# PPCHTEX

typesetting chemistry in ConTEXt LMTX

work in progress

Hans Hagen
Ton Otten
Alan Braslau

October 2025

# Contents

# Introduction

This manual is an update of the ppchTEX manual. Being one of the first public packages in what became the ConTEXt suite, it covers a set of coherent macros that can be used to typeset chemical structure formulas. The first version of ppchTEX was ready for use in 1995 and the second release came with the previous versions of this manual. Some 15 years later, in the process of upgrading ConTEXt to suit LuaTEX, it made sense to reconsider the usage of ppchTEX.[1] First of all, we want basic chemical support in the kernel, so it made sense to take the relevant code and turn it into a kernel module. This means that in ConTEXt MkIV one does not need to load any module. For compatibility we keep the old module around.

The old code can also be used outside ConTEXt, but the integrated code not. This permits us to get away from the old PICTEX based approach and combine the power of TEX, MetaPost and Lua. The differences for ConTEXt users are not that noticeable as we already default to MetaPost for rendering the lines and curves, but in ConTEXt MkIV users could notice that the runtime is no longer influenced by callouts to MetaPost because there we use the integrated library. We also changed a couple of aspects like dealing with fonts and provide more control over color, as we no longer need to be generic. The fact that we operate within ConTEXt also meant that we can use all its facilities (like interaction) without special precautions.

This means that this manual is just an overview as before but done a bit different. When we moved on to LuaMetaTEX and MkXL, aka LMTX, the chemical modules didn't change apart from adaptations to more efficient MetaPost integration. This manual is now targetting that version but still is mostly an overview. It is important to keep in mind that the code is not compatible with MkII, mostly because we decided to make the interface a bit more flexible for which we also needed a bit more consistency across structures, especially the usage of plus and minus signs.

As a final introductory note, chemical typesetting has its own rules and practices as well as variants. Using the chemical macros described here, it may be possible to construct structures that do not make any chemical sense, or to produce representations that are incorrect, as no chemical knowledge is built-in. Indeed, certain operations are available for completeness and by symmetry and may not have any reality in chemical practice. The user is responsible for a correct and consistant usage and should not count on the macros to ensure the construction of valid structures.

[1] The original module was written by J. Hagen and A.F. Otten (with contributions by T. Burnus and D. Kuypers) and provided a set of macros based on PICTEX or pstricks for the actual drawing of graphics. It could be used with ConTEXt as well as with LATEX.

# 1 Structures

The number of commands that is used to typeset chemical structure formulas is relatively small once you see the pattern. Often most you need is the following.

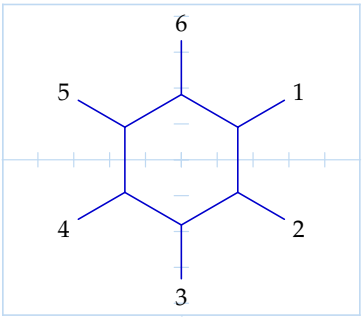



**Figure 1.1**

**Example 1**

```
\setupchemical[axis=on,frame=on,width=5000,axis=on]

\startchemical
  \chemical[SIX,B,+SR,RZ][1,2,3,4,5,6]
\stopchemical
```

With `\setupchemical` we can influence the makeup of the formulas. These setups influence all the following formulas, unless they are superceded by local setup variables.

The set up variables can be defined right after `\startchemical`. In that case the set up is only applied to one structure formula.

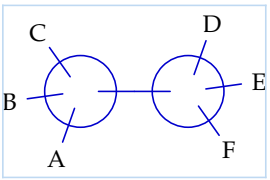



**Figure 1.2**

**Example 2**

```
\startchemical[frame=on,width=fit,height=fit]
  \chemical[CARBON,CB1][A,B,C,D,E,F]
\stopchemical
```

Both examples show that `\chemical` is the essential command. This command, that may be used more than once within a `\start-\stop`-pair, is accompanied with two arguments. These arguments are written between `[ ]`. The first argument is used for defining the chemical bonds, the second argument for the atoms and molecules that make up the structure.

> If you use(d) this module in the early days you will notice that we use `+SR` instead of just `R` here. The reason is that it is more consistent to have `R` for the full-length radical line in all constructs, at the TEX end as well as in MetaPost. This means that older code has to be adapted.

The carbon example looks simple but in fact is pretty complex. It actually is

```
\chemical[CARBON,C,SB,Z234,1.5MOV1,MIRO,C,SB,Z234][A,B,C,D,E,F]
```

and the simplification comes from `CB1` being a shortcut definition in the `CARBON` namespace. This will be discussed later, so consider it a teaser.

Text is typeset in mathematical mode, this means that you may type anything that normally is allowed between `$ $` or with `\im`.

We will explore the examples in more detail. The key `SIX` means that we want to draw a sixring. In analogy we could type `ONE`, `THREE`, `FOUR` and `FIVE`, `EIGHT`, `CARBON`, `NEWMAN`, `CHAIR`, some alternatives on these keys and some symbols.

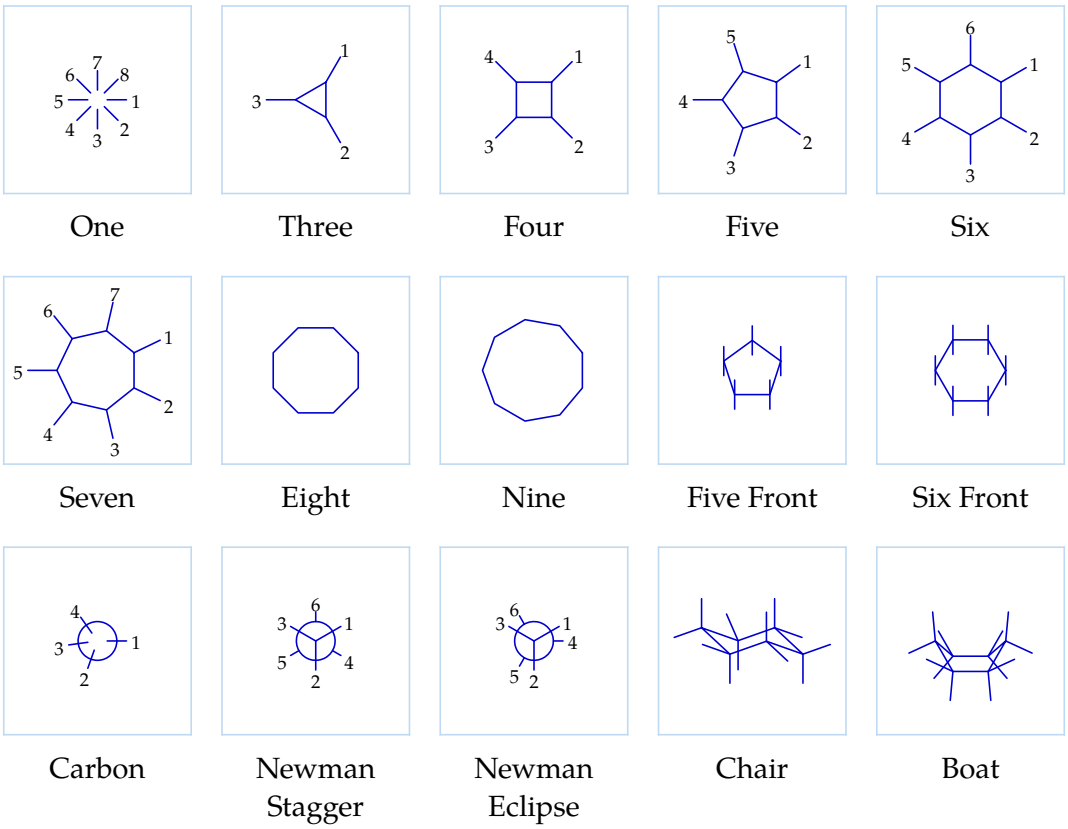



**Figure 1.3** A few examples

The dimensions of `CHAIR` and `BOAT` are somewhat different from the others. This structure is also different in other ways. Rotation for example is not possible.

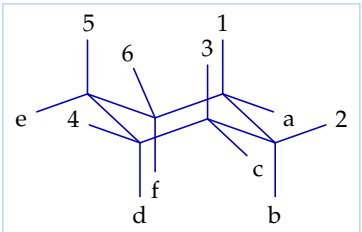



**Figure 1.4**

**Example 3**

```
\startchemical[frame=on,width=5000]
  \chemical
    [CHAIR,B,+LSR,+RSR,+LRZ,+RRZ]
    [a,b,c,d,e,f,1,2,3,4,5,6]
\stopchemical
```

Within a structure the chemical bonds between the C–atoms are defined in the same way. In this example we use `B` and `SR`. Bonds within a structure are numbered and can be defined by:

```
\chemical[SIX,B1,B2,B3,B4,B5,B6]
\chemical[SIX,B135]
\chemical[SIX,B1..5]
```

These keys draw parts of a sixring. With R and RZ we place substituents on the ring. The key R draws the bond from a ring corner to the substituent ($\angle \sim 120°$). The corner is also identified with a number.

```
\chemical[SIX,B1..6,+SR1..6]
```

The definition above draws the six bonds in the sixring and the bonds to the substituents. The substituents are identified by the key RZ. Again numbers are used to mark the position. The substituents themselves are defined as text in the second argument.

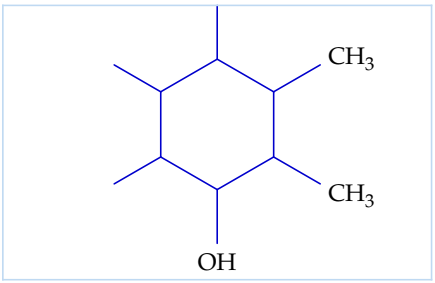



**Figure 1.5**

**Example 4**

```
\startchemical[frame=on,width=6000]
  \chemical
    [SIX,B1..6,+SR1..6,RZ1..3]
    [CH_3,CH_3,OH]
\stopchemical
```

When the second argument is left out no text (substituents) are placed on the ring and the key RZ1..3 has no effect.

Splitting the specification part from the text has the advantage that we can predefine structures (as we will see later) that we call within a structure and that pick up text from the calling structure. In such a nested case the texts are put on a sort of stack and fetching a text can involve the whole stack.

The disadvantage is that in complex structures it might be hard to see which text belongs to what specification. Therefore we provide a key/value variant as well.

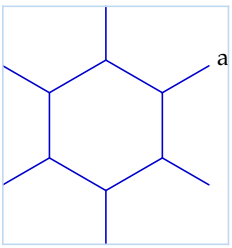



**Figure 1.6**

**Example 5**

```
\startchemical[frame=on]
  \chemical[SIX,B,+SR,RZ1=a]
\stopchemical
```

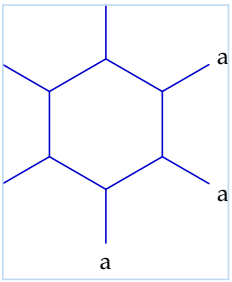



**Figure 1.7**

**Example 6**

```
\startchemical[frame=on]
  \chemical[SIX,B,+SR,RZ1..3=a]
\stopchemical
```

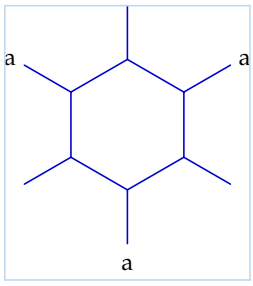



**Figure 1.8**

**Example 7**

```
\startchemical[frame=on]
  \chemical[SIX,B,+SR,RZ135=a]
\stopchemical
```

As you can see here, the text is repeated in the case of a key specification that involves more bonds. Compare this with the split definition approach:

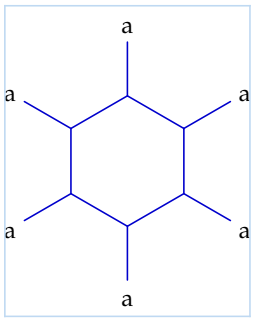



**Figure 1.9**

**Example 8**

```
\startchemical[frame=on]
  \chemical[SIX,B,+SR,RZ=a]
\stopchemical
```

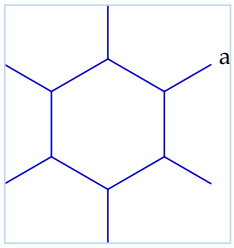



**Figure 1.10**

**Example 9**

```
\startchemical[frame=on]
  \chemical[SIX,B,+SR,RZ] [a]
\stopchemical
```

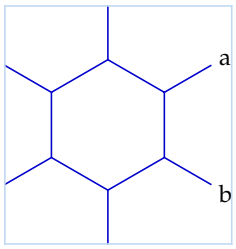



**Figure 1.11**

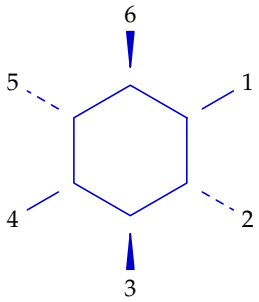
**Example 10**

```
\startchemical[frame=on]
  \chemical[SIX,B,+SR,RZ] [a,b]
\stopchemical
```

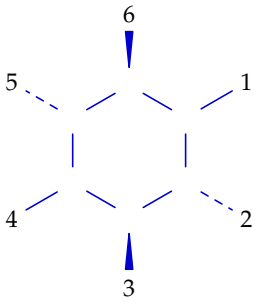
# 2 Bonds

*Let me know if some bolds are not mentioned here an needs to gbe discussed. As the is an update from the older syntax there might be errors, so you can also send me fixes.*

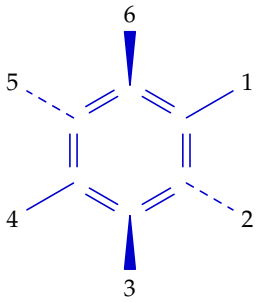
This chapter gives an overview of the bonds you can use in structures. From the examples throughout this manual the use of the different keys will become more meaningful. There is a clear pattern in the specifications so we start with an example where `S` means single and `D` refers to double. The signs influence the attachment (or length for that matter).




`B,SR14,RD25,RB36,RZ`




`SB,SR14,RD25,RB36,RZ`




`DB,+SR14,+RD25,+RB36,RZ`

Bonds always have two alternatives: a long and a short version. The shortened bonds leave room to place atoms within a structure. A number of bonds can be shortened on both sides left (`-`) or right (`+`).

| | | | |
|---|---|---|---|
| `B` | Bond | `SB` | Single Bond |
| `BB` | Bold Bond | `-SB` | Left Single Bond |
| `HB` | Hydrogen Bond | `+SB` | Right Single Bond |

**Table 2.1** Single bonds.

The example below shows a number of bonds combined within one structure:

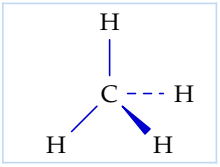



**Figure 2.1**

**Example 11**

```
\startchemical[frame=on,width=3000]
  \chemical[ONE,SD1,SB4,BB2,SB7,Z01247][C,H,H,H,H]
\stopchemical
```

A bond can be followed by one or more numbers or a range, for example: `B1`, `B135` and `B1..5`. When you want to draw all bonds you can type `B`.

Within a ring structure you can define extra bonds between atoms, for example a double or triple bond.

| | | | |
|---|---|---|---|
| EB | Extra Bond | DB | Double Bond |
| | | TB | Triple Bond |

**Table 2.2** Multiple bonds.

Free electrons and electron pairs can be defined in different ways. The accompanying keywords start with an E.

| | | | |
|---|---|---|---|
| ES | Extra Single | ED | Extra Double |
| EP | Extra Pair | ET | Extra Triple |

**Table 2.3** Free electrons and electron pairs.

The example below shows a carbon atom with 8 outer electrons arranged in a chemically very peculiar way.

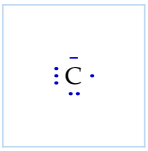


**Figure 2.2**

**Example 12**

```
\startchemical[frame=on,width=2000,height=2000]
  \chemical[ONE,Z0,ES1,ED3,ET5,EP7][C]
\stopchemical
```

Within a ring structure you can make a shortcut from one atom to another. In that case the atom that you want to skip has to be identified. As a replacement of the double bonds in an aromatic sixring a circle can be drawn and charges can be placed within the ring.

| | | | |
|---|---|---|---|
| SS | Short Shortcut | S | Shortcut |
| -SS | Left Short Shortcut | MID | Open Mid Shortcut |
| +SS | Right Short Shortcut | MIDS | Closed Mid Shortcut |

**Table 2.4** Special bonds.

| | | | |
|---|---|---|---|
| C | Circle | CD | Dashed Circle |
| CC | Shifted Circle | CCD | Dashed Shifted Circle |

**Table 2.5** Circle bonds.

An example will explain the use of the circular bond and the use of displaced charges.

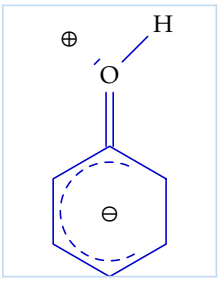



**Figure 2.3**

**Example 13**

```
\startchemical[frame=on,width=3000]
  \chemical
    [SIX,B,+ER6,CCD12,Z0,SUB6,ONE,SB8,EP6,Z0,ZT6,Z8]
    [\ominus,O,\oplus,H]
\stopchemical
```

Substituents can be connected to all corners of a structure. A substituent can be anything you want. It depends on the presence of atoms or molecules whether the bonds are long or short. In the examples you will see a great number of the keys that are used to define substituents.

| | | | |
|---|---|---|---|
| R | Radical | SR | Single Radical |
| -R | Left Radical | -SR | Single Left Radical |
| +R | Right Radical | +SR | Single Right Radical |

**Table 2.6** Bonds to substituents.

There are a few alternatives to draw bridges.

| | | | |
|---|---|---|---|
| RD | Radical Dashed | RB | Radical Bold |
| -RD | Left Radical Dashed | -RB | Left Radical Bold |
| +RD | Right Radical Dashed | +RB | Right Radical Bold |

**Table 2.7** Special bonds to substituents.

Radicals can be drawn in three ways.[2] Some alternatives are seldom used.

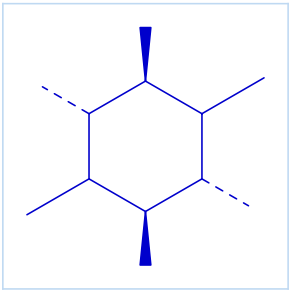


**Figure 2.4**

**Example 14**

```
\startchemical[frame=on,width=4000,height=4000]
  \chemical[SIX,B,R14,+RD25,+RB36]
\stopchemical
```

| | | | |
|---|---|---|---|
| SD | Single Dashed | LDD | Left Double Dashed |
| OE | Open Ended | RDD | Right Double Dashed |

**Table 2.8** More special bonds to substituents.

An example of an *Open Ended* is defined below. We see a sixring (SIX) with a number of consecutive ONEs. The use of PB is explained later.

[2] The word radical schould not be interpreted chemically, but typographically.

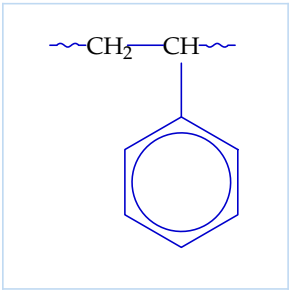



**Figure 2.5**

**Example 15**

```
\startchemical
    [width=4000,right=1500,
     top=2500,bottom=1500,
     frame=on]
  \chemical
    [SIX,
     B,C,+SR6,
     SUB6,ONE,CZ0,OE1,SB5,MOV5,CZ0,OFF5,OE5]
    [CH,CH_2]
\stopchemical
```

It's obvious that substituents can be attached to the structure by means of double bonds.

| | | | |
|---|---|---|---|
| ER | Extra Radical | DR | Double Radical |

**Table 2.9** Double bonds to substituents.

You can comment on a bond. Text is typed in the second argument of `\chemical`.

| | | | |
|---|---|---|---|
| Z | Atom | RZ | Radical Atom |
| CRZ | Center Atom | -RZ | Left Radical Atom |
| MIDZ | Mid Atom | +RZ | Right Radical Atom |

**Table 2.10** Atoms and molecules (radicals).

From these keys `RZ` is an addition to the key `R`. The key `MID` is only available in combination with a sixring (`SIX`). In the example below we see the effects of `MID` and `MIDZ`. These keys have no positioning parameter.

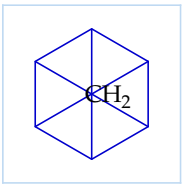



**Figure 2.6**

**Example 16**

```
\startchemical[frame=on,width=2500,height=2500]
  \chemical[SIX,B,MID,MIDZ][\SL{CH_2}]
\stopchemical
```

Atoms and molecules are numbered clockwise. Combinations are also allowed. Position `0` (zero) is the middle of a structure.

We can attach labels and numbers to an atom or a bond. This is done with `ZN` and `ZT`:

| | | | |
|---|---|---|---|
| ZN | Atom Number | ZT | Atom Text |

**Table 2.11** Labels and numbers.

In case of a `SIX` and a `FIVE` we can also attach text to radicals. We use `RN` and `RT`.

| | | | |
|---|---|---|---|
| RN | Radical Number | RT | Radical Text |
| RTN | Radical Top Number | RTT | Radical Top Text |
| RBN | Radical Bottom Number | RBT | Radical Bottom Text |

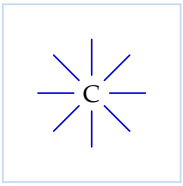
**Table 2.12** Labels and numbers.

The structure ONE has also a top and bottom alternative.

| | | | |
|---|---|---|---|
| ZTN | Atom Top Number | ZTT | Atom Top Text |
| ZBN | Atom Bottom Number | ZBT | Atom Bottom Text |

**Table 2.13** Extra labels and numbers.

With the keys ZTN and ZBN numbers are generated automatically. The other keys will use the typed text of the second argument of chemical.




**Example 17**

```
\startchemical[frame=on,width=2500,height=2500]
  \chemical[ONE,SB,Z0,ZTT][C,a,b,c,d,e,f,g,h]
\stopchemical
```

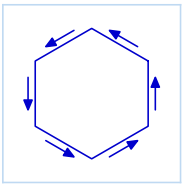
**Figure 2.7**

You can also add some symbols to the structure.

| | | | |
|---|---|---|---|
| AU | Arrow Up | AD | Arrow Down |

**Table 2.14** Indications.

The arrows are positioned between the atoms in a structure.



**Example 18**

```
\startchemical[frame=on,width=2500,height=2500]
  \chemical[SIX,B,AU]
\stopchemical
```

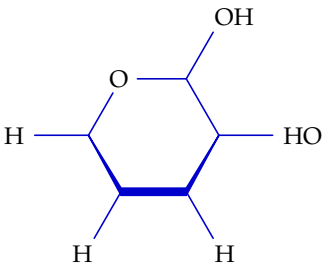
**Figure 2.8**

We want to add that while typesetting atoms and molecules the dimensions of these atoms and molecules are taken into account. The width of C and the height of $C_m^n$ play an important role during positioning. This mechanism may be refined in a later stage.

# 3 Frontviews

Structures `FIVE` and `SIX` can be displayed in a frontview. However there are some limitations. Frontviews can not be rotated. Also the coupling of several structures is limited.

We illustrate the frontview keys in two examples.



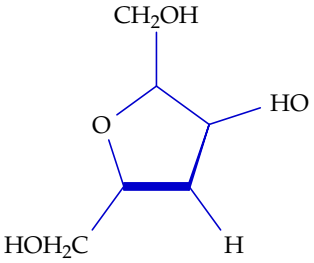

$\beta$-D-Fructopyranose

**Figure 3.1**

**Example 19**

```
\startchemical[height=4500,bottom=2500]
  \bottext{$\beta$-D-Fructopyranose}
  \chemical
    [SIX,FRONT,BB1236,+SB4,-SB5,Z5][O]
  \chemical
    [SIX,FRONT,+R12346,+RZ12346][HO,H,H,H,OH]
\stopchemical
```

Positioning the radicals is an optimization of feasability and quality. The next example will illustrate this.




$\alpha$-D-Fructofuranose

**Figure 3.2**

**Example 20**

```
\startchemical[height=4500,bottom=2500]
  \bottext{$\alpha$-D-Fructofuranose}
  \chemical
    [FIVE,FRONT,BB125,+SB3,-SB4,Z4][O]
  \chemical
    [FIVE,FRONT,+R1235,+RZ1235][HO,H,HOH_2C,CH_2OH]
\stopchemical
```

# 4 Definitions

It is possible to build a library of structures. These predefined structures can be used in a later stage, for example as a component of a more complex structure. Predefinition can be done with the TEX–primitive `\def`.

```
\def\sixring{\chemical[SIX,B,+R,RZ]}
```

However it is better to use the command `\definechemical`. In that case a message will occur during processing if a duplicate name is found.

```
\definechemical[sixring]
  {\chemical[SIX,B,+R,RZ]}
```

Recalling `\chemical[sixring]` will display a bare sixring without substituents.

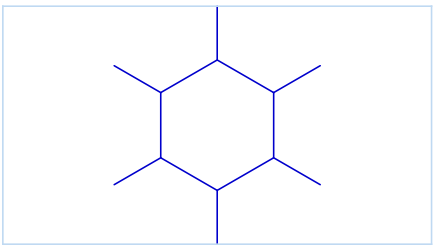


**Figure 4.1**

**Example 21**

```
\definechemical[sixring]
  {\chemical[SIX,B,+R,RZ]}

\startchemical[frame=on,width=6000]
  \chemical[sixring]
\stopchemical
```

If we want to attach six substituents in a later stage to a sixring we could type:

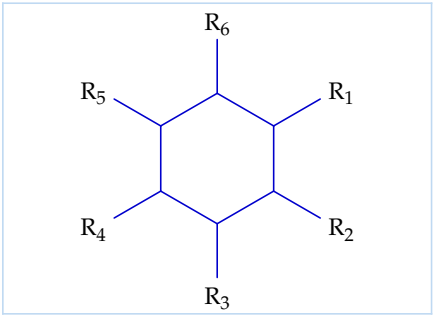



**Figure 4.2**

**Example 22**

```
\definechemical[sixring]
  {\chemical[SIX,B,+R,RZ]}

\startchemical[frame=on,width=6000]
  \chemical[sixring][R_1,R_2,R_3,R_4,R_5,R_6]
\stopchemical
```

The structure `sixring` can be defined without substituents (`RZ`). We could attach them after recalling `\chemical[sixring]`.

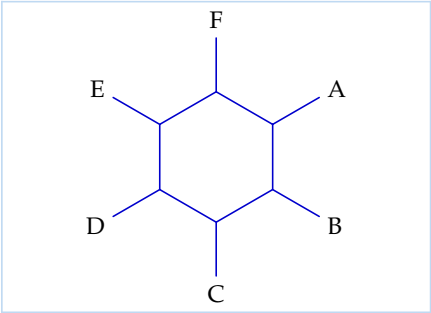



**Figure 4.3**

**Example 23**

```
\definechemical[sixring]
  {\chemical[SIX,B,+R]}

\startchemical[frame=on,width=6000]
  \chemical[sixring,RZ][A,B,C,D,E,F]
\stopchemical
```

In principal the possibilities are unlimited. However, you should remember that atoms and molecules are selected from the second argument in the order of definition in the first argument.

A definition may contain atoms and molecules (texts).

```
\definechemical[sixring]
  {\chemical[SIX,B,+R,RZ135][R_1,R_3,R_5]}
```

In the example above there will always be three substituents. If we want to attach more substituents we have to indicate explicitly that we want to continue with the sixring (`SIX`).

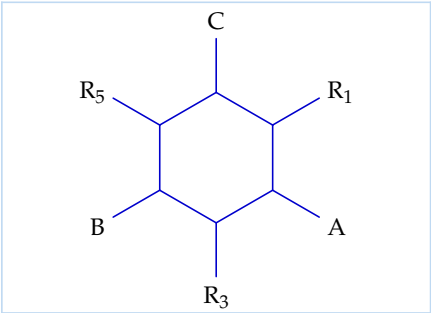



**Figure 4.4**

**Example 24**

```
\definechemical[sixring]
  {\chemical[SIX,B,+R,RZ135][R_1,R_3,R_5]}

\startchemical[frame=on,width=6000]
  \chemical[sixring,SIX,RZ246][A,B,C]
\stopchemical
```

In a definition `\chemical[]` has a global scope (this means that `SIX` is remembered) and `\chemical[][]` has a local scope. The idea behind this is that in the first case a range of keys can be added and in the second case a complete structure.

In a definition `\chemical` may be used more than once. The last example could have been defined thus:

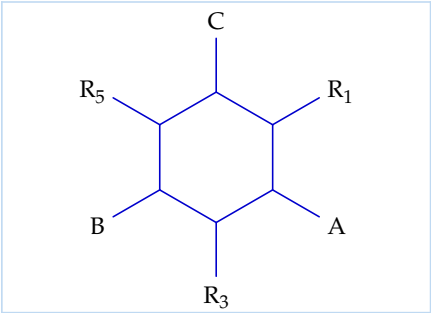



**Figure 4.5**

**Example 25**

```
\definechemical[sixring]
  {\chemical[SIX,B,+R,RZ135][R_1,R_3,R_5]
   \chemical[SIX,RZ246]}

\startchemical[frame=on,width=6000]
  \chemical[sixring][A,B,C]
\stopchemical
```

When TEX announces that an `unknown command` has occurred, you may have forgotten to type `SIX`, `FIVE` or a comparable key.

# 5 Combinations

Structures can be combined to more complex compounds. Moving one structure in relation to another structure is done by `MOV`, `ROT`, `ADJ` and `SUB`.

| | | |
|---|---|---|
| `MOV` | Move | moving a comparable structure structure in the direction of a bond |
| `ADJ` | Adjace | moving another structure in the direction of the $x$– or $y$–axis, adjacent to a bond |
| `SUB` | Substitute | moving one structure in relation to another in the direction of the $x$– or $y$–axis |
| `ROT` | Rotate | rotating a structure |

**Table 5.1** Moving and rotating.

The four keys mentioned above have different effects when they are applied to different structures. The angle of rotation in `\chemical[FIVE,ROT1,B]` differs from that in `\chemical[SIX,ROT1,B]`.

With the structure `ONE` you can use `MOV` but the key `DIR` is also available. Both keys have the same effect but differ in spacing. Small adjustments are possible with `OFF`.

| | | |
|---|---|---|
| `DIR` | Direction | moving a structure in a diagonal direction |
| `OFF` | Offset | moving atoms and molecules over small distances |

**Table 5.2** Moving and rotating.

The structure `CARBON` can be mirrored with `MIR`.

| | | |
|---|---|---|
| `MIR` | Mirror | mirroring a structure |

**Table 5.3** Mirroring.

We use a number to indicate the direction of a movement or the level of rotation. These keys are closely related with the structure. Therefore they have to be defined before bonds are drawn and texts are placed. So definition `\chemical[FIVE,B,ROT1,R]` and `\chemical[FIVE,ROT1,B,R]` will not have the same result. The first definition will give an undesirable result.

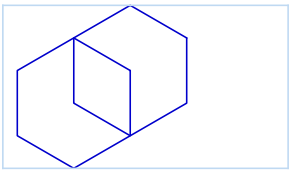


**Figure 5.1**

**Example 26**

```
\startchemical[frame=on,width=4000,right=3000]
  \chemical[SIX,B,MOV1,B]
\stopchemical
```

In this example a sixring is drawn because of `SIX,B`. Then a movement in the direction of bond 1 takes place and a second sixring is drawn: `B` (`SIX` is stil in effect).

A movement with `MOV` in a sixring can occur in six directions. A movement with `ADJ` will take place in only four axis–directions $(x, -x, y, -y)$. It is a coincidence that in a sixring some of these movements have the same effect. The example above could have been drawn with: `[SIX,B,ADJ1,B]`.

Structures can be combined. It is possible for example to combine structure `FIVE` with structure `SIX` in such a way that they have one mutual bond. Luckily the mechanism that takes care of these kinds of combinations is hidden for the user. In the next example you will see a sixring drawn by `SIX,B`. Then a movement in the positive $x$–direction is done by `ADJ1`. At last a rotated fivering is drawn: `FIVE,ROT3,B`.

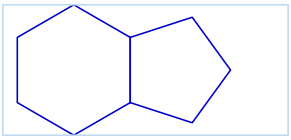


**Figure 5.2**

**Example 27**

```
\startchemical[frame=on,width=4000,right=3000]
  \chemical[SIX,B,ADJ1,FIVE,ROT3,B]
\stopchemical
```

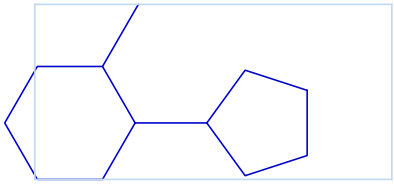


**Figure 5.3**

**Example 28**

```
\startchemical[frame=on,width=5000,right=4500]
  \chemical[SIX,ROT2,B,R6,SUB1,FIVE,B,R4]
\stopchemical
```

To go from one structure to an adjacent one is done with `ADJ`. Most of the time one of these structures will have to be rotated to obtain a good attachment. This is done by `ROT`. Rotations are always clockwise in steps of 90.°C When a structure is attached with a bond you will have to use `SUB`. Movements with `ADJ` and `SUB` take place in the four directions of the $x$- and $y$–axis.

The next examples illustrate that the dimensions of the smaller structures are determined by the larger structures, especially `SIX`. You will notice that `EIGHT` has fewer possibilities than `SIX`.

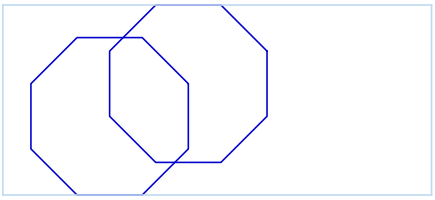


**Figure 5.4**

**Example 29**

```
\startchemical[width=6000,left=1500,frame=on]
  \chemical[EIGHT,B,MOV1,B]
\stopchemical
```

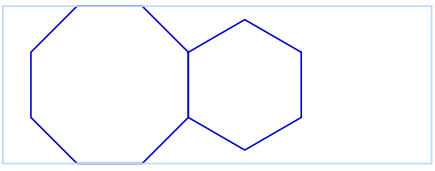


**Figure 5.5**

**Example 30**

```
\startchemical[width=6000,left=1500,frame=on]
  \chemical[EIGHT,B,ADJ1,SIX,B]
\stopchemical
```

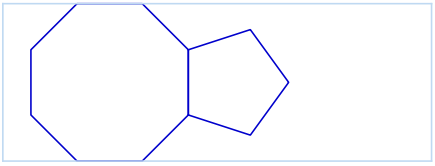


**Figure 5.6**

**Example 31**

```
\startchemical[width=6000,left=1500,frame=on]
  \chemical[EIGHT,B,ADJ1,FIVE,ROT3,B]
\stopchemical
```

It will be clear by now that the order in which the keys are defined makes a lot of difference. The order should be:

```
\chemical
  [structure,                              % SIX, FIVE, ...
   bonds within the structure,             % B, C, EB, ...
   bonds pointing to substituents,         % R, DR, ...
   atoms within the structure,             % Z
   subsituents attached to the structure]  % RZ, -RZ, ...
  [atoms,
   substituents]
```

Most of the time putting structures together is done by translating and rotating. You could automate this process. In earlier versions this was done automatically, however this led to misinterpretations of users concerning the positions of bonds, atoms and substituents. A structure that consists of more than one component can best be defined per component, translations included. Rotations should wait until the last step.

