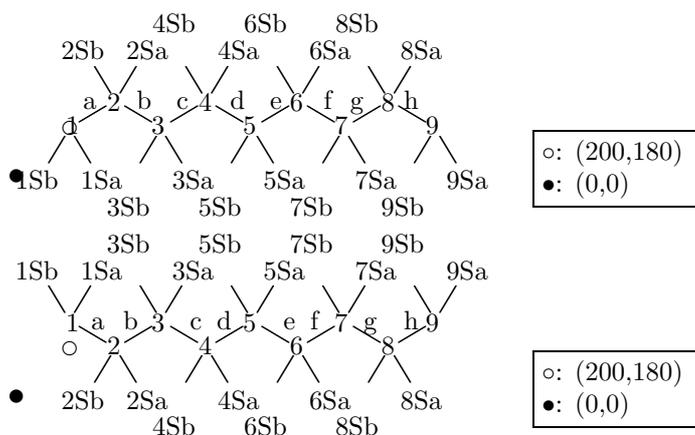


9.8 Nonamethylenes

The macros `\nonamethylene` and `\nonamethylenei` have two arguments `ATOMLIST` and `SUBSLIST` as well as an optional argument `BONDLIST` (`methylen.sty`).

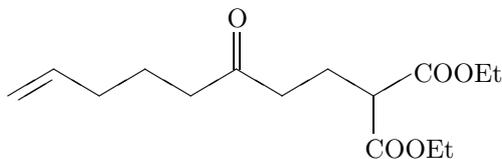
```
\nonamethylene[BONDLIST]{ATOMLIST}{SUBSLIST}
\nonamethylenei[BONDLIST]{ATOMLIST}{SUBSLIST}
```

The following diagram shows the numbering for designating substitution positions:

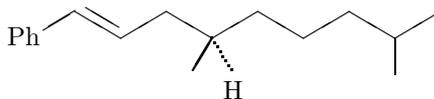


Examples:

```
\nonamethylene[a]{}{6D==O;9W==COOEt;9==COOEt}
```



```
\nonamethylenei[a]{1W==Ph;4SB==\null;4SA==H;8==\null}
```

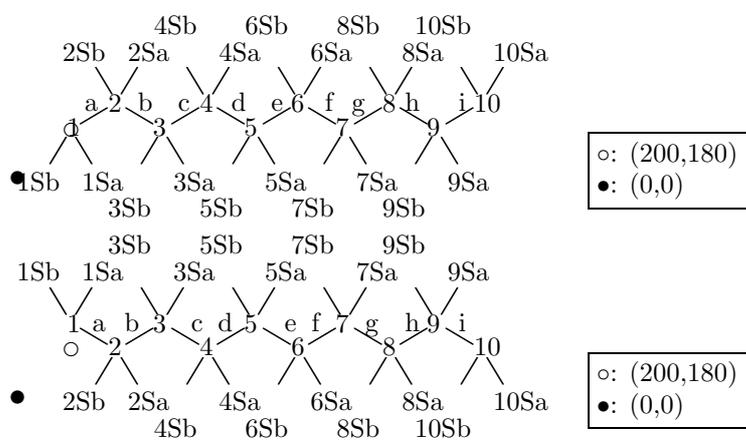


9.9 Decamethylenes

The macros `\decamethylene` and `\decamethylenei` have two arguments `ATOMLIST` and `SUBSLIST` as well as an optional argument `BONDLIST` (`methylen.sty`).

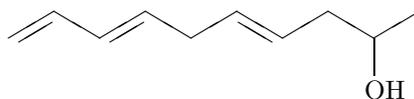
```
\decamethylene[BONDLIST]{ATOMLIST}{SUBSLIST}
\decamethylenei[BONDLIST]{ATOMLIST}{SUBSLIST}
```

The following diagram shows the numbering for designating substitution positions:

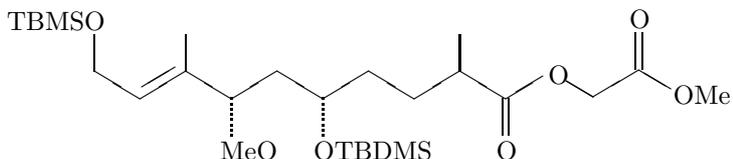


Examples:

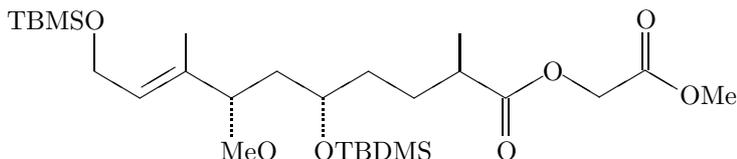
```
\decamethylene[acf]{9==OH}
```



```
\decamethylenei[b]{1==\lmoiety{TBMSO};3==\null;4A==MeO;6A==OTBDMS;%
9B==\null;10D==0;%
10W==\ryl(4==0){5==\dimethylene}{1==(y1);2D==0;2W==OMe}}
```



```
\decamethylenei[b]{1==\lmoiety{TBMSO};3==\null;4A==MeO;6A==OTBDMS;%
9B==\null;10D==0;%
10W==\trimethylenei{1==0}{1==(y1);3D==0;3W==OMe}}
```



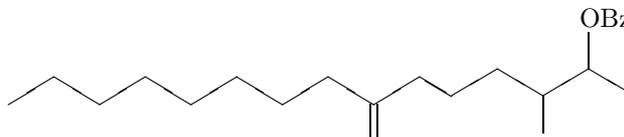
9.10 Longer Polymethylene Chains

A polymethylene chain longer than ten carbons should be written by combining two or more units selected from the above-mentioned di- to deca-methylenes.

