AT&T Bell Laboratories
Murray Hill, New Jersey 07974

Computing Science Technical Report No. 122

CHEM — A Program for Typesetting Chemical Diagrams: User Manual

Jon L. Bentley
Lynn W. Jelinski
Brian W. Kernighan

October 6, 1992

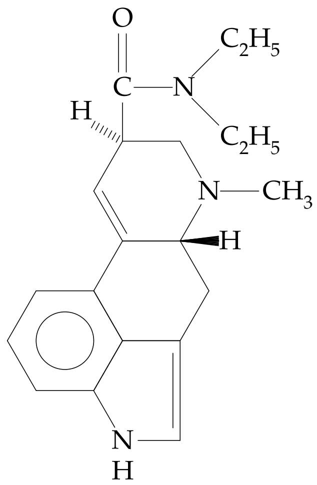
CHEM — A Program for Typesetting Chemical Diagrams: User Manual

Jon L. Bentley
Lynn W. Jelinski
Brian W. Kernighan

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

chem is a *troff* preprocessor like *eqn* or *tbl*, for producing chemical structure diagrams like this one:



chem is intended to make it possible for chemists to include structure diagrams in their manuscripts with the same ease that they can include tables or mathematics. *chem* has provisions for rings, heterocycles, adjustable bond lengths and angles, and stereochemistry. It can be used for papers and for viewgraphs.

October 6, 1992

CHEM — A Program for Typesetting Chemical Diagrams: User Manual

Jon L. Bentley
Lynn W. Jelinski
Brian W. Kernighan

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

1. What is *chem*?

chem is a *troff* preprocessor like *eqn* or *tbl* or *pic*, for producing chemical structure diagrams like the ones in this manual. It is intended as a natural language for chemists; it attempts to capture the way that a chemist might describe a molecule to a colleague over the telephone. *chem* converts the textual description of a structure diagram into a sequence of commands in the *pic* graphics language, and these in turn are converted into commands for the *troff* formatter.

chem is still experimental: it is slow, error checking is minimal, and it certainly doesn't do everything. Right now it is best suited for producing viewgraphs and publication-quality structures of organic and bio-organic molecules.

2. How to Input Structure Descriptions

The commands that tell *chem* how to draw a chemical structure can be embedded in the text of a manuscript, or they can stand alone. In either case, the commands must be surrounded by `.cstart` and `.cend`, with the ''`.`'' always in the first column. (This is analogous to the use of `.EQ` and `.EN` for equations or `.TS` and `.TE` for tables.) For example:

```
.cstart
    CH3
    bond
    CH2
    bond
.cend
```

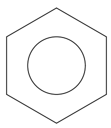
will draw the ethyl group

$$CH_3 — CH_2 —$$

and

```
.cstart
    benzene
.cend
```

draws a benzene ring:

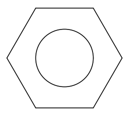


Whatever is between `.cstart` and `.cend` is converted into *pic* commands by *chem*; everything else is passed along untouched.

Sometimes it is helpful to include comments among the commands that specify the structure, especially when describing complex molecules. A comment begins with a #; any characters from there to the end of the line are thrown away. For example:

```
.cstart
    benzene pointing right     # a rotated benzene ring
.cend
```

produces the following diagram:



3. How to Get Output

Once you've created a file that contains a *chem* description, it must be converted into *troff* output. To do this, you must run your file through *chem*, *pic*, *troff* and an output filter.

Let's take the simplest case first. If you are only doing a single diagram, there's no need to use a macro package, so you should type the command line:

```
chem filename | pic | troff >filename.out
```

`filename.out` then contains your structure. There are a number of ways that you can see the results. You can get Imagen output by typing:

```
dcan filename.out
```

For APS-5 typesetter output:

```
apsend filename.out
```

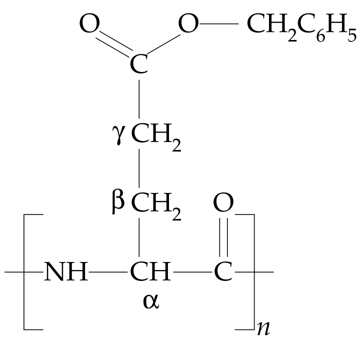
With a Blit, running *mux* or *layers*:

```
proof filename.out
```

With an AT&T PC 6300 using PC-PLOT (4014 emulation):

```
tc filename.out
```

Here is a fairly complex structure, poly(benzyl glutamic acid), printed first on a typesetter, then on the various lower-quality previewers. You can see that *tc* is only marginally satisfactory.



Typesetter (apsend)

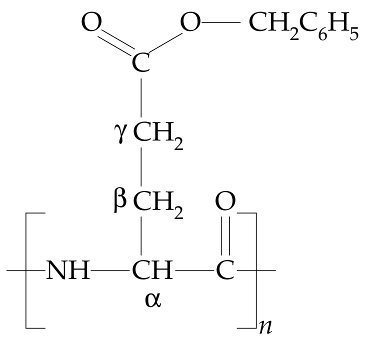


Imagen laser printer (dcan)

Teletype 5620 (proof)

Epson printer (tc)

Now, back to more complicated command lines. If your formula contains *eqn* characters (such as α, β, and γ), you should include *eqn* in the command line:

```
chem filename | pic | eqn '-d$$' | troff >filename.out
```

(If you use the *eqn* argument `-d$$`, you don't need `delim $$` in your input file).

If your structures are embedded in the usual type of chemistry manuscript that contains equations and tables and uses a macro package, your command line will look something like:

```
chem filename | pic | tbl | eqn | troff -mm ...
```

4. Building Chemical Formulas

The *chem* input language is rather small. It provides for rings, heterocycles, bonds, atoms, and stereochemistry. Objects are normally connected together in the order that you write them down. *chem* tries to connnect them at the natural places, but provides ways to specify precisely where things are placed and how they are connected for those situations where it isn't smart enough to guess right. If all else fails, because *chem* is a *pic* preprocessor, it is possible to include *pic* statements in the middle of a diagram to draw things that are not provided for by *chem* itself.

Each component of the diagram is written on a single line; blank lines and comments can be used freely to make the description easy to read later on.

Bonds, Directions, and Atoms

Bonds are specified in the following general form, where the brackets specify optional qualifiers:

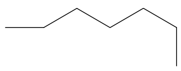
bondtype [direction] [`length` *n*] [`from` *name*] [`to` *name*] *[attribute]*

The options must appear in this order. Let's look at them one at a time. First, the *bondtype*. *chem* knows about single, double, and triple bonds. These are invoked by the commands `bond`, `double bond`, and `triple bond`, respectively. *chem* also understands `front bond` and `back bond` to specify stereochemistry.