A sixring may have substituents consisting of a carbon chain. In those situations we use `DIR` to build the chain.

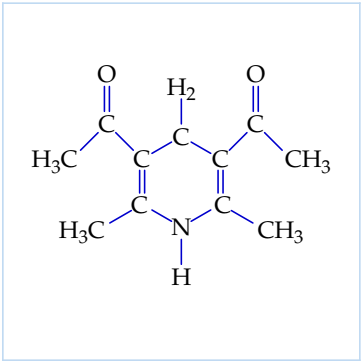



**Figure 5.7**

**Example 32**

```
\startchemical
  [scale=small,width=6000,height=6000,frame=on]
  \chemical[SIX,SB2356,DB14,Z2346,SR36,RZ36]
    [C,N,C,C,H,H_2]
  \chemical[PB:Z1,ONE,Z0,MOV8,Z0,SB24,DB7,Z27,PE]
    [C,C,CH_3,O]
  \chemical[PB:Z5,ONE,Z0,MOV6,Z0,SB24,DB7,Z47,PE]
    [C,C,H_3C,O]
  \chemical[SR24,RZ24]
    [CH_3,H_3C]
\stopchemical
```

Because chains have no predefined format the chains are build and positioned as a substructure. For positioning we use the keys `PB` and `PE`.

| | | |
|---|---|---|
| PB:.. | Picture Begin | beginning a substructure |
| PE | Picture End | ending a substructure |

**Table 5.4** Positioning.

Directly after PB you will have to define the location where the substructure is positioned. The first following atom is centered on that location. Always use a central atom on this location.

These keys were introduced after trying to obtain structures that are typeset in an acceptable quality. There are different ways to define structures. The following alternative would have resulted in:

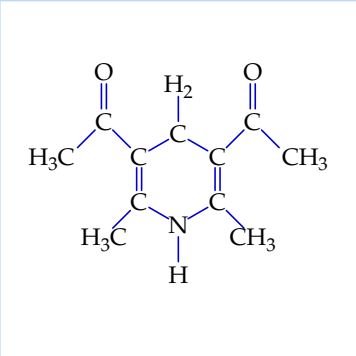



**Figure 5.8**

**Example 33**

```
\startchemical
    [scale=small,width=6000,height=6000,frame=on]
  \chemical
    [SIX,SB2356,DB14,Z36,SR36,RZ36][N,C,H,H_2]
  \chemical
    [PB:Z1,ONE,Z0,MOV8,Z0,SB24,DB7,Z27,PE][C,C,CH_3,O]
  \chemical
    [PB:Z5,ONE,Z0,MOV6,Z0,SB24,DB7,Z47,PE][C,C,H_3C,O]
  \chemical
    [PB:Z2,ONE,Z0,MOV2,SB6,CZ0,PE][C,CH_3]
  \chemical
    [PB:Z4,ONE,Z0,MOV4,SB8,CZ0,PE][C,H_3C]
\stopchemical
```

You may have noticed that the measurements of the structure is determined by the substituents. The chains are not taken into account. This leads to a consistent build-up of a structure.

The differences in outcome when using SUB in stead of PB are very small. However compare the following formulas.

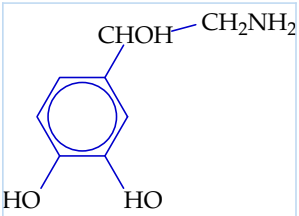



**Figure 5.9**

**Example 34**

```
\startchemical
  [width=fit,frame=on,scale=small]
  \chemical
    [SIX,ROT2,B,C,+R236,RZ23,
     SUB6,ONE,OFF1,Z0,4OFF1,SB1,Z1]
    [HO,HO,CHOH,CH_2NH_2]
\stopchemical
```

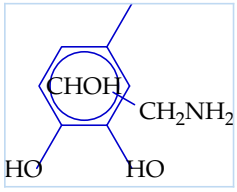



Figure 5.10

**Example 35**

```
\startchemical
  [width=fit,frame=on,scale=small]
  \chemical
    [SIX,ROT2,B,C,R236,RZ23,
     PB:RZ6,ONE,Z0,3OFF1,SB1,Z1,PE]
    [HO,HO,CHOH,CH_2NH_2]
\stopchemical
```

We try to predict rotations and shifts as good as possible but it might take some trial and error, but we assume that at some poitn one has (and shares) definitions.

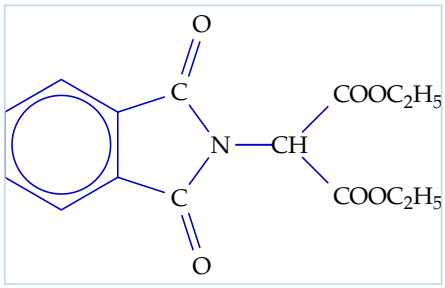



Figure 5.11

**Example 36**

```
\startchemical
  [width=fit,frame=on]
  \chemical
    [SIX,B,C,
     ADJ1,FIVE,ROT3,SB34,+SB2,-SB5,Z345,DR35,SR4,RZ35,
     SUB4,ONE,SB258,Z0,Z28]
    [C,N,C,O,O,
     CH,COOC_2H_5,COOC_2H_5]
\stopchemical
```

We will show another example, produced in two ways. When choosing a method you should take into account the consistency throughout your document.

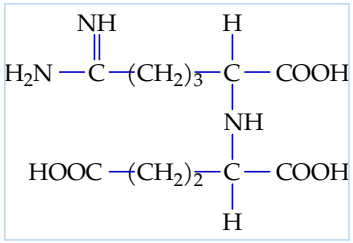



Figure 5.12

**Example 37**

```
\startchemical
  [width=fit,height=fit,frame=on,
   scale=small]
  \chemical
    [ONE,SB15,DB7,Z057,3OFF1,MOV1,Z0,3OFF1,MOV1,
     Z017,SB1357,MOV3,Z0,MOV3,SB1357,Z013,3OFF5,
     MOV5,Z0,3OFF5,SB5,Z5]
    [C,H_2N,NH,(CH_2)_3,C,COOH,H,\SL{NH},C,COOH,H,
     (CH_2)_2,HOOC]
\stopchemical
```

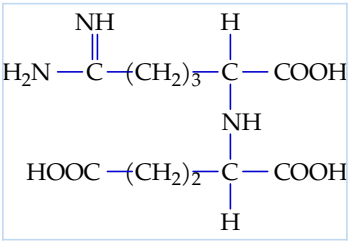


**Figure 5.13**

**Example 38**

```
\startchemical
  [width=fit,height=fit,frame=on,
   scale=small]
  \chemical [ONE,SB15,DB7,Z057,3OFF1][C,H_2N,NH]
  \chemical [MOV1,Z0,3OFF1]          [(CH_2)_3]
  \chemical [MOV1,Z017,SB1357]       [C,COOH,H]
  \chemical [MOV3,Z0]                [\SL{NH}]
  \chemical [MOV3,SB1357,Z013,3OFF5] [C,COOH,H]
  \chemical [MOV5,Z0,3OFF5,SB5,Z5]   [(CH_2)_2,HOOC]
\stopchemical
```

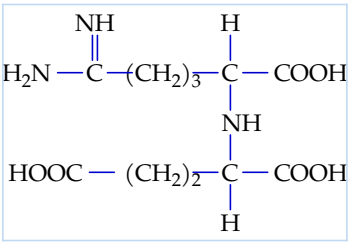


**Figure 5.14**

**Example 39**

```
\startchemical
  [width=fit,height=fit,frame=on,
   scale=small]
  \chemical
    [ONE,Z0,SAVE,MOV7,SB1357,Z017,3OFF5,MOV5,Z0,
     3OFF5,MOV5,SB15,DB7,Z057,RESTORE,
     MOV3,SB1357,Z013,MOV5,3OFF5,Z0,6OFF5,SB5,Z5]
    [\SL{NH},C,COOH,H,(CH_2)_3,C,H_2N,NH,C,COOH,H,
     (CH_2)_2,HOOC]
\stopchemical
```

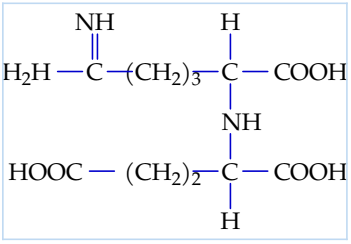


**Figure 5.15**

**Example 40**

```
\startchemical
  [width=fit,height=fit,frame=on,scale=small]
  \chemical
    [ONE,Z0,MOV7,SB1357,Z017,3OFF5,MOV5,Z0,
     3OFF5,MOV5,SB15,DB7,Z057,MOV0,MOV3,SB1357,
     Z013,MOV5,3OFF5,Z0,6OFF5,SB5,Z5]
    [\SL{NH},C,COOH,H,(CH_2)_3,C,H_2H,NH,C,COOH,H,
     (CH_2)_2,HOOC]
\stopchemical
```

Notice the use of `SAVE` and `RESTORE`. These keys enable you to save a location in a structure and return to that location in another stage. As an extra we will show you a combination of `SIX` and `FIVE`.

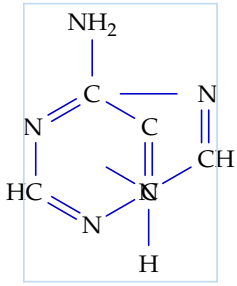



**Figure 5.16**

**Example 41**

```
\startchemical
    [width=fit,height=fit,frame=on]
  \chemical
    [SIX,DB135,SB246,Z,SR6,RZ6][C,C,N,\SR{HC},N,C,NH_2]
  \chemical
    [SIX,MOV1,DB1,SB23,SS6,Z1..3,SR3,RZ3][N,\SL{CH},N,H]
\stopchemical
```

# 6 Extra text

We can add text and symbols in and around structures. For example:

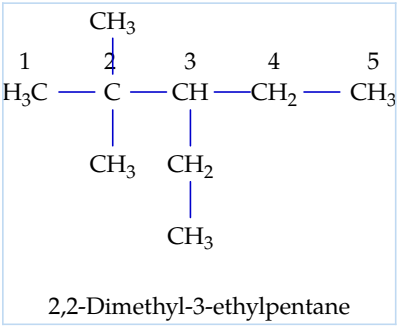



**Figure 6.1**

**Example 42**

```
\startchemical
  [height=4500,top=1250,width=fit,frame=on]
  \bottext
    {2,2-Dimethyl-3-ethylpentane}
  \chemical
    [ONE,Z3570,SB1357]
    [CH_3,\T{1}{H_3C},CH_3,\SR{\LT{2}{C}}]
  \chemical
    [MOV1,OFF1,Z0,SB3]
    [\T{3}{CH}]
  \chemical
    [MOV3,Z0,SB3,MOV3,Z0,MOV7,MOV7]
    [CH_2,CH_3]
  \chemical
    [OFF1,SB1,MOV1,OFF1,Z0,2OFF1,SB1,Z1]
    [\T{4}{CH_2},\T{5}{CH_3}]
\stopchemical
```

There is a range of keys like \T. In a number of cases the arguments are optional. Charges can be displayed in Roman by means of \+ and \- or directly by means of \1 up to \7.

| | |
|---|---|
| \+{number} | positive charge in Roman |
| \-{number} | negative charge in Roman |
| \1 | I (without sign) |
| \7 | II, III, IV, V, VI and VII |

**Table 6.1** Text: charges.

A charge is centered above the atom. For example:

```
\placeformula
  \startformula
      \chemical{S}
      \chemical{+}
      \chemical{O_2}
      \chemical{GIVES}{violent}
      \chemical{\+{4}{S}\-{2}{O_2}}
  \stopformula
```

will result in:

$$\mathrm{S} \quad + \quad \mathrm{O_2} \quad \longrightarrow \quad \mathit{violent}\overset{+\mathrm{IV}\text{-}\mathrm{II}}{\mathrm{SO_2}} \tag{6.1}$$

Compare this with:

```
\placeformula
  \startformula
    \startchemicalformula
      \chemical{S}
      \chemical{+}
      \chemical{O_2}
      \chemical{GIVES}{violent}
      \chemical{\+{4}{S}\-{2}{O_2}}
    \stopchemicalformula
  \stopformula
```

which gives:

$$\mathrm{S} \quad + \quad \mathrm{O_2} \quad \xrightarrow[\text{violent}]{} \quad \overset{+\mathrm{IV}\text{-}\mathrm{II}}{\mathrm{SO_2}} \tag{6.2}$$

# 7 Axis

Structures can be typeset in a frame that is divided by axis. The dimensions of the axis and the location of the origin can be defined in the set up. The axis and the frame can be made visible.



**Figure 7.1**

**Example 43**

```
\startchemical
  [axis=on,
   width=6000,height=4000]
\stopchemical
```



**Figure 7.2**

**Example 44**

```
\startchemical
  [axis=on,
   frame=on,
   width=6000,height=4000]
\stopchemical
```

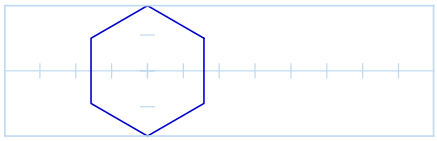


**Figure 7.3**

**Example 45**

```
\startchemical
  [axis=on,
   frame=on,
   left=2000,right=4000]
   \chemical[SIX,B]
\stopchemical
```



**Figure 7.4**

**Example 46**

```
\startchemical
  [axis=on,
   frame=on,
   width=6000,top=1000,bottom=3000]
\stopchemical
```

The dimensions of the total structure determine the dimensions of the axis. When `width=fit` and/or `height=fit` is typed the dimensions are determined by the real dimensions so there need to be content. Your choice will depend on how you want to place the structure in the text. As shown here, the axis model currently uses rather large numbers. This is an inheritance of the previous implementation and we deciced to keep it this way.

# 8 Setups

After \startchemical and \setupchemical you can type the setup.

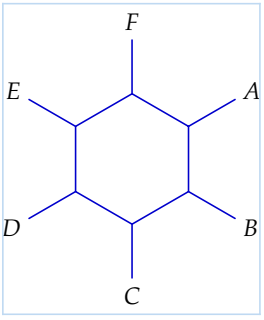
| parameter | values | default |
|---|---|---|
| width | number | 4000 |
| height | number | 4000 |
| left | number | |
| right | number | |
| top | number | |
| bottom | number | |
| bodyfont | 8pt 9pt 10pt etc. | \bodyfontsize |
| textstyle | \rm \bf etc. | \rm |
| style | \rm \bf etc. | \rm |
| color | colorname | |
| rulecolor | colorname | |
| scale | number | medium |
| size | small medium big | medium |
| axis | on off | off |
| frame | on off | off |

**Table 8.1** Setups for structures.

The axis range from −2000 upto +2000, height as well as width. The parameter Z0 is at (0, 0). Other divisions can be set up with left, right, top and/or bottom in combination with width and height.

You can use the key size to setup the current text size and with scale you setup the dimensions of the structure itself. The scale is also determined by the parameter bodyfont. The values small, medium and big are proportionally related.

Contrary to the old implementation we no longer operate in math mode as it makes not much sense. The mathematical symbols are available in text mode anyway and control over super and subscripts in implemented differently. With the parameter style an alternative style can be chosen. In **example 47** the substituents are typeset *slanted*.



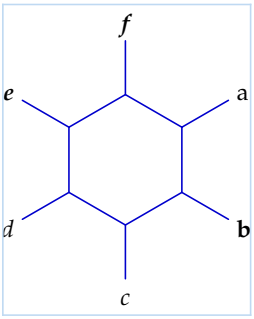

**Example 47**

```
\startchemical[frame=on,style=\sl]
  \chemical[SIX,B,+R,RZ][A,B,C,D,E,F]
\stopchemical
```

**Figure 8.1**

The `style` option is valid for chemical structures in a picture and in the text. The sub- and superscripts are changed accordingly. This is illustrated in $CH_4$, $\mathbf{CH_4}$ and $\mathit{CH_4}$, in which the setups are `\rm`, `\bf` and `\sl`. Italic `\it` formulas lead to a bigger linewidth. The commands adjust automatically to the actual fontstyle: $\mathit{CH_4}$, $\mathit{CH_4}$, $\mathit{CH_4}$ etc. (`\ss`, `\rm`, `\tt`).

It is also possible to set the style at the instant you type them in the argument.[3]

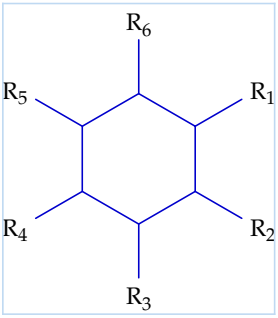



**Example 48**

```
\startchemical[frame=on]
  \chemical[SIX,B,+R,RZ][\tf a,\bf b,\it c,\sl d,\bi e,\bs
f]
\stopchemical
```

**Figure 8.2**

The parameter `frame` and `axis` need no further explanation.



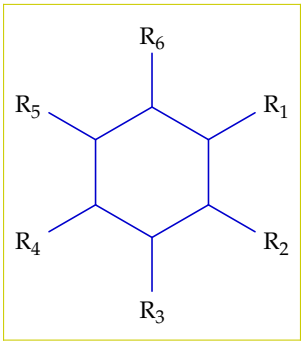

**Example 49**

```
\startchemical[frame=on]
  \chemical[SIX,B,+R,RZ][R_1,R_2,R_3,R_4,R_5,R_6]
\stopchemical
```

**Figure 8.3**

A more controled framing is possible by setting up the encompassing frame:

[3] Not yet documented but already available is support for the rendering prefix `mystyle->a` and alike. This permits a more general setup of styles.

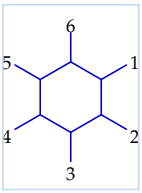



**Example 50**

```
\setupchemicalframed[frame=on,offset=1ex]
\startchemical
  \chemical[SIX,B,+R,RZ][R_1,R_2,R_3,R_4,R_5,R_6]
\stopchemical
```

**Figure 8.4**

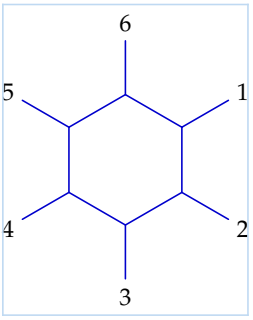
A structure can be displayed in different sizes. This is done with `size` and `scale`.




**Example 51**

```
\startchemical[frame=on,scale=small,size=small]
  \chemical[SIX,B,+R,RZ][1,2,3,4,5,6]
\stopchemical
```

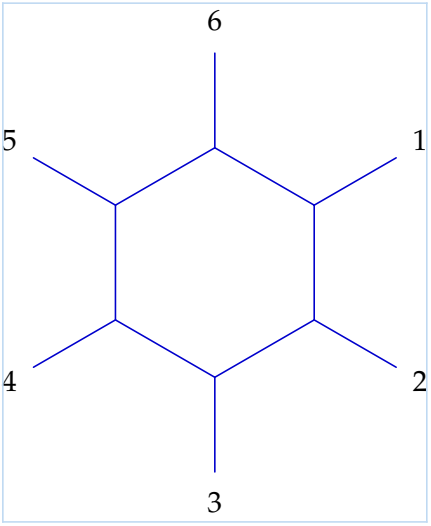
**Figure 8.5**




**Example 52**

```
\startchemical[frame=on,scale=medium,size=medium]
  \chemical[SIX,B,+R,RZ][1,2,3,4,5,6]
\stopchemical
```

**Figure 8.6**




**Example 53**

```
\startchemical[frame=on,scale=big,size=big]
  \chemical[SIX,B,+R,RZ][1,2,3,4,5,6]
\stopchemical
```

**Figure 8.7**

The values belonging to `small`, `medium` or `big` are proportionally related.

# 9 Symbols

There are some symbols that can be used to display reactions. The reaction below is typed by:

```
\setupchemical
  [size=small,
   scale=small,
   width=fit,
   height=5500,
   bottom=1500]

\hbox
  {\startchemical
     \chemical[SIX,B,ER6,RZ6][O]
   \stopchemical
   \startchemical
     \chemical[SPACE,PLUS,SPACE]
   \stopchemical
   \startchemical
     \chemical[FIVE,ROT4,B125,+SB3,-SB4,Z4,SR4,RZ4][N,H]
   \stopchemical
   \startchemical
     \chemical[SPACE,GIVES,SPACE][?]
   \stopchemical
   \startchemical
     \chemical[SIX,B,EB6,R6,SUB4,FIVE,ROT4,B125,+SB3,-SB4,Z4][N]
   \stopchemical
   \startchemical
     \chemical[SPACE,PLUS,SPACE,CHEM][H_2O]
   \stopchemical}
```

The `\hbox` is necessary to align the structures. The symbols `GIVES` and `PLUS` need no further explanation. With `SPACE` more room can be created between the structures and symbols.

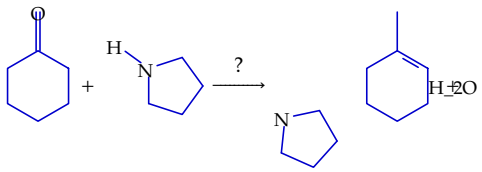



**Figure 9.1**

An equilibrium can be displayed with `EQUILIBRIUM`. Over `GIVES` and `EQUILIBRIUM` you can place text. In the example the text is just a '?'. In addition `MESOMERIC` is also available. Braces used for displaying complexes can be created with `OPENCOMPLEX` and `CLOSECOMPLEX`.

# 10 Positioning

When you are combining atoms or molecules, for example with SUB, some positions and dimensions change their value. To overcome this problem it is possible to save a location with SAVE and return to that location with RESTORE.

| | | |
|---|---|---|
| SAVE | Save Status | save actual status |
| RESTORE | Restore Status | restore actual status |

**Table 10.1** Positioning.

The keys SAVE and RESTORE are used with substituents. When placing radicals we use PB and PE. This example also illustrates the possibility to create chains.

```
\definechemical[molecule]
  {\chemical
     [ONE,Z0,SB1357,
      SAVE,SUB3,SIX,B,+R6,C,RESTORE,
      MOV1,Z0,SB137,
      MOV1,Z0,SB37,
      MOV1]
     [C,C,C]}

\startchemical[width=fit,height=fit]
  \chemical[molecule,molecule,molecule]
\stopchemical
```

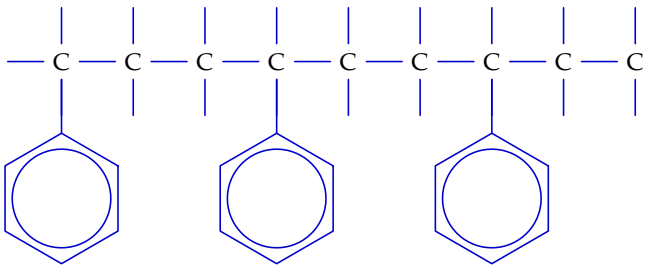



**Figure 10.1**

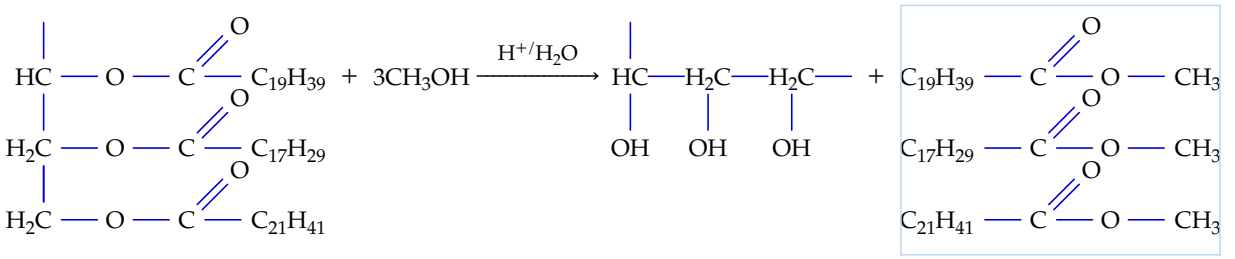
The example below is more complicated and show a complete reaction. The set up of bottom and top is essential in this example.

```
\placeformula
  \startformula
    \setupchemical
      [width=fit,top=1000,bottom=2500,
]%       scale=small,size=small]
    \startchemical
```

```
  \chemical
    [ONE,
     SAVE,
        Z0,SB7,SB3,SB1,MOV1,Z0,SB1,MOV1,Z0,DB8,CZ8,SB1,Z1,
     RESTORE,
     SUB3,ONE,
     SAVE,
        Z0,SB3,SB1,MOV1,Z0,SB1,MOV1,Z0,DB8,CZ8,SB1,Z1,
     RESTORE,
     SUB3,ONE,
        Z0,SB7,SB1,MOV1,Z0,SB1,MOV1,Z0,DB8,CZ8,SB1,Z1
    ]
    [\SR{HC},O,C,O,C_{19}H_{39},
     \SR{H_{2}C},O,C,O,C_{17}H_{29},
     \SR{H_{2}C},O,C,O,C_{21}H_{41}]
\stopchemical
\startchemical
  \chemical[SPACE,PLUS,SPACE]
\stopchemical
\startchemical[right=600]
  \chemical[ONE,CZ0][3CH_{3}OH]
\stopchemical
\startchemical
  \chemical[SPACE,GIVES,SPACE,SPACE][H^+/H_2O]
\stopchemical
\startchemical
  \chemical
    [ONE,Z0,SB7,SB3,SB1,Z3,OFF1,
     SUB1,
     ONE,Z0,OFF1,SAVE,-OFF1,SB3,Z3,RESTORE,SB1,OFF1,
     SUB1,
     ONE,Z0,OFF1,SAVE,-OFF1,SB3,Z3,RESTORE,SB1,OFF1]
    [HC,OH,
     H_{2}C,OH,
     H_{2}C,OH]
\stopchemical
\startchemical
  \chemical[SPACE,PLUS,SPACE]
\stopchemical
\startchemical[frame=on]
  \chemical
    [ONE,
     SAVE,
        Z0,
        DB8,CZ8,SB1,SB5,Z5,MOV1,Z0,SB1,Z1,
     RESTORE,
```

```
        SUB3,ONE,
        SAVE,
           Z0,DB8,CZ8,SB1,SB5,Z5,MOV1,Z0,SB1,Z1,
        RESTORE,
        SUB3,ONE,
           Z0,DB8,CZ8,SB1,SB5,Z5,MOV1,Z0,SB1,Z1
       ]
       [C,O,C_{19}H_{39},O,CH_{3},
        C,O,C_{17}H_{29},O,CH_{3},
        C,O,C_{21}H_{41},O,CH_{3}]
  \stopchemical
\stopformula
```

This definition might have been more compact if we would have typed `SB731` in stead of `SB7,SB3,SB1`. But in this way the definition is readable. Complex structures can best be defined in its respective components.

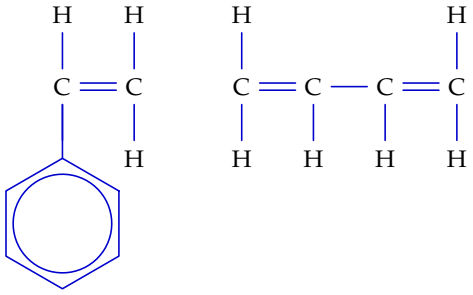


(10.1)

Just two more examples where we place text under a structure.

```
\placeformula
  \startformula
    \setupchemical
      [width=fit,top=1500,bottom=3500]
    \startchemical
      \chemical
        [ONE,Z0,DB1,SB3,SB7,Z7,MOV1,Z0,SB3,SB7,Z3,Z7,
         MOV0,SUB3,SIX,B,+R6,C]
        [C,H,C,H,H]
      \bottext{styreen}
    \stopchemical
    \quad\quad\quad
    \startchemical
      \chemical
        [ONE,Z0,DB1,SB3,SB7,Z3,Z7,
         MOV1,Z0,SB1,SB3,Z3,
         MOV1,Z0,DB1,SB3,Z3,
         MOV1,Z0,SB3,SB7,Z3,Z7]
        [C,H,H,C,H,C,H,C,H,H]
      \bottext{1,3-butadieen}
```

```
\stopchemical
\stopformula
```



styreen 1,3-butadieen (10.2)

The use of OFF can be very subtle. The examples below illustrate this and show minor shifts of ONE.

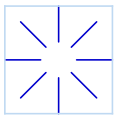


Figure 10.2

**Example 54**

```
\startchemical[height=fit,frame=on]
  \chemical[ONE,SB]
\stopchemical
```

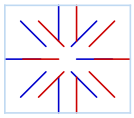


Figure 10.3

**Example 55**

```
\startchemical[height=fit,frame=on]
  \chemical[ONE,SB,3OFF1,rulecolor,SB][red]
\stopchemical
```

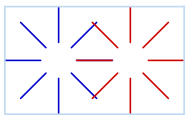


Figure 10.4

**Example 56**

```
\startchemical[height=fit,frame=on]
  \chemical[ONE,SB,MOV1,rulecolor,SB][red]
\stopchemical
```

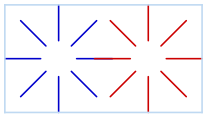


Figure 10.5

**Example 57**

```
\startchemical[height=fit,frame=on]
  \chemical[ONE,SB,3OFF1,MOV1,rulecolor,SB][red]
\stopchemical
```

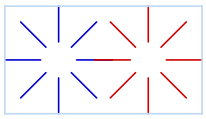


Figure 10.6

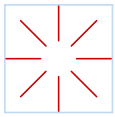
**Example 58**

```
\startchemical[height=fit,frame=on]
  \chemical[ONE,SB,MOV1,3OFF1,rulecolor,SB][red]
\stopchemical
```

**Example 59**

```
\startchemical[height=fit,frame=on]
  \chemical[ONE,MOV1,3OFF1,OFF0,rulecolor,SB][red]
\stopchemical
```

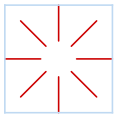
**Figure 10.7**



**Example 60**

```
\startchemical[height=fit,frame=on]
  \chemical[ONE,MOV1,3OFF1,MOV0,rulecolor,SB][red]
\stopchemical
```

**Figure 10.8**



**Example 61**

```
\startchemical[height=fit,frame=on]
  \chemical[ONE,MOV1,MOV0,rulecolor,SB][red]
\stopchemical
```

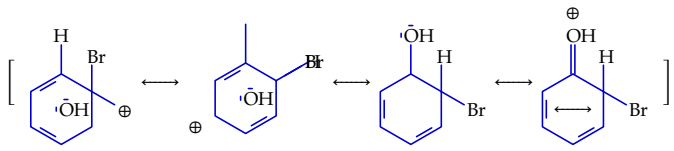
**Figure 10.9**

The next example shows the definition of complexes. Pay special attention to the use of RBT. Normally an extra spacing is not necessary but we use here —the command is not visible— a smaller bodyfont to prevent the structure to run in the margin.

```
\startformula
\setupchemical[scale=small,width=fit]
\startchemical
  \chemical[OPENCOMPLEX,SPACE]
\stopchemical
\startchemical
  \chemical[SIX,B,EB35,+R6,+LR1,+RR1]
  \chemical[SIX,ONE,SAVE,OFF1,Z0,EP57,RESTORE][\SL{OH}]
  \chemical[SIX,RZ6,LRZ1,RRZ1,RT2][H,Br,\oplus]
\stopchemical
\startchemical
  \chemical[SPACE,MESOMERIC,SPACE]
\stopchemical
\startchemical
  \chemical[SIX,B,EB25,R6,-R1,+R1]
  \chemical[SIX,PB:RZ6,ONE,OFF1,Z0,EP57,PE][\SL{OH}]
  \chemical[SIX,-RZ1,+RZ1,RT4][H,Br,\oplus]
\stopchemical
\startchemical
  \chemical[SPACE,MESOMERIC,SPACE]
\stopchemical
\startchemical
  \chemical[SIX,B,EB24,+R6,+LR1,+RR1]
```

```
  \chemical[SIX,SAVE,SUB6,ONE,Z0,EP57,RESTORE][\SL{OH}]
  \chemical[SIX,LRZ1,RRZ1,RT6][H,Br,\L{\oplus}] % \L takes upto two arguments
\stopchemical
\startchemical
  \chemical[SPACE,MESOMERIC,SPACE]
\stopchemical
\startchemical
  \chemical[SPACE,MESOMERIC,SPACE]
  \chemical[SIX,B,EB24,+ER6,+LR1,+RR1]
  \chemical[SIX,SAVE,SUB6,ONE,Z0,Z0,ZT7,RESTORE][\SL{OH},\T{\oplus}]
  \chemical[SIX,LRZ1,RRZ1][H,Br]
\stopchemical
\startchemical
  \chemical[SPACE,CLOSECOMPLEX]
\stopchemical
\stopformula
```

Without the use of SPACE the seperate structures would merge. Most of the time the optimization of such a reaction is an iterative proces.



# 11 Reactions

Not only the typesetting of chemical structures is supported but also the typesetting of normal reactions. The command `\chemical` has three other appearances:

```
\chemical{formula}
\chemical{formula}{bottom text}
\chemical{formula}{top text}{bottom text}
```

This command adapts itself to text mode. That means that it 'knows' whether it is used in:

- text–mode
- mathematical text–mode
- mathematical display–mode

When the command is used in running text it will automatically be surrounded by `$ $`. Typing `\chemical{NH_4^+}` will result in $NH_4^+$.

The result would be the same if we would place the command between `$ $`. In both cases the second and third argument can be left out. If we place the command between `$$ $$` (or `\startformula` and `\stopformula`) both arguments do have a function. First a simple example. The command `\placeformula` is a ConTEXt command and handles the positioning and numbering of the formula.

```
\placeformula
  \startformula
    \chemical{2H_2} \chemical{PLUS} \chemical{O_2}
    \chemical{GIVES} \chemical{2H_2O}
  \stopformula
```

$$2H_2 \quad + \quad O_2 \quad \longrightarrow \quad 2H_2O \tag{11.1}$$

The definition of the chemical part could be somewhat shorter:

```
\chemical{2H_2,PLUS,O_2,GIVES,2H_2O}
```

or even:

```
\chemical{2H_2,+,O_2,->,2H_2O}
```

A TEX–addict will notice from these examples that the plus sign and the arrow are on the baseline. Compare for example + and $+$. In the reaction you will see that the $+$ and the $\rightarrow$ are vertically aligned.

You can use `PLUS`, `GIVES` and `EQUILIBRIUM` (`<->`) in this command. With `MESOMERIC` or `<>` you will get $\leftrightarrow$.

The reaction can be placed in the text. In that case a more compact display is used: $2H_2 + O_2 \rightarrow 2H_2O$. Some finetuning with `\,` would result in 2 H_2 .

It is also possible to display bonds in textmode. For example if you want H—CH=HC—H you should type `\chemical{H,SINGLE,CH,DOUBLE,HC,SINGLE,H}` or something like this `\chemical{H,-,CH,--,HC,-,H}`. A triple bond can be defined as `TRIPLE` or `---`: HC≡CH.