To do this task, we regard one unit as a substituent of another unit. In this method, the code for the former unit is written in the SUBSLIST of the code for the latter. For example, the code,

```
\decamethylene{-9D==\null;%
10W==\pentamethylene{-1==(y1);3==\null;4==OBz}}
```

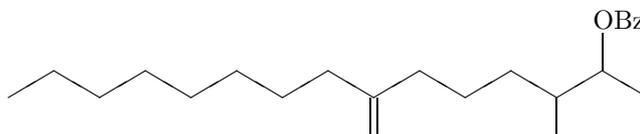
generates the following formula:



Alternatively, we regard one unit as a replacement part of another unit, where the code for the former unit is written in the BONDLIST of the code for the latter (see spiro compounds). The same formula with slightly different appearance can be typeset by the code,

```
\decamethylene{10s==\hexamethylenei{-1==(y1);4==\null;5==OBz}%
-9D==\null}
```

which gives



9.11 Cisoid Tetramethylenes

The macros `\tetramethylenecup` and `\tetramethylenecap` have two arguments `ATOMLIST` and `SUBSLIST` as well as an optional argument `BONDLIST` (`methylen.sty`).

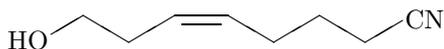
```
\tetramethylenecup[BONDLIST]{ATOMLIST}{SUBSLIST}
\tetramethylenecap[BONDLIST]{ATOMLIST}{SUBSLIST}
```

The following diagram shows the numbering for designating substitution positions:

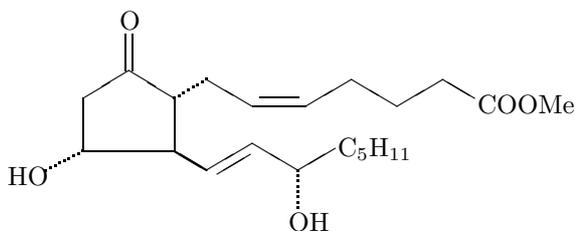


Examples:

```
\tetramethylenecap[b]{1s==\dimethylenei{}{1W==HO;2==(y1)}};%
4s==\trimethylene{}{3W==CN;1==(y1)}}{-}
```



```
\cyclopentanevi{1D==0;4A==HO;%
2A==\tetramethylenecup[b]{%
4s==\trimethylenei{}{1==(y1);3W==COOMe}}{1==(y1)}};%
3B==\trimethylene[a]{}{1==(y1);3A==OH;3W==C$_{5}$H$_{11}$}}
```

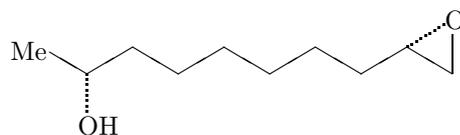


9.12 Ring Fusion to Polymethylenes

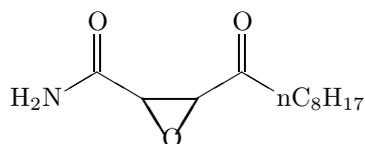
The BONDLIST of each “methylene” command can accept bond fusion.

Examples:

```
\nonamethylene[{h}\threefusehi({cA}){3==0}]{a}}{-}{1W==Me;1A==OH}
```



```
\tetramethylenecup[{\b\threefusev({aB}{cB}){1==0}{-}{B}}]%
  {}{1D==0;1W==H$_{2}$N;4D==0;4W==nC$_{8}$H$_{17}$}
```



```
\pentamethylenei[{\c\threefusehi({bA}{cA}){3==0}{-}{a}}]{1W==Ph;5W==OH}
```

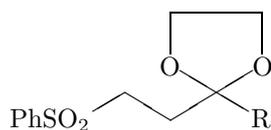


9.13 Ring Replacement to Polymethylenes

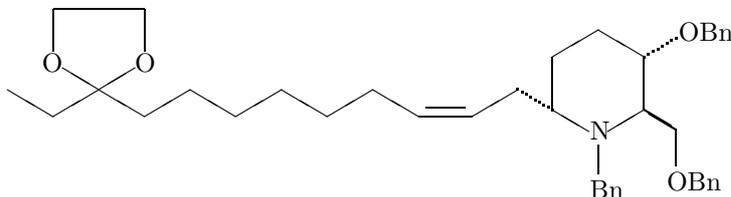
The ATOMLIST of each “methylene” command can accept atom or ring replacement.

Examples:

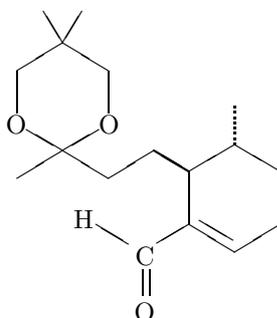
```
\trimethylenei{3s==\fiveheterov{2==0;5==0}{1==(y1)}}{1W==PhSO$_{2}$;3W==R}
```



```
\tetramethylenecup[b]{%
1s==\nonamethylenei{3s==\fiveheterov{2==0;5==0}{1==(y1)}}{9==(y1)};%
4s==\ryl{5A==\sixheterovi{1==N}{1==Bn};%
2B==\ryl{8==OBn};3A==OBn;6==(y1)}}{-}
```



```
\tetramethylene{%
2s==\sixheterovi{2==0;6==0}{4Sa==\null;4Sb==\null;1==(y1)};%
4s==\ryl{5B==\cyclohexanev[d]{6==(y1);1A==\null;%
5==\Utrigonal{0==C;1D==0;2==(y1);3==H}}}{-}
```