The *direction* of bonds in *chem* is handled in several ways. *chem* recognizes `up`, `down`, `left` and `right`, as well as the corresponding compass points `n`, `s`, `w` and `e`. When the direction isn't one of these, it's easiest to use actual angles. Zero corresponds to `up` or `n`, 90 to `right`, –90 to `left`, and so on. For example, this input:

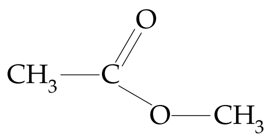
```
.cstart
    bond right
    bond 60
    bond 120
    bond 60
    bond 120
    bond down
.cend
```

produces the following stick structure:



Of course bonds usually connect atoms or groups, as in this description of methyl acetate:

```
.cstart
CH3      # the 3 is automatically turned into a subscript
bond     # the implicit direction is right
         # implicit connection is to right side of CH3
C
double bond 30     # by default, from the substituent C
O
bond 120 from C    # must be "from C"; otherwise would leave from O
O
bond right
CH3
.cend
```



These commands could be written more compactly by putting the substituents on the same lines as bonds, separated by semicolons (spaces matter!), as in:

```
.cstart
    CH3
    bond ; C
    double bond 30 ; O
    bond 120 from C ; O
    bond right ; CH3
.cend
```

A group or moiety like CH_3 is written as a string of characters. A group must begin with a capital letter, as in HCl or N(C2H5)2. Numbers are automatically converted to subscripts and dots are centered (unless the combination appears to be a fractional value, as in N2.5H). For example:

```
.cstart
    HCl.H2O
.cend
```

produces

$$HCl{\cdot}H_2O$$

but attempting to do the same with $CaSO_4{\cdot}2H_2O$, i.e.,

```
.cstart
    CaSO4.2H2O
.cend
```

produces

$$CaSO_{4.2}H_2O$$

You will have to use *eqn* or the "`right of`" construct in unusual cases like this.

Normally a group is placed right after the last thing mentioned, but it may be manually positioned by *pic*-like commands, e.g.,

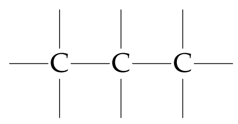
```
    CH3 at C + (0.5, 0.5)
```

We'll see more examples of this later.

In the methyl acetate example, the carbon atom C was used both to draw something and as a *name* for a place. An atom or a group always defines a name for a place. (When the group or moiety contains special characters, like the parentheses in N(C2H5)2, the name is what is left after the special characters are stripped out.) Later we'll see that we sometimes need to invent names for rings and bonds so we can refer back to them.

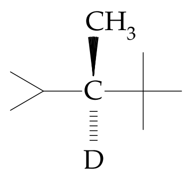
Each new occurrence of a name overrides the definition of the previous one:

```
.cstart
    bond ; C            # 1st definition of C
    bond up from C
    bond down from C
    bond right from C ; C         # 2nd definition of C
    bond up from C
    bond down from C
    bond right from C ; C         # 3rd definition of C
    bond up from C
    bond down from C
    bond right from C
.cend
```



BP is a special name that is used to specify a ''branch point,'' a place where no moiety is printed. Here's an example of its use:

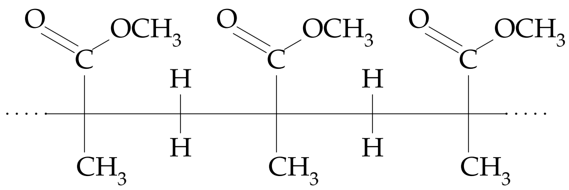
```
.cstart
# this is the isopropyl group
    bond 120 ; BP       # BP is right end of this bond
    bond -120 from BP
    bond right from BP ; C
    front bond up ; CH3
    back bond down from C ; D
    bond right from C ; BP
# redefine BP to mean the center carbon of this t-butyl group
    bond up from BP
    bond right from BP
    bond down from BP
.cend
```



chem takes considerable pains to connect bonds to the proper parts of moieties, for example by connecting to carbon atoms where possible. It is possible to override the automatic connections to carbon, as we'll see later in this section.

The default bond length is 0.2 inches, but it can be adjusted by specifying a length in inches. For example, in this structure of isotactic poly(methyl methacrylate), variable bond lengths are used to produce an uncrowded structure. Note also the bond attribute ''dotted''; ''dashed'' bonds are also available.

```
.cstart
    bond dotted
    bond right ; BP
    bond up from BP ; C
    double bond -60 from C ; O
    bond 60 length .1 from C ; OCH3
    bond down from BP ; CH3
# begin second segment of polymer
    bond right length .5 from BP ; BP
    bond up length .1 from BP ; H
    bond down length .1 from BP ; H
# begin third segment of polymer
    bond right length .5 from BP ; BP
    bond up from BP ; C
    double bond -60 from C ; O
    bond 60 length .1 from C ; OCH3
    bond down from BP ; CH3
# begin fourth segment of polymer
    bond right length .5 from BP ; BP
    bond up length .1 from BP ; H
    bond down length .1 from BP ; H
# begin fifth segment of polymer
    bond right length .5 from BP ; BP
    bond up from BP ; C
    double bond -60 from C ; O
    bond 60 length .1 from C ; OCH3
    bond down from BP ; CH3
    bond right from BP
    bond dotted
.cend
```



Rings

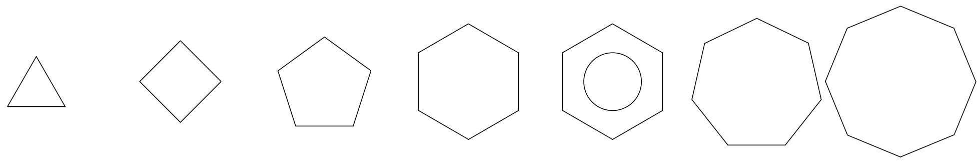
Rings are specified in the following general form:

[ring name:] `ring`*[n] [direction] [heteroatoms] [doublebond pattern]*

where the optional parts are enclosed in square brackets. We'll look at each of these in more detail.

Cyclopropane through cyclooctane can be obtained by simply saying `ring3`, `ring4`, and so on:

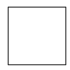
```
.cstart
R3: ring3
R4: ring4 at R3 + (.75,0)
R5: ring5 at R4 + (.75,0)
R6: ring6 at R5 + (.75,0)
B:  benzene at R6 + (.75,0)
R7: ring7 at B + (.75,0)
R8: ring8 at R7 + (.75,0)
.cend
```



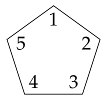
''`ring`'' is a synonym for `ring6`. The attribute ''`at` *name* `+ (.75,0)`'' is a *pic* command that puts each object 3/4 inch to the right of (the center of) the previous one.