We return two the display–mode. The second and third argument can be used to add text to the reaction:

```
\placeformula
  \startformula
    \chemical{2H_2}{hydrogen} \chemical{PLUS} \chemical{O_2}{oxygen}
    \chemical{GIVES}{heat} \chemical{2H_2O}{water}
   \stopformula
```

So we can also place text over and under symbols!

$$2H_2 \quad + \quad O_2 \underset{oxygen}{} \quad \longrightarrow \quad \overset{heat}{} 2H_2O \underset{water}{} \tag{11.2}$$

The last argument is placed under the compound.

$$\underset{\text{liquid}}{H_2O} \quad \text{c.q.} H_2O \underset{water}{} \tag{11.3}$$

The formula above is defined with:

```
\placeformula
  \startformula
    \chemical{H_2O}{liquid}{water}
    \hbox{c.q.}
    \chemical{H_2O}{water}
  \stopformula
```

The size of the formulas or reactions in the running text can be set up with:

| parameter | set up | default |
|---|---|---|
| size | small medium big | big |

**Table 11.1** Set up in text formulas.

The definition `\chemical{H,SINGLE,CH,DOUBLE,HC,SINGLE,H}` result with `big`, `medium` and `small` in the following formulas:

H—CH=HC—H H—CH=HC—H H—CH=HC—H

# 12 Subscripts

As we no longer typeset the consituents in math mode, we also no longer need to worry about the alignment of super and subscripts. We just use the same mechanism as in running text, using `\low`, `\high` and `\lohi`. These are applied automatically.

# 13 Fonts

The remarks concerning fonts in the old manual no longer apply. In principle you can use any font you want. Of course you can also use any character in the font as no encoding is involved.

# 14 Color

In ConTEXt you can colorize parts of a structure. In example ?? and **63** we see a directive being used for the rules but for text we just use TEX commands.

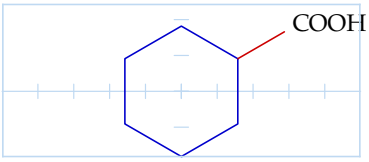



**Figure 14.1**

**Example 62**

```
\startchemical[axis=on,frame=on,width=5000]
    \chemical[SIX,B,rulecolor,+R1,RZ1][red,COOH]
\stopchemical
```

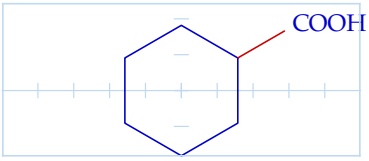



**Figure 14.2**

**Example 63**

```
\startchemical[axis=on,frame=on,width=5000]
    \chemical
        [SIX,B,rulecolor,+R1,color,RZ1]
        [red,{\blue COOH}]
\stopchemical
```

# 15 Interaction

We do support interactive texts. An interactive text is a text that can be consulted on a computerscreen and contains many hyperlinked textareas. This means that clicking on such an area will result in a jump to the target area.

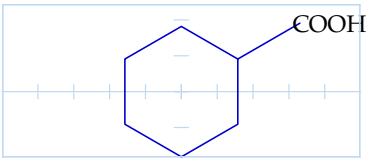



**Figure 15.1**

**Example 64**

```
\startchemical[axis=on,frame=on,width=5000]
  \chemical[SIX,B]
  \chemical[sub:cooh][SIX,R1,RZ1][COOH]
\stopchemical
```

We see a new argument: the reference `[sub:cooh]`. This means that we can refer from the text COOH to the structure with:

```
... text ... \goto{\chemical{COOH}}[sub:cooh] ... text ...
```

In this definition `\goto` is a ConTEXt–command. We can also refer from the structure to a particular part of the text.

Clicking in COOH in the structure results in a jump to the text that is marked with:

```
\paragraph[txt:cooh]{Substituents}

... text ... \chemical{COOH} ... text ...
```

A combination is also possible. In that case it is necessary to mark the reference with `\chemical` and to refer in the text with `\gotochemical`.

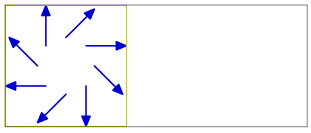
# 16 Summary

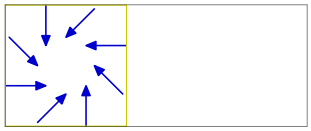
| structure | # | category | keyword | arguments |
|---|---|---|---|---|
| alkyl | 4 | one | ALKYL | |
| boat | 6 | front | BOAT | |
| carbon | 4 | one | CARBON | |
| chair | 6 | front | CHAIR | |
| eight | 8 | ring | EIGHT | |
| five | 5 | ring | FIVE | |
| fivefront | 5 | front | FIVEFRONT | |
| four | 4 | ring | FOUR | |
| newmaneclipsed | 6 | one | NEWMAN,ECLIPSED | |
| newmanstagger | 6 | one | NEWMAN | |
| nine | 9 | ring | NINE | |
| one | 8 | one | ONE | |
| seven | 7 | ring | SEVEN | |
| six | 6 | ring | SIX | |
| sixfront | 6 | front | SIXFRONT | |
| three | 3 | ring | THREE | |

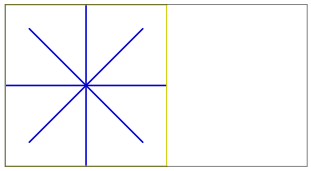
pb
pe
save
restore
reset
mp

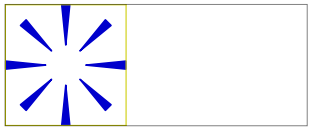
# 17 One

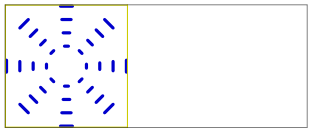
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ad | line | dir | transform | move | transform | rt | text |
| adj | transform | dr | line | number | number | rz | text |
| arrow | line | ed | line | oe | line | sb | line |
| au | line | ep | line | off | transform | sd | line |
| b | line | es | line | r | line | sr | line |
| bb | line | et | line | rb | line | sub | transform |
| bd | line | hb | line | rbd | line | tb | line |
| bw | line | label | text | rd | line | z | text |
| c | fixed | ldb | line | rdb | line | zln | number |
| cc | line | ldd | line | rdd | line | zlt | text |
| ccd | line | line | line | rh | line | zn | number |
| cd | fixed | lrn | number | rm | transform | zrn | number |
| cz | text | lrt | text | rn | number | zrt | text |
| dash | line | mark | transform | rot | transform | zt | text |
| db | line | mir | transform | rrn | number | | |
| diff | transform | mov | transform | rrt | text | | |

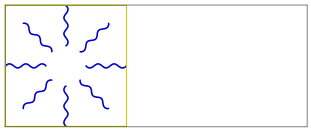
AD  [ONE,AD]

AU  [ONE,AU]

B  [ONE,B]

BB  [ONE,BB]

BD  [ONE,BD]

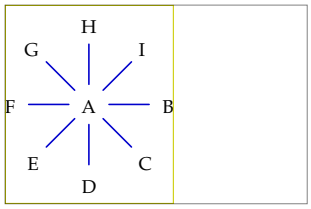
BW  [ONE,BW]

C  [ONE,C]

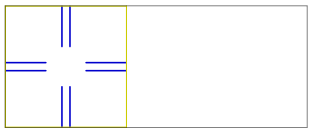
CC  [ONE,CC32]

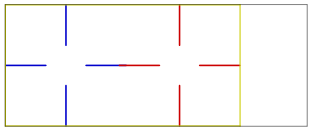
CCD  [ONE,CCD12]

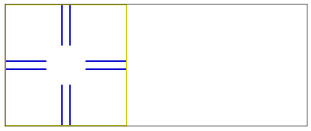
CD  [ONE,CD]

CZ 

[ONE,SB1357,SB2468,Z0,CZ] [A,B,C,D,E,F,G,H,I]

DB  [ONE,DB1357]

DIR  [ONE,SB1357,DIR1,rulecolor,SB1357] [red]

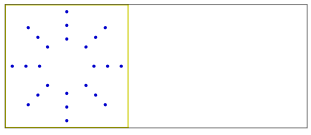
DR  [ONE,DR1357]

ED  [ONE,ED1357]

EP  [ONE,EP1357]

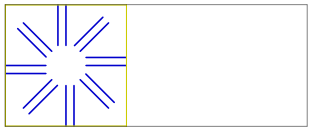
ES  [ONE,ES1357]

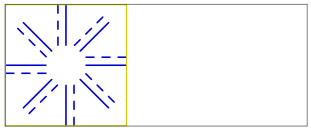
ET  [ONE,ET1357]

HB  [ONE,HB]

LABEL 

[ONE,C,LABEL] [A]

LDB 
[ONE,LDB]

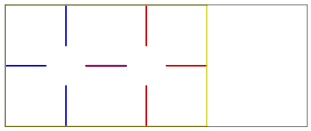
LDD 
[ONE,LDD]

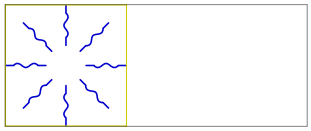
LRN 

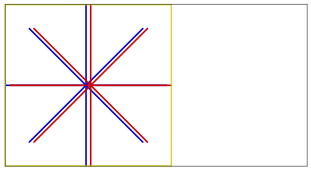
[ONE,LRN]

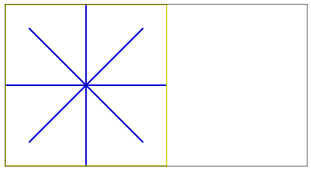
LRT 

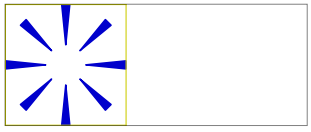
[ONE,LRT] [A,B,C,D,E,F,G,H,I]

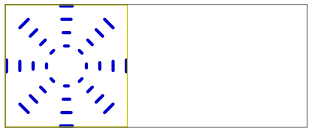
MOV 
[ONE,SB1357,MOV1,rulecolor,SB1357] [red]

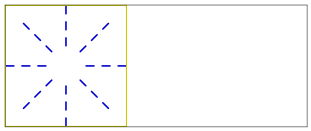
OE 
[ONE,OE]

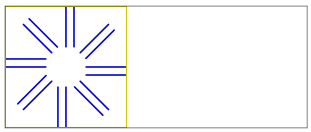
OFF 
[ONE,B,OFF,rulecolor,ONE,B] [red]

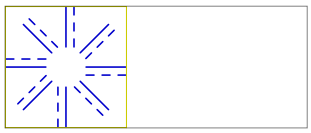
R 
[ONE,R]

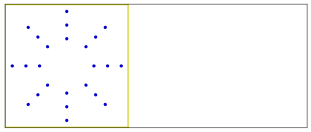
RB 
[ONE,RB]

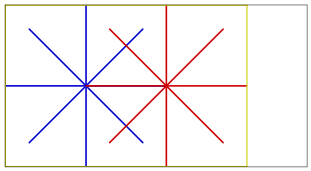
RBD 
[ONE,RBD]

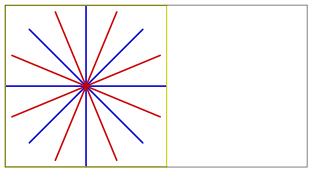
RD  [ONE,RD]

RDB  [ONE,RDB]

RDD  [ONE,RDD]

RH  [ONE,RH]

RM  [ONE,B,RM,rulecolor,ONE,B] [red]

ROT  [ONE,B,ROT,rulecolor,ONE,B] [red]

RRN 

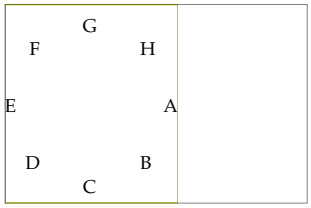
[ONE,RRN]

RRT 

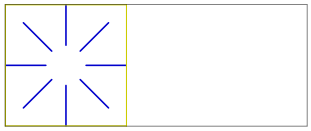
[ONE,RRT] [A,B,C,D,E,F,G,H,I]

RT 

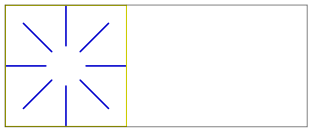
[ONE,RT] [A,B,C,D,E,F,G,H,I]

RZ 

[ONE,RZ] [A,B,C,D,E,F,G,H,I]

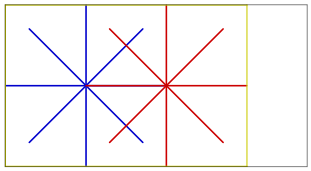
SB  [ONE,SB]

SR



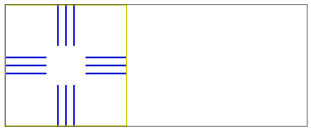
[ONE,SR]

SUB



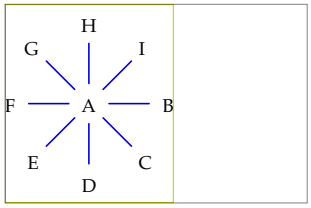
[ONE,B,SUB,rulecolor,ONE,B] [red]

TB



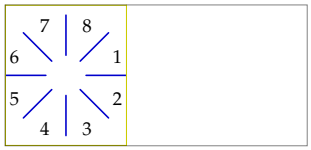
[ONE,TB1357]

Z



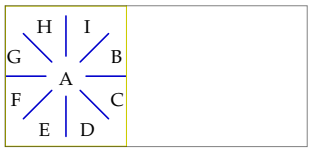

[ONE,SB1357,SB2468,Z0,Z] [A,B,C,D,E,F,G,H,I]

ZLN



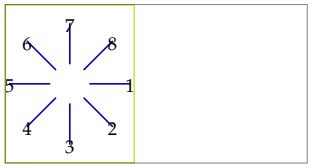

[ONE,SB,ZLN]

ZLT



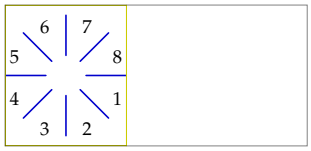

[ONE,SB1357,SB2468,Z0,ZLT] [A,B,C,D,E,F,G,H,I]

ZN



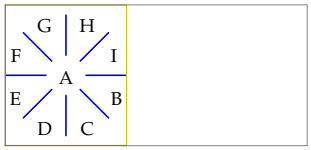

[ONE,SB,ZN]

ZRN



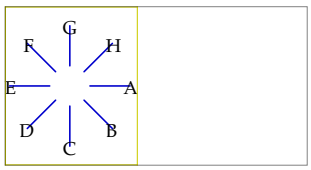

[ONE,SB,ZRN]

ZRT




[ONE,SB1357,SB2468,Z0,ZRT] [A,B,C,D,E,F,G,H,I]

ZT

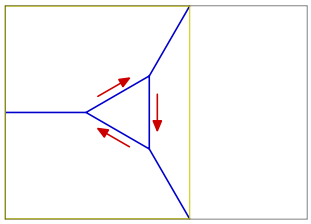



[ONE,SB,ZT] [A,B,C,D,E,F,G,H]

# 18 Three

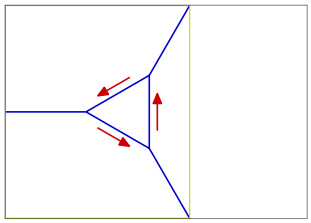
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ad | line | label | text | move | transform | rrz | text |
| adj | transform | line | line | number | number | rsr | line |
| arrow | line | lr | line | off | transform | rsub | transform |
| au | line | lrb | line | r | line | rt | text |
| b | line | lrbd | line | rb | line | rz | text |
| c | fixed | lrd | line | rbd | line | s | line |
| cc | line | lrh | line | rd | line | sb | line |
| ccd | line | lrn | number | rh | line | sr | line |
| cd | fixed | lrt | text | rm | transform | ss | line |
| crz | text | lrz | text | rn | number | sub | transform |
| dash | line | lsr | line | rot | transform | z | text |
| db | line | lsub | transform | rr | line | zln | number |
| diff | transform | mark | transform | rrb | line | zlt | text |
| dr | line | mid | line | rrbd | line | zn | number |
| eb | line | mids | line | rrd | line | zrn | number |
| ed | line | midz | text | rrh | line | zrt | text |
| er | line | mir | transform | rrn | number | zt | text |
| hb | line | mov | transform | rrt | text | | |

AD



```
[THREE,B,R,rulecolor,AD] [red]
```

AU



```
[THREE,B,R,rulecolor,AU] [red]
```

B

```
[THREE,B]
```

C

```
[THREE,B,rulecolor,C] [red]
```

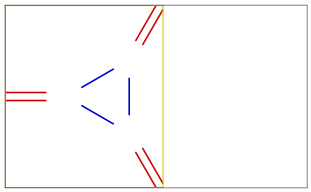
CC

```
[THREE,B,rulecolor,CC1..5] [red]
```

CCD  [THREE,B,rulecolor,CCD1..5] [red]

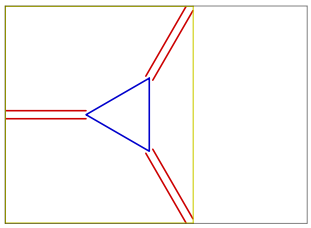
CD  [THREE,B,rulecolor,CD] [red]

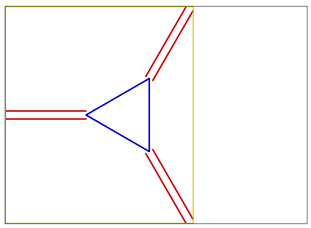
DB  [THREE,DB]

DR  [THREE,SB,rulecolor,DR] [red]

EB  [THREE,B,rulecolor,EB135] [red]

ED  [THREE,B,rulecolor,ED135] [red]

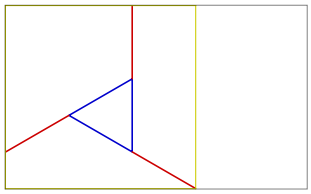
ER  [THREE,B,rulecolor,ER135] [red]

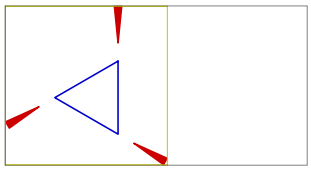
ER  [THREE,B,rulecolor,ER] [red]

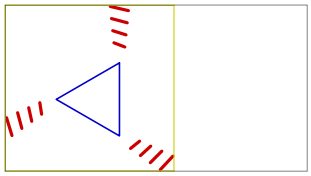
HB  [THREE,HB]

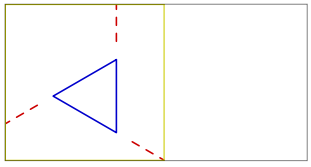
LABEL 

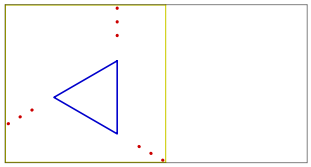
[THREE,B,LABEL] [A]

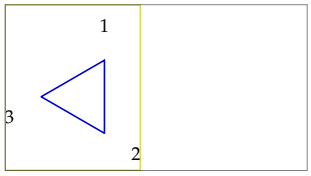
LR  [THREE,B,rulecolor,LR] [red]

LRB  [THREE,B,rulecolor,LRB] [red]

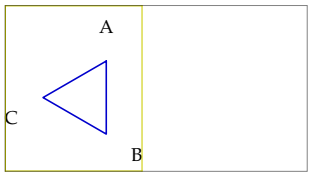
LRBD  [THREE,B,rulecolor,LRBD] [red]

LRD  [THREE,B,rulecolor,LRD] [red]

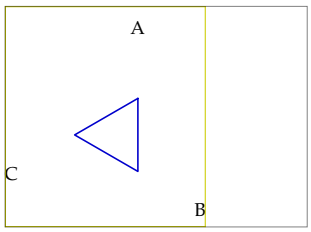
LRH  [THREE,B,rulecolor,LRH] [red]

LRN 

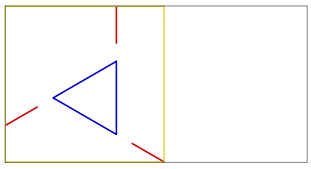
[THREE,B,LRN]

LRT 

[THREE,B,LRT] [A,B,C]

LRZ 

[THREE,B,LRZ] [A,B,C]

LSR  [THREE,B,rulecolor,LSR] [red]

MID  [THREE,B,rulecolor,MID135] [red]

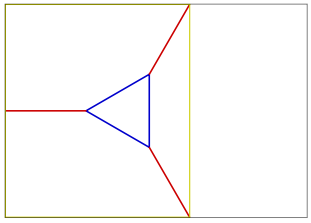
MIDS  [THREE,B,rulecolor,MIDS135] [red]

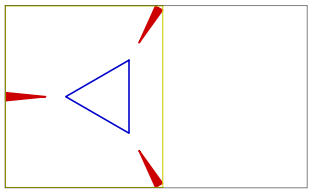
MIDZ 

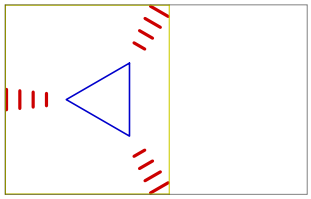
[THREE,B,MIDZ] [A]

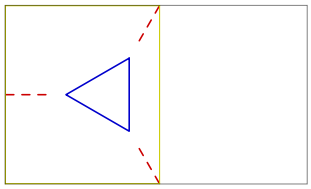
MOV  [THREE,B,MOV,rulecolor,THREE,B] [red]

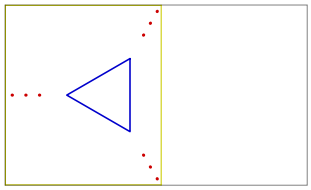
OFF  [THREE,B,OFF,rulecolor,THREE,B] [red]

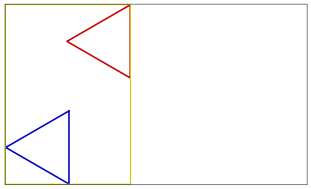
R  [THREE,B,rulecolor,R] [red]

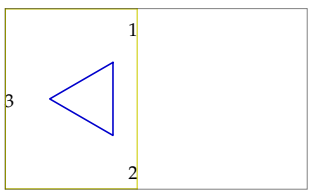
RB  [THREE,B,rulecolor,RB] [red]

RBD  [THREE,B,rulecolor,RBD] [red]

RD  [THREE,B,rulecolor,RD] [red]

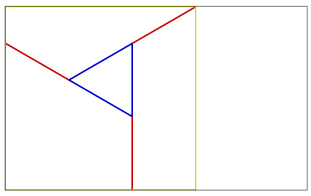
RH  [THREE,B,rulecolor,RH] [red]

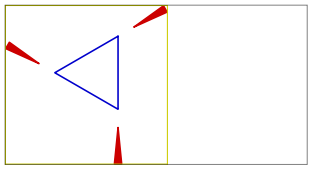
RM  [THREE,B,RM,rulecolor,THREE,B] [red]

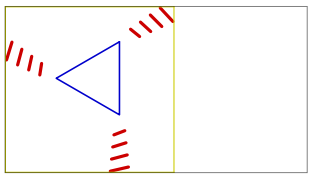
RN 

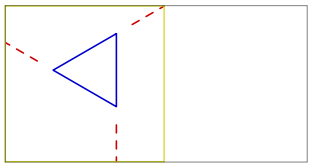
[THREE,B,RN]

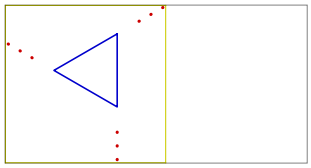
ROT  `[THREE,B,ROT,rulecolor,THREE,B] [red]`

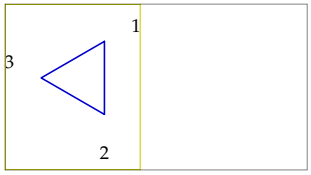
RR  `[THREE,B,rulecolor,RR] [red]`

RRB  `[THREE,B,rulecolor,RRB] [red]`

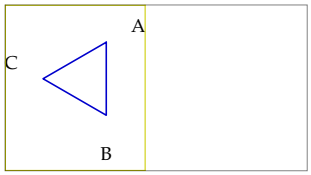
RRBD  `[THREE,B,rulecolor,RRBD] [red]`

RRD  `[THREE,B,rulecolor,RRD] [red]`

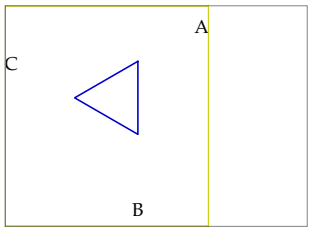
RRH  `[THREE,B,rulecolor,RRH] [red]`

RRN 

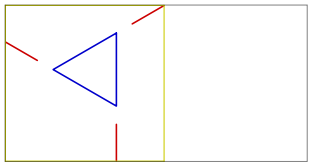
`[THREE,B,RRN]`

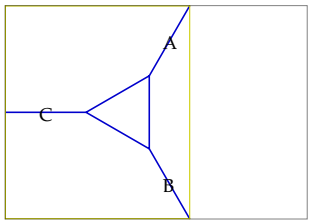
RRT 

`[THREE,B,RRT] [A,B,C]`

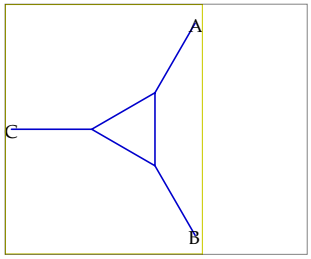
RRZ 

`[THREE,B,RRZ] [A,B,C]`

RSR  [THREE,B,rulecolor,RSR] [red]

RT 

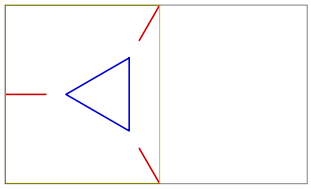
[THREE,B,R,RT] [A,B,C]

RZ 

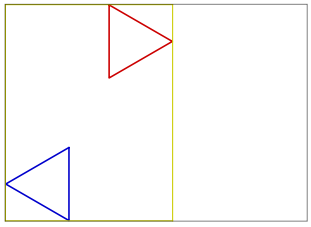
[THREE,B,R,RZ] [A,B,C]

S  [THREE,B,rulecolor,S] [red]

SB  [THREE,SB]

SR  [THREE,B,rulecolor,SR] [red]

SS  [THREE,B,rulecolor,SS135] [red]

SUB  [THREE,B,SUB,rulecolor,THREE,B] [red]

Z 

[THREE,SB,Z] [A,B,C]

ZLN 

[THREE,B,ZLN] [A,B,C]

ZLT 

[THREE,B,ZLT] [A,B,C]

ZN 

[THREE,B,ZN] [A,B,C]

ZRN 

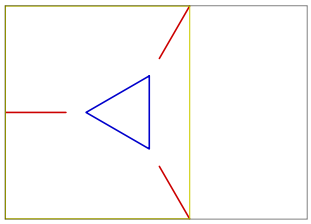
[THREE,B,ZRN] [A,B,C]

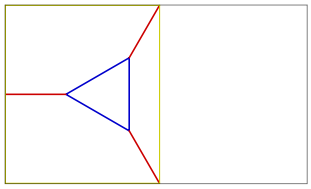
ZRT 

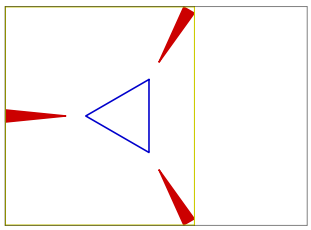
[THREE,B,ZRT] [A,B,C]

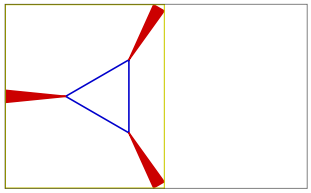
ZT 

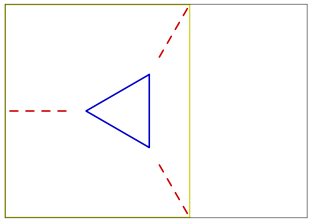
[THREE,B,ZT] [A,B,C]

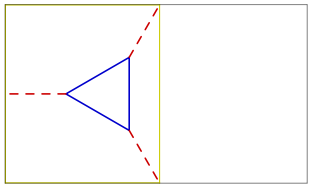
-R 
[THREE,B,rulecolor,-R] [red]

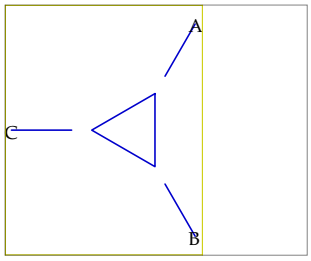
+R 
[THREE,B,rulecolor,+R] [red]

-RB 
[THREE,B,rulecolor,-RB] [red]

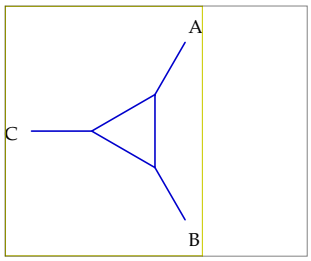
+RB 
[THREE,B,rulecolor,+RB] [red]

-RD 
[THREE,B,rulecolor,-RD] [red]

+RD 
[THREE,B,rulecolor,+RD] [red]

-RZ 

[THREE,B,-SR,-RZ] [A,B,C]

+RZ 

[THREE,B,+SR,+RZ] [A,B,C]

-S 
[THREE,B,rulecolor,-S246] [red]

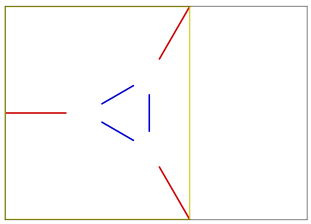
+S 
[THREE,B,rulecolor,+S246] [red]

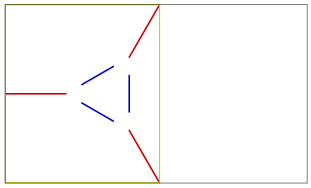
-SB 
[THREE,-SB1,rulecolor,B2..6] [red]

--SB 
[THREE,--SB1,rulecolor,B2..6] [red]

+SB 
[THREE,+SB1,rulecolor,B2..6] [red]

++SB 
[THREE,++SB1,rulecolor,B2..6] [red]

-SR 
[THREE,SB,rulecolor,-SR] [red]

+SR 
[THREE,SB,rulecolor,+SR] [red]

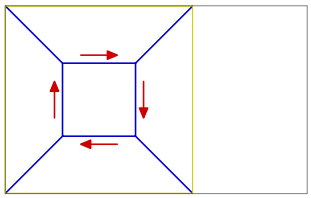
-SS 
[THREE,B,rulecolor,-SS246] [red]

+SS 
[THREE,B,rulecolor,+SS246] [red]

# 19 Four

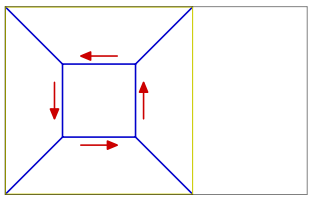
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ad | line | label | text | move | transform | rrz | text |
| adj | transform | line | line | number | number | rsr | line |
| arrow | line | lr | line | off | transform | rsub | transform |
| au | line | lrb | line | r | line | rt | text |
| b | line | lrbd | line | rb | line | rz | text |
| c | fixed | lrd | line | rbd | line | s | line |
| cc | line | lrh | line | rd | line | sb | line |
| ccd | line | lrn | number | rh | line | sr | line |
| cd | fixed | lrt | text | rm | transform | ss | line |
| crz | text | lrz | text | rn | number | sub | transform |
| dash | line | lsr | line | rot | transform | z | text |
| db | line | lsub | transform | rr | line | zln | number |
| diff | transform | mark | transform | rrb | line | zlt | text |
| dr | line | mid | line | rrbd | line | zn | number |
| eb | line | mids | line | rrd | line | zrn | number |
| ed | line | midz | text | rrh | line | zrt | text |
| er | line | mir | transform | rrn | number | zt | text |
| hb | line | mov | transform | rrt | text | | |

AD



```
[FOUR,B,R,rulecolor,AD] [red]
```

AU



```
[FOUR,B,R,rulecolor,AU] [red]
```

B

```
[FOUR,B]
```

C

```
[FOUR,B,rulecolor,C] [red]
```

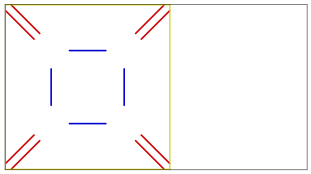
CC

```
[FOUR,B,rulecolor,CC1..5] [red]
```

CCD  [FOUR,B,rulecolor,CCD1..5] [red]

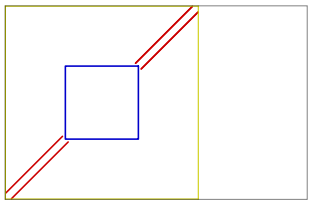
CD  [FOUR,B,rulecolor,CD] [red]

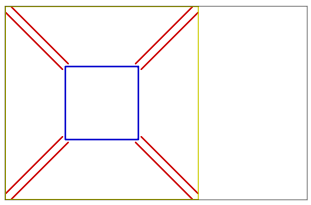
DB  [FOUR,DB]

DR  [FOUR,SB,rulecolor,DR] [red]

EB  [FOUR,B,rulecolor,EB135] [red]

ED  [FOUR,B,rulecolor,ED135] [red]

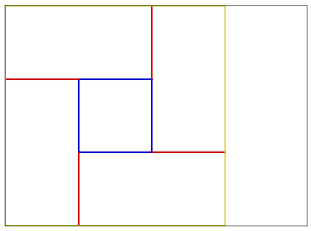
ER  [FOUR,B,rulecolor,ER135] [red]

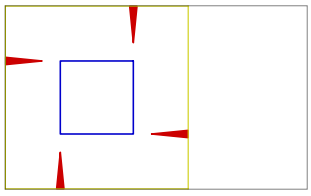
ER  [FOUR,B,rulecolor,ER] [red]

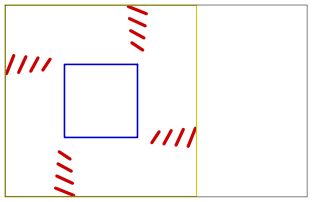
HB  [FOUR,HB]

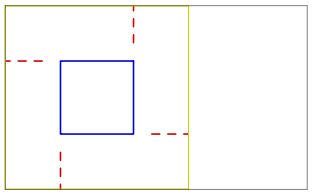
LABEL 

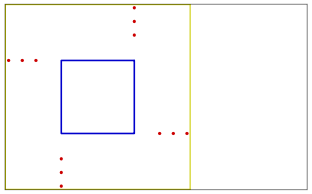
[FOUR,B,LABEL] [A]

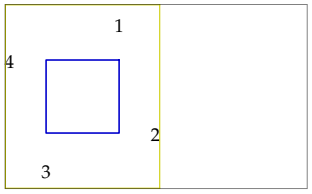
LR  [FOUR,B,rulecolor,LR] [red]

LRB  [FOUR,B,rulecolor,LRB] [red]

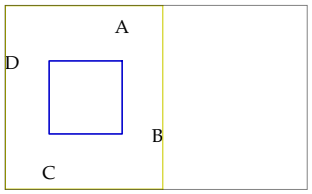
LRBD  [FOUR,B,rulecolor,LRBD] [red]

LRD  [FOUR,B,rulecolor,LRD] [red]