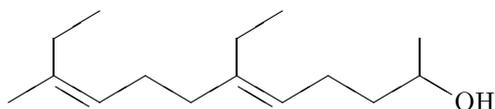


9.14 Branched Chains

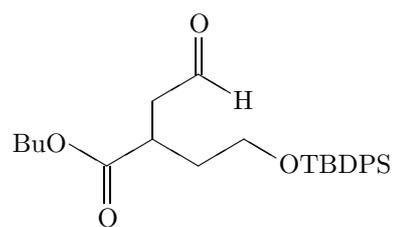
Branched chains can be drawn by using a “methylene” command with the “yl”-function.

Examples:

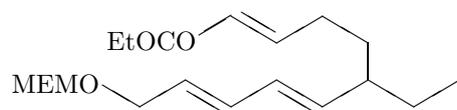
```
\decamethylene[bf]{-%
2==\dimethylene{-}{1==(y1)};6==\dimethylene{-}{1==(y1)};%
{10}W==OH;{10}==\null}
```



```
\tetramethylene{-}{1W==Bu0;1D==0;4W==OTBDPS;%
2==\dimethylene{-}{1==(y1);2D==0;2W==H}}
```



```
\octamethylene[bd]{1W==MEMO;%  
6==\tetramethylenei[a]{4==(y1);1W==EtOCO}}
```



Chapter 10

Enhanced Functions of Commands for General Use

10.1 Expanded Format

Commands for general use, e.g. `\sixheterov`, have originally taken a common format:

```
\genCOM[BONDLIST]{ATOMLIST}{SUBSLIST}
```

where `\genCOM` represents a command name such as `\sixheterov`. In \LaTeX version 2.00, we add a top optional argument `SKBONDLIST` to treat stereochemical information as well as an end optional argument `OMIT` to treat a bond-deleted skeleton. Thus, the expanded format of each command for general use is represented by

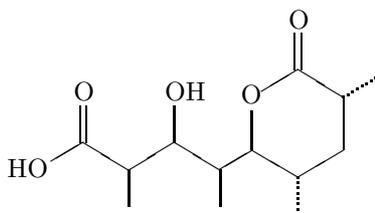
```
\genCOM(SKBONDLIST) [BONDLIST] {ATOMLIST}{SUBSLIST} [OMIT]
```

The argument `SKBONDLIST` contains pairs of two alphabets in braces, where each pair consists of a bond specifier (a lowercase letter) and an uppercase letter (A or B). The letter A represents an α (downward) bond, while B represents a β (upward) bond. For example, an `SKBONDLIST`, `({aA}{cB})`, represents that bond 'a' is an α bond in a dotted form and that bond 'c' is a β bond in a boldfaced form. The argument `OMIT` is a list of bond specifiers, each of which designates a bond to be deleted. As a matter of course, `SKBONDLIST` and `OMIT` take no common bond specifiers.

10.2 Boldfaced and Dotted Bonds

The following example shows that the `\sixheterov` command takes an optional `SKBONDLIST`, `({eB})`, which typesets a boldfaced bond at 'e' in the resulting tetrahydropyran ring.

```
\sixheterov({eB}){6==0}{1D==0;2A==\null;4A==\null;%  
5==\tetramethylenei}{1W==HO;1D==0;2B==\null;3B==OH;4B==\null;4==(y1)}
```

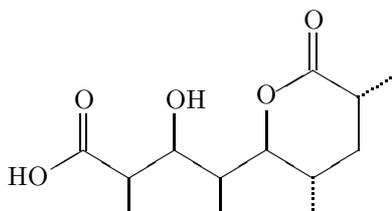


This is an example of the substitution technique in which the side-chain is based on `\tetramethylei` written in the SUBSLIST of the outer `\sixheterov` command.

The same structural formula can alternatively drawn by means of the replacement technique, in which the BONDLIST of the `\sixheterov` command is used for specifying the side-chain. Thus, the code,

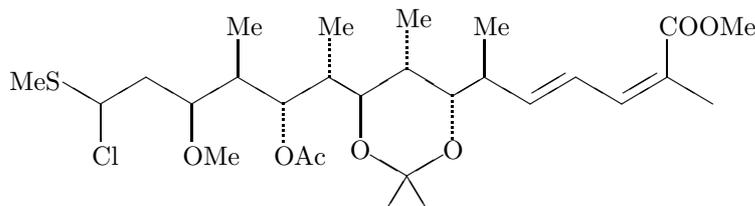
```
\sixheterov({eB}){6==0;%
5s==\pentamethylei}{1W==HO;1D==0;2B==\null;3B==OH;4B==\null;5==(y1)}%
}{1D==0;2A==\null;4A==\null}
```

generates the following formula:

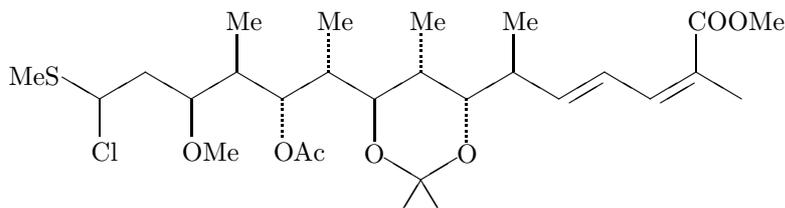


We have further examples in which the `\sixheterov` command takes an optional SKBONDLIST. The following two examples show the comparison between the substitution and the replacement technique, giving formulas of chemically equivalence with slightly different bond lengths.

```
\sixheterov({bA}{eB}){3==0;5==0}{1A==Me;4Sa==\null;4Sb==\null;%
6==\hexamethylene}{1W==MeS;1==Cl;3B==OMe;4B==Me;5A==OAc;6A==Me;6==(y1)};
2==\hexamethylei[bd]}{1==(y1);1B==Me;5==COOMe}}
```