Note that all of these rings are ''pointing up''. That is, the top-most vertex, or vertex 1, points toward the top of the page. This is especially obvious for cyclobutane (''`ring4`''), which is oriented in a manner that at first appears strange to a chemist. The normal orientation has the ''point'' or vertex 1 rotated 45 degrees, so it is written as

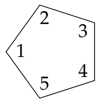
```
ring4 pointing 45
```



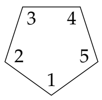
The vertices of rings are always numbered the same way: the ''point'' is the first vertex, which is called `.V1`, and `.V2`, `.V3`, etc., continue clockwise around the ring. For example, the following structures show that the ''point'' in cyclopentane, as in any other ring, is always numbered 1, but its position follows the direction in which the ring points:



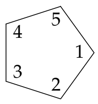
ring5 (or ring5 pointing up)



ring5 pointing left

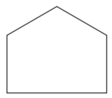


ring5 pointing down

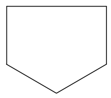


ring5 pointing right

There are two types of 5-membered rings: the pentagonal `ring5` above, and the "`flatring`." The `flatring` is useful for fusing onto the sides of other rings.



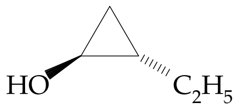
flatring pointing up



flatring pointing down

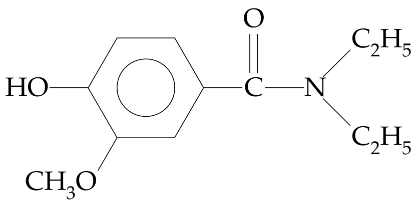
It is handy (and sometimes essential!) to name the rings so you can properly add the substituents. For example:

```
.cstart
    R:   ring3
         back bond 120 from R.V2 ; C2H5
         front bond -120 from R.V3 ; HO
.cend
```



The ring itself is named `R`; its second vertex is named `R.V2`. As another example,

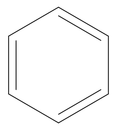
```
.cstart
    R:   benzene pointing right
         bond left from R.V4 ; HO
         bond -150 from R.V3 ; CH3O
         bond right from R.V1 ; C
         double bond up from C ; O
         bond right from C ; N
         bond 45 ; C2H5
         bond 135 from N ; C2H5
.cend
```



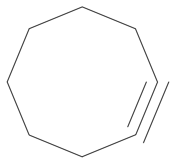
If you do not specify an explicit connection point for a substituent bond, *chem* will try to deduce it from the direction of the bond. For example, in the two diagrams immediately above, it is possible to omit all of the `.V`*n*'s.

Double bonds within rings are specified by naming vertices between which they appear:

```
.cstart
    ring double 1,2 3,4 5,6
.cend
```

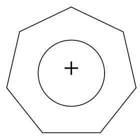


```
.cstart
    ring8 triple 3,4
.cend
```



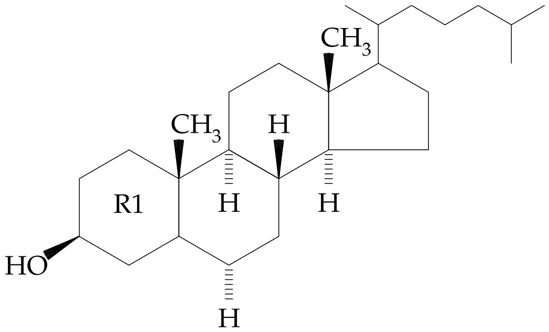
A circle can be placed inside any ring by using the modifier `aromatic`:

```
.cstart
R:  aromatic ring7
    "+" at R
.cend
```



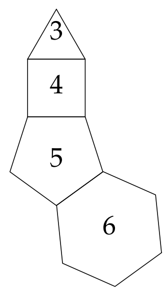
Fused ring structures are formed by specifying a common vertex on each ring. For example, consider cholestanol:

```
.cstart
R1: ring6
    "R1" at R1     # this puts a label at R1
    front bond -120 from R1.V5 ; HO
         # the following line says "fuse the next six-
         # membered ring with its 6th vertex joining
         # the second vertex of R1"
R2: ring6 with .V6 at R1.V2
    front bond up from R2.V6 ; CH3
    back bond down from R2.V4 ; H
    back bond down from R2.V1 ; H
    front bond up from R2.V2 ; H
R3: ring6 with .V4 at R2.V2
R4: flatring with .V5 at R3.V2
    front bond up from R4.V5 ; CH3
    back bond down from R4.V4 ; H # this is the alkyl chain
    bond up from R4.V1 ; BP
    bond -60 from BP
    bond 60 from BP
    bond 120
    bond 60
    bond 120 ; BP
    bond down from BP
    bond 60 from BP
.cend
```



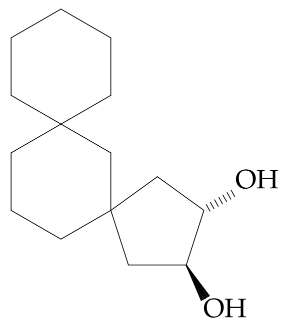
The next example shows how the different sized rings fuse together. Note that the fusion of the five- and six-membered rings requires an unusual angle:

```
.cstart
R3: ring3
R4: ring4 pointing 45 with .V1 at R3.V2
R5: ring5 pointing down with .V4 at R4.V2
R6: ring6 pointing 54 with .V6 at R5.V5
    # the following lines specify the labels inside the rings
    "3" at R3
    "4" at R4
    "5" at R5
    "6" at R6
.cend
```



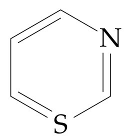
Spiro ring junctions are formed in an analogous way:

```
.cstart
    R1:  ring6
    R2:  ring6 with .V1 at R1.V4
    R3:  ring5 with .V5 at R2.V3
         back bond 60 from R3.V2 ; OH
         front bond 150 from R3.V3 ; OH
.cend
```



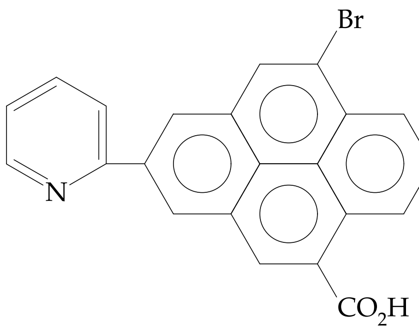
Heteroatoms in rings are written as "`put` *X* `at` *V*," where *X* is the heteroatom and *V* is the vertex number. For example:

```
.cstart
    ring put N at 2 put S at 4 double 2,3 4,5 6,1
.cend
```



As a more complex example, this polycyclic aromatic compound is produced with the following input:

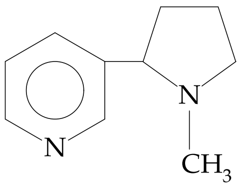
```
.cstart
R1: benzene pointing right
    bond 30 from R1.V6 ; Br
R2: benzene pointing right with .V5 at R1.V1
R3: benzene pointing right with .V1 at R2.V3
    bond 150 from R3.V2 ; CO2H
R4: benzene pointing right with .V1 at R1.V3
# next line names bond B1 so we can refer to its end
B1: bond left from R4.V4
    ring6 put N at 4 double 2,3 4,5 6,1 with .V3 at B1.end
R5: benzene with .V5 at B2.end
    ring6 put N at 4 double 1,2 3,4 with .V5 at R5.V3
.cend
```



Quite arbitrarily, we started with the bromo-substituted ring but, as is usual in *chem*, there are many other equally good ways to draw this structure.