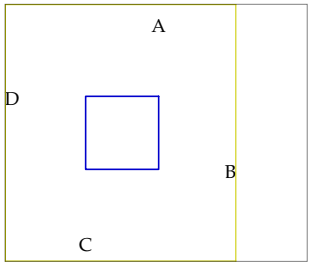
LRH  [FOUR,B,rulecolor,LRH] [red]

LRN 

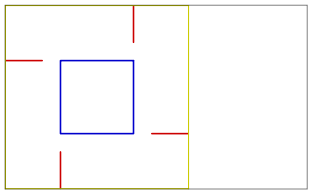
[FOUR,B,LRN]

LRT 

[FOUR,B,LRT] [A,B,C,D]

LRZ 

[FOUR,B,LRZ] [A,B,C,D]

LSR  [FOUR,B,rulecolor,LSR] [red]

MID  [FOUR,B,rulecolor,MID135] [red]

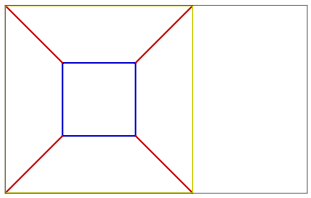
MIDS  [FOUR,B,rulecolor,MIDS135] [red]

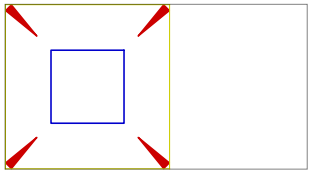
MIDZ 

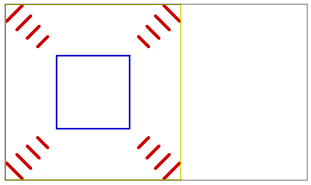
[FOUR,B,MIDZ] [A]

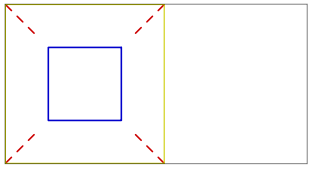
MOV 
[FOUR,B,MOV,rulecolor,FOUR,B] [red]

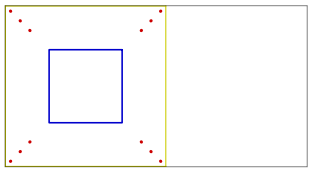
OFF 
[FOUR,B,OFF,rulecolor,FOUR,B] [red]

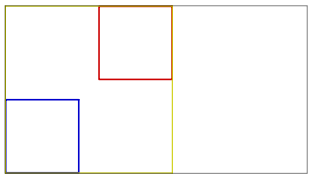
R 
[FOUR,B,rulecolor,R] [red]

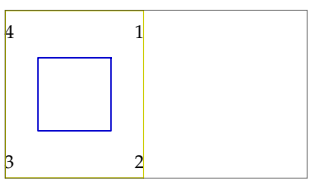
RB 
[FOUR,B,rulecolor,RB] [red]

RBD 
[FOUR,B,rulecolor,RBD] [red]

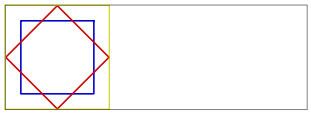
RD 
[FOUR,B,rulecolor,RD] [red]

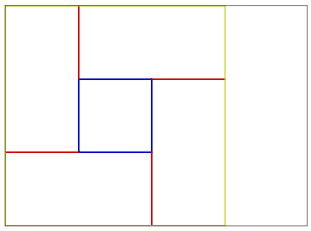
RH 
[FOUR,B,rulecolor,RH] [red]

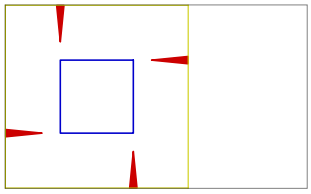
RM 
[FOUR,B,RM,rulecolor,FOUR,B] [red]

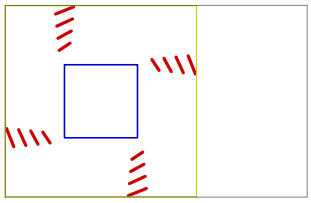
RN 

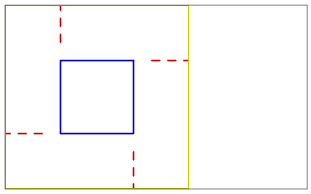
[FOUR,B,RN]

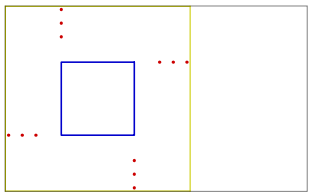
ROT  [FOUR,B,ROT,rulecolor,FOUR,B] [red]

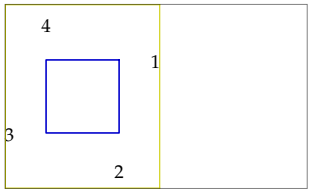
RR  [FOUR,B,rulecolor,RR] [red]

RRB  [FOUR,B,rulecolor,RRB] [red]

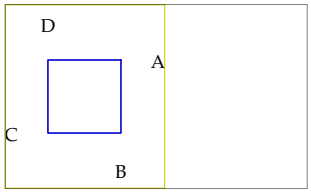
RRBD  [FOUR,B,rulecolor,RRBD] [red]

RRD  [FOUR,B,rulecolor,RRD] [red]

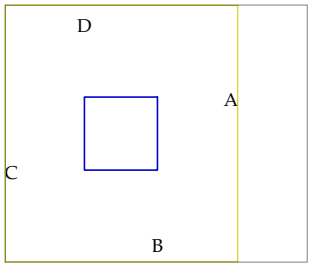
RRH  [FOUR,B,rulecolor,RRH] [red]

RRN 

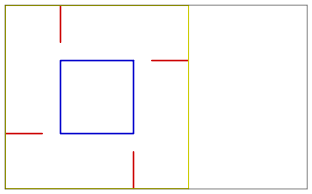
[FOUR,B,RRN]

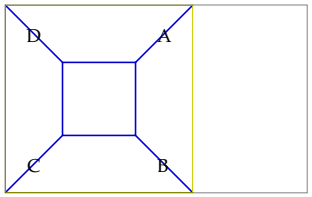
RRT 

[FOUR,B,RRT] [A,B,C,D]

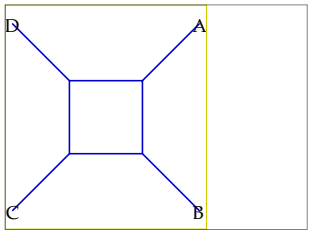
RRZ 

`[FOUR,B,RRZ] [A,B,C,D]`

RSR 
`[FOUR,B,rulecolor,RSR] [red]`

RT 

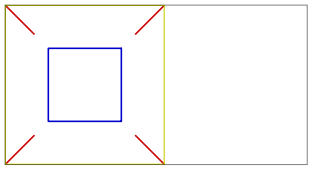
`[FOUR,B,R,RT] [A,B,C,D]`

RZ 

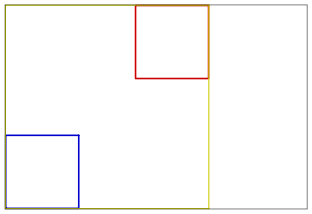
`[FOUR,B,R,RZ] [A,B,C,D]`

S 
`[FOUR,B,rulecolor,S] [red]`

SB 
`[FOUR,SB]`

SR 
`[FOUR,B,rulecolor,SR] [red]`

SS 
`[FOUR,B,rulecolor,SS135] [red]`

SUB 
`[FOUR,B,SUB,rulecolor,FOUR,B] [red]`

Z 

`[FOUR,SB,Z] [A,B,C,D]`

ZLN 

`[FOUR,B,ZLN] [A,B,C,D]`

ZLT 

`[FOUR,B,ZLT] [A,B,C,D]`

ZN 

`[FOUR,B,ZN] [A,B,C,D]`

ZRN 

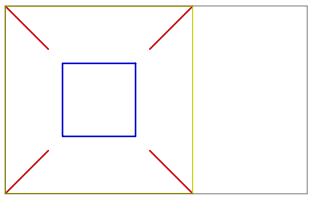
`[FOUR,B,ZRN] [A,B,C,D]`

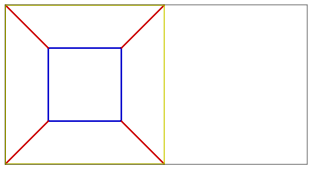
ZRT 

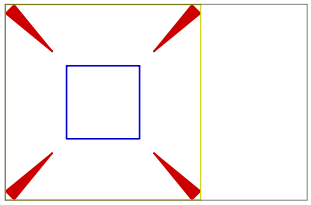
`[FOUR,B,ZRT] [A,B,C,D]`

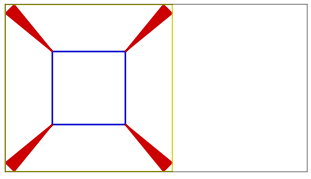
ZT 

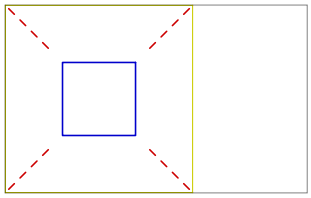
`[FOUR,B,ZT] [A,B,C,D]`

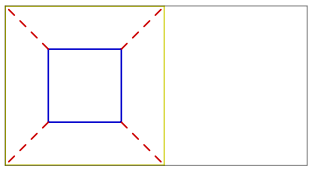
-R 
`[FOUR,B,rulecolor,-R] [red]`

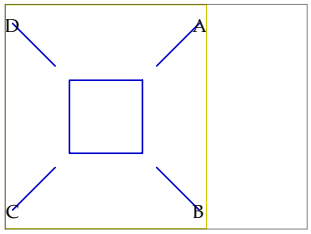
+R 
`[FOUR,B,rulecolor,+R] [red]`

-RB 
`[FOUR,B,rulecolor,-RB] [red]`

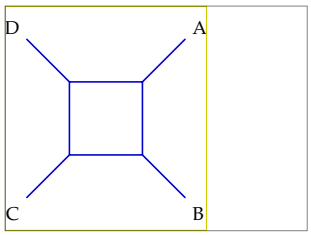
+RB 
`[FOUR,B,rulecolor,+RB] [red]`

-RD  [FOUR,B,rulecolor,-RD] [red]

+RD  [FOUR,B,rulecolor,+RD] [red]

-RZ 

[FOUR,B,-SR,-RZ] [A,B,C,D]

+RZ 

[FOUR,B,+SR,+RZ] [A,B,C,D]

-S  [FOUR,B,rulecolor,-S246] [red]

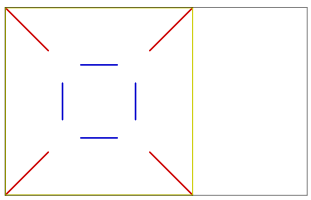
+S  [FOUR,B,rulecolor,+S246] [red]

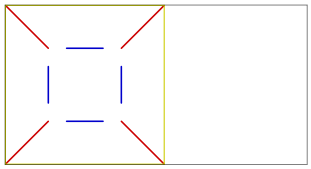
-SB  [FOUR,-SB1,rulecolor,B2..6] [red]

--SB  [FOUR,--SB1,rulecolor,B2..6] [red]

+SB  [FOUR,+SB1,rulecolor,B2..6] [red]

++SB  [FOUR,++SB1,rulecolor,B2..6] [red]

-SR  [FOUR,SB,rulecolor,-SR] [red]

+SR  [FOUR,SB,rulecolor,+SR] [red]

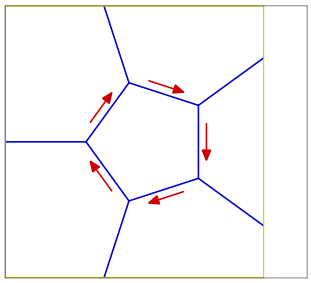
-SS  [FOUR,B,rulecolor,-SS246] [red]

+SS  [FOUR,B,rulecolor,+SS246] [red]

# 20 Five

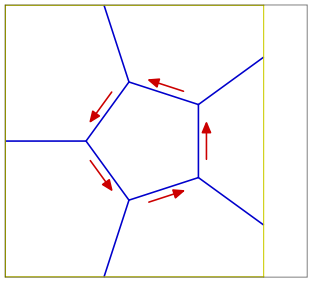
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ad | line | label | text | move | transform | rrz | text |
| adj | transform | line | line | number | number | rsr | line |
| arrow | line | lr | line | off | transform | rsub | transform |
| au | line | lrb | line | r | line | rt | text |
| b | line | lrbd | line | rb | line | rz | text |
| c | fixed | lrd | line | rbd | line | s | line |
| cc | line | lrh | line | rd | line | sb | line |
| ccd | line | lrn | number | rh | line | sr | line |
| cd | fixed | lrt | text | rm | transform | ss | line |
| crz | text | lrz | text | rn | number | sub | transform |
| dash | line | lsr | line | rot | transform | z | text |
| db | line | lsub | transform | rr | line | zln | number |
| diff | transform | mark | transform | rrb | line | zlt | text |
| dr | line | mid | line | rrbd | line | zn | number |
| eb | line | mids | line | rrd | line | zrn | number |
| ed | line | midz | text | rrh | line | zrt | text |
| er | line | mir | transform | rrn | number | zt | text |
| hb | line | mov | transform | rrt | text | | |

AD



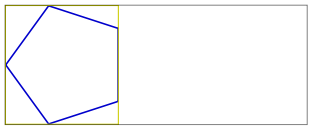
```
[FIVE,B,R,rulecolor,AD] [red]
```

AU

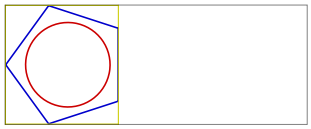


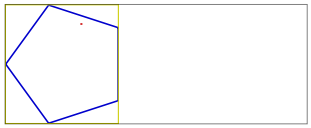
```
[FIVE,B,R,rulecolor,AU] [red]
```

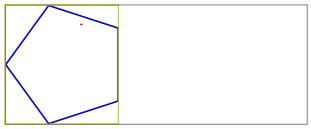
B

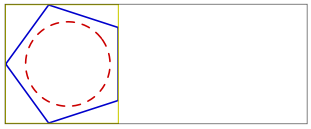


```
[FIVE,B]
```

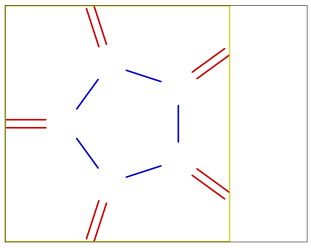
C  [FIVE,B,rulecolor,C] [red]

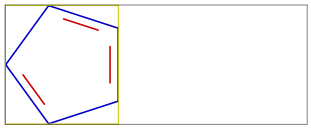
CC  [FIVE,B,rulecolor,CC1..5] [red]

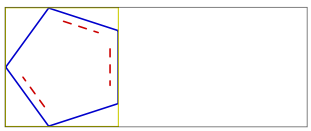
CCD  [FIVE,B,rulecolor,CCD1..5] [red]

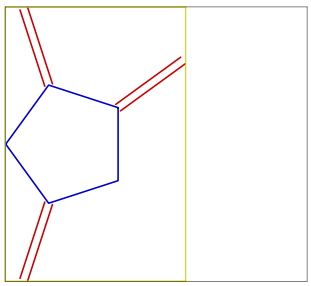
CD  [FIVE,B,rulecolor,CD] [red]

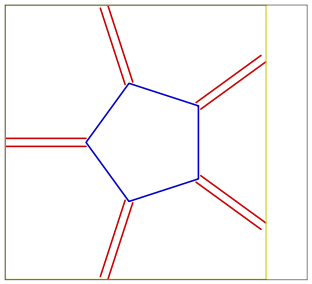
DB  [FIVE,DB]

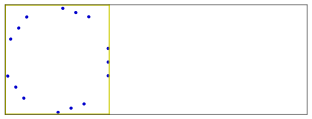
DR  [FIVE,SB,rulecolor,DR] [red]

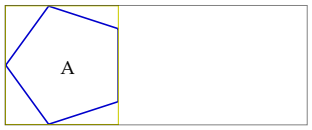
EB  [FIVE,B,rulecolor,EB135] [red]

ED  [FIVE,B,rulecolor,ED135] [red]

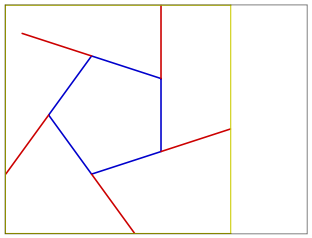
ER  [FIVE,B,rulecolor,ER135] [red]

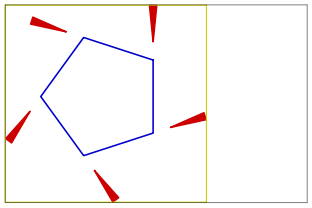
ER  [FIVE,B,rulecolor,ER] [red]

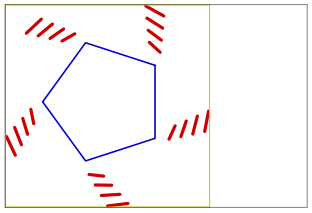
HB  [FIVE,HB]

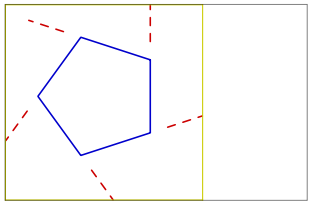
LABEL 

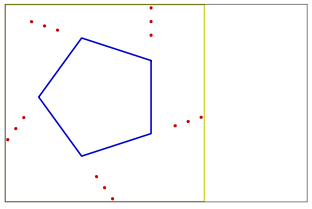
[FIVE,B,LABEL] [A]

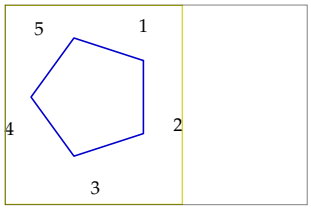
LR  [FIVE,B,rulecolor,LR] [red]

LRB  [FIVE,B,rulecolor,LRB] [red]

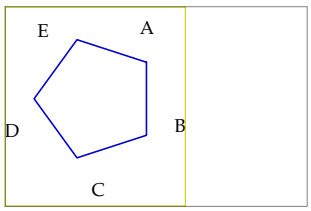
LRBD  [FIVE,B,rulecolor,LRBD] [red]

LRD  [FIVE,B,rulecolor,LRD] [red]

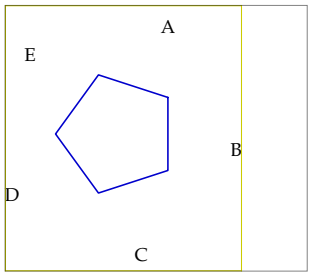
LRH  [FIVE,B,rulecolor,LRH] [red]

LRN 

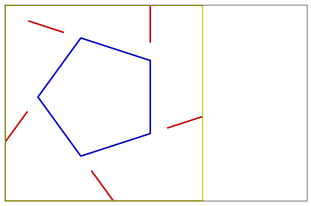
[FIVE,B,LRN]

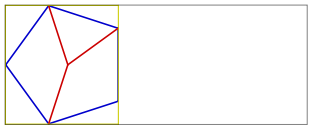
LRT 

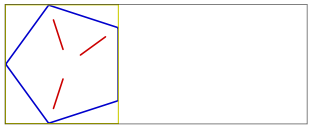
[FIVE,B,LRT] [A,B,C,D,E]

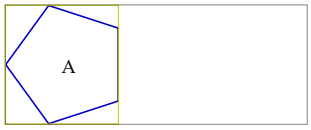
LRZ 

[FIVE,B,LRZ] [A,B,C,D,E]

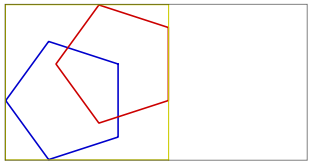
LSR 
[FIVE,B,rulecolor,LSR] [red]

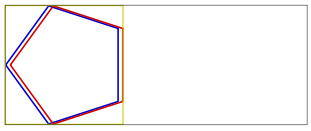
MID 
[FIVE,B,rulecolor,MID135] [red]

MIDS 
[FIVE,B,rulecolor,MIDS135] [red]

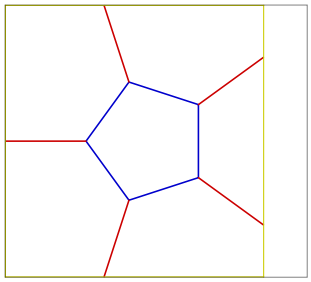
MIDZ 

[FIVE,B,MIDZ] [A]

MOV 
[FIVE,B,MOV,rulecolor,FIVE,B] [red]

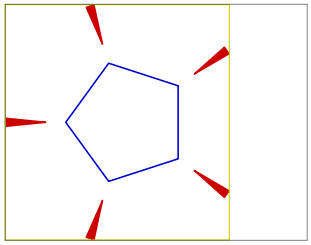
OFF 
[FIVE,B,OFF,rulecolor,FIVE,B] [red]

R



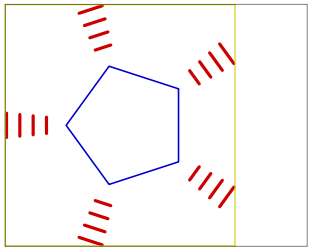
`[FIVE,B,rulecolor,R] [red]`

RB



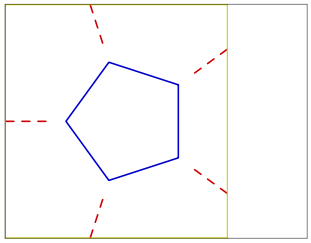
`[FIVE,B,rulecolor,RB] [red]`

RBD



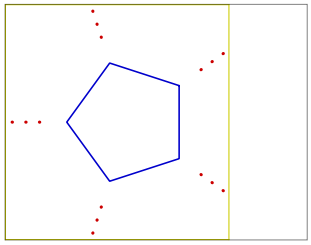
`[FIVE,B,rulecolor,RBD] [red]`

RD



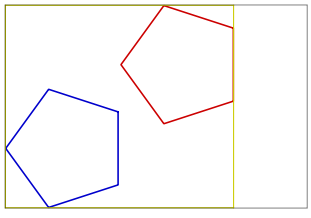
`[FIVE,B,rulecolor,RD] [red]`

RH

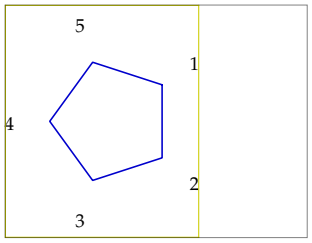


`[FIVE,B,rulecolor,RH] [red]`

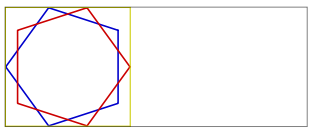
RM

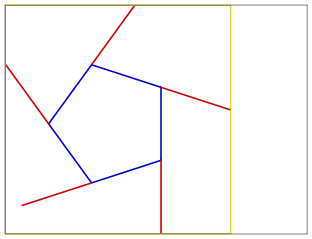


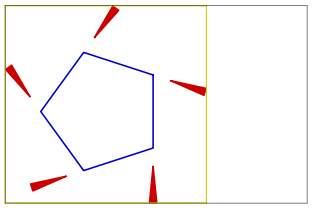
`[FIVE,B,RM,rulecolor,FIVE,B] [red]`

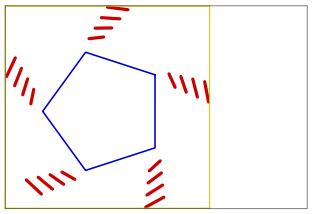
RN 

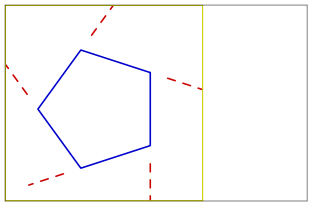
[FIVE,B,RN]

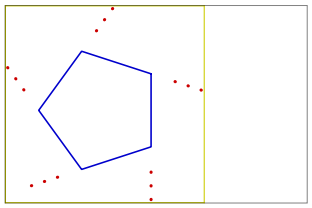
ROT 
[FIVE,B,ROT,rulecolor,FIVE,B] [red]

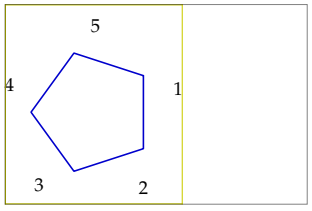
RR 
[FIVE,B,rulecolor,RR] [red]

RRB 
[FIVE,B,rulecolor,RRB] [red]

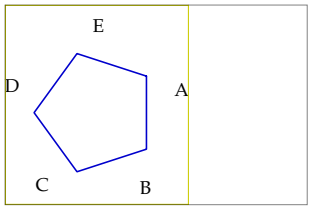
RRBD 
[FIVE,B,rulecolor,RRBD] [red]

RRD 
[FIVE,B,rulecolor,RRD] [red]

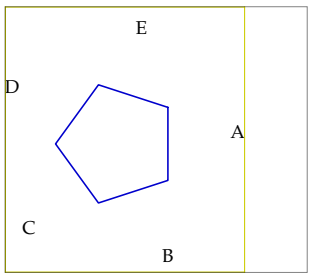
RRH 
[FIVE,B,rulecolor,RRH] [red]

RRN 

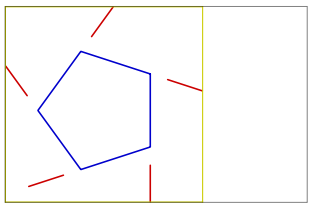
[FIVE,B,RRN]

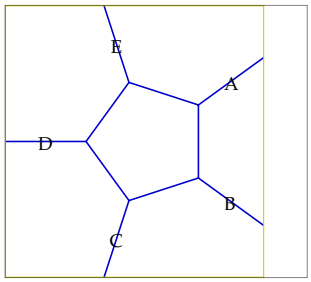
RRT 

`[FIVE,B,RRT] [A,B,C,D,E]`

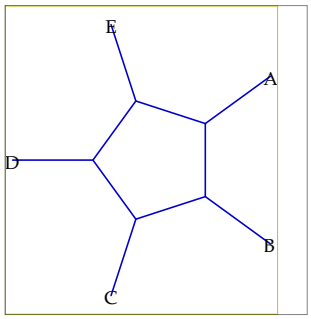
RRZ 

`[FIVE,B,RRZ] [A,B,C,D,E]`

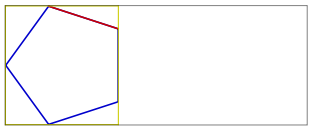
RSR 
`[FIVE,B,rulecolor,RSR] [red]`

RT 

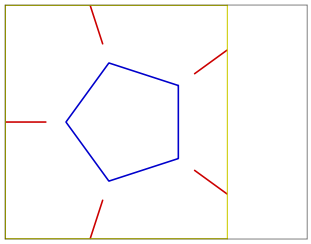
`[FIVE,B,R,RT] [A,B,C,D,E]`

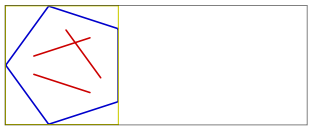
RZ 

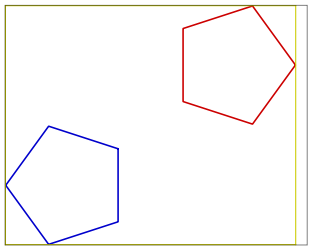
`[FIVE,B,R,RZ] [A,B,C,D,E]`

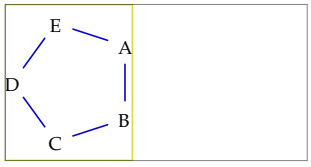
S 
`[FIVE,B,rulecolor,S] [red]`

SB 
`[FIVE,SB]`

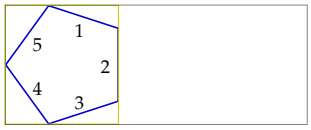
SR  [FIVE,B,rulecolor,SR] [red]

SS  [FIVE,B,rulecolor,SS135] [red]

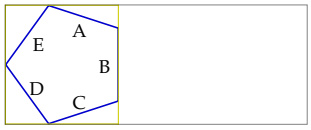
SUB  [FIVE,B,SUB,rulecolor,FIVE,B] [red]

Z 

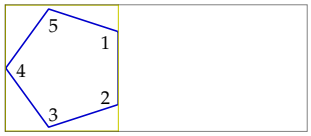
[FIVE,SB,Z] [A,B,C,D,E]

ZLN 

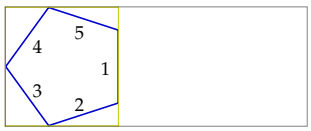
[FIVE,B,ZLN] [A,B,C,D,E]

ZLT 

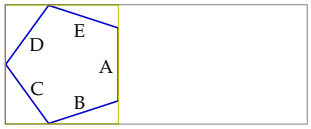
[FIVE,B,ZLT] [A,B,C,D,E]

ZN 

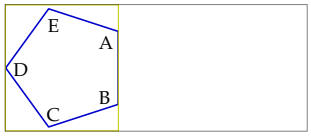
[FIVE,B,ZN] [A,B,C,D,E]

ZRN 

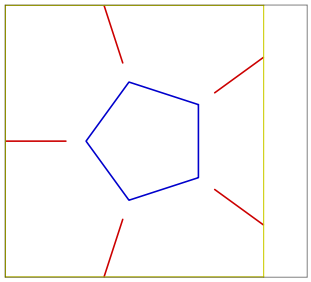
[FIVE,B,ZRN] [A,B,C,D,E]

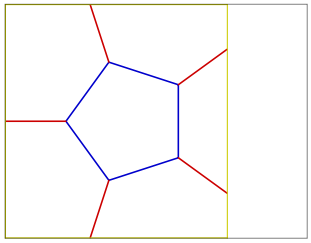
ZRT 

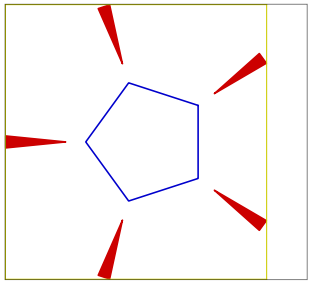
[FIVE,B,ZRT] [A,B,C,D,E]

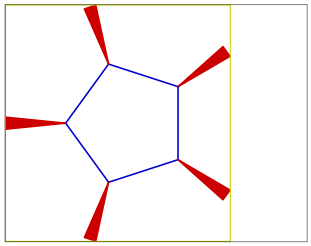
ZT 

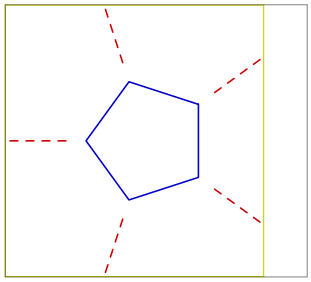
[FIVE,B,ZT] [A,B,C,D,E]

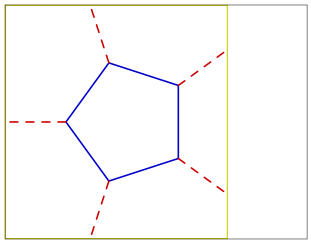
-R  [FIVE,B,rulecolor,-R] [red]

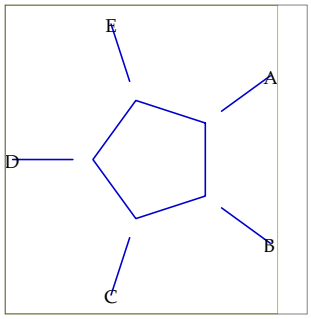
+R  [FIVE,B,rulecolor,+R] [red]

-RB  [FIVE,B,rulecolor,-RB] [red]

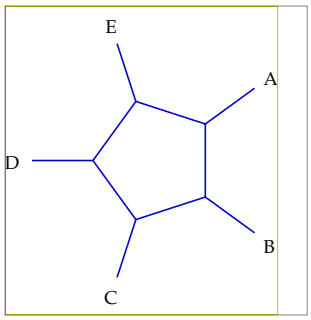
+RB  [FIVE,B,rulecolor,+RB] [red]

-RD  [FIVE,B,rulecolor,-RD] [red]

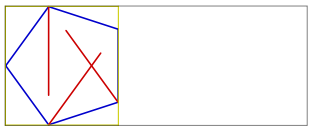
+RD  [FIVE,B,rulecolor,+RD] [red]

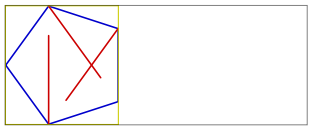
-RZ 

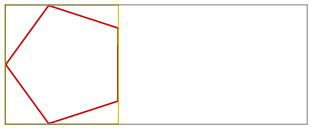
[FIVE,B,-SR,-RZ] [A,B,C,D,E]

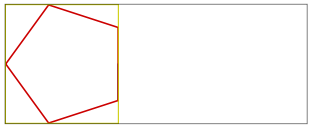
+RZ 

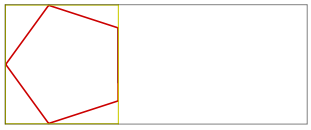
[FIVE,B,+SR,+RZ] [A,B,C,D,E]

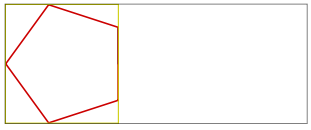
-S 
[FIVE,B,rulecolor,-S246] [red]

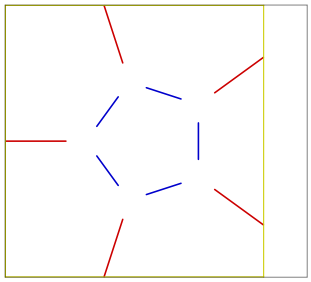
+S 
[FIVE,B,rulecolor,+S246] [red]

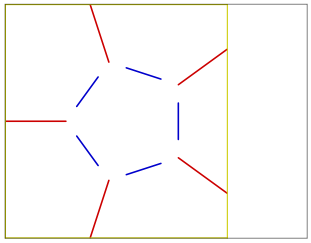
-SB 
[FIVE,-SB1,rulecolor,B2..6] [red]

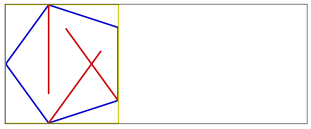
--SB 
[FIVE,--SB1,rulecolor,B2..6] [red]

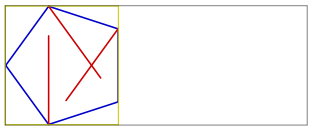
+SB 
[FIVE,+SB1,rulecolor,B2..6] [red]

++SB 
[FIVE,++SB1,rulecolor,B2..6] [red]

-SR  [FIVE,SB,rulecolor,-SR] [red]

+SR  [FIVE,SB,rulecolor,+SR] [red]

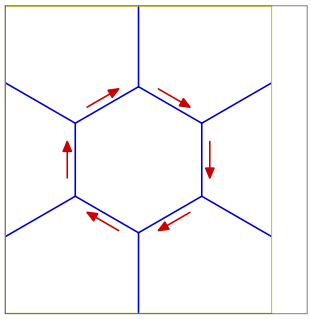
-SS  [FIVE,B,rulecolor,-SS246] [red]

+SS  [FIVE,B,rulecolor,+SS246] [red]