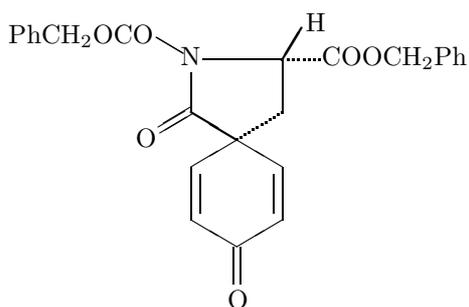


```
\sixheterov({bA}{eB}){3==0;5==0;%
6s==\heptamethylene}{1W==MeS;1==Cl;3B==OMe;4B==Me;5A==OAc;6A==Me;7==(y1)};
2s==\heptamethylene[ce]}{1==(y1);2B==Me;6==COOMe}%
}{1A==Me;4Sa==\null;4Sb==\null}
```



The following structure shows the use of SKBONDLIST in drawing a spiro ring.

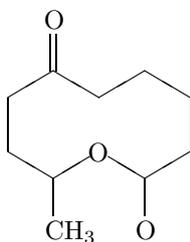
```
\sixheterov[be]{%
1s==\fiveheterov({aA}{eB}){4==N}%
{4==PhCH$_{2}$OCO;3SB==H;3SA==COOCH$_{2}$Ph;5D==0;1==(y1)}%
}{4D==0}
```



10.3 Bond Deletion

The OMIT argument of each command for general use is used to draw a large ring. The following example is a simple case in which one bond is deleted:

```
\decaheterov{9==0}{4==0;8D==0;5==CH$_{3}$}[k]
```



The absence and presence of the OMIT argument give different formulas as follows.

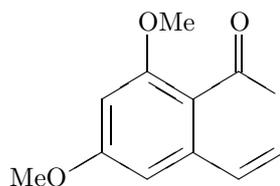
```
\decaheterov[{\k\threefuseh{}{}{b}}]{}{}
\decaheterov[{\k\threefuseh{}{}{b}}]{}{}[k]
```



A complicated case contains a ring fusion as follows. First, the code

```
\decaheterov[cegi]{2==\null}{6==MeO;8==OMe;1D==0}[b]
```

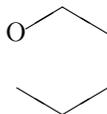
generates the following formula:



where [b] indicates the deletion of bond 'b'. A similar mechanism is also available in a fusing unit, `\sixunitv`. The code,

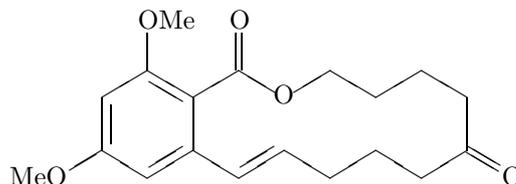
```
\sixfusev{6==0}{E}[b]
```

generates a formula:



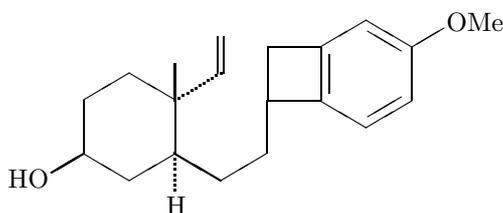
where bond 'e' is deleted by means of the FUSE argument (E) and bond 'b' is deleted by means of the OMIT argument (b). Finally, we have the structural formula of zearalenone:

```
\decaheterov[cegi%
{b\sixfusev[%
{b\sixfusev{}{3D==0}{E}}%
]{6==0}{E}[b]}%
]{2==\null%
}{6==MeO;8==OMe;1D==0}[b]
```

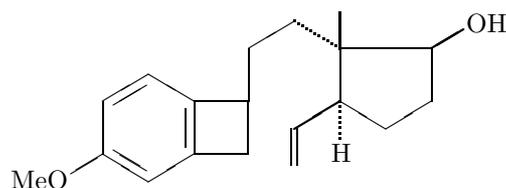


Intermediates for steroid synthesis via intermolecular cycloadditions of *o*-quinodimethane derivatives (Kametani, et al. *J. Org. Chem.*, 1980, **45**, 2204; Grieco, et al. *J. Org. Chem.*, 1980, **45**, 2247) can be drawn by the bond deletion of `\decaheterov` and `\nonaheterov`.

```
\decaheterov({jA}{dB}){%
2s==\fourhetero[{b\sixfusev[ace]{}{2==OMe}{e}}]%
}{1==(y1)}%
}{6B==HO;9A==H;{10}B==\null;1D==\null}[a]
```

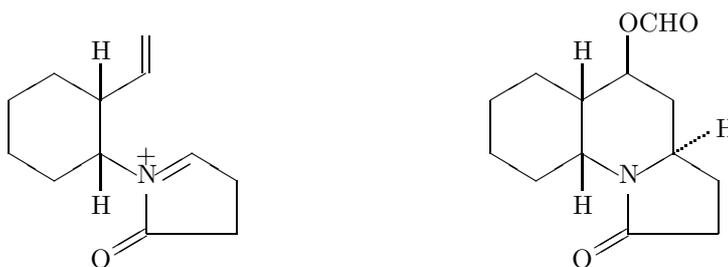


```
\nonaheterov({dA}{hB}){%
6s==\fourhetero[%
{d\sixfusev[bdf]{}{5==MeO}{b}}]%
}{3==(y1)}%
}{3B==OH;8B==\null;7D==\null;9A==H}[g]
```



A remarkable merit of using a skeleton with deleted bonds appears in drawing a starting compound with an acyclic part along with the resulting product via cyclization, since their codes are akin to each other.