Substituents are placed on heteroatoms just as they are placed on non-heterocycles. The nicotine molecule provides an example of bond positioning:

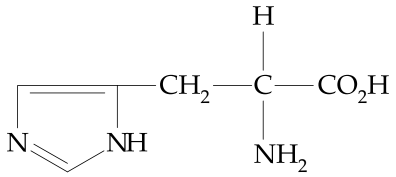
```
.cstart
    benzene put N at 4
    bond right
    ring5 pointing down put N at 1
    bond down from .N ; CH3        # or .V1
.cend
```



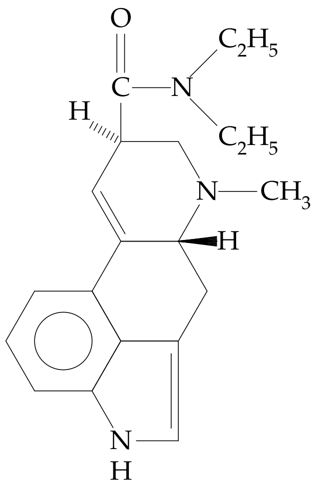
The second bond refers to `.N`, which is an alternative name for `.V1`. `.N` is taken to refer to the immediately preceding object, the unnamed `ring5`.

When drawing heterocyclic rings, it is often useful to position a group or atom in the vicinity of the heteroatom. This is done with the commands `above`, `below`, `right of`, and `left of`, as in the NH of the imidazole ring of histidine:

```
.cstart
R1: flatring pointing down put N at 2 put N at 5 double 1,2 3,4
    H right of R1.V5
    bond right from R1.V4 ; CH2
    bond right ; C
    bond up from C ; H
    bond down from C ; NH2
    bond right from C ; CO2H
.cend
```



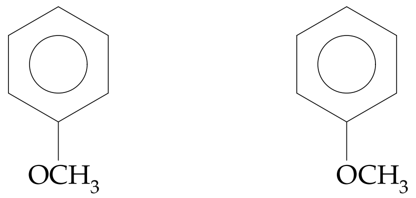
The lysergic acid diethylamide structure shown in the abstract provides another example of positioning substituents on heteroatoms:



```
.cstart
B:  benzene pointing right
F:  flatring pointing left put N at 5 double 3,4 with .V1 at B.V2
    H below F.N
R:  ring pointing right with .V4 at B.V6
    front bond right from R.V6 ; H
W:  ring pointing right with .V2 at R.V6 put N at 1 double 3,4
    bond right from W.N ; CH3
    back bond -60 from W.V5 ; H
    bond up from W.V5 ; C
    double bond up from C ; O
    bond right from C ; N
    bond 45 from N ; C2H5
    bond 135 from N ; C2H5
.cend
```

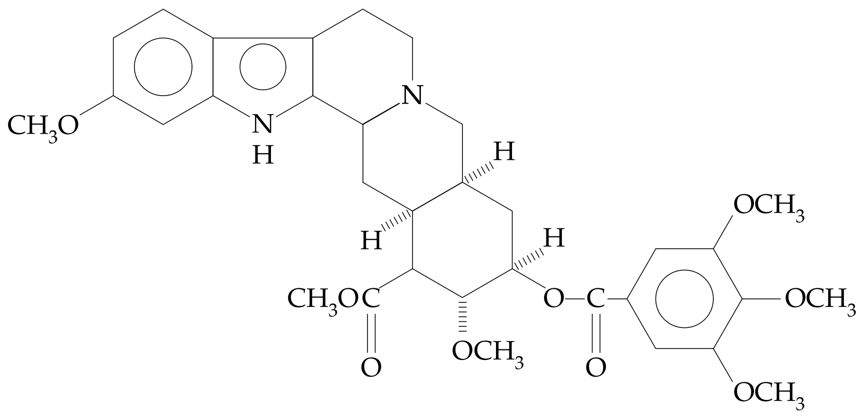
We mentioned before that *chem* attempts to connect bonds to the proper parts of moieties, connecting to carbon atoms when possible. Once in a while this may provide an unwanted effect, as shown in the left diagram of anisole. The CH_3 can be manually positioned by using `right of`, as shown in the diagram on the right.

```
.cstart
R1: benzene
    bond down from R1.V4 ; OCH3
R2: benzene at R1 + (1.5,0)
    bond down from R2.V4 ; O
    CH3 right of O
.cend
```



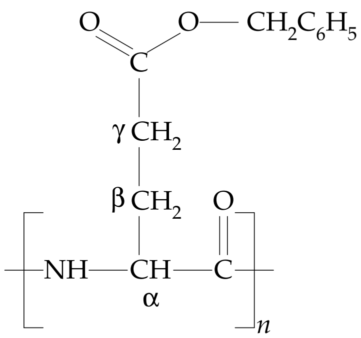
The following diagram of reserpine shows that *chem* gets the connectivity right most of the time; it is only when the OCH_3 is connected to a `bond up` or a `bond down` that there is a problem.

```
.cstart
    CH3O
    bond 60
R1: benzene
R2: aromatic flatring5 pointing down put N at 1 with .V3 at R1.V2
    H below R2.V1
R3: ring put N at 3 with .V5 at R2.V5
R4: ring put N at 1 with .V1 at R3.V3
    back bond -120 from R4.V4 ; H
    back bond 60 from R4.V3 ; H
R5: ring with .V1 at R4.V3
    bond -120 ; C
    double bond down from C ; O
    CH3O left of C
    back bond 60 from R5.V3 ; H
    back bond down from R5.V4 ; O
    CH3 right of O
    bond 120 from R5.V3 ; O
    bond right lenght .1 from O ; C
    double bond down ; O
    bond right length .1 from C
B:  benzene pointing right
    bond 30 from B ; OCH3
    bond right from B ; OCH3
    bond 150 from B ; OCH3
.cend
```



Brackets and Text

It is possible to make brackets of any size but only in a somewhat awkward fashion. The trick is to use the branch point, BP. This is poly(benzyl glutamic acid), which was also shown near the beginning of this manual:

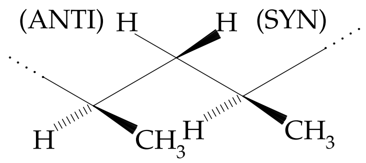


Anything that is enclosed in double quotes is treated as text and is passed through *chem* untouched. In particular, text intended for *eqn* is handled this way.

```
.cstart
# a left bracket
    bond right length .1 ; BP
    bond up length .3
    bond right length .1
    bond down length .3 from BP
    bond right length .1
# this is the mainchain amide structure
    bond right length .1 from BP ; NH
    bond right ; CH
# label the CH with an alpha, intended for eqn.
# this line says "put the north edge of the alpha at the
# south edge of the CH"
    "$alpha$" with .n at CH.s
    bond right from CH ; C
    double bond up from C ; O
    bond right length .1 from C ; BP
# a right bracket
    bond up length .3
    bond left length .1
    bond right length .1 from BP
    bond down length .3 from BP ; BP
    bond left length .1

# label the degree of polymerization
    "$n$" with .w at BP.se
# this is the sidechain
    bond up from CH ; CH2
    "$beta$" with .e at CH2.w
    bond up from CH2 ; CH2
    "$gamma$" with .e at CH2.w
    bond up from CH2 ; C
# this is the benzyl ester part
    double bond -60 from C ; O
    bond 60 from C ; O
    bond right ; CH2C6H5
.cend
```