# 21 Six

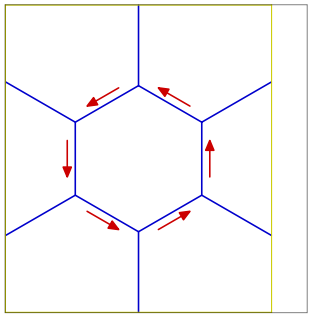
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ad | line | label | text | move | transform | rrz | text |
| adj | transform | line | line | number | number | rsr | line |
| arrow | line | lr | line | off | transform | rsub | transform |
| au | line | lrb | line | r | line | rt | text |
| b | line | lrbd | line | rb | line | rz | text |
| c | fixed | lrd | line | rbd | line | s | line |
| cc | line | lrh | line | rd | line | sb | line |
| ccd | line | lrn | number | rh | line | sr | line |
| cd | fixed | lrt | text | rm | transform | ss | line |
| crz | text | lrz | text | rn | number | sub | transform |
| dash | line | lsr | line | rot | transform | z | text |
| db | line | lsub | transform | rr | line | zln | number |
| diff | transform | mark | transform | rrb | line | zlt | text |
| dr | line | mid | line | rrbd | line | zn | number |
| eb | line | mids | line | rrd | line | zrn | number |
| ed | line | midz | text | rrh | line | zrt | text |
| er | line | mir | transform | rrn | number | zt | text |
| hb | line | mov | transform | rrt | text | | |

AD

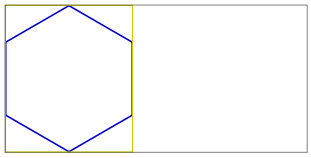


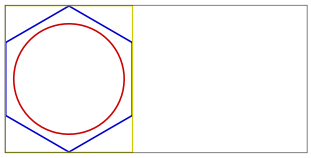
```
[SIX,B,R,rulecolor,AD] [red]
```

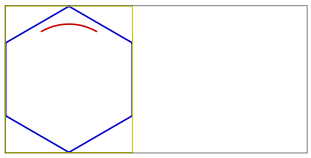
AU

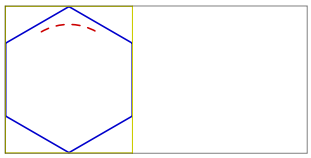


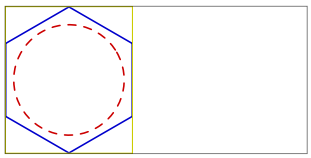
```
[SIX,B,R,rulecolor,AU] [red]
```

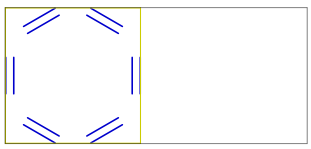
B  [SIX,B]

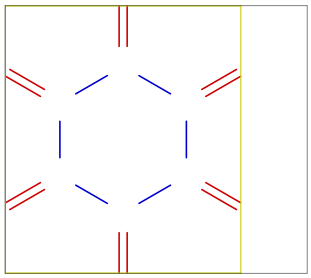
C  [SIX,B,rulecolor,C] [red]

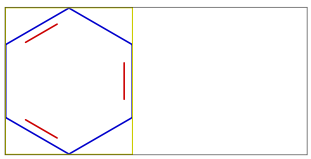
CC  [SIX,B,rulecolor,CC1..5] [red]

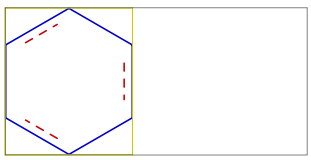
CCD  [SIX,B,rulecolor,CCD1..5] [red]

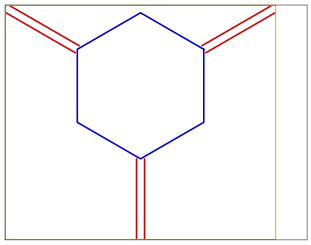
CD  [SIX,B,rulecolor,CD] [red]

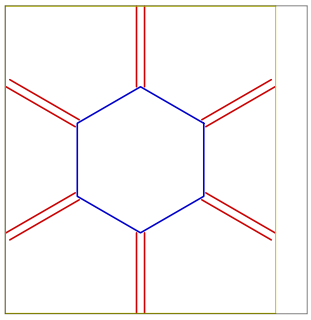
DB  [SIX,DB]

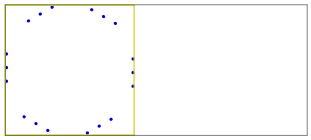
DR  [SIX,SB,rulecolor,DR] [red]

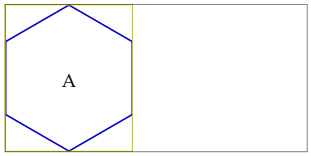
EB  [SIX,B,rulecolor,EB135] [red]

ED  [SIX,B,rulecolor,ED135] [red]

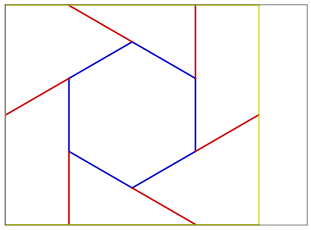
ER  [SIX,B,rulecolor,ER135] [red]

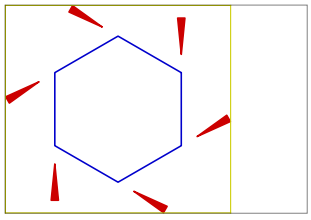
ER  [SIX,B,rulecolor,ER] [red]

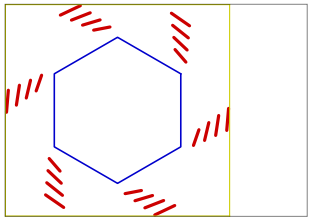
HB  [SIX,HB]

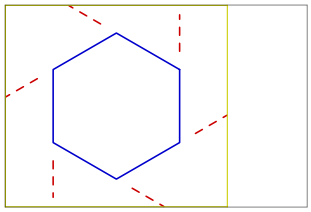
LABEL 

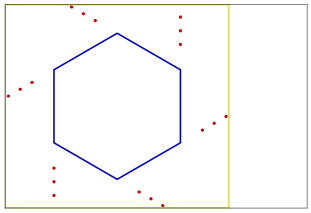
[SIX,B,LABEL] [A]

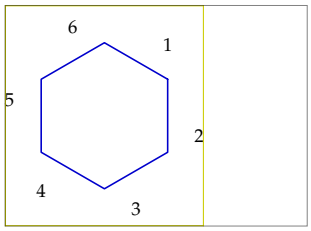
LR  [SIX,B,rulecolor,LR] [red]

LRB  [SIX,B,rulecolor,LRB] [red]

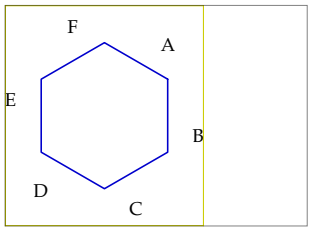
LRBD  [SIX,B,rulecolor,LRBD] [red]

LRD  `[SIX,B,rulecolor,LRD] [red]`

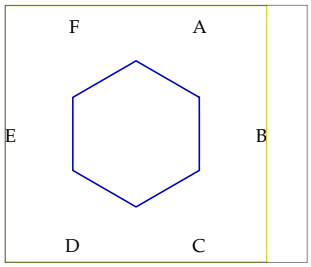
LRH  `[SIX,B,rulecolor,LRH] [red]`

LRN 

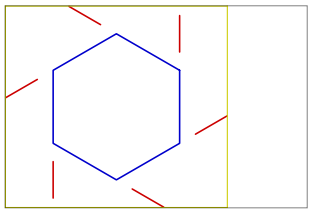
`[SIX,B,LRN]`

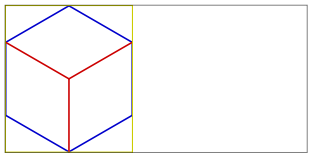
LRT 

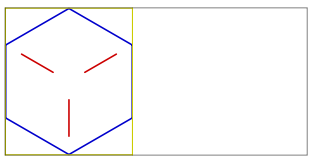
`[SIX,B,LRT] [A,B,C,D,E,F]`

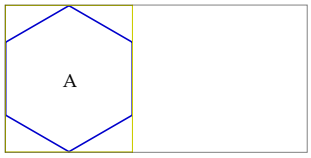
LRZ 

`[SIX,B,LRZ] [A,B,C,D,E,F]`

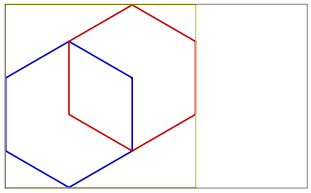
LSR  `[SIX,B,rulecolor,LSR] [red]`

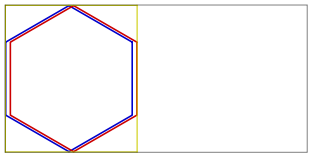
MID  `[SIX,B,rulecolor,MID135] [red]`

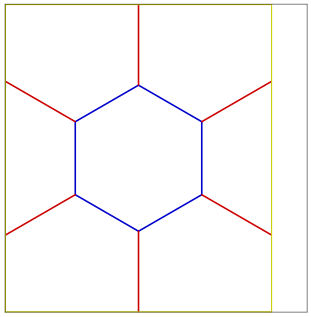
MIDS  `[SIX,B,rulecolor,MIDS135] [red]`

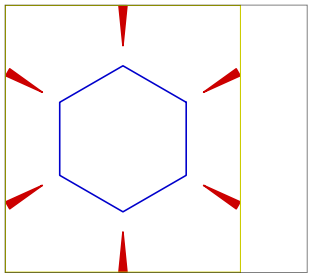
MIDZ 

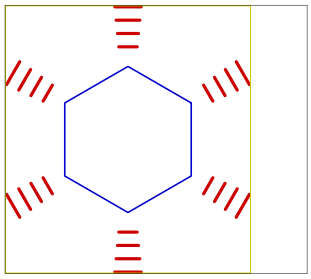
[SIX,B,MIDZ] [A]

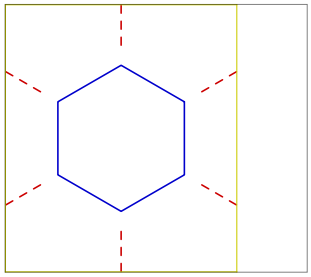
MOV 
[SIX,B,MOV,rulecolor,SIX,B] [red]

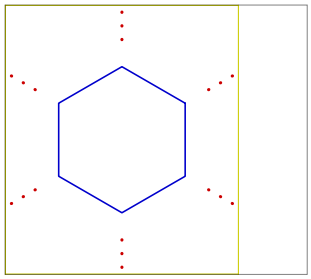
OFF 
[SIX,B,OFF,rulecolor,SIX,B] [red]

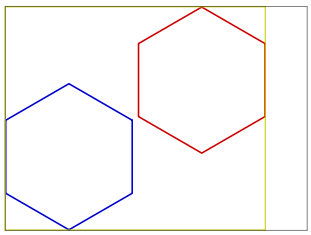
R 
[SIX,B,rulecolor,R] [red]

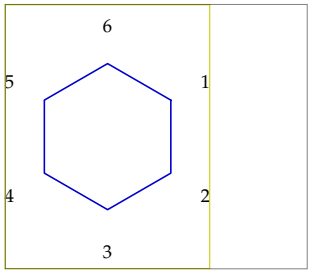
RB 
[SIX,B,rulecolor,RB] [red]

RBD 
[SIX,B,rulecolor,RBD] [red]

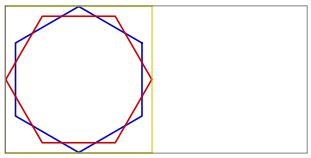
RD 
[SIX,B,rulecolor,RD] [red]

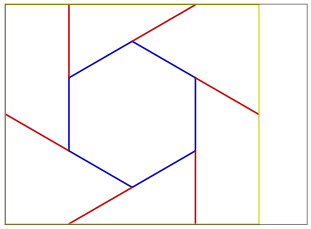
RH  [SIX,B,rulecolor,RH] [red]

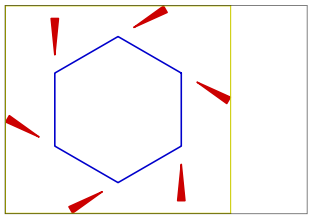
RM  [SIX,B,RM,rulecolor,SIX,B] [red]

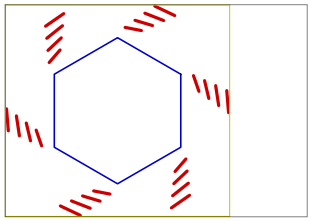
RN 

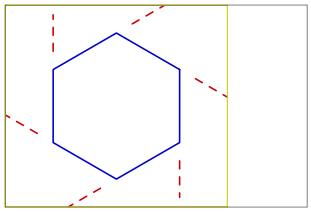
[SIX,B,RN]

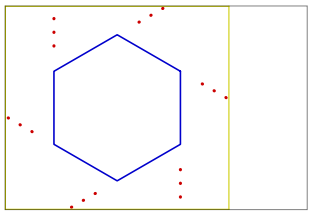
ROT  [SIX,B,ROT,rulecolor,SIX,B] [red]

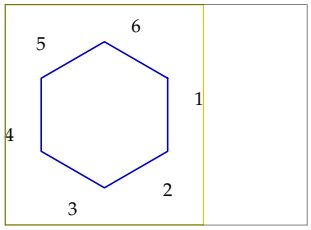
RR  [SIX,B,rulecolor,RR] [red]

RRB  [SIX,B,rulecolor,RRB] [red]

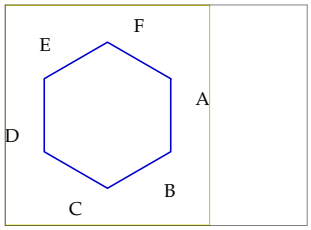
RRBD  [SIX,B,rulecolor,RRBD] [red]

RRD  [SIX,B,rulecolor,RRD] [red]

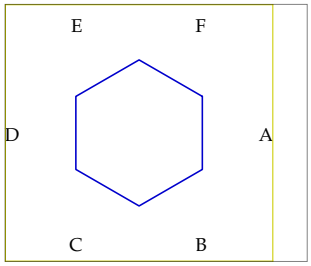
RRH  [SIX,B,rulecolor,RRH] [red]

RRN 

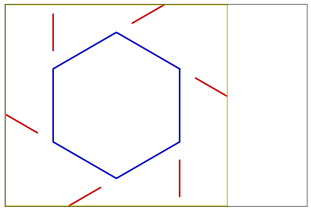
[SIX,B,RRN]

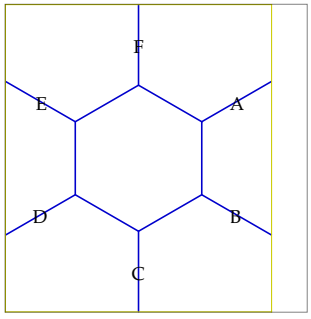
RRT 

[SIX,B,RRT] [A,B,C,D,E,F]

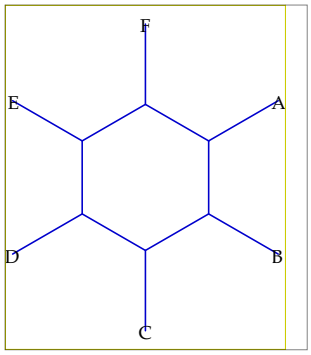
RRZ 

[SIX,B,RRZ] [A,B,C,D,E,F]

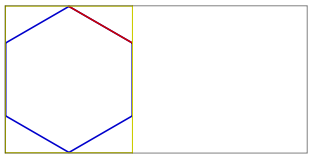
RSR  [SIX,B,rulecolor,RSR] [red]

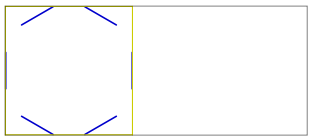
RT 

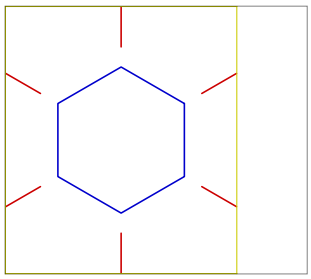
[SIX,B,R,RT] [A,B,C,D,E,F]

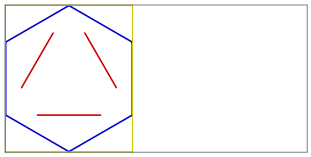
RZ 

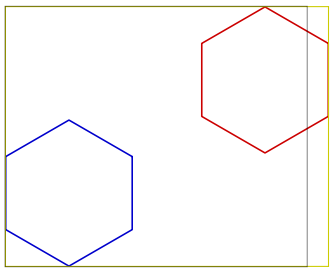
`[SIX,B,R,RZ] [A,B,C,D,E,F]`

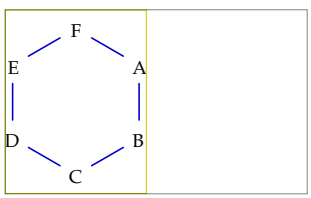
S 
`[SIX,B,rulecolor,S] [red]`

SB 
`[SIX,SB]`

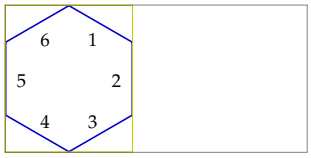
SR 
`[SIX,B,rulecolor,SR] [red]`

SS 
`[SIX,B,rulecolor,SS135] [red]`

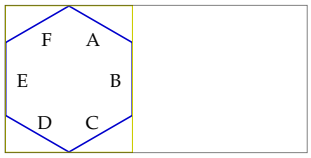
SUB 
`[SIX,B,SUB,rulecolor,SIX,B] [red]`

Z 

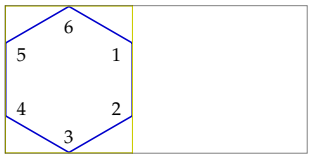
`[SIX,SB,Z] [A,B,C,D,E,F]`

ZLN 

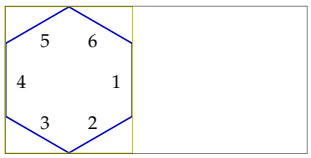
`[SIX,B,ZLN] [A,B,C,D,E,F]`

ZLT 

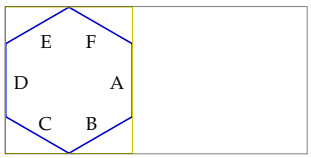
`[SIX,B,ZLT] [A,B,C,D,E,F]`

ZN 

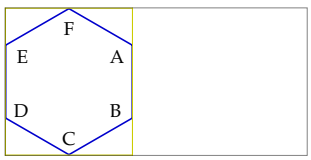
`[SIX,B,ZN] [A,B,C,D,E,F]`

ZRN 

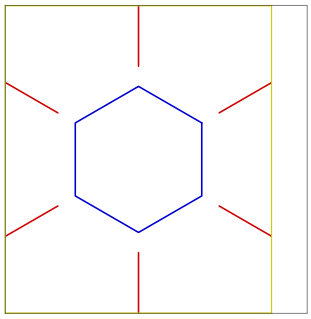
`[SIX,B,ZRN] [A,B,C,D,E,F]`

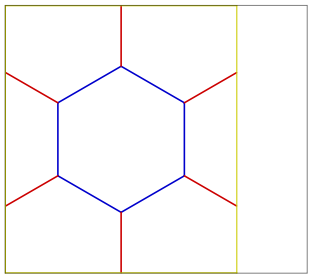
ZRT 

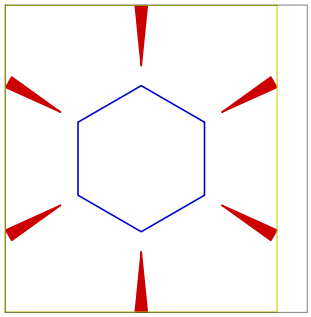
`[SIX,B,ZRT] [A,B,C,D,E,F]`

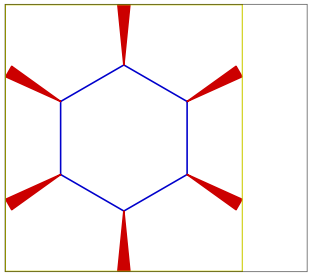
ZT 

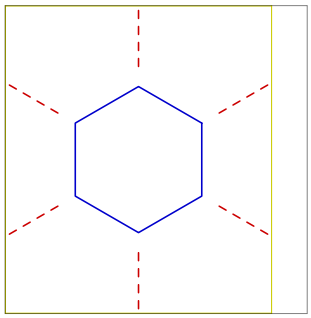
`[SIX,B,ZT] [A,B,C,D,E,F]`

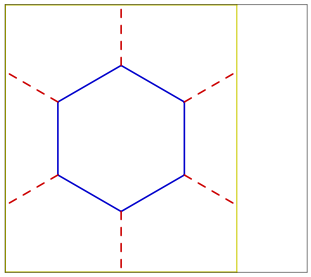
-R 
`[SIX,B,rulecolor,-R] [red]`

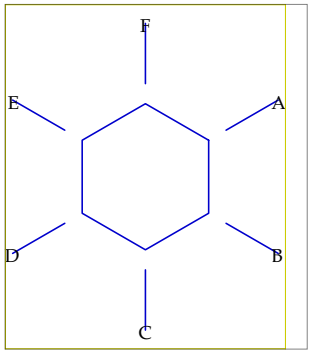
+R 
`[SIX,B,rulecolor,+R] [red]`

-RB  [SIX,B,rulecolor,-RB] [red]

+RB  [SIX,B,rulecolor,+RB] [red]

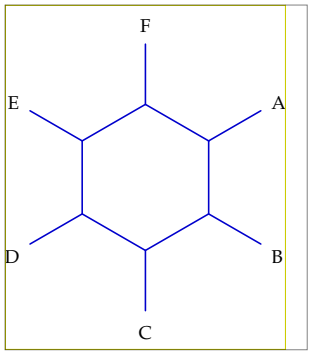
-RD  [SIX,B,rulecolor,-RD] [red]

+RD  [SIX,B,rulecolor,+RD] [red]

-RZ 

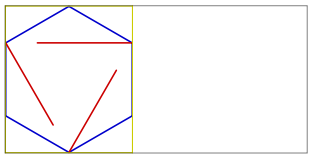
[SIX,B,-SR,-RZ] [A,B,C,D,E,F]

+RZ



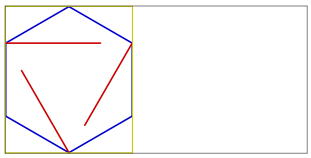

[SIX,B,+SR,+RZ] [A,B,C,D,E,F]

-S



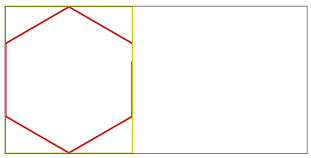
[SIX,B,rulecolor,-S246] [red]

+S



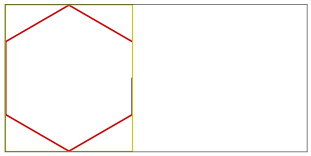
[SIX,B,rulecolor,+S246] [red]

-SB



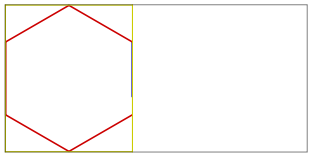
[SIX,-SB1,rulecolor,B2..6] [red]

--SB



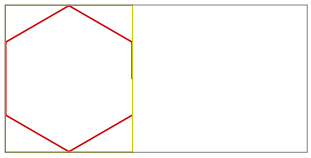
[SIX,--SB1,rulecolor,B2..6] [red]

+SB



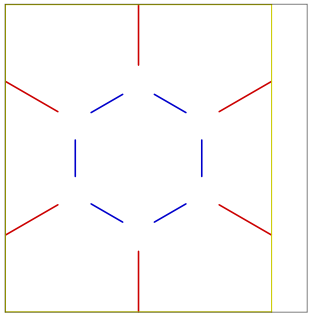
[SIX,+SB1,rulecolor,B2..6] [red]

++SB

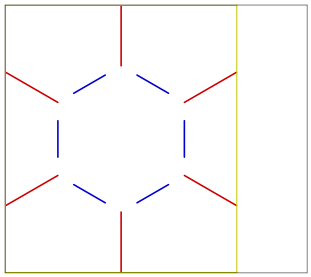


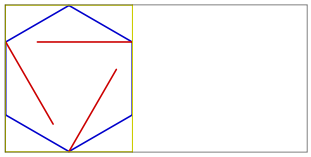
[SIX,++SB1,rulecolor,B2..6] [red]

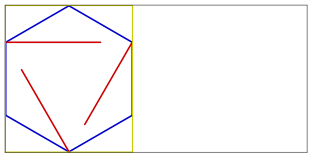
-SR



[SIX,SB,rulecolor,-SR] [red]

+SR  `[SIX,SB,rulecolor,+SR] [red]`

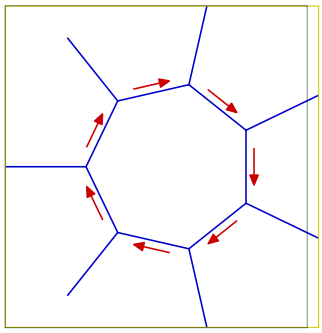
-SS  `[SIX,B,rulecolor,-SS246] [red]`

+SS  `[SIX,B,rulecolor,+SS246] [red]`

# 22 Seven

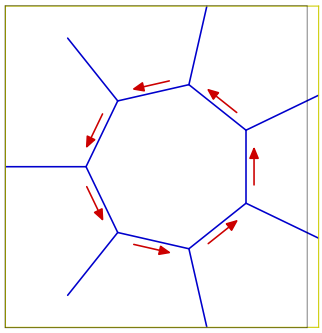
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ad | line | label | text | move | transform | rrz | text |
| adj | transform | line | line | number | number | rsr | line |
| arrow | line | lr | line | off | transform | rsub | transform |
| au | line | lrb | line | r | line | rt | text |
| b | line | lrbd | line | rb | line | rz | text |
| c | fixed | lrd | line | rbd | line | s | line |
| cc | line | lrh | line | rd | line | sb | line |
| ccd | line | lrn | number | rh | line | sr | line |
| cd | fixed | lrt | text | rm | transform | ss | line |
| crz | text | lrz | text | rn | number | sub | transform |
| dash | line | lsr | line | rot | transform | z | text |
| db | line | lsub | transform | rr | line | zln | number |
| diff | transform | mark | transform | rrb | line | zlt | text |
| dr | line | mid | line | rrbd | line | zn | number |
| eb | line | mids | line | rrd | line | zrn | number |
| ed | line | midz | text | rrh | line | zrt | text |
| er | line | mir | transform | rrn | number | zt | text |
| hb | line | mov | transform | rrt | text | | |

AD

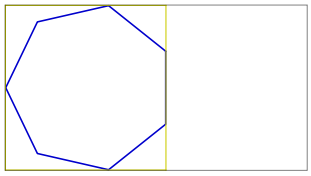


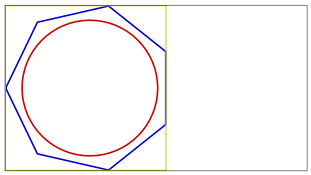
```
[SEVEN,B,R,rulecolor,AD] [red]
```

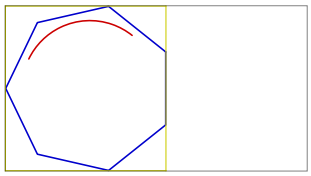
AU

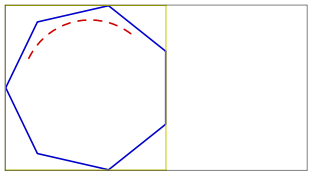


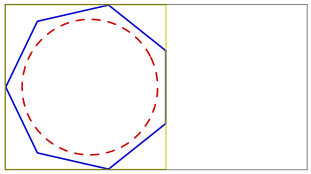
```
[SEVEN,B,R,rulecolor,AU] [red]
```

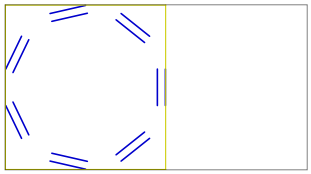
B  `[SEVEN,B]`

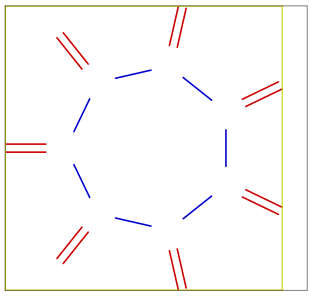
C  `[SEVEN,B,rulecolor,C] [red]`

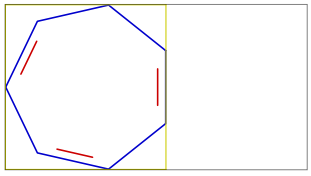
CC  `[SEVEN,B,rulecolor,CC1..5] [red]`

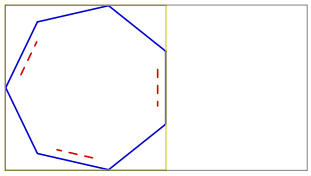
CCD  `[SEVEN,B,rulecolor,CCD1..5] [red]`

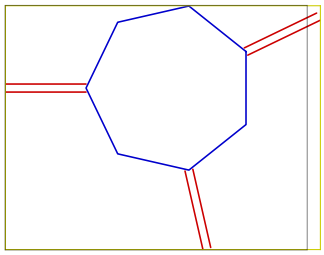
CD  `[SEVEN,B,rulecolor,CD] [red]`

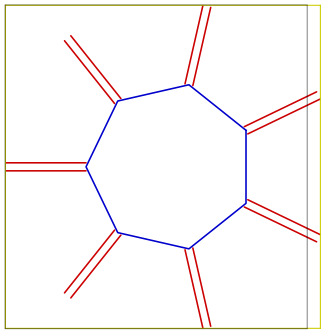
DB  `[SEVEN,DB]`

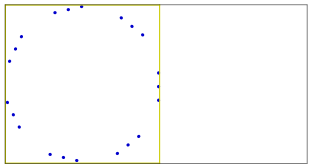
DR  `[SEVEN,SB,rulecolor,DR] [red]`

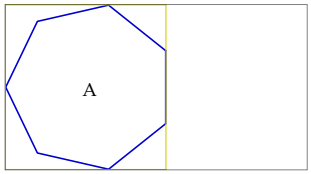
EB  `[SEVEN,B,rulecolor,EB135] [red]`

ED  [SEVEN,B,rulecolor,ED135] [red]

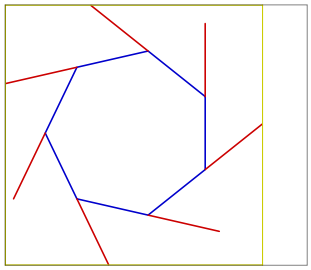
ER  [SEVEN,B,rulecolor,ER135] [red]

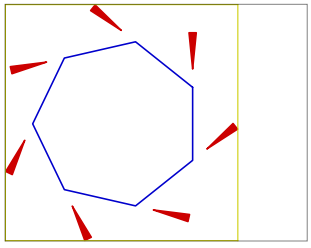
ER  [SEVEN,B,rulecolor,ER] [red]

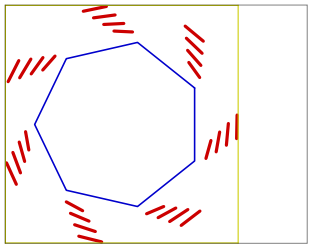
HB  [SEVEN,HB]

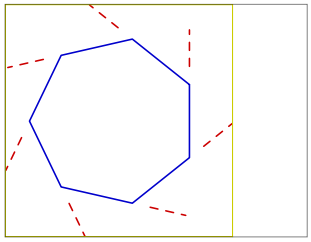
LABEL 

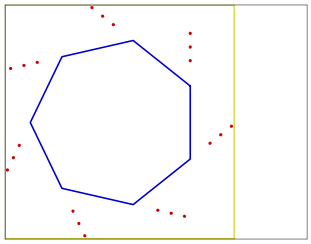
[SEVEN,B,LABEL] [A]

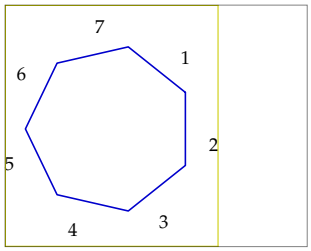
LR  [SEVEN,B,rulecolor,LR] [red]

LRB  [SEVEN,B,rulecolor,LRB] [red]

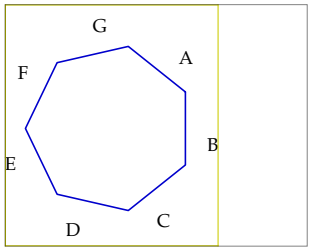
LRBD  [SEVEN,B,rulecolor,LRBD] [red]

LRD  [SEVEN,B,rulecolor,LRD] [red]

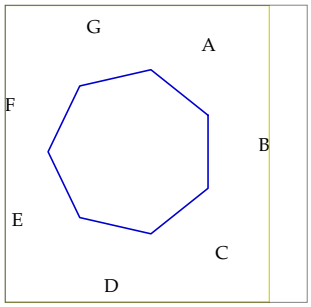
LRH  [SEVEN,B,rulecolor,LRH] [red]

LRN 

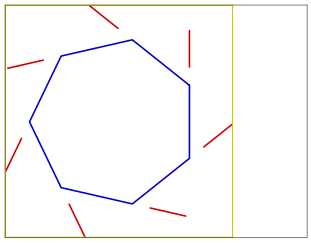
[SEVEN,B,LRN]

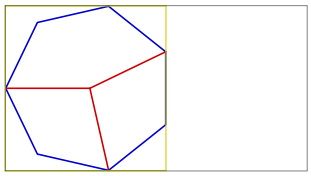
LRT 

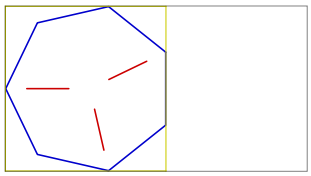
[SEVEN,B,LRT] [A,B,C,D,E,F,G]

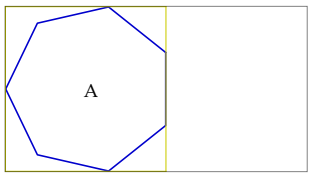
LRZ 

[SEVEN,B,LRZ] [A,B,C,D,E,F,G]

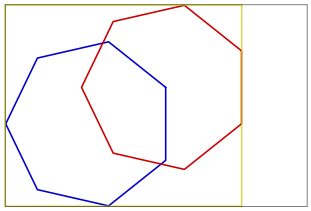
LSR  [SEVEN,B,rulecolor,LSR] [red]

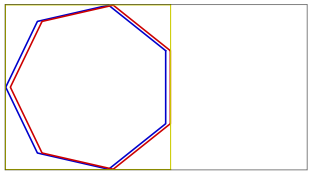
MID  [SEVEN,B,rulecolor,MID135] [red]

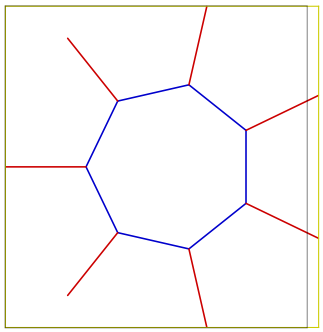
MIDS  [SEVEN,B,rulecolor,MIDS135] [red]