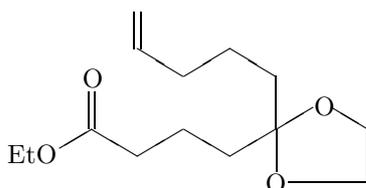
```
\decaheterov[4+}%
{c\fivefusevi[e]{5==\null}{4D==0}{E}}%
]{4==N}{1D==\null;9B==H;{10}B==H}[ab]
\hskip2cm
\decaheterov[%
{c\fivefusevi{5==\null}{4D==0}{E}}%
]{4==N}{1B==OCHO;9B==H;{10}B==H;3FA==H}
```



The latter compound was obtained by the cyclization of the former (D. J. Hart, et al., *J. Am. Chem. Soc.*, 1980, **102**, 397).

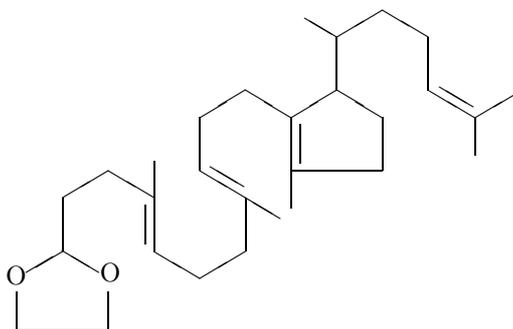
Some polymethylene chains are drawn in a folded form. The bond-deletion technique can be applied to drawing such folded formulas.

```
\sixheterov{%
3s==\fiveheterovi{1==0;4==0}{5==(y1)};%
6s==\dimethylenei{}{1D==\null;2==(y1)};%
5s==\trimethylenei{}{1W==EtO;1D==0;3==(y1)}%
}{}[e]
```



The following formula, which is an intermediate for synthesizing steroid skeletons, can also be drawn by this technique.

```
\decaheterov[k%
{f\fivefusevi{2==\null;5==0}{A}}%
{a\sixfusev[d%
{b\fivefusevi[d%
{a\sixfusev{%
3s==\trimethylenei[a]{1==(y1);2==\null}%
}{6==\null}{D}[c]}%
]{}{D}}%
]{}{3G==\null}{D}[c]}%
]{5==0}{10}{Sb==\null;2G==\null}[ej]
```



Chapter 11

Enhanced Functions of Commands for Ring Fusion

11.1 Expanded Format

Commands for ring fusion, e.g. `\sixfusev`, have originally taken a common format (version 1.02 not released):

```
\fuseCOM[BONDLIST]{ATOMLIST}{SUBSLIST}{FUSE}
```

where `\fuseCOM` represents a command name such as `\sixfusev`. In \LaTeX version 2.00, we add a top optional argument `SKBONDLIST` to treat stereochemical information as well as an end optional argument `OMIT` to treat a bond-deleted skeleton. Thus, the expanded format of each command for general use is represented by

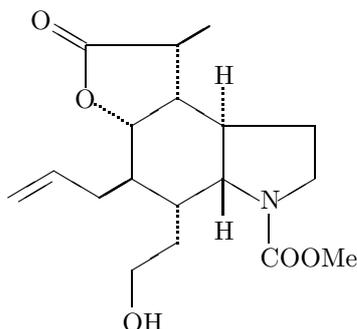
```
\fuseCOM(SKBONDLIST) [BONDLIST] {ATOMLIST} {SUBSLIST} {FUSE} [OMIT]
```

The argument `SKBONDLIST` contains pairs of two alphabets in braces, where (1) each pair consists of a bond specifier (a lowercase letter) and an uppercase letter (A or B); and (2) the letter A represents an α (downward) bond, while B represents a β (upward) bond. The argument `OMIT` is a list of bond specifiers, each of which designates a bond to be deleted. As a matter of course, `SKBONDLIST` takes no common bond specifiers with `FUSE` and `OMIT`.

11.2 Boldfaced and Dotted Bonds

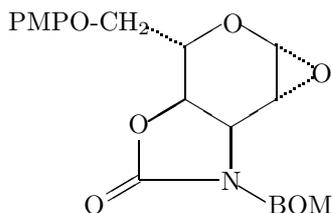
The first example shows that the command `\fivefusev` with a `SKBONDLIST` generates a formula with dotted bonds at fused positions.

```
\nonaheterov[%  
{e\fivefusev({bA}{eA){5==0}{3B==\null;4D==0}{A}}%  
]{1==N}{1==COOMe;8A==H;9B==H;%  
6B==\trimethylene[a]{3==(y1)};%  
7A==\dimethylene{2==(y1);1==OH}}
```



The next example shows the use of the SKBONDLISTS of `\threefuseh` and `\fivefusevi` to indicate stereochemical information.

```
\sixheterov[%
{b\threefuseh({aA}{cA}){1==0}{B}}%
{d\fivefusevi({bB}{eB}){3==N;5==0}{3==BOM;4D==0}{A}}%
]{1==0}{6A==PMPO-CH$_{2}$}
```