Text can be positioned with various *pic* commands, as shown in the Greek letters above, and in this example:

```
.cstart
    bond 120 dotted
    bond 120 length .3 ; BP
    back bond -120 length .25 from BP ; H
    front bond 120 length .25 from BP ; CH3
    bond 60 length .5 from BP ; BP
    bond -60 length .25 from BP ; H
# note the pic move command to position the text
    move left .35 ; "(ANTI)"
    front bond 60 length .25 from BP ; H
# another positioning of text
    move right .35 ; "(SYN)"
    bond 120 length .4 from BP ; BP
    back bond -120 length .25 from BP ; H
    front bond 120 length .25 from BP ; CH3
    bond 60 length .5 from BP
    bond 60 dotted
.cend
```

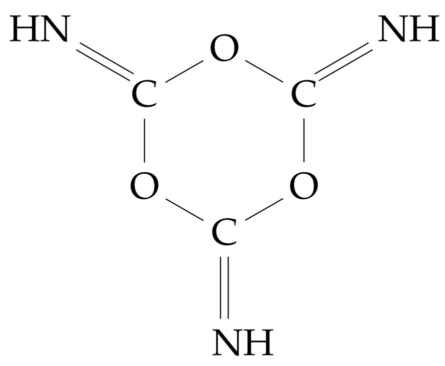


5. Changing the Size of Structures

The default point size for *chem* is 10; this paper is printed in size 10. The size of bonds, rings, etc., can be adjusted within the `.cstart` and `.cend` pair by specifying `size` *n*, where *n* is the desired point size. The text size itself is *not* changed, however; you must do that separately with *troff* `.ps` commands. The following example shows how *chem* could be used to make a diagram that is suitable for a viewgraph.

```
.cstart
.ps 14
size 16
R: ring6 put O at 1 put C at 2 put O at 3 put C at 4 put O at 5 put C at 6
   double bond 60 from R.V2 ; NH
   double bond down from R.V4 ; NH
   double bond -60 from R.V6 ; HN
size 10  # if you are doing more than one
.ps 10
.cend
```

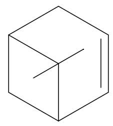
Don't forget that the `.ps` must begin in the first column, and that you need to reset the point size after the structure has been drawn.



6. Communicating with Pic

(This section can be skipped on a first reading.) Since *chem* translates your commands into *pic*, it is often possible to slip in a few *pic* commands of your own to draw things that *chem* can't handle itself. We have seen a few examples, such as the `at` commands in the section on rings. Here are some more.

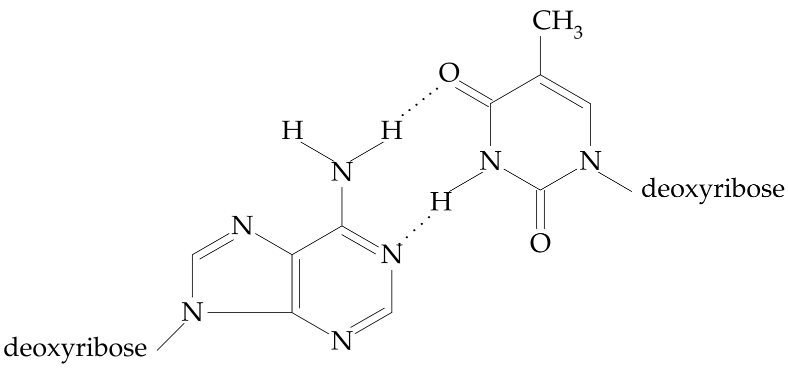
```
.cstart
R:  ring double 2,3
    line from R.V6 to R.C
    line from R.C to R.V4
X1: 1/2 <R.V5,R.C>
X2: 1/2 <R.C,R.V2>
    bond from X1 to X2
.cend
```



This relies on knowing that the center of a ring `R` is called `R.C`. The *pic* construction α`<`p_1`,`p_2`>` defines a point α of the way from p_1 to p_2.

The following diagram of part of DNA shows some more *pic* commands:

```
.cstart
P: [
R1: flatring pointing up put N at 1 put N at 4 double 5,1
    bond -135 from R1.V4 ; BP
    "deoxyribose" rjust with .e at BP.w
R2: ring6 put N at 2 put N at 4 double 1,2 3,4 5,6 with .V6 at R1.V2
    pic Conn: R2.V2.ne  #because naming is too restricted in pic
    bond up from R2.V1 ; N
    bond -60 from N ; H
    bond 60 from N ; H
]
    # thymine
Q: [
R3: ring6 put N at 3 put N at 5 double 1,2
    bond up from R3.V1 ; CH3
    bond 120 from R3.V3 ; BP
    "deoxyribose" ljust with .w at BP.e
    double bond down from R3.V4 ; O
    double bond -60 from R3.V6 ; O
    bond -120 from R3.V5 ; H
] with .O at P.H + (.3,.3)

    bond from Q.O.sw to P.H.ne dotted
    bond from Q.H.sw to P.Conn dotted
.cend
```



Here the trick is to use *pic*'s bracketing construct to make two pieces that are glued together at the desired position.

7. In Case of Difficulty

Chem does not provide much in the way of error diagnostics: if you make an error, the most likely result will be a complaint from *pic*, expressed in its own terms, not those of your diagram. For example, a typo like:

```
.cstart
rng5 put N at 2
```

produces the error message

```
pic: no such variable as rng5 near line 6, file -
 context is
     Last:  rng5 >>>  put <<<  N at 2
```

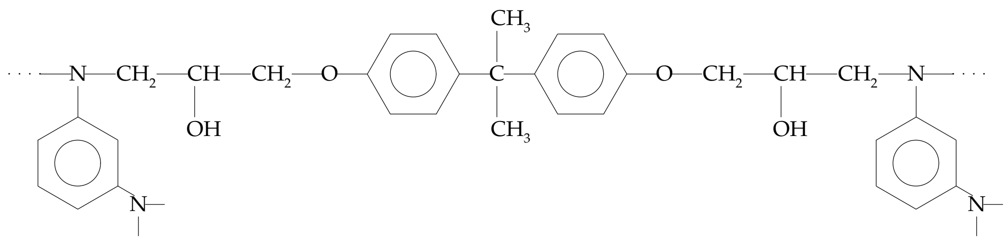
The brackets surround the first symbol that *pic* didn't recognize, which is usually close to the actual error. (The line number refers to *chem*'s output, not input.)