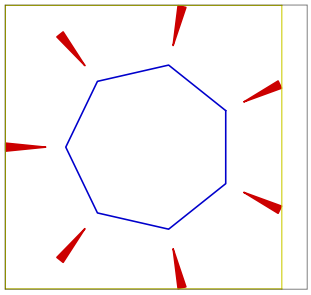
MIDZ 

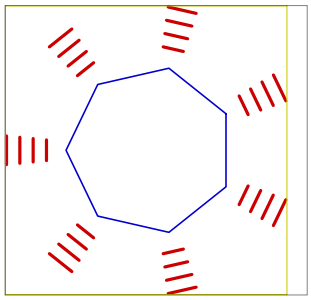
[SEVEN,B,MIDZ] [A]

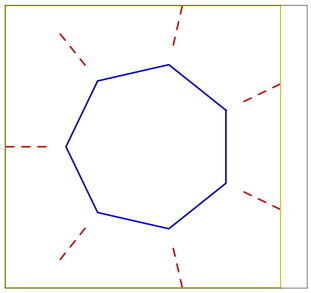
MOV  [SEVEN,B,MOV,rulecolor,SEVEN,B] [red]

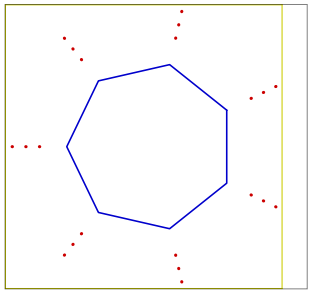
OFF  [SEVEN,B,OFF,rulecolor,SEVEN,B] [red]

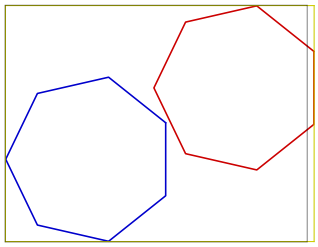
R  [SEVEN,B,rulecolor,R] [red]

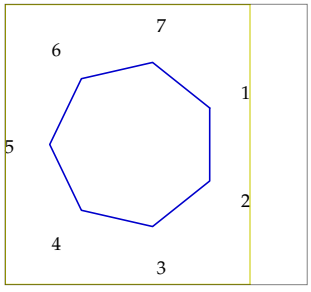
RB  [SEVEN,B,rulecolor,RB] [red]

RBD  [SEVEN,B,rulecolor,RBD] [red]

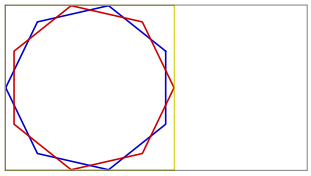
RD  [SEVEN,B,rulecolor,RD] [red]

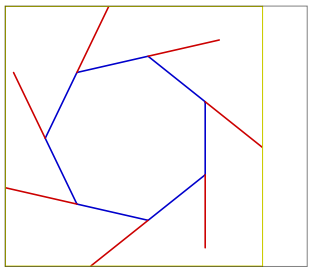
RH  [SEVEN,B,rulecolor,RH] [red]

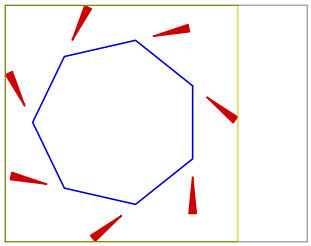
RM  [SEVEN,B,RM,rulecolor,SEVEN,B] [red]

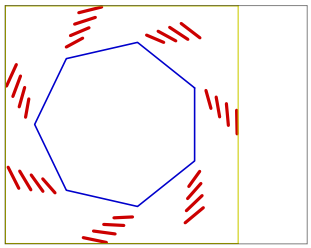
RN 

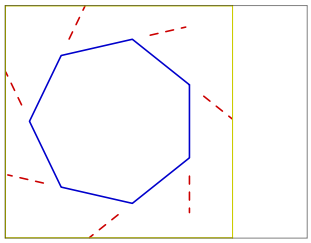
[SEVEN,B,RN]

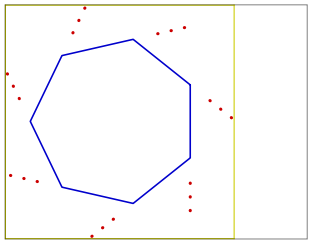
ROT  [SEVEN,B,ROT,rulecolor,SEVEN,B] [red]

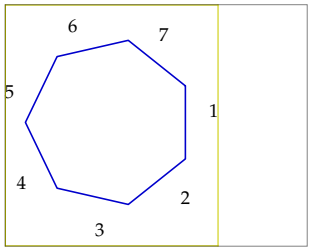
RR  [SEVEN,B,rulecolor,RR] [red]

RRB  [SEVEN,B,rulecolor,RRB] [red]

RRBD  [SEVEN,B,rulecolor,RRBD] [red]

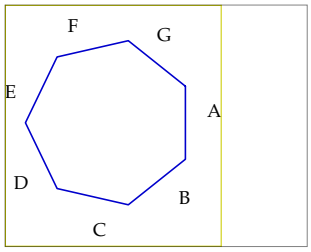
RRD  [SEVEN,B,rulecolor,RRD] [red]

RRH  [SEVEN,B,rulecolor,RRH] [red]

RRN 

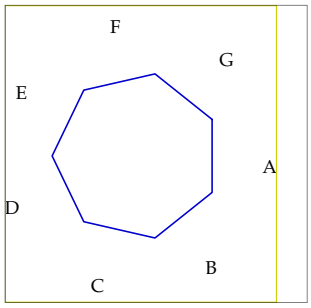
[SEVEN,B,RRN]

RRT



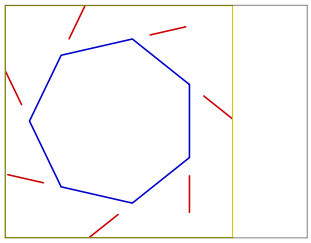

[SEVEN,B,RRT] [A,B,C,D,E,F,G]

RRZ



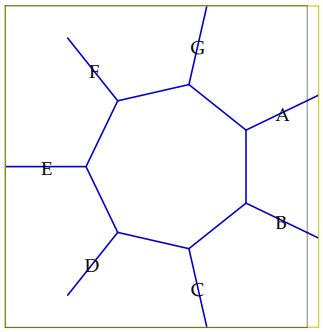

[SEVEN,B,RRZ] [A,B,C,D,E,F,G]

RSR



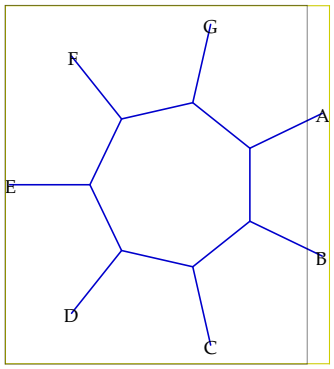
[SEVEN,B,rulecolor,RSR] [red]

RT



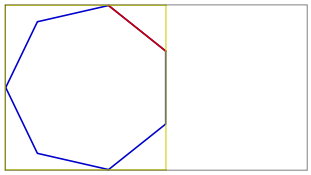

[SEVEN,B,R,RT] [A,B,C,D,E,F,G]

RZ

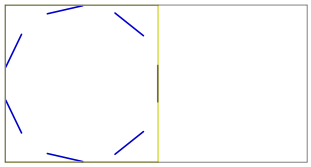


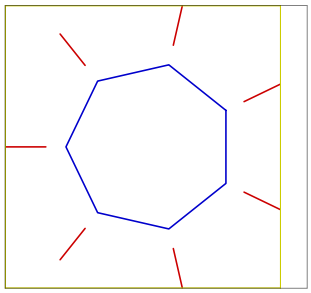

[SEVEN,B,R,RZ] [A,B,C,D,E,F,G]

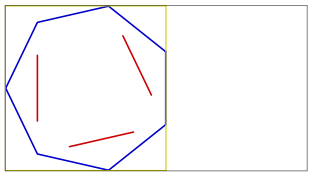
S

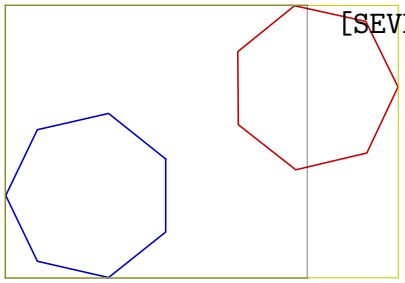


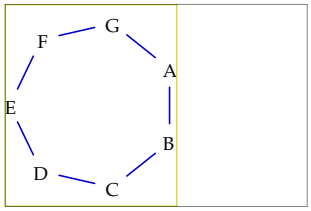
[SEVEN,B,rulecolor,S] [red]

SB  [SEVEN,SB]

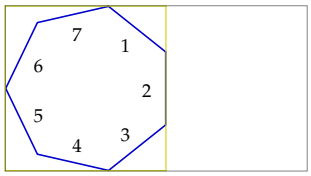
SR  [SEVEN,B,rulecolor,SR] [red]

SS  [SEVEN,B,rulecolor,SS135] [red]

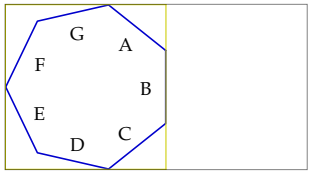
SUB  [SEVEN,B,SUB,rulecolor,SEVEN,B] [red]

Z 

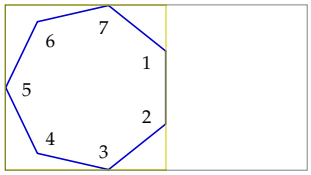
[SEVEN,SB,Z] [A,B,C,D,E,F,G]

ZLN 

[SEVEN,B,ZLN] [A,B,C,D,E,F,G]

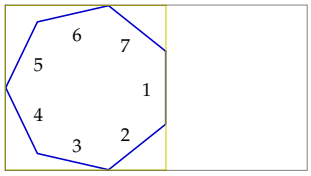
ZLT 

[SEVEN,B,ZLT] [A,B,C,D,E,F,G]

ZN 

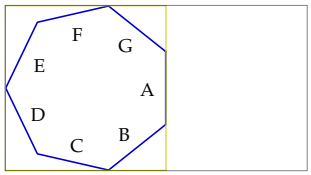
[SEVEN,B,ZN] [A,B,C,D,E,F,G]

ZRN



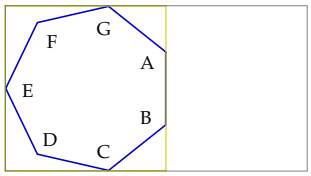

[SEVEN,B,ZRN] [A,B,C,D,E,F,G]

ZRT



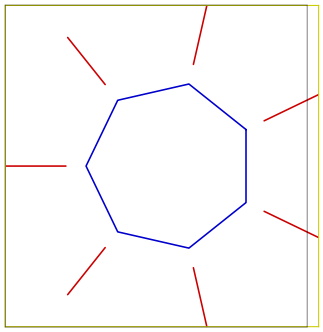

[SEVEN,B,ZRT] [A,B,C,D,E,F,G]

ZT



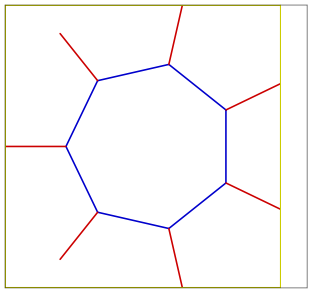

[SEVEN,B,ZT] [A,B,C,D,E,F,G]

-R



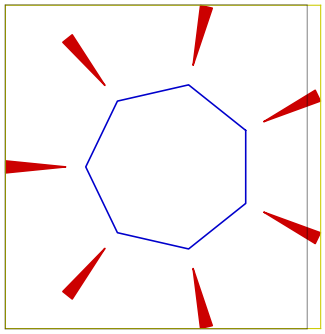
[SEVEN,B,rulecolor,-R] [red]

+R

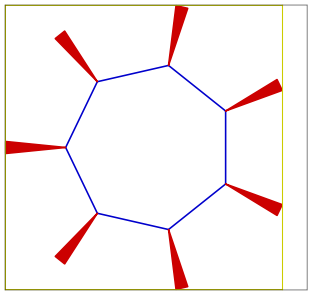


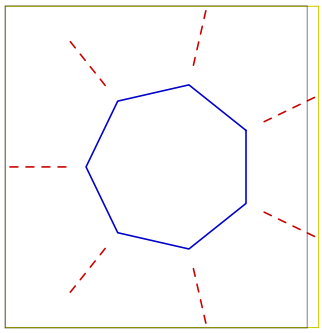
[SEVEN,B,rulecolor,+R] [red]

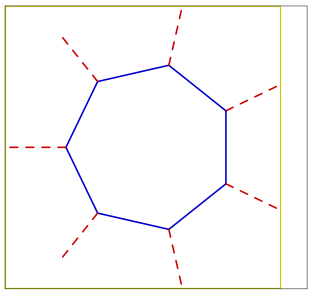
-RB

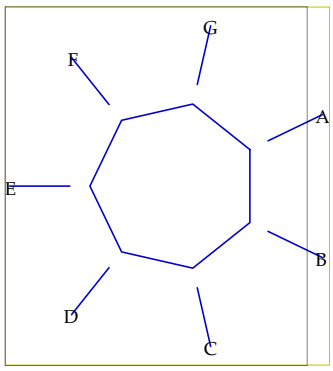


[SEVEN,B,rulecolor,-RB] [red]

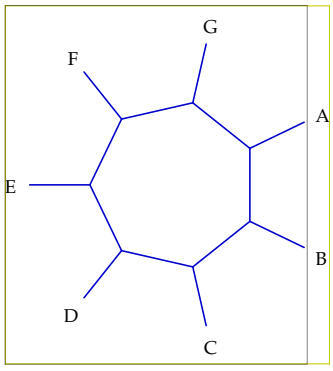
+RB  [SEVEN,B,rulecolor,+RB] [red]

-RD  [SEVEN,B,rulecolor,-RD] [red]

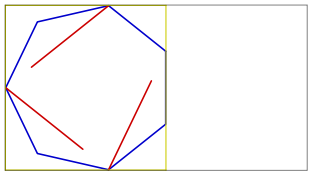
+RD  [SEVEN,B,rulecolor,+RD] [red]

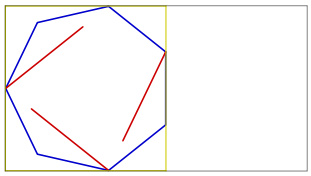
-RZ 

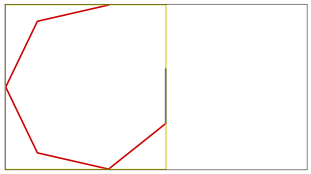
[SEVEN,B,-SR,-RZ] [A,B,C,D,E,F,G]

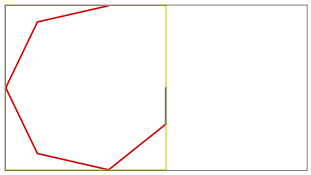
+RZ 

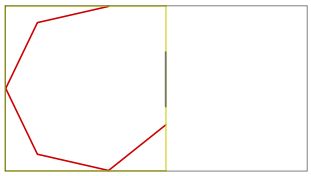
[SEVEN,B,+SR,+RZ] [A,B,C,D,E,F,G]

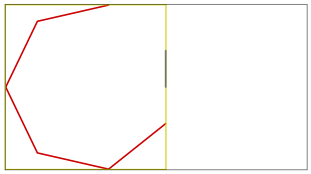
-S  [SEVEN,B,rulecolor,-S246] [red]

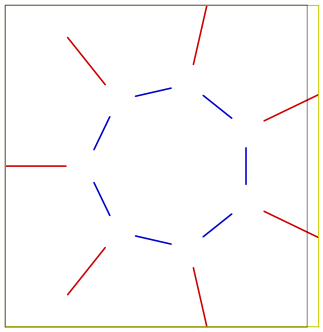
+S  [SEVEN,B,rulecolor,+S246] [red]

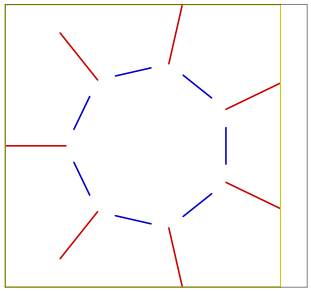
-SB  [SEVEN,-SB1,rulecolor,B2..6] [red]

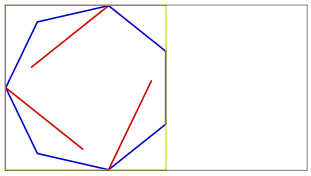
--SB  [SEVEN,--SB1,rulecolor,B2..6] [red]

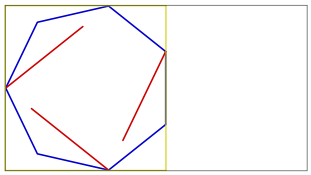
+SB  [SEVEN,+SB1,rulecolor,B2..6] [red]

++SB  [SEVEN,++SB1,rulecolor,B2..6] [red]

-SR  [SEVEN,SB,rulecolor,-SR] [red]

+SR  [SEVEN,SB,rulecolor,+SR] [red]

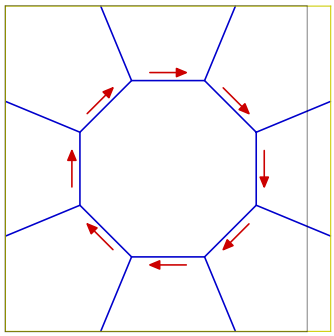
-SS  [SEVEN,B,rulecolor,-SS246] [red]

+SS  [SEVEN,B,rulecolor,+SS246] [red]

# 23 Eight

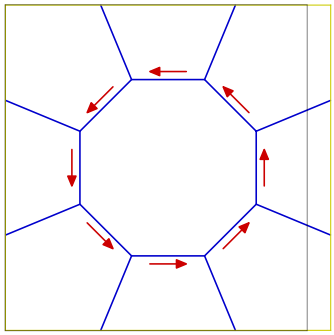
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ad | line | label | text | move | transform | rrz | text |
| adj | transform | line | line | number | number | rsr | line |
| arrow | line | lr | line | off | transform | rsub | transform |
| au | line | lrb | line | r | line | rt | text |
| b | line | lrbd | line | rb | line | rz | text |
| c | fixed | lrd | line | rbd | line | s | line |
| cc | line | lrh | line | rd | line | sb | line |
| ccd | line | lrn | number | rh | line | sr | line |
| cd | fixed | lrt | text | rm | transform | ss | line |
| crz | text | lrz | text | rn | number | sub | transform |
| dash | line | lsr | line | rot | transform | z | text |
| db | line | lsub | transform | rr | line | zln | number |
| diff | transform | mark | transform | rrb | line | zlt | text |
| dr | line | mid | line | rrbd | line | zn | number |
| eb | line | mids | line | rrd | line | zrn | number |
| ed | line | midz | text | rrh | line | zrt | text |
| er | line | mir | transform | rrn | number | zt | text |
| hb | line | mov | transform | rrt | text | | |

AD

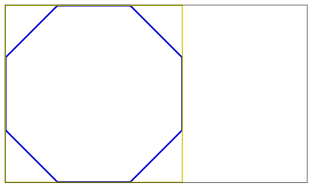


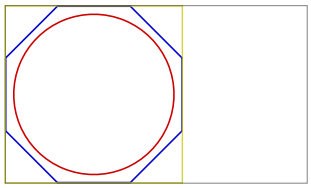
```
[EIGHT,B,R,rulecolor,AD] [red]
```

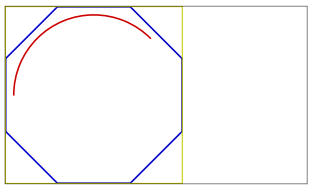
AU

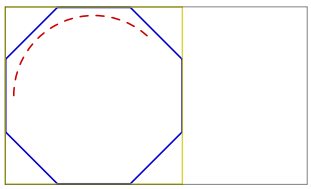


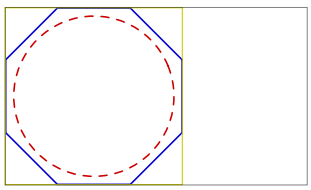
```
[EIGHT,B,R,rulecolor,AU] [red]
```

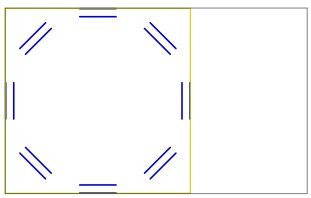
B  [EIGHT,B]

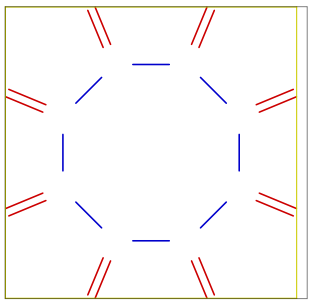
C  [EIGHT,B,rulecolor,C] [red]

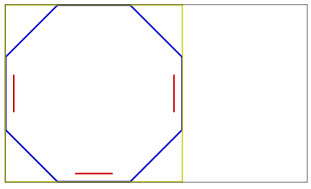
CC  [EIGHT,B,rulecolor,CC1..5] [red]

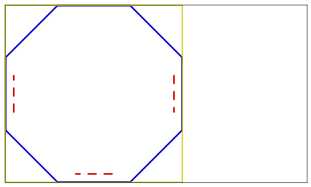
CCD  [EIGHT,B,rulecolor,CCD1..5] [red]

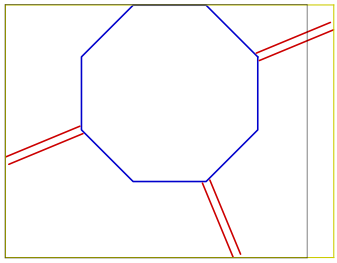
CD  [EIGHT,B,rulecolor,CD] [red]

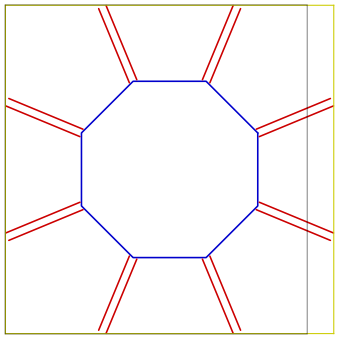
DB  [EIGHT,DB]

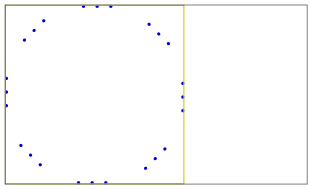
DR  [EIGHT,SB,rulecolor,DR] [red]

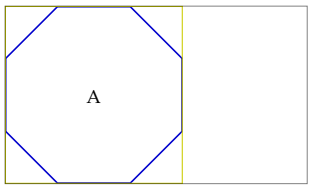
EB  [EIGHT,B,rulecolor,EB135] [red]

ED  [EIGHT,B,rulecolor,ED135] [red]

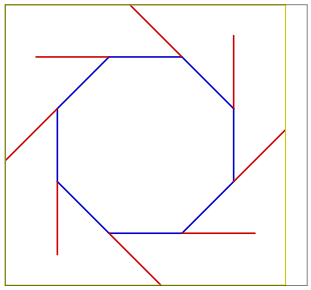
ER  [EIGHT,B,rulecolor,ER135] [red]

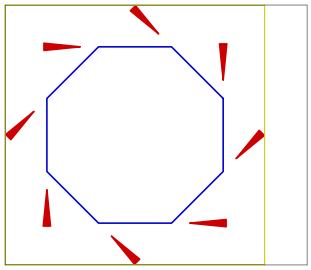
ER  [EIGHT,B,rulecolor,ER] [red]

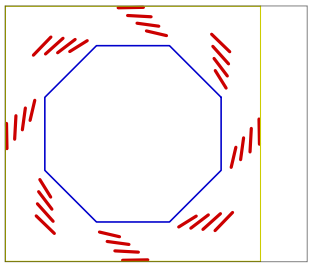
HB  [EIGHT,HB]

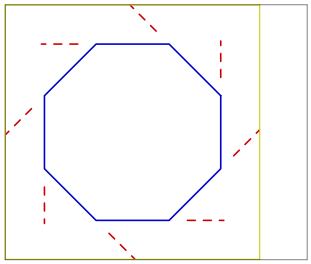
LABEL 

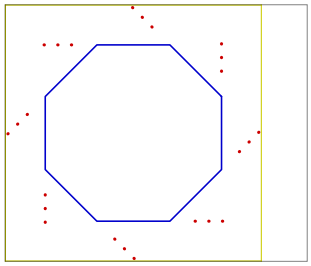
[EIGHT,B,LABEL] [A]

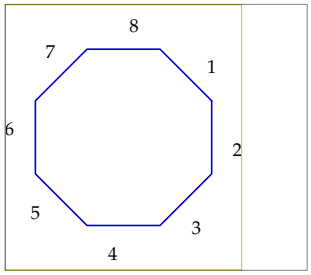
LR  [EIGHT,B,rulecolor,LR] [red]

LRB  [EIGHT,B,rulecolor,LRB] [red]

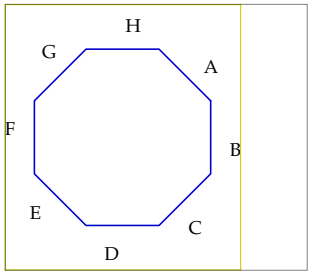
LRBD  [EIGHT,B,rulecolor,LRBD] [red]

LRD  [EIGHT,B,rulecolor,LRD] [red]

LRH  [EIGHT,B,rulecolor,LRH] [red]

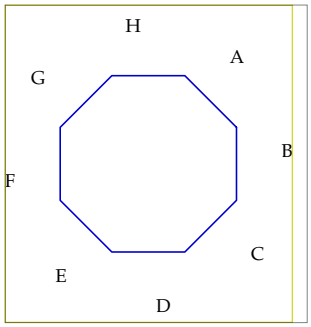
LRN 

[EIGHT,B,LRN]

LRT 

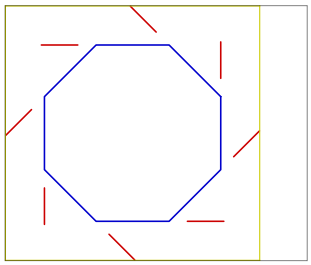
[EIGHT,B,LRT] [A,B,C,D,E,F,G,H]

LRZ



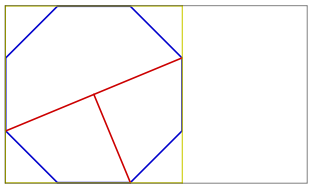

```
[EIGHT,B,LRZ] [A,B,C,D,E,F,G,H]
```

LSR



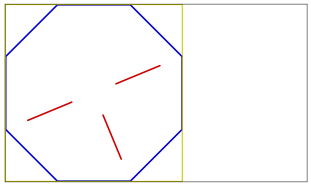
```
[EIGHT,B,rulecolor,LSR] [red]
```

MID



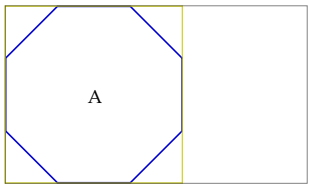
```
[EIGHT,B,rulecolor,MID135] [red]
```

MIDS



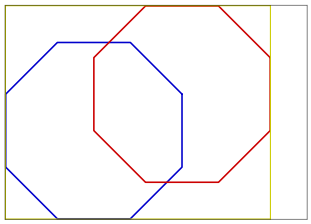
```
[EIGHT,B,rulecolor,MIDS135] [red]
```

MIDZ



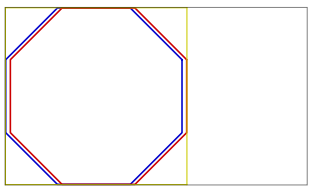

```
[EIGHT,B,MIDZ] [A]
```

MOV

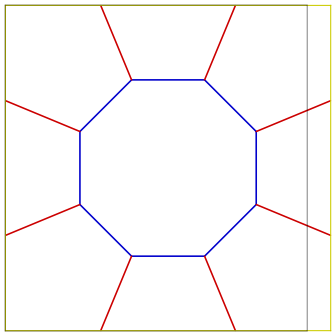


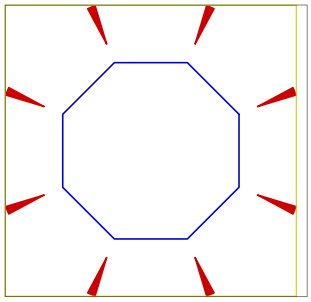
```
[EIGHT,B,MOV,rulecolor,EIGHT,B] [red]
```

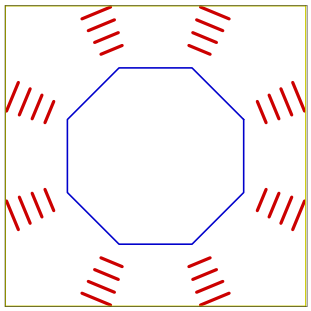
OFF

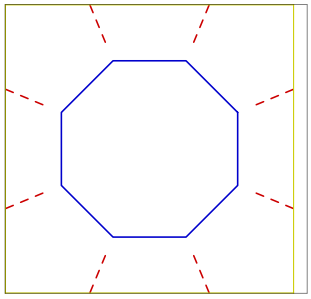


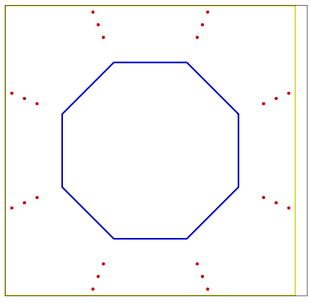
```
[EIGHT,B,OFF,rulecolor,EIGHT,B] [red]
```

R  [EIGHT,B,rulecolor,R] [red]

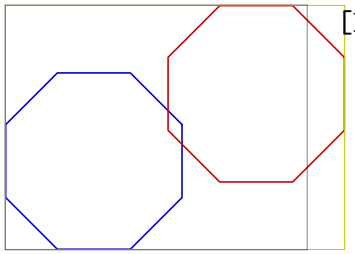
RB  [EIGHT,B,rulecolor,RB] [red]

RBD  [EIGHT,B,rulecolor,RBD] [red]

RD  [EIGHT,B,rulecolor,RD] [red]

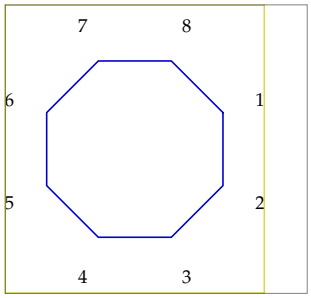
RH  [EIGHT,B,rulecolor,RH] [red]

RM



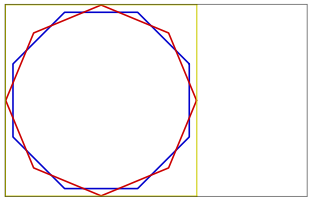
[EIGHT,B,RM,rulecolor,EIGHT,B] [red]

RN



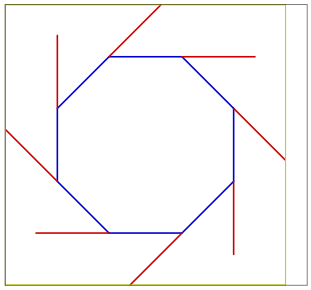

[EIGHT,B,RN]

ROT



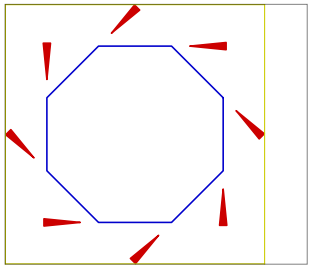
[EIGHT,B,ROT,rulecolor,EIGHT,B] [red]

RR



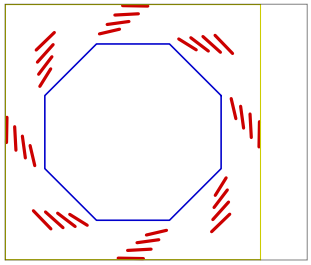
[EIGHT,B,rulecolor,RR] [red]

RRB

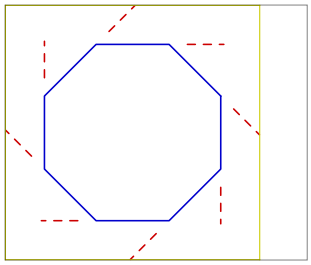


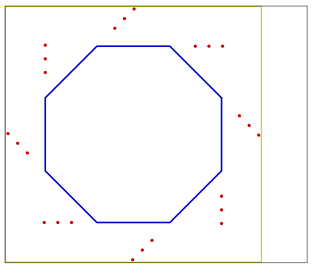
[EIGHT,B,rulecolor,RRB] [red]

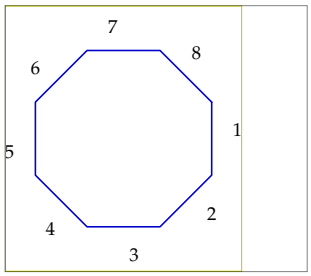
RRBD



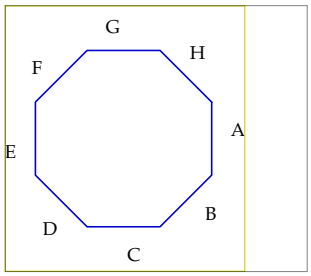
[EIGHT,B,rulecolor,RRBD] [red]

RRD  [EIGHT,B,rulecolor,RRD] [red]

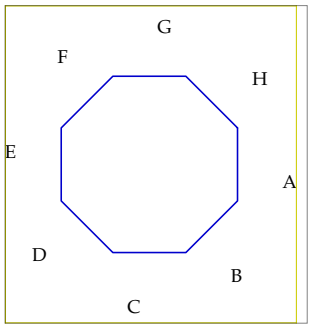
RRH  [EIGHT,B,rulecolor,RRH] [red]

RRN 

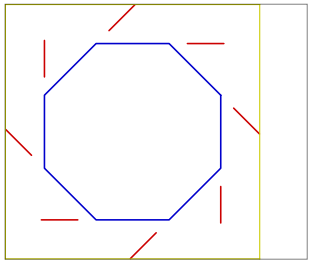
[EIGHT,B,RRN]

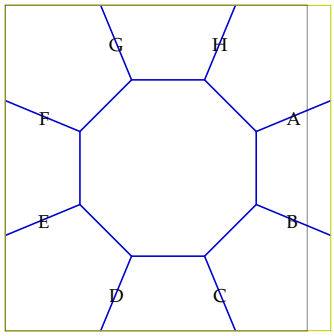
RRT 

[EIGHT,B,RRT] [A,B,C,D,E,F,G,H]

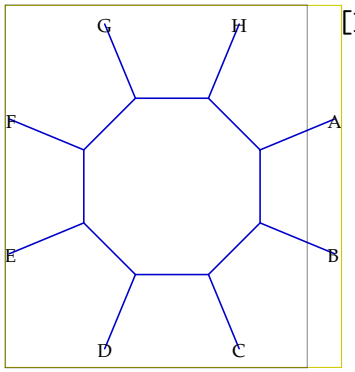
RRZ 

[EIGHT,B,RRZ] [A,B,C,D,E,F,G,H]