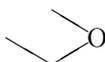
11.3 Bond Deletion

11.3.1 Larger Rings from Two or More Three-Membered Rings

To draw a fused four-membered ring, we can use two `\threefuseh(i)` commands in a nested fashion. For example, the code

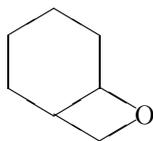
```
\threefusehi [{b\threefuseh{1==0}{b}}]{c}[b]
```

generates a four-membered unit:



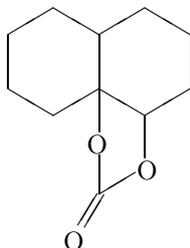
The resulting unit is used to draw a four-membered fused ring, as shown below:

```
\sixheterov[%
{c\threefusehi [{b\threefuseh{1==0}{b}}]
}{c}[b]}%
]{}
```



In a similar way, a five-membered fusing unit can be drawn by combining three `\threefuseh(i)` commands, as shown in the following example:

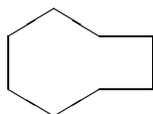
```
\decaheterov[%
{d\threefuseh[%
{a\threefusehi[%
{a\threefuseh{1==\null;3==\null}{2D==0}{c}}%
]{2==0;1==\null}{c}[a]}%
]{2==0}{C}[a]}%
]{}
```



11.3.2 Further Rings

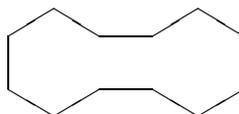
A six-membered ring fused by a four-membered unit gives an eight-membered ring as follows:

```
\sixheterov[{b\fourfuse{}{}{d}}]{b}
```



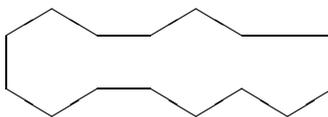
The bond 'b' of the four-membered unit in the resulting ring is deleted and used as an acceptor ring of a six-membered fusing unit. Then, we have a twelve-membered ring:

```
\sixheterov[{b\fourfuse[{b\sixfusev{}{}{e}]}{}{d}[b]}]{b}
```



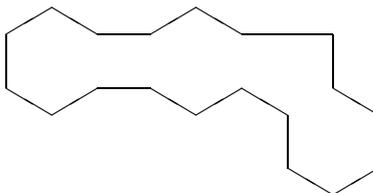
After applying the bond-deletion technique to the twelve-membered ring, this is used as an acceptor of a five-membered fusing unit. Then we have a fifteen-membered ring:

```
\sixheterov[{\b\fourfuse[{\b\sixfusev[{\b\fivefusev{}-{}-d}]%
]{}-{}-e}[b]]{}-{}-d}[b]]{}-{}[b]
```



A further fusion of a six-membered unit gives a nineteen-membered ring:

```
\sixheterov[{\b\fourfuse[{\b\sixfusev[{\b\fivefusev[{\a\sixfusev{}-{}-f}]%
]{}-{}-d}[a]]%
]{}-{}-e}[b]]{}-{}-d}[b]]{}-{}[b]
```



Chapter 12

Reaction Schemes

12.1 Compound Numbers

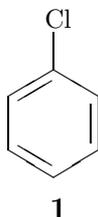
The XyMcompd environment has two functions:

1. for giving a compound number and specifying a reference key and
2. for specifying the size of a domain to draw a structural formula.

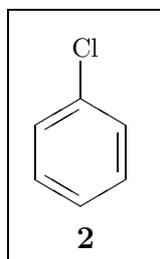
For example, the code:

```
¥begin{XyMcompd}(400,750)(220,200){cPhCL}{}  
¥bzdrv{1=Cl}  
¥end{XyMcompd}
```

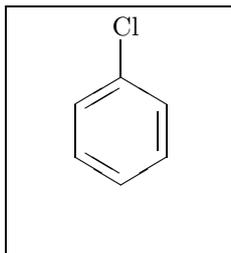
produces the following formula,



The compound number (**1**) can be referred to by designating `¥cref{cPhCL}`. The code (400,750) specifies the size of the drawing domain and the code (220,200) represents x- and y-shift values. When the XyMcompd environment is surrounded by a frame generated by the `¥fbox` command, we obtain the following diagram:



The original `¥bzdrv` command has a domain to accommodate substituents as follows:



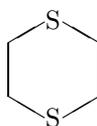
If such adjustment and cross-reference are unnecessary, we write the code:

```

\begin{XyMcompd}(,)(,){}{}
\sixheterov{1==S;4==S}{}
\end{XyMcompd}

```

Thereby, we obtain the formula of the original specification:



which is the same formula generated by the code:

```

\sixheterov{1==S;4==S}{}

```

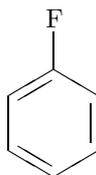
The last argument of the XyMcompd environment is to specify the subnumber of a compound number. For example, the code:

```

\begin{XyMcompd}(400,750)(220,200){PhF}{a}
\bzdrv{1==F}
\end{XyMcompd}

```

produces the following formula,



3a

Derivatives of a compound numbered in the XyMderiv environment are designated by subnumbering using a $\backslash\text{derivlist}$ command in the XyMderiv environment. For example, the code:

```

\begin{XyMderiv}
\begin{XyMcompd}(400,750)(220,200){PhX}{}
\bzdrv{1==X}
\end{XyMcompd}
\derivlist{X = Cl;X = NO$_{2}$;X = F}
\end{XyMderiv}