Chem is implemented as an *awk* program. If your system does not have a current version of *awk*, you will get syntax error messages from *awk* itself. Similar messages will appear if your version of *pic* is too antiquated.

Appendix A: Lots More Examples

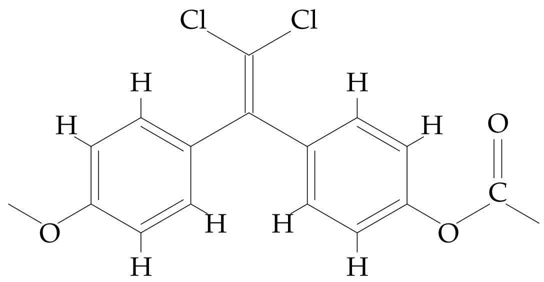
The first example shows how to put all the pieces together to make a fairly complex polymer repeat unit. Note that the bond direction does not need to be specified in the first three lines because the implicit direction is right. On the other hand, it doesn't hurt to specify the bond direction if you like.

```
 .cstart
# epoxy based on the m-phenyldiamine cured bisphenol-A
 size 8
 .ps 8
     bond dotted
     bond ; N
     bond ; CH2
     bond down from N
R1:  benzene
     bond 120 length .1 from R1.V3 ; N
     bond right length .1 from N
     bond down length .1 from N
# back to the CH2
     bond right from CH2 ; CH
     bond down from CH ; OH
     bond right from CH ; CH2
     bond right ; O
     bond right
     benzene pointing right
     bond right ; C
     bond up from C ; CH3
     bond down from C ; CH3
     bond right from C
     benzene pointing right
     bond right ; O
     bond right from O ; CH2
     bond right ; CH
     bond down from CH ; OH
     bond right from CH ; CH2
     bond right ; N
     bond right from N
     bond dotted
     bond down from N
R2:  benzene
     bond 120 length .1 from R2.V3 ; N
     bond right length .1 from N
     bond down length .1 from N
 .cend
```



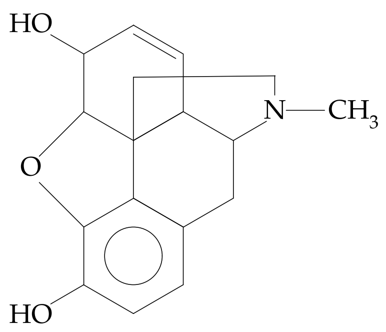
This input for chloral shows but one of many possible ways to build this structure. We chose to start with the vinyl chloro groups.

```
.cstart
    Cl
    bond 120 length .25 ; BP
    bond 60 length .25 from BP ; Cl
B1: double bond down length .3 from BP
    bond 120 length .35 ; BP
# now comes the ring
R1: ring6 double 1,2 3,4 5,6 with .V6 at BP
    bond up length .1 from R1.V1 ; H
    bond 60 length .1 from R1.V2 ; H
    bond 120 from R1.V3 ; O
    bond 60 from O ; C
    double bond up from C ; O
    bond 120 from C
# continue decorating the ring
    bond down length .1 from R1.V4 ; H
    bond -120 length .1 from R1.V5 ; H
# now go back and do the left hand ring
    bond -120 length .35 from B1.end ; BP
R2: ring6 double 1,2 3,4 5,6 with .V2 at BP
    bond up length .1 from R2.V1 ; H
    bond -60 length .1 from R2.V6 ; H
    bond -120 from R2.V5 ; O
    bond -60 from O
    bond down length .1 from R2.V4 ; H
    bond 120 length .1 from R2.V3 ; H
.cend
```



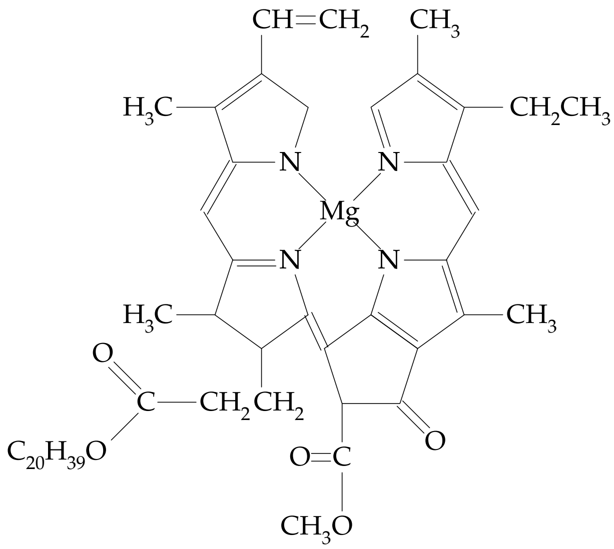
Sometimes it is necessary to draw bonds that connect to something, but you don’t know the exact angle for the bond. Morphine provides an example: the long horizontal bond near the nitrogen is specified in the last three lines of the input.

```
 .cstart
 R1: ring6 double 1,2
     bond -60 from R1.V6 ; HO
 R2: ring6 with .V1 at R1.V3
     bond 60 from R2.V2 ; N
     bond right from N ; CH3
 R3: benzene with .V1 at R2.V5
     bond -120 from R3.V5 ; HO
 # this is the furan ring
     bond -135 length .33 from R1.V5 ; O
     bond -45 length .33 from R3.V6
 # this is the odd ring
     bond up length .1 from N ; BP
B1:  bond up length .33 from R1.V4
     bond to BP
 .cend
```



This diagram of chlorophyll shows how *chem* can be used to produce very complex molecules. It was necessary to use `right of` and `left of` to get the CH_2CH_2 and the OCH_3 to position correctly. Also note how the bonds from Mg to N are specified. (We could have used dashed bonds, or we could have used `arrow` in *pic* to specify different bonds.)

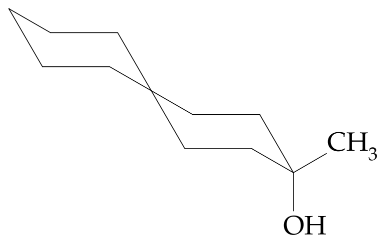
```
.cstart
    Mg
    bond 45 ; N
R1: ring5 pointing up put N at 4 double 1,2 4,5 with .V4 at N
         bond up from R1.V1 ; CH3
         bond right from R1.V2 ; CH2CH3
    bond 135 from Mg ; N
R2: ring5 pointing down put N at 3 double 1,2 4,5 with .V3 at N
         bond right from R2.V5 ; CH3
    bond 225 from Mg ; N
R3: ring5 pointing down put N at 4 double 3,4 with .V4 at N
    bond -45 from Mg ; N
R4: ring5 pointing up put N at 3 double 1,5 with .V3 at N
         bond left  from R4.V5 ; H3C
         bond up from R4.V1 ; CH
         double bond right length .1 from CH ; CH2
    double bond 150 length .3 from R1.V3
    bond to R2.V4
R5: ring5 pointing 72 with .V5 at R2.V2
         double bond 135 from R5.V2 ; O
         bond down from R5.V3 ; C
         double bond left length .1 from C ; O
         bond down from C ; O
         CH3 left of O
    double bond -25 from R5.V4
    bond down from R3.V1 ; CH2
    CH2 left of CH2
    bond left ; C
    double bond -45 ; O
    bond -135 from C ; C20H39O
    bond left from R3.V2 ; H3C
    double bond -150 length .3 from R4.V4
    bond to R3.V3
.cend
```