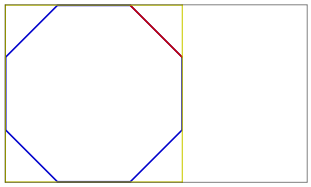
RSR  [EIGHT,B,rulecolor,RSR] [red]

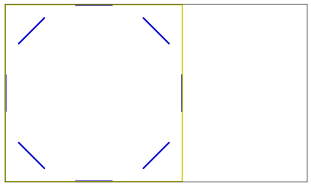
RT 

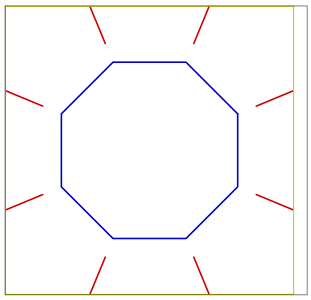
[EIGHT,B,R,RT] [A,B,C,D,E,F,G,H]

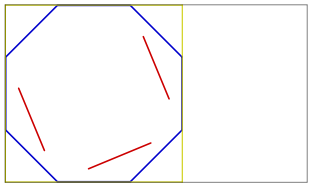
RZ 

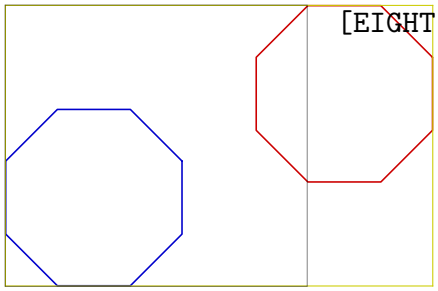
[EIGHT,B,R,RZ] [A,B,C,D,E,F,G,H]

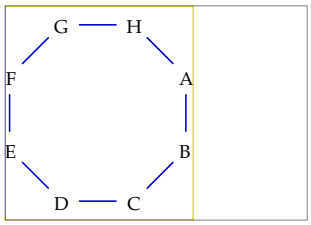
S 
[EIGHT,B,rulecolor,S] [red]

SB 
[EIGHT,SB]

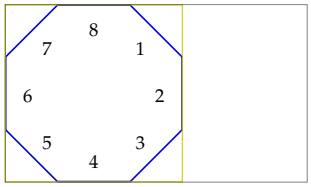
SR 
[EIGHT,B,rulecolor,SR] [red]

SS 
[EIGHT,B,rulecolor,SS135] [red]

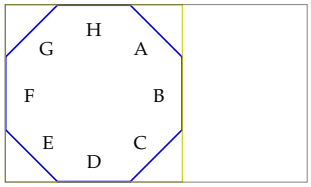
SUB  [EIGHT,B,SUB,rulecolor,EIGHT,B] [red]

Z 

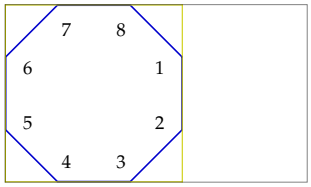
[EIGHT,SB,Z] [A,B,C,D,E,F,G,H]

ZLN 

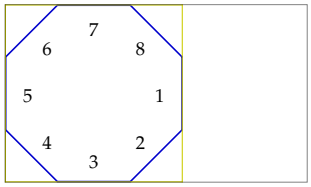
[EIGHT,B,ZLN] [A,B,C,D,E,F,G,H]

ZLT 

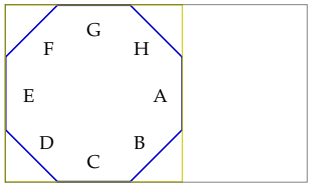
[EIGHT,B,ZLT] [A,B,C,D,E,F,G,H]

ZN 

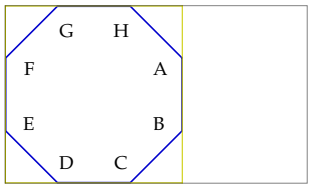
[EIGHT,B,ZN] [A,B,C,D,E,F,G,H]

ZRN 

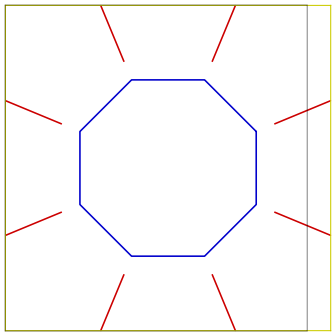
[EIGHT,B,ZRN] [A,B,C,D,E,F,G,H]

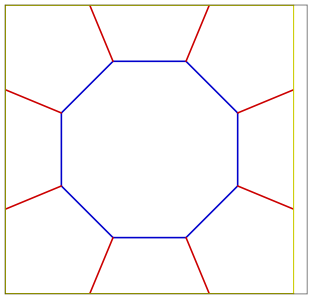
ZRT 

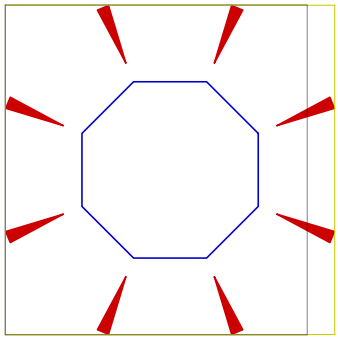
[EIGHT,B,ZRT] [A,B,C,D,E,F,G,H]

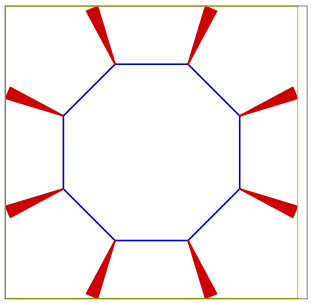
ZT 

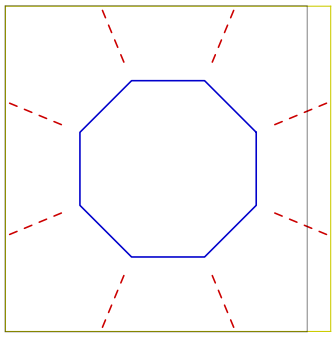
[EIGHT,B,ZT] [A,B,C,D,E,F,G,H]

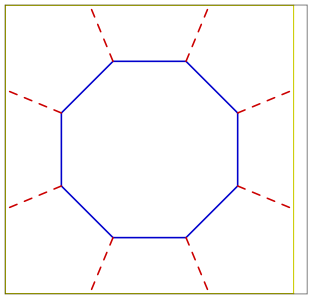
-R  [EIGHT,B,rulecolor,-R] [red]

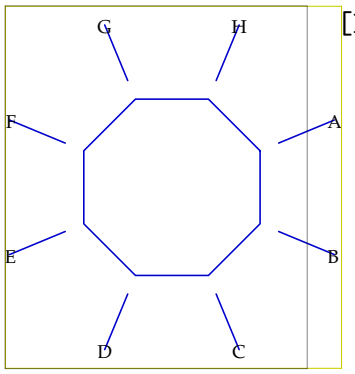
+R  [EIGHT,B,rulecolor,+R] [red]

-RB  [EIGHT,B,rulecolor,-RB] [red]

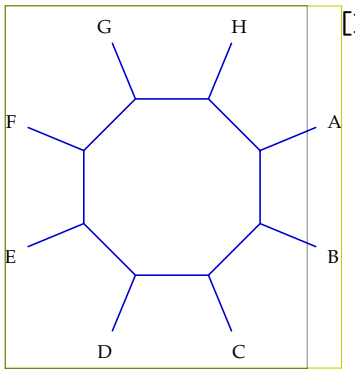
+RB  [EIGHT,B,rulecolor,+RB] [red]

-RD  [EIGHT,B,rulecolor,-RD] [red]

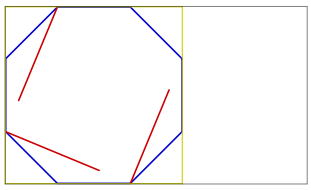
+RD  [EIGHT,B,rulecolor,+RD] [red]

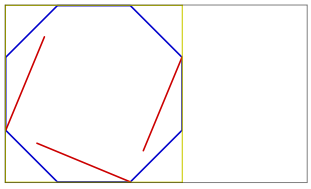
-RZ 

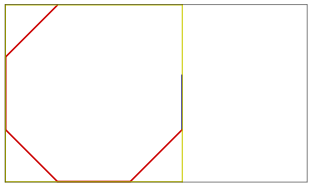
[EIGHT,B,-SR,-RZ] [A,B,C,D,E,F,G,H]

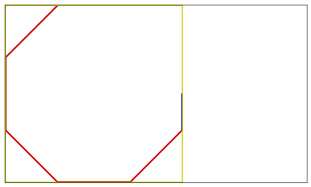
+RZ 

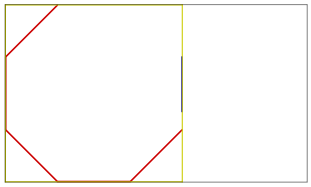
[EIGHT,B,+SR,+RZ] [A,B,C,D,E,F,G,H]

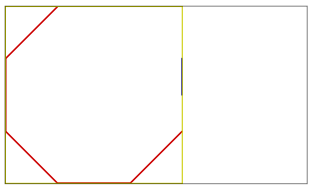
-S  [EIGHT,B,rulecolor,-S246] [red]

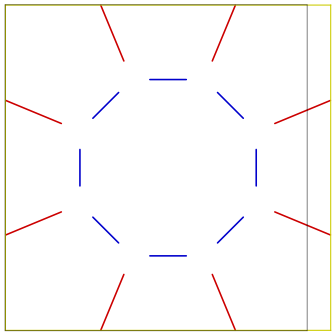
+S  [EIGHT,B,rulecolor,+S246] [red]

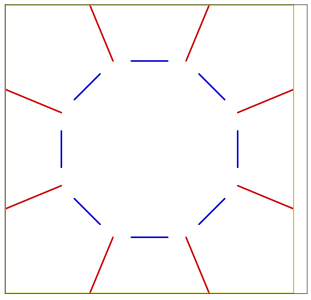
-SB  [EIGHT,-SB1,rulecolor,B2..6] [red]

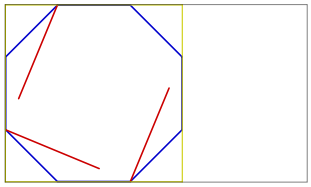
--SB  [EIGHT,--SB1,rulecolor,B2..6] [red]

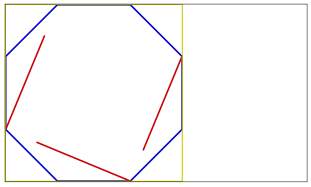
+SB  [EIGHT,+SB1,rulecolor,B2..6] [red]

++SB  [EIGHT,++SB1,rulecolor,B2..6] [red]

-SR  [EIGHT,SB,rulecolor,-SR] [red]

+SR  [EIGHT,SB,rulecolor,+SR] [red]

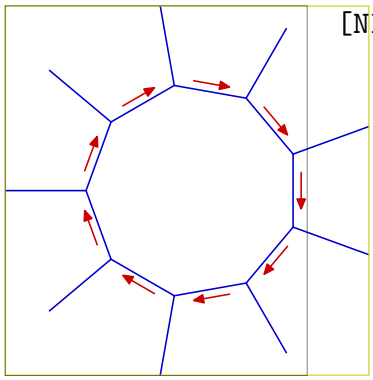
-SS  [EIGHT,B,rulecolor,-SS246] [red]

+SS  [EIGHT,B,rulecolor,+SS246] [red]

# 24 Nine

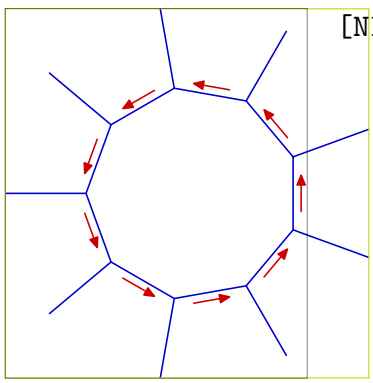
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ad | line | label | text | move | transform | rrz | text |
| adj | transform | line | line | number | number | rsr | line |
| arrow | line | lr | line | off | transform | rsub | transform |
| au | line | lrb | line | r | line | rt | text |
| b | line | lrbd | line | rb | line | rz | text |
| c | fixed | lrd | line | rbd | line | s | line |
| cc | line | lrh | line | rd | line | sb | line |
| ccd | line | lrn | number | rh | line | sr | line |
| cd | fixed | lrt | text | rm | transform | ss | line |
| crz | text | lrz | text | rn | number | sub | transform |
| dash | line | lsr | line | rot | transform | z | text |
| db | line | lsub | transform | rr | line | zln | number |
| diff | transform | mark | transform | rrb | line | zlt | text |
| dr | line | mid | line | rrbd | line | zn | number |
| eb | line | mids | line | rrd | line | zrn | number |
| ed | line | midz | text | rrh | line | zrt | text |
| er | line | mir | transform | rrn | number | zt | text |
| hb | line | mov | transform | rrt | text | | |

AD

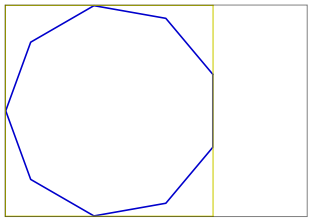


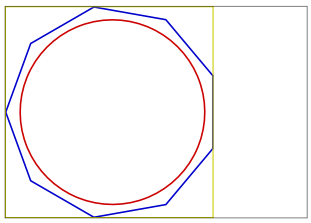
[NINE,B,R,rulecolor,AD] [red]

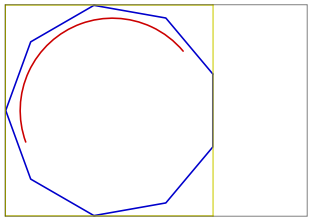
AU

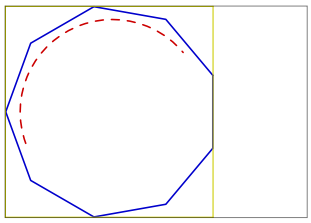


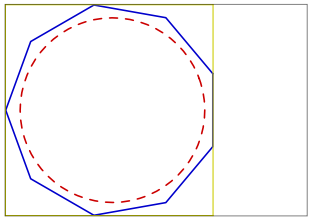
[NINE,B,R,rulecolor,AU] [red]

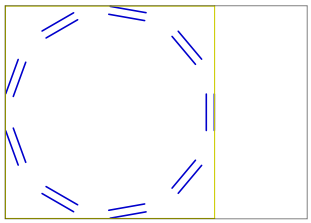
B  [NINE,B]

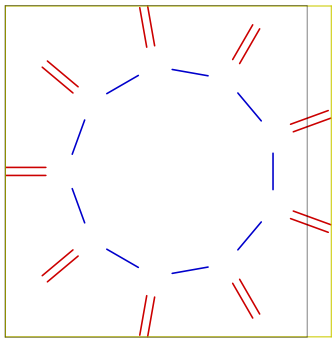
C  [NINE,B,rulecolor,C] [red]

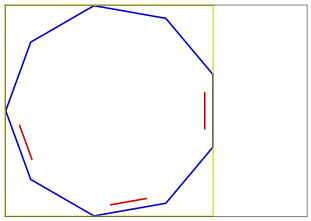
CC  [NINE,B,rulecolor,CC1..5] [red]

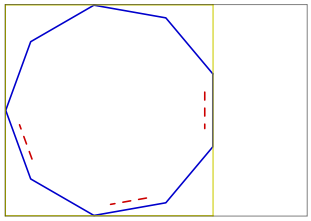
CCD  [NINE,B,rulecolor,CCD1..5] [red]

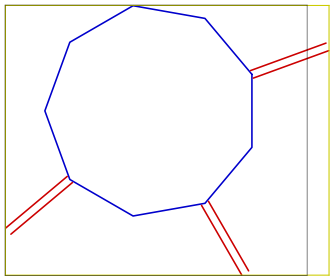
CD  [NINE,B,rulecolor,CD] [red]

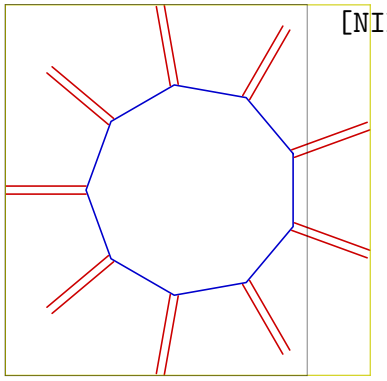
DB  [NINE,DB]

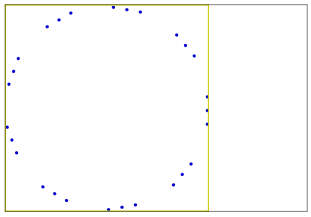
DR  [NINE,SB,rulecolor,DR] [red]

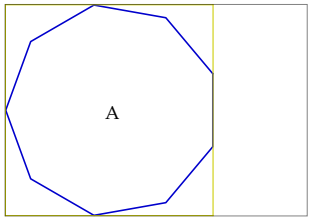
EB  [NINE,B,rulecolor,EB135] [red]

ED  [NINE,B,rulecolor,ED135] [red]

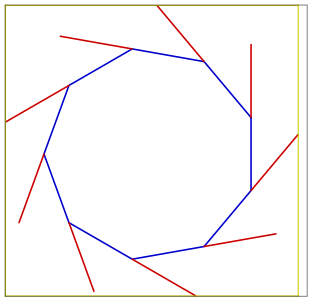
ER  [NINE,B,rulecolor,ER135] [red]

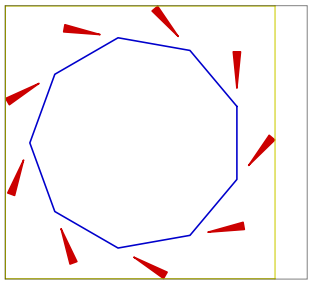
ER  [NINE,B,rulecolor,ER] [red]

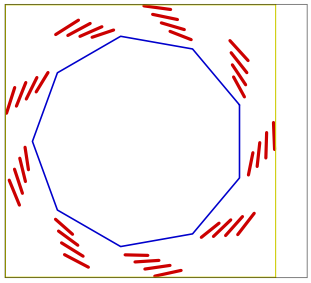
HB  [NINE,HB]

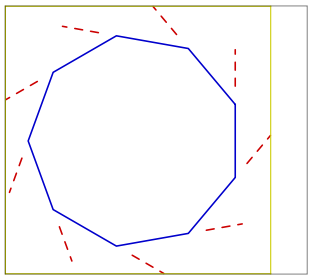
LABEL 

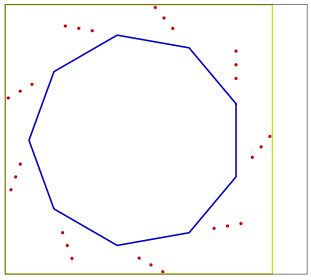
[NINE,B,LABEL] [A]

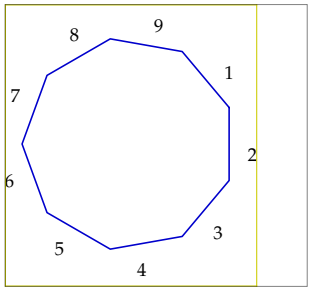
LR  [NINE,B,rulecolor,LR] [red]

LRB  [NINE,B,rulecolor,LRB] [red]

LRBD  [NINE,B,rulecolor,LRBD] [red]

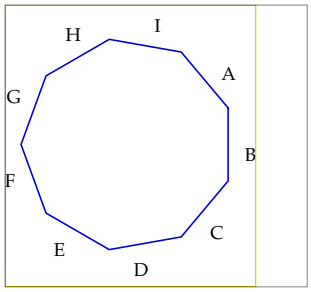
LRD  [NINE,B,rulecolor,LRD] [red]

LRH  [NINE,B,rulecolor,LRH] [red]

LRN 

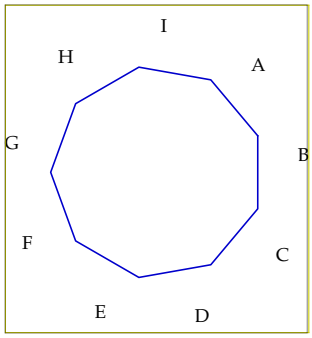
[NINE,B,LRN]

LRT



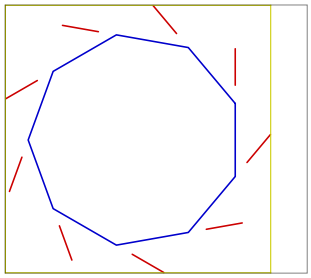

[NINE,B,LRT] [A,B,C,D,E,F,G,H,I]

LRZ



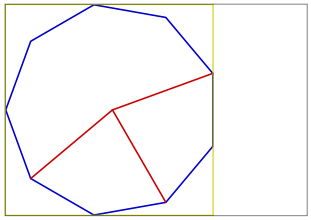

[NINE,B,LRZ] [A,B,C,D,E,F,G,H,I]

LSR



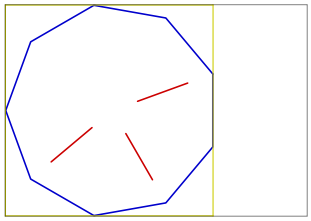
[NINE,B,rulecolor,LSR] [red]

MID



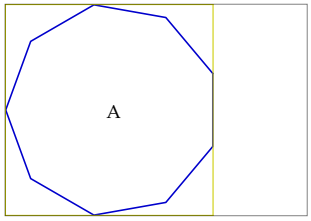
[NINE,B,rulecolor,MID135] [red]

MIDS



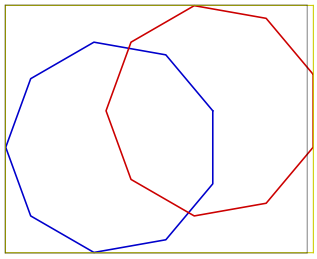
[NINE,B,rulecolor,MIDS135] [red]

MIDZ



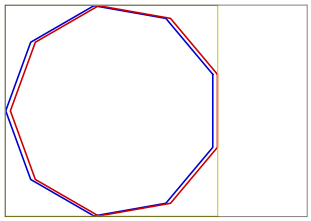

[NINE,B,MIDZ] [A]

MOV



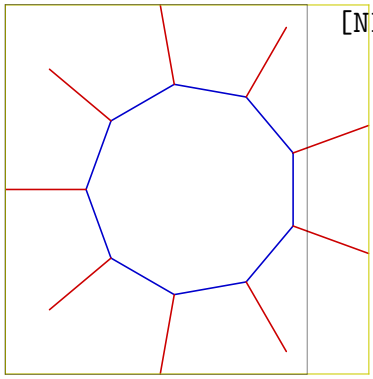
[NINE,B,MOV,rulecolor,NINE,B] [red]

OFF



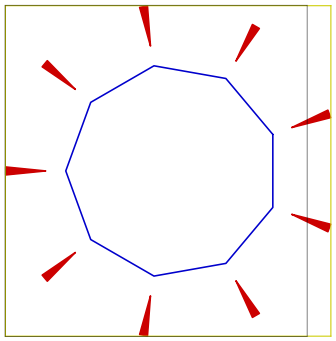
[NINE,B,OFF,rulecolor,NINE,B] [red]

R



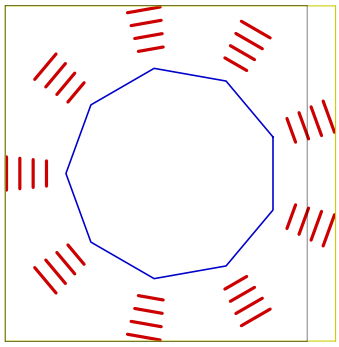
[NINE,B,rulecolor,R] [red]

RB



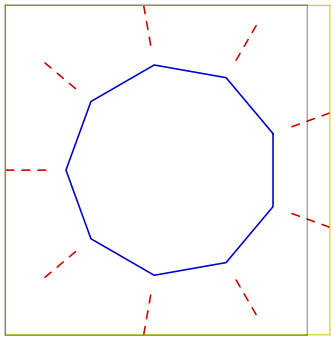
[NINE,B,rulecolor,RB] [red]

RBD



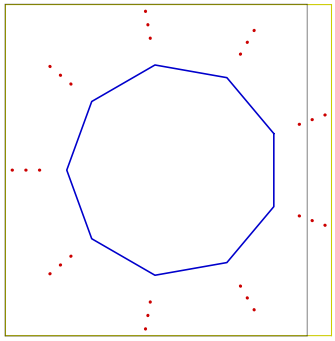
[NINE,B,rulecolor,RBD] [red]

RD



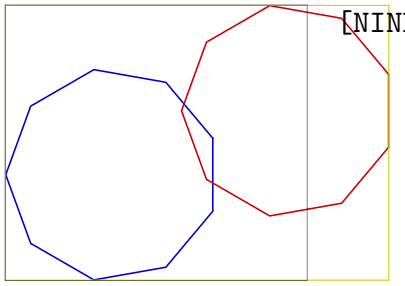
[NINE,B,rulecolor,RD] [red]

RH



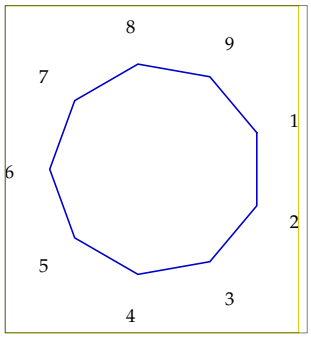
[NINE,B,rulecolor,RH] [red]

RM



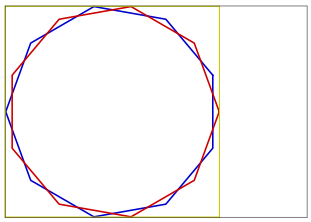
[NINE,B,RM,rulecolor,NINE,B] [red]

RN

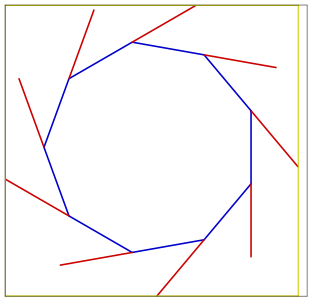


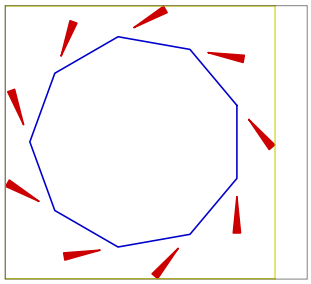

[NINE,B,RN]

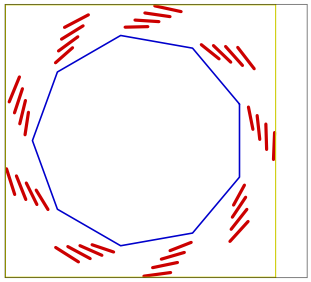
ROT

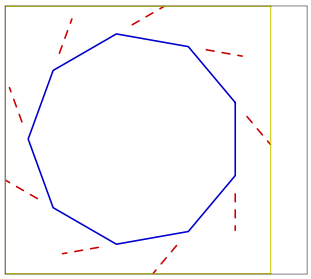


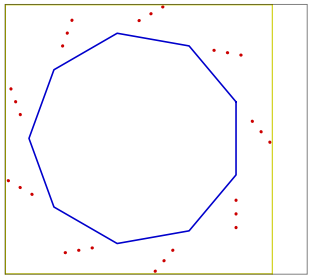
[NINE,B,ROT,rulecolor,NINE,B] [red]

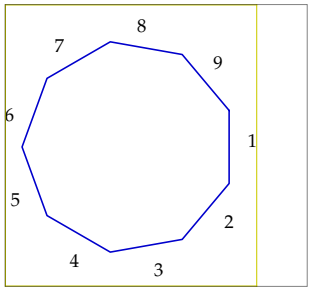
RR  [NINE,B,rulecolor,RR] [red]

RRB  [NINE,B,rulecolor,RRB] [red]

RRBD  [NINE,B,rulecolor,RRBD] [red]

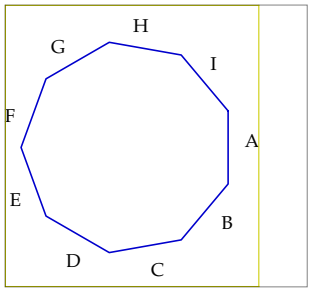
RRD  [NINE,B,rulecolor,RRD] [red]

RRH  [NINE,B,rulecolor,RRH] [red]

RRN 

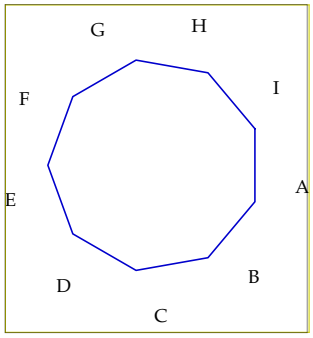
[NINE,B,RRN]

RRT



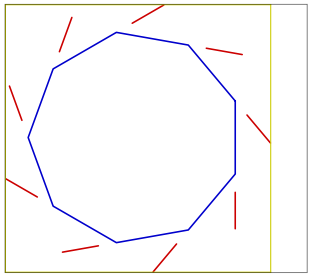

[NINE,B,RRT] [A,B,C,D,E,F,G,H,I]

RRZ



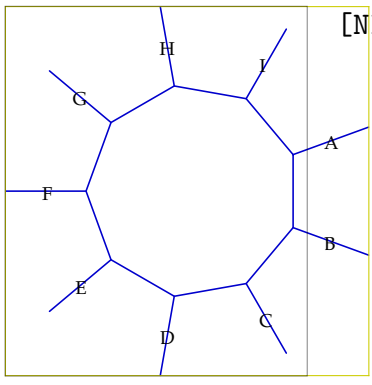

[NINE,B,RRZ] [A,B,C,D,E,F,G,H,I]

RSR



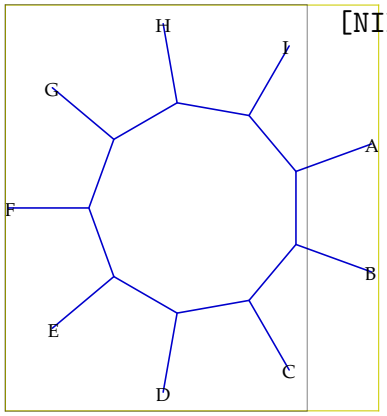
[NINE,B,rulecolor,RSR] [red]

RT



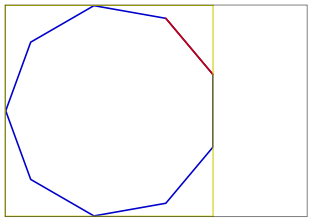

[NINE,B,R,RT] [A,B,C,D,E,F,G,H,I]

RZ



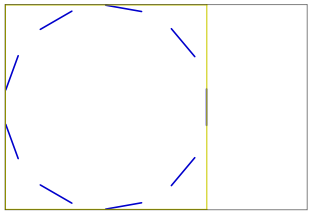

[NINE,B,R,RZ] [A,B,C,D,E,F,G,H,I]

S



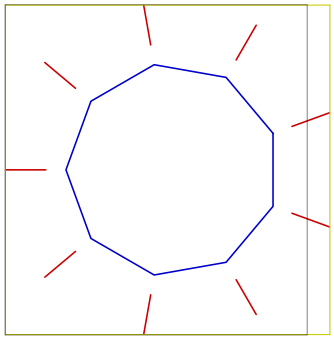
[NINE,B,rulecolor,S] [red]

SB



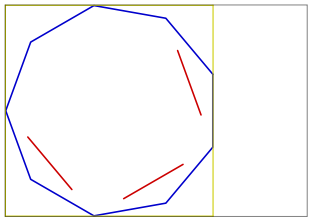
[NINE,SB]

SR



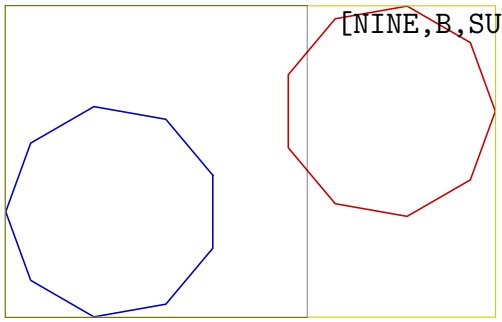
[NINE,B,rulecolor,SR] [red]

SS



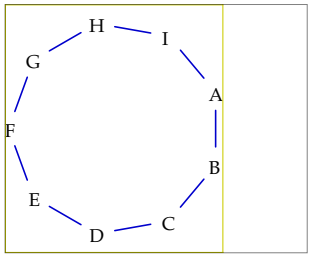
[NINE,B,rulecolor,SS135] [red]

SUB

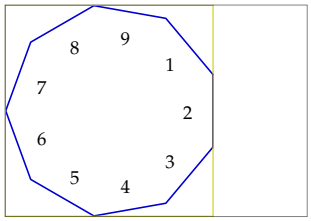


[NINE,B,SUB,rulecolor,NINE,B] [red]

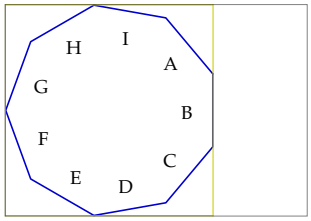
Z



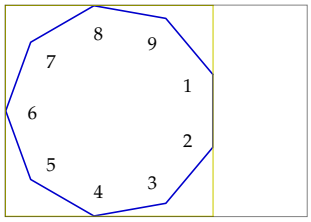

[NINE,SB,Z] [A,B,C,D,E,F,G,H,I]

ZLN 

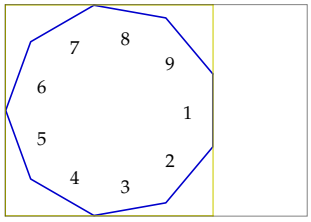
[NINE,B,ZLN] [A,B,C,D,E,F,G,H,I]

ZLT 

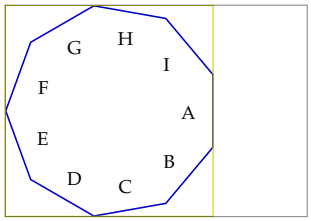
[NINE,B,ZLT] [A,B,C,D,E,F,G,H,I]

ZN 

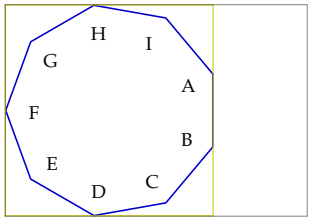
[NINE,B,ZN] [A,B,C,D,E,F,G,H,I]

ZRN 

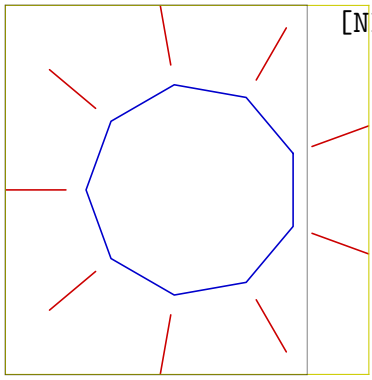
[NINE,B,ZRN] [A,B,C,D,E,F,G,H,I]