```

produces the following formula:

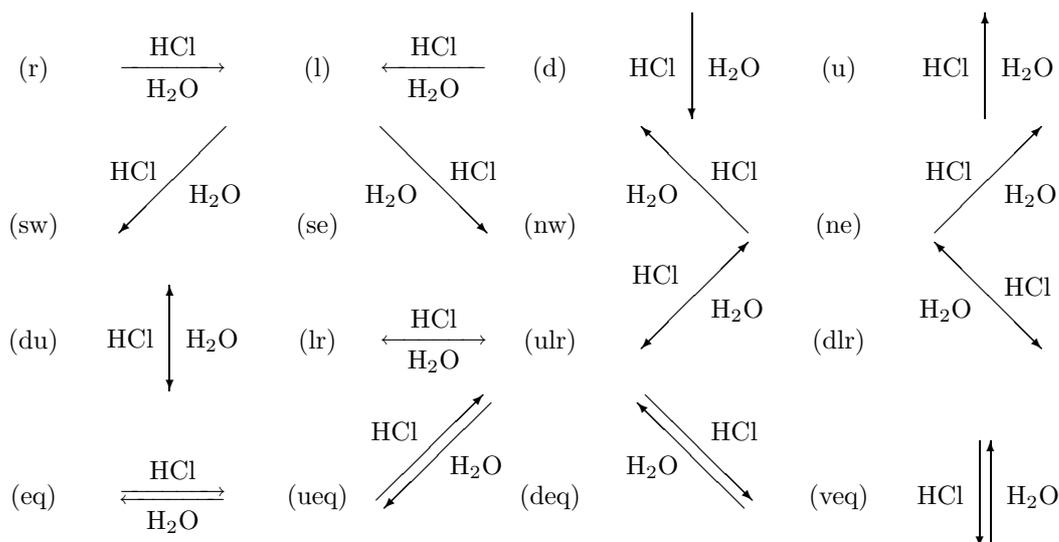
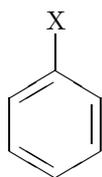


Figure 12.1: Reaction arrows of various types



4

- a:** X = Cl
b: X = NO₂
c: X = F

12.2 Reaction Arrows

In addition of the reaction arrows described in Ref. [13], we have added further reaction arrows shown in Fig. 12.1. They are defined in the package `chemist.sty`. Each arrow command is the following format:

$$\text{\textbackslash}\text{ARROWNAME}[\text{xshift}]{\text{yshift}}{\text{length}}{\text{itemover}}{\text{itemunder}}$$

where `\ARROWNAME` represents a command name; `xshift` is an optional argument to show a horizontal adjustment value; `yshift` is an argument to show a vertical adjustment value; `length` is an argument to designate the length of the arrow; and the arguments `itemover` and `itemunder` represent items placed over and under the arrow. The name (`\ARROWNAME`) of each reaction arrow take the format of `\react...arrow` in which `...` is selected from the following list: `r` = right arrow, `l` = left arrow, `lr` = leftright arrow, `d` = down arrow, `u` = up arrow, `du` = down up arrow, `eq` = equilibrium arrow, `veq` = vertical equilibrium arrow, `deq` = down equilibrium arrow, `leq` = up equilibrium arrow, `dlr` = down leftright arrow, `ulr` = up leftright arrow, `sw` = southwest arrow, `se` = southeast arrow, `nw` = northwest arrow, and `ne` = northeast arrow.

12.3 Display Formulas and Tabular Schemes

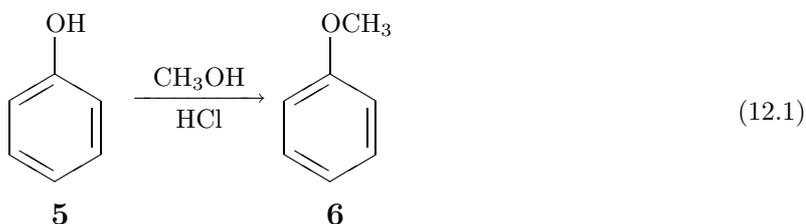
Display formulas containing structural formulas and reaction arrows are drawn by using the equation environment of $\tilde{L}\tilde{T}\tilde{E}\tilde{X}$ or the chemeqn environment of the chemist package. For example, the code,

```

\begin{equation}\label{EQ1}
\begin{XyMcompd}(400,750)(220,200){BPHOH}{-}
\begin{bmatrix}1==OH\end{bmatrix}
\end{XyMcompd}
\reactarrow[10\unitlength]{60\unitlength}{500\unitlength}
{CH\mbox{$_3$}OH}{HCl}
\begin{XyMcompd}(400,750)(220,200){PHOME}{-}
\begin{bmatrix}1==OCH\mbox{$_3$}\end{bmatrix}
\end{XyMcompd}
\end{equation}

```

produces the following display formula:



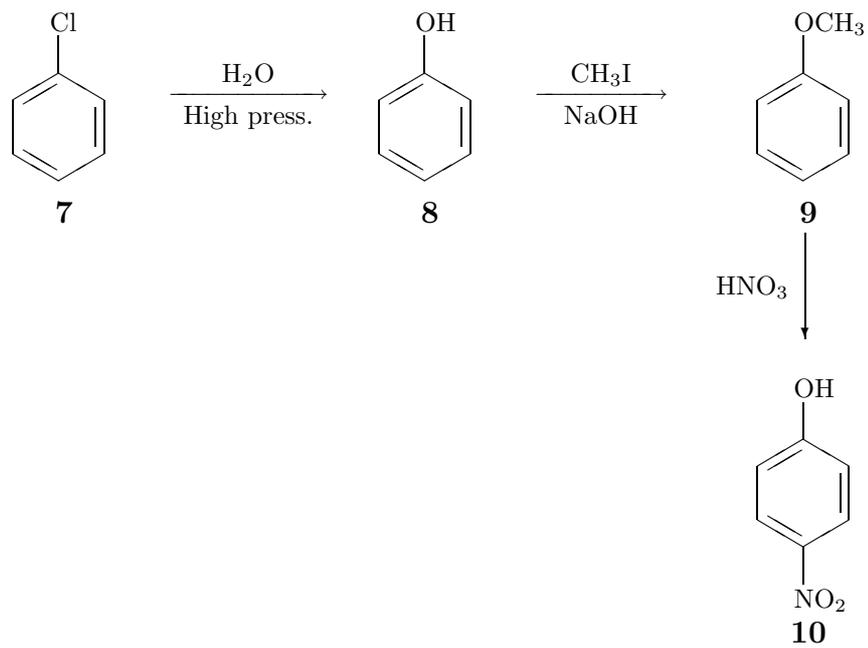
Tabular schemes containing structural formulas and reaction arrows are drawn by using the XyMtab environment of the chemist package. For example, the code,

```

\begin{XyMtab}{cccccc}
\begin{XyMcompd}(400,750)(220,200){AAPHCL}{-}
\begin{bmatrix}1==Cl;\end{bmatrix}
\end{XyMcompd}
&
\reactarrow[10\unitlength]{60\unitlength}{600\unitlength}
{\strut{H}\mbox{$_2$}O}{\strut{High press.}}&
%
\begin{XyMcompd}(400,750)(220,200){AAPHOH}{-}
\begin{bmatrix}1==OH;\end{bmatrix}
\end{XyMcompd}
&
\reactarrow[10\unitlength]{60\unitlength}{500\unitlength}
{\strut{CH}\mbox{$_3$}I}{\strut{NaOH}}&
%
\begin{XyMcompd}(400,750)(220,200){AAPHOME}{-}
\begin{bmatrix}1==OCH\mbox{$_3$};\end{bmatrix}
\end{XyMcompd}
%
&\&\&
\reactswarrow[0\unitlength]{300\unitlength}{400\unitlength}
{\strut{HNO}\mbox{$_3$}}{\strut{}}&
%
\begin{XyMcompd}(400,850)(220,0){APHNO2}{-}
\begin{bmatrix}1==OH;{4}==NO\mbox{$_2$};\end{bmatrix}
\end{XyMcompd}
&\&
\end{XyMtab}

```

generates a tabular scheme as follows:



Bibliography

- [1] NIFTY-Serve achieves, FPRINT library No. 7, Item Nos. 201, 202, 204.
- [2] CTAN, tex-archive/macros/latex209/contrib/xymtex/.
- [3] Fujita S., “Typesetting structural formulas with the text formatter $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ”, *Comput. Chem.*, **18**, 109 (1994).
- [4] Fujita S., “ $\text{X}^{\text{M}}\text{T}_{\text{E}}\text{X}$ for Drawing Chemical Structural Formulas”, *TUGboat*, **16** (1), 80 (1995).
- [5] Lamport L., *L^AT_EX. A document Preparation System*, 2nd ed. for $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X } 2_{\epsilon}$, Addison-Wesley, Reading (1994). See also Lamport L., *L^AT_EX. A document Preparation System*, Addison-Wesley, Reading (1986).
- [6] Goossens, M., Mittelbach, F., & Samarin, A., *The L^AT_EX Companion*, Addison-Wesley, Reading (1994).
- [7] NIFTY-Serve achieves, FPRINT library No. 7, Item Nos. 385, 386.
- [8] <http://imt.chem.kit.ac.jp/fujita/fujitas/fujita.html>
- [9] Fujita, S., *X^MT_EX—Typesetting Chemical Structural Formulas*, Addison-Wesley, Tokyo (1997). The book title is abbreviated as “ $\text{X}^{\text{M}}\text{T}_{\text{E}}\text{X}$ book” in the present manual.
- [10] For the $\text{T}_{\text{E}}\text{X}$ system, see Knuth D. E., *The T_EXbook*, Addison-Wesley, Reading (1984).
- [11] For the Chem $\text{T}_{\text{E}}\text{X}$ macros, see Haas R. T. & O’Kane K. C., *Comput. Chem.*, **11**, 251 (1987).
- [12] For drawing chemical formulas by $\text{T}_{\text{E}}\text{X}$, see Ramek, M., in Clark, M. (ed), $\text{T}_{\text{E}}\text{X}$: Applications, Uses, Methods, Ellis Horwood, London (1990), p. 277.
- [13] For chemical application of the $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ system, see Fujita S., *Kagakusha-Seikagakusha no tame no L^AT_EX (L^AT_EX for Chemists and Biochemists)*, Tokyo Kagaku Dozin, Tokyo (1993).
- [14] For epic macros, see Podar S., “Enhancements to the picture environment of $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ”, Manual for Version 1.2 dated July 14, 1986.
- [15] For graphic applications of $\text{T}_{\text{E}}\text{X}$, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ and relevant systems, see Goossens, M., Rahtz, S., & Mittelbach, F., *L^AT_EX Graphics Companion*, Addison Wesley Longman, Reading (1997).