The next example shows how to escape into *pic* to make structures that are not provided for in *chem*. First we define the cyclohexane chair structure as `chair` in *pic*, and we then glue several of the chairs together to make the desired structure. Substituents can be added to these *pic*-

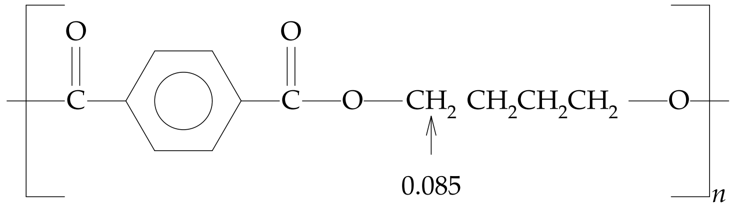
defined structures, just as with rings.

```
.cstart
pic define chair { [
    V1:  bond 120 length .25
    V2:  bond right length .35
    V3:  bond 150 length .35
    V4:  bond -60 length .25
    V5:  bond left length .35
    V6:  bond to V1.start
pic ] }

R1: chair
R2: chair with .V1 at R1.V4.start

bond 60 from R2.V4.start ; CH3
bond down from R2.V4.start ; OH
.cend
```



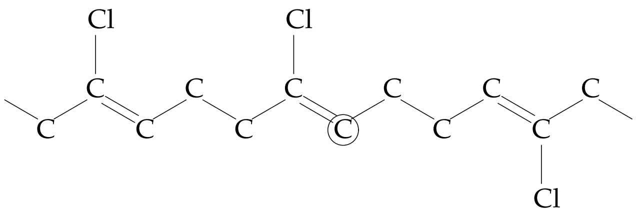
The next example shows how to position arrows and text beneath certain parts of a structure. The ''up arrow'' is of default length, and is drawn with the `line <-` statement.

```
.cstart
    bond length .1 ; BP
    bond up length .5
    bond right
    bond down length .5 from BP
    bond right
    bond right from BP ; C
    double bond up ; O
    bond right from C
    benzene pointing right
    bond right ; C
    double bond up from C ; O
    bond right from C ; O
    bond right ; CH2
# this is the statement to make the arrow
    line <- from CH2.s down
    move down .1 ; "0.085"
    CH2CH2CH2 right of CH2
    bond right ; O
    bond right length .1 ; BP
    bond up length .5 from BP
    bond left
    bond right length .1 from BP
    bond down length .5 from BP ; BP
    bond left
    "$n$" with .w at BP.se
.cend
```



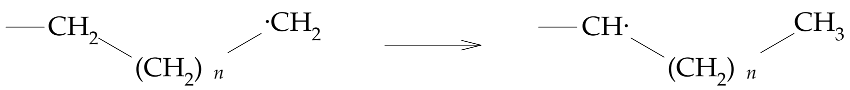
We can draw circles around atoms or groups of atoms as is shown in the following structure.

```
.cstart

bond 120 ; C
bond 60 ; C
    bond up ; Cl
double bond 120 from C ; C
bond 60 ; C
bond 120 ; C
bond 60 ; C
    bond up ; Cl
double bond 120 from C ; C
    circle at C rad .08
bond 60 from C ; C
bond 120 ; C
bond 60 ; C
double bond 120 ; C
    bond down ; Cl
bond 60 from C ; C
bond 120
.cend
```



Chemical equations can be drawn by specifying the various components between square brackets, and then using *pic* to make arrows and to position text.

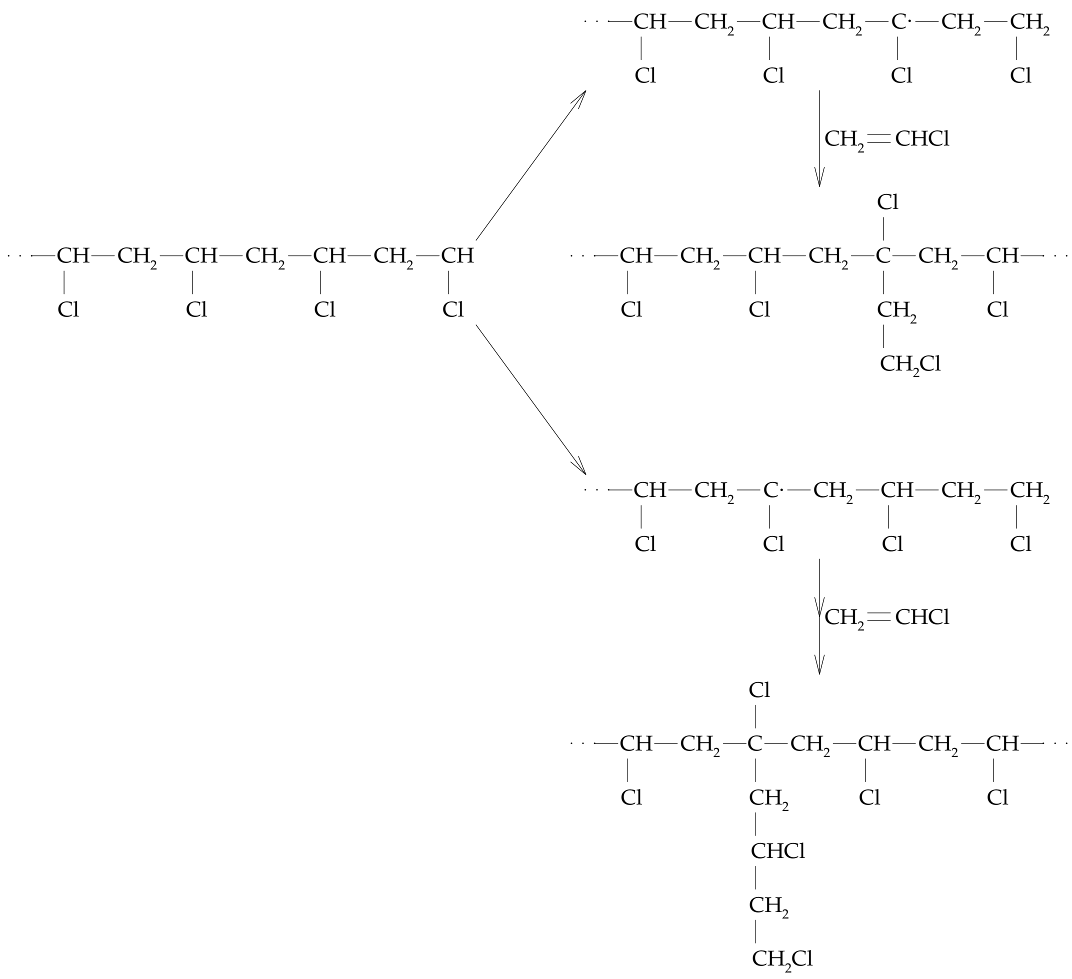
```
.cstart
[
    bond right ; CH2
    bond 120 ; (CH2)
    "$nothing sub n$"
    bond 60 ; .CH2
]
# now put the arrow in
    move right .3
    arrow .5
    move right .3
# begin second structure
[
    bond right ; CH.
    bond 120 ; (CH2)
    "$nothing sub n$"
    bond 60 ; CH3
]
.cend
```



The final example is a long and repetitive exercise, poly(vinyl chloride).

```
 .ps 8
 .cstart
db = .12
cwid = .095
A: [
	bond dotted
	bond right ; CH
	bond down ; Cl
	bond right from CH ; CH2
	bond ; CH
	bond down ; Cl
	bond right from CH ; CH2
	bond ; C.
	bond down ; Cl
	bond right from C ; CH2
	bond ; CH2
	bond down ; Cl
]
"	(6.13a)" ljust at A.e

arrow down .5 from A.s
[
	CH2
	double bond right ; CHCl
] with .w at last arrow.c

B: [
	bond dotted
	bond right ; CH
	bond down ; Cl
	bond right from CH ; CH2
	bond ; CH
	bond down ; Cl
	bond right from CH ; CH2
	bond ; C
	bond up ; Cl
	bond down from C ; CH2
	bond ; CH2Cl
	bond right from C ; CH2
	bond ; CH
	bond down ; Cl
	bond right from CH
	bond dotted
] with .n at end of last arrow
"	(6.13b)" ljust at B.e

C: [
	bond dotted
	bond right ; CH
	bond down ; Cl
	bond right from CH ; CH2
	bond ; C.
	bond down ; Cl
	bond right from C ; CH2
	bond ; CH
	bond down ; Cl
	bond right from CH ; CH2
	bond ; CH2
	bond down ; Cl
] with .n at B.s - (0,.5)
"	(6.14a)" ljust at C.e

arrow down .3 from C.s
[
	CH2
	double bond right
	CHCl
] with .w at last arrow.s
arrow down .3 from last arrow.s

D: [
	bond dotted
	bond right ; CH
	bond down ; Cl
	bond right from CH ; CH2
	bond ; C
	bond up ; Cl
	bond down from C ; CH2
	bond ; CHCl
	bond ; CH2
	bond ; CH2Cl
	bond right from C ; CH2
	bond ; CH
	bond down ; Cl
	bond right from CH ; CH2
	bond ; CH
	bond down ; Cl
	bond right from CH
	bond dotted
] with .n at last arrow.s
"	(6.14b)" ljust at D.e

E: [
	bond dotted
	bond ; CH
	bond down ; Cl
	bond right from CH ; CH2
	bond ; CH
	bond down ; Cl
	bond right from CH ; CH2
	bond ; CH
	bond down ; Cl
	bond right from CH ; CH2
	bond ; CH
	bond down ; Cl
] with .e at B.w - (.5,0)
```

```
arrow from E.ne to A.sw
arrow from E.se to C.nw
 .cend
```



(6.13a)

(6.13b)

(6.14a)

(6.14b)

Appendix B: *chem* Command Summary

Structure descriptions begin with `.cstart` and end with `.cend`, with the ‘`.`’ always in the first column. Labels begin with a capital letter and are followed by ‘`:`’. The command `size` *n* scales the diagram to size *n*; default is 10. Comments begin with a #; any characters from the # to the end of the line are ignored. Any line beginning with a `.` is copied verbatim; any line beginning with `pic` is copied verbatim after the `pic` is removed. All other lines are assumed to be valid *pic* commands and are passed through labeled but otherwise verbatim.

`chem` *filenames* `| pic` *[* `| tbl | eqn` *]* `| troff`

Atoms or Groups

Atoms or groups must begin with a capital letter. Numbers are automatically converted to subscripts and dots are centered on the line. An atom or group always defines a name for a place. If the group contains special characters, the name is what is left after the special characters are stripped out. The name `BP` (‘‘branch point’’) does not draw anything.

Examples:

```
.cstart
    SiO2              # name = SiO2
    move right 1
    CH3CH2NH2.HCl     # name = CH3CH2NH2HCl
.cend
```

SiO_2 $CH_3CH_2NH_2{\cdot}HCl$

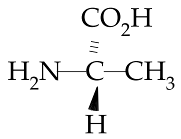
Bonds

type **bond** *[direction] [* `length` *n] [* `from` *name] [* `to` *name] [attribute]*

type:	**double**, **triple**, **front**, **back**
direction:	**up**, **down**, **left**, **right**, angle in degrees (0° is up, 90° is right)
length *n*:	*n* in inches, default 0.2
attribute:	**dotted**, **dashed**

Example:

```
.cstart
    C
    frontbond -170 from C ; H
    backbond 10 from C ; CO2H
    bond left length .15 from C ; H2N
    bond right from C ; CH3
.cend
```



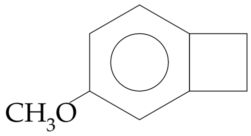
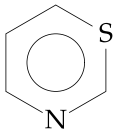
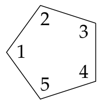
Rings

[modifier] `ring`*n [* `pointing` *direction] [heteroatoms] [doublebonds]*

modifier:	**aromatic**
ring*n*:	**ring3**, ... **ring8**, **flatring** (5-membered ring) or **benzene**
direction:	**pointing up** (default), **down**, **left**, **right** or angle in degrees
heteroatoms:	**put** *X* **at** *V*, where X is the heteroatom and V is the vertex
vertices:	numbered clockwise; the ''point'' is always **.V1** (vertex 1)
double bonds:	**double** i,j draws a double bond between vertices i and j **triple** i,j draws a triple bond
fused rings:	**ring***n* **with .V***i* **at R.V**j fuses ith vertex to jth vertex of previous ring R

Examples:

```
.cstart
  R1:   benzene
        bond -120 from R1.V5 ; CH3O
  R2:   ring4 pointing 45 with .V4 at R1.V2
  R3:   aromatic ring6 put N at 4 put S at 2 at R2 + (.75,0)
  R4:   ring5 pointing left at R3 + (.75,0)
        label R4
.cend
```

Positioning Substituents

group position name

position: **right of**, **left of**, **above**, **below**

Examples:

```
.cstart
   R1: flatring pointing down put N at 2 put N at 5 double 1,2 3,4
       H right of R1.V5
   R2: ring7 put N at 1 at R1 + (1.0,0)
       H above R2.V1
.cend
```