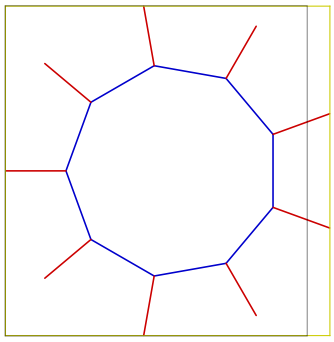
ZRT 

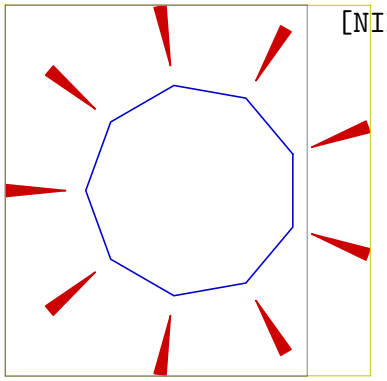
[NINE,B,ZRT] [A,B,C,D,E,F,G,H,I]

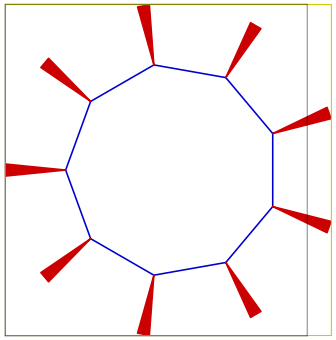
ZT 

[NINE,B,ZT] [A,B,C,D,E,F,G,H,I]

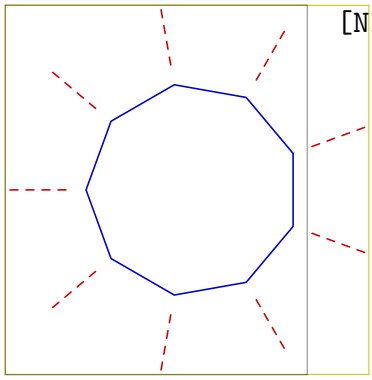
-R  [NINE,B,rulecolor,-R] [red]

+R  [NINE,B,rulecolor,+R] [red]

-RB  [NINE,B,rulecolor,-RB] [red]

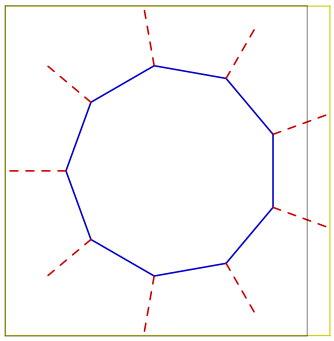
+RB  [NINE,B,rulecolor,+RB] [red]

-RD



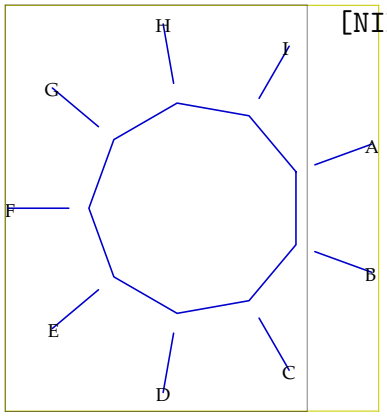
[NINE,B,rulecolor,-RD] [red]

+RD



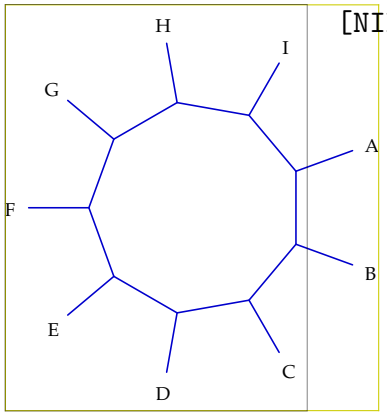
[NINE,B,rulecolor,+RD] [red]

-RZ

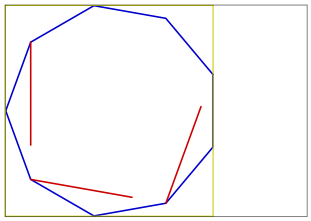


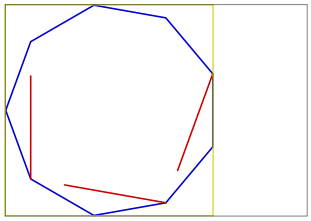

[NINE,B,-SR,-RZ] [A,B,C,D,E,F,G,H,I]

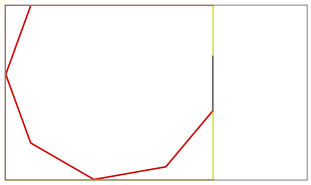
+RZ

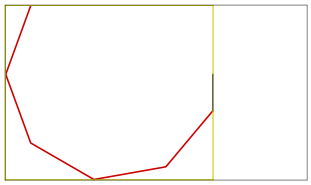


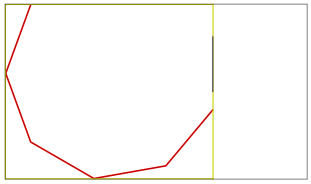

[NINE,B,+SR,+RZ] [A,B,C,D,E,F,G,H,I]

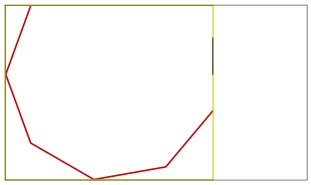
-S  [NINE,B,rulecolor,-S246] [red]

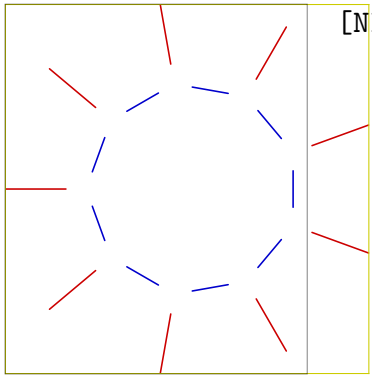
+S  [NINE,B,rulecolor,+S246] [red]

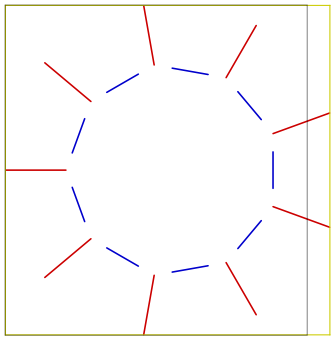
-SB  [NINE,-SB1,rulecolor,B2..6] [red]

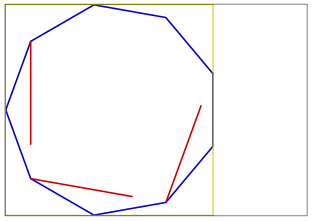
--SB  [NINE,--SB1,rulecolor,B2..6] [red]

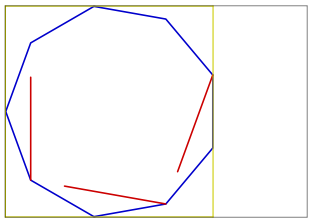
+SB  [NINE,+SB1,rulecolor,B2..6] [red]

++SB  [NINE,++SB1,rulecolor,B2..6] [red]

-SR  [NINE,SB,rulecolor,-SR] [red]

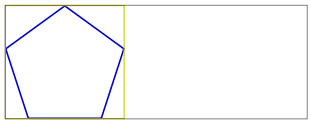
+SR  [NINE,SB,rulecolor,+SR] [red]

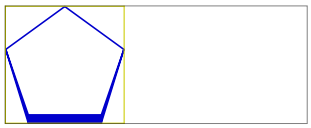
-SS  [NINE,B,rulecolor,-SS246] [red]

+SS  [NINE,B,rulecolor,+SS246] [red]

# 25 Five Front

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| adj | transform | lrb | line | r | line | rsr | line |
| arrow | line | lrbd | line | rb | line | rsub | transform |
| b | line | lrd | line | rbd | line | rt | text |
| bb | line | lrh | line | rd | line | rz | text |
| c | fixed | lrn | number | rh | line | sb | line |
| cc | line | lrt | text | rn | number | sr | line |
| ccd | line | lrz | text | rr | line | sub | transform |
| cd | fixed | lsr | line | rrb | line | z | text |
| dash | line | lsub | transform | rrbd | line | zln | number |
| diff | transform | mark | transform | rrd | line | zlt | text |
| eb | line | mov | transform | rrh | line | zn | number |
| label | text | move | transform | rrn | number | zrn | number |
| line | line | number | number | rrt | text | zrt | text |
| lr | line | off | transform | rrz | text | zt | text |

B  [FIVE,FRONT,B]

BB  [FIVE,FRONT,BB]

C  [FIVE,FRONT,C]

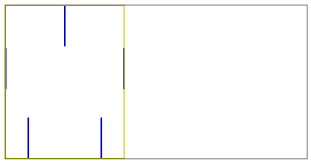
CC  [FIVE,FRONT,CC1]

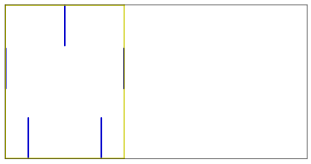
CCD  [FIVE,FRONT,CCD1]

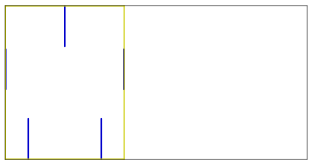
CD  [FIVE,FRONT,CD]

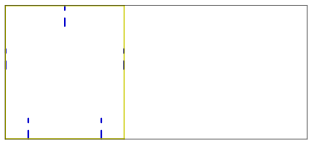
EB  [FIVE,FRONT,EB]

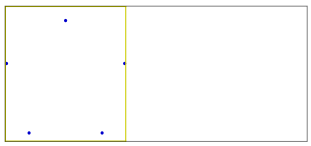
LINE  [FIVE,FRONT,LINE]

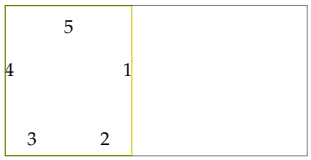
LR 
[FIVE,FRONT,LR]

LRB 
[FIVE,FRONT,LR]

LRBD 
[FIVE,FRONT,LR]

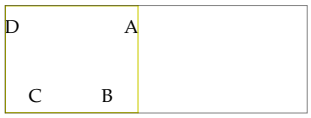
LRD 
[FIVE,FRONT,LRD]

LRH 
[FIVE,FRONT,LRH]

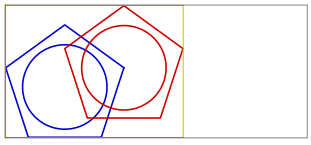
LRN 

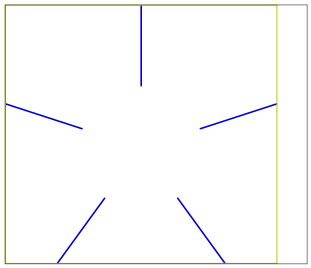
[FIVE,FRONT,LRN]

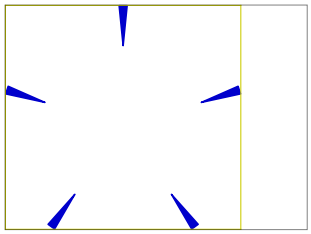
LRT
[FIVE,FRONT,LRT]

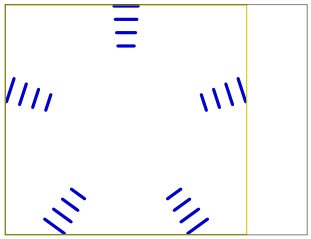
LRZ 

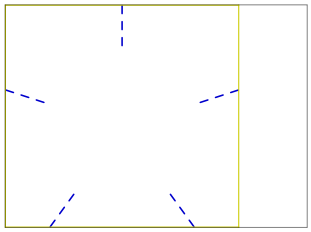
[FIVE,FRONT,LRZ] [A,B,C,D]

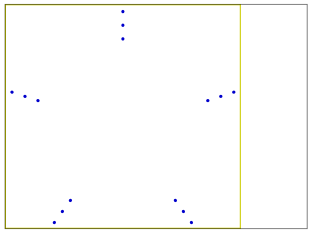
MOV 
[FIVE,FRONT,B,C,MOV1,rulecolor,FIVE,FRONT,B,C] [red]

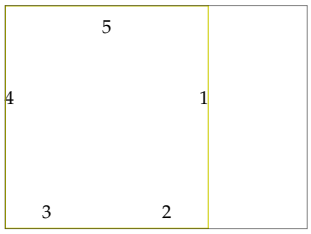
R 
[FIVE,FRONT,R]

RB  [FIVE,FRONT,RB]

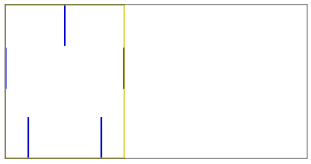
RBD  [FIVE,FRONT,RBD]

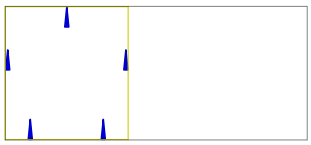
RD  [FIVE,FRONT,RD]

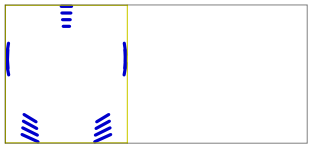
RH  [FIVE,FRONT,RH]

RN 

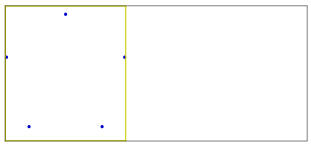
[FIVE,FRONT,RN]

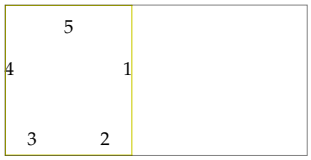
RR  [FIVE,FRONT,RR]

RRB  [FIVE,FRONT,RRB]

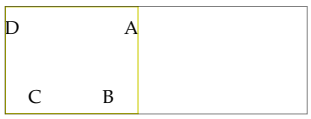
RRBD  [FIVE,FRONT,RRBD]

RRD  [FIVE,FRONT,RRD]

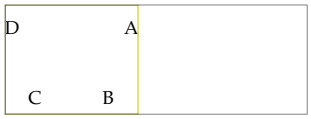
RRH  [FIVE,FRONT,RRH]

RRN 

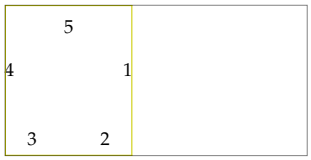
[FIVE,FRONT,RRN]

RRT 

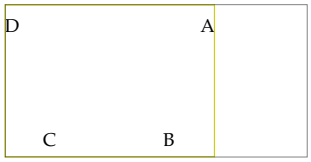
[FIVE,FRONT,RRT] [A,B,C,D]

RRZ 

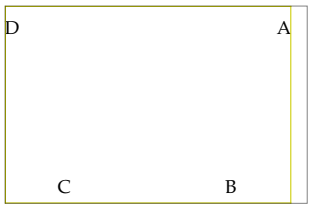
[FIVE,FRONT,RRZ] [A,B,C,D]

RSR 

[FIVE,FRONT,RRN]

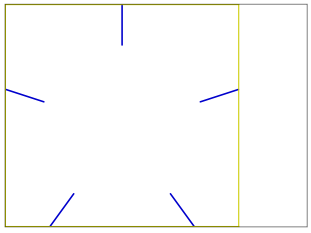
RT 

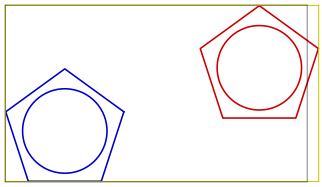
[FIVE,FRONT,RT] [A,B,C,D]

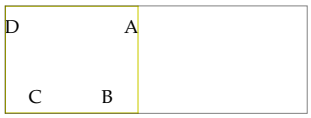
RZ 

[FIVE,FRONT,RZ] [A,B,C,D]

SB  [FIVE,FRONT,SB]

SR  [FIVE,FRONT,SR]

SUB  [FIVE,FRONT,B,C,SUB1,rulecolor,FIVE,FRONT,B,C] [red]

Z 

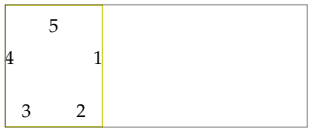
[FIVE,FRONT,Z] [A,B,C,D]

ZLN 

[FIVE,FRONT,ZLN]

ZLT 

[FIVE,FRONT,ZLT] [A,B,C,D]

ZN 

[FIVE,FRONT,ZN]

ZRN 

[FIVE,FRONT,ZRN]

ZRT 

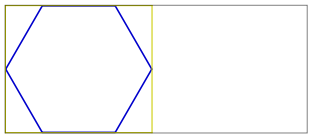
[FIVE,FRONT,ZRT] [A,B,C,D]

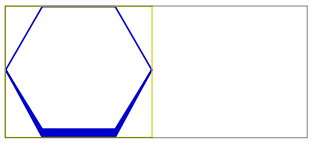
ZT 

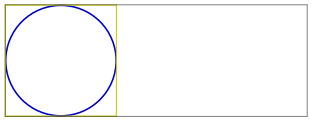
[FIVE,FRONT,ZT] [A,B,C,D]

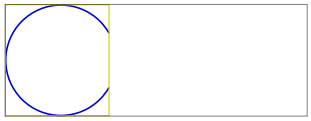
# 26 Six Front

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| adj | transform | lrb | line | r | line | rsr | line |
| arrow | line | lrbd | line | rb | line | rsub | transform |
| b | line | lrd | line | rbd | line | rt | text |
| bb | line | lrh | line | rd | line | rz | text |
| c | fixed | lrn | number | rh | line | sb | line |
| cc | line | lrt | text | rn | number | sr | line |
| ccd | line | lrz | text | rr | line | sub | transform |
| cd | fixed | lsr | line | rrb | line | z | text |
| dash | line | lsub | transform | rrbd | line | zln | number |
| diff | transform | mark | transform | rrd | line | zlt | text |
| eb | line | mov | transform | rrh | line | zn | number |
| label | text | move | transform | rrn | number | zrn | number |
| line | line | number | number | rrt | text | zrt | text |
| lr | line | off | transform | rrz | text | zt | text |

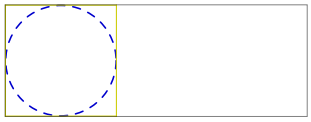
B  [SIX,FRONT,B]

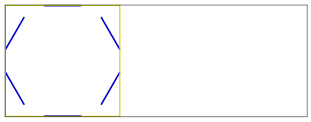
BB  [SIX,FRONT,BB]

C  [SIX,FRONT,C]

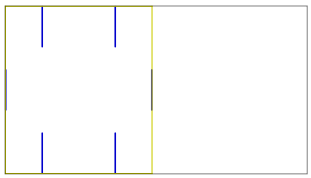
CC  [SIX,FRONT,CC1]

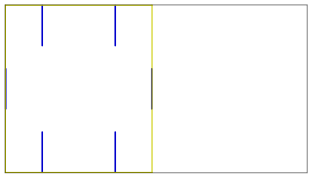
CCD  [SIX,FRONT,CCD1]

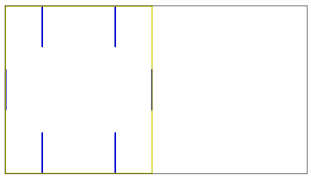
CD  [SIX,FRONT,CD]

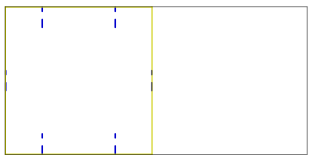
EB  [SIX,FRONT,EB]

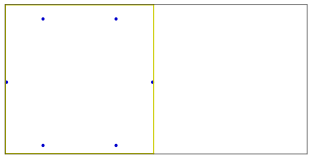
LINE [SIX,FRONT,LINE]

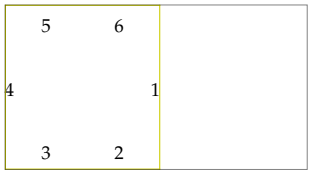
LR  [SIX,FRONT,LR]

LRB  [SIX,FRONT,LR]

LRBD  [SIX,FRONT,LR]

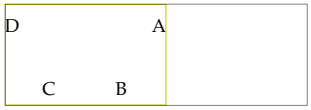
LRD  [SIX,FRONT,LRD]

LRH  [SIX,FRONT,LRH]

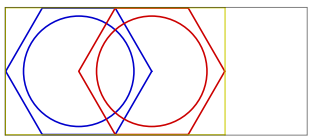
LRN 

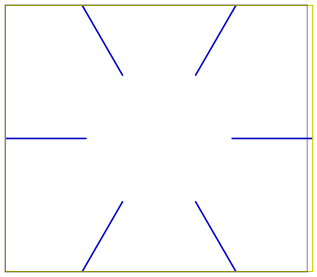
[SIX,FRONT,LRN]

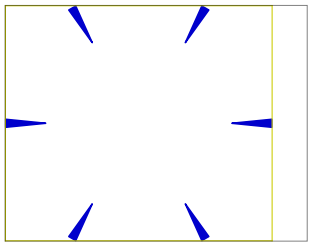
LRT [SIX,FRONT,LRT]

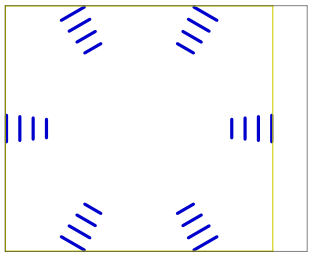
LRZ 

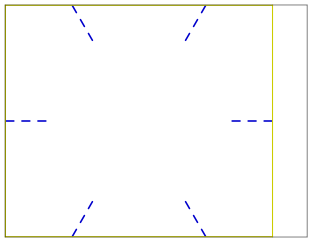
[SIX,FRONT,LRZ] [A,B,C,D]

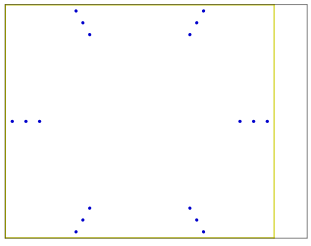
MOV  [SIX,FRONT,B,C,MOV1,rulecolor,SIX,FRONT,B,C] [red]

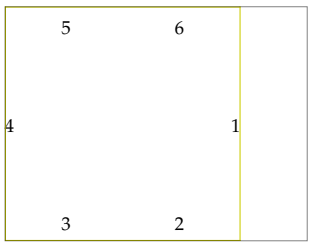
R  [SIX,FRONT,R]

RB  [SIX,FRONT,RB]

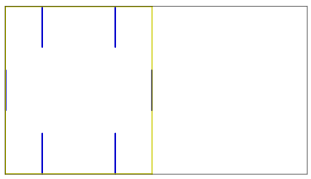
RBD  [SIX,FRONT,RBD]

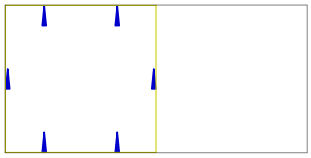
RD  [SIX,FRONT,RD]

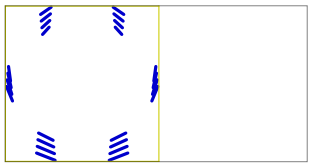
RH  [SIX,FRONT,RH]

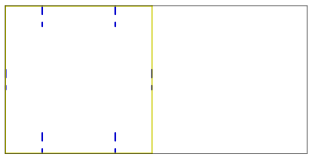
RN 

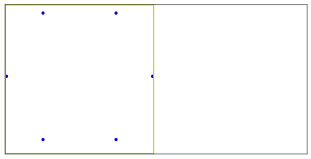
[SIX,FRONT,RN]

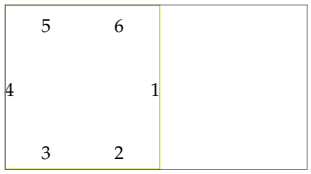
RR  [SIX,FRONT,RR]

RRB  [SIX,FRONT,RRB]

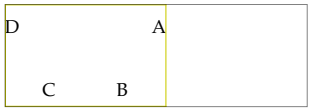
RRBD  [SIX,FRONT,RRBD]

RRD  [SIX,FRONT,RRD]

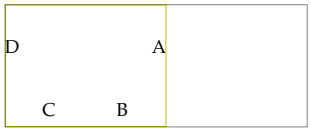
RRH  [SIX,FRONT,RRH]

RRN 

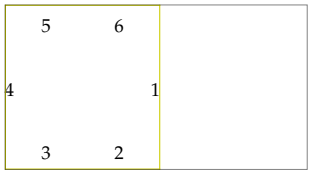
[SIX,FRONT,RRN]

RRT 

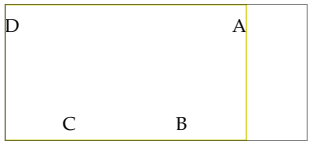
[SIX,FRONT,RRT] [A,B,C,D]

RRZ 

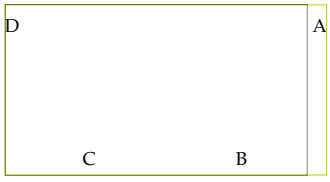
[SIX,FRONT,RRZ] [A,B,C,D]

RSR 

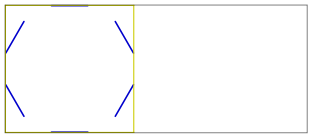
[SIX,FRONT,RRN]

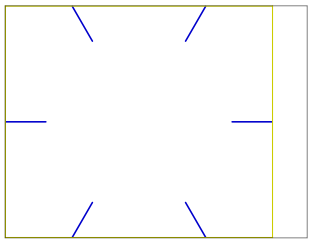
RT 

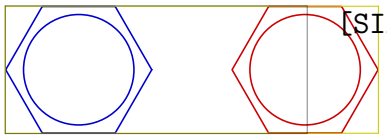
[SIX,FRONT,RT] [A,B,C,D]

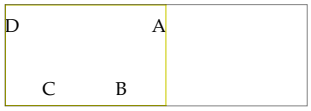
RZ 

[SIX,FRONT,RZ] [A,B,C,D]

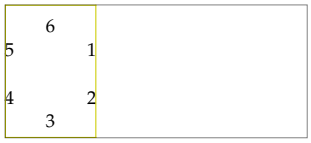
SB  [SIX,FRONT,SB]

SR  [SIX,FRONT,SR]

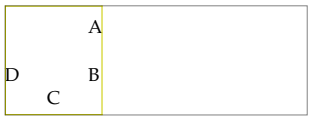
SUB  [SIX,FRONT,B,C,SUB1,rulecolor,SIX,FRONT,B,C] [red]

Z 

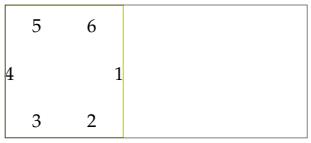
[SIX,FRONT,Z] [A,B,C,D]

ZLN 

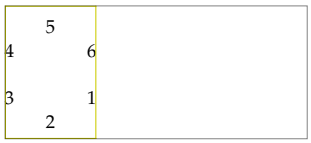
[SIX,FRONT,ZLN]

ZLT 

[SIX,FRONT,ZLT] [A,B,C,D]

ZN 

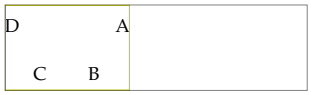
[SIX,FRONT,ZN]

ZRN 

[SIX,FRONT,ZRN]

ZRT 

[SIX,FRONT,ZRT] [A,B,C,D]

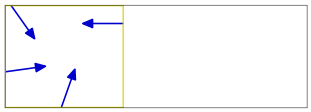
ZT 

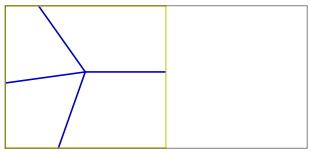
[SIX,FRONT,ZT] [A,B,C,D]

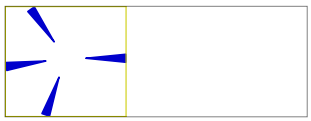
# 27 Carbon

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ad | line | dir | transform | move | transform | rt | text |
| adj | transform | dr | line | number | number | rz | text |
| arrow | line | ed | line | oe | line | sb | line |
| au | line | ep | line | off | transform | sd | line |
| b | line | es | line | r | line | sr | line |
| bb | line | et | line | rb | line | sub | transform |
| bd | line | hb | line | rbd | line | tb | line |
| bw | line | label | text | rd | line | z | text |
| c | fixed | ldb | line | rdb | line | zln | number |
| cc | line | ldd | line | rdd | line | zlt | text |
| ccd | line | line | line | rh | line | zn | number |
| cd | fixed | lrn | number | rm | transform | zrn | number |
| cz | text | lrt | text | rn | number | zrt | text |
| dash | line | mark | transform | rot | transform | zt | text |
| db | line | mir | transform | rrn | number | | |
| diff | transform | mov | transform | rrt | text | | |

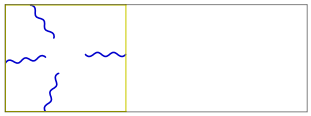
AD  [CARBON,AD]

AU  [CARBON,AU]

B  [CARBON,B]

BB  [CARBON,BB]

BD  [CARBON,BD]

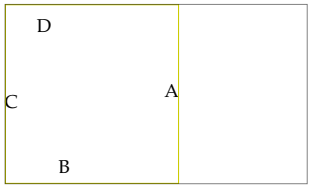
BW  [CARBON,BW]

C  [CARBON,C]

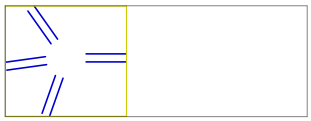
CC  [CARBON,CC1]

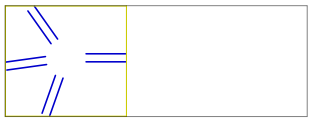
CCD  [CARBON,CCD1]

CD  [CARBON,CD]

CZ 

[CARBON,CZ] [A,B,C,D]

DB  [CARBON,DB]

DR  [CARBON,DR]

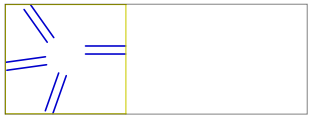
ED  [CARBON,ED]

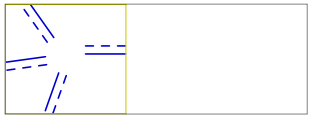
EP  [CARBON,EP]

ES  [CARBON,ES]

ET  [CARBON,ET]

HB  [CARBON,HB]

LDB  [CARBON,LDB]

LDD  [CARBON,LDD]

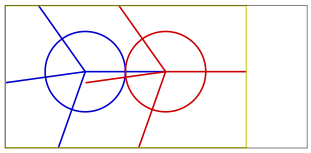
LINE  [CARBON,LINE]

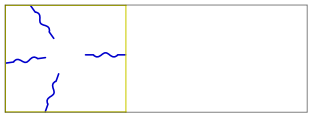
LRN 

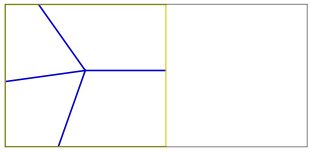
[CARBON,LRN]

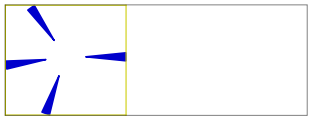
LRT 

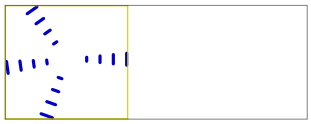
[CARBON,LRT] [A,B,C,D]

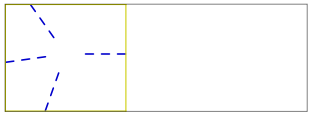
MOV 
[CARBON,B,C,MOV1,rulecolor,CARBON,B,C] [red]

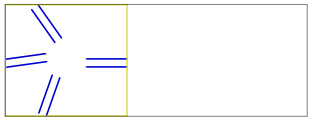
OE 
[CARBON,OE]

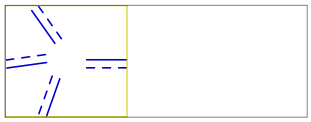
R 
[CARBON,R]

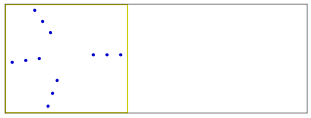
RB 
[CARBON,RB]

RBD 
[CARBON,RBD]

RD 
[CARBON,RD]

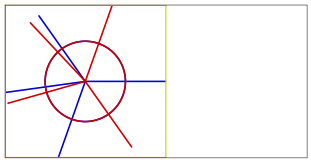
RDB 
[CARBON,RDB]

RDD 
[CARBON,RDD]

RH 
[CARBON,RH]

RN 

[CARBON,RN]

ROT  [CARBON,B,C,ROT1,rulecolor,CARBON,B,C] [red]

RRN 

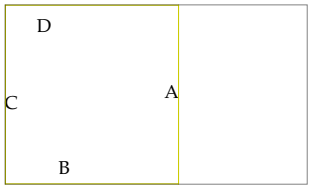
[CARBON,RRN]

RRT 

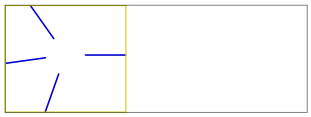
[CARBON,RRT] [A,B,C,D]

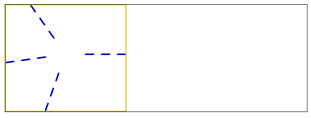
RT 

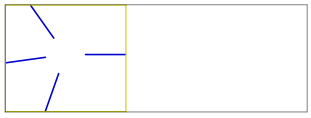
[CARBON,RT] [A,B,C,D]

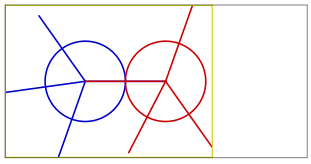
RZ 

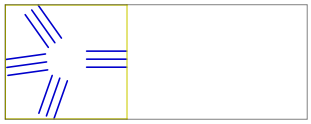
[CARBON,RZ] [A,B,C,D]

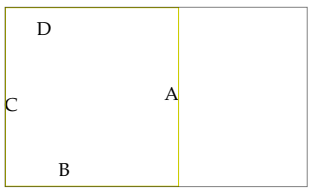
SB 
[CARBON,SB]

SD 
[CARBON,SD]

SR 
[CARBON,SR]

SUB 
[CARBON,B,C,SUB1,rulecolor,CARBON,B,C] [red]

TB 
[CARBON,TB]

Z 

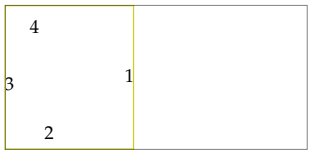
[CARBON,Z] [A,B,C,D]

ZLN 

[CARBON,ZLN]

ZLT 

[CARBON,ZLT] [A,B,C,D]

ZN 

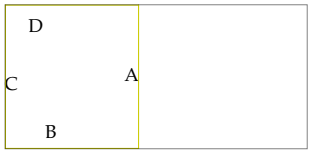
[CARBON,ZN]

ZRN 

[CARBON,ZRN]

ZRT 

[CARBON,ZRT] [A,B,C,D]

ZT 

[CARBON,ZT] [A,B,C,D]

# 28 Newman Stagger

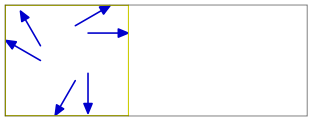
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ad | line | dir | transform | move | transform | rt | text |
| adj | transform | dr | line | number | number | rz | text |
| arrow | line | ed | line | oe | line | sb | line |
| au | line | ep | line | off | transform | sd | line |
| b | line | es | line | r | line | sr | line |
| bb | line | et | line | rb | line | sub | transform |
| bd | line | hb | line | rbd | line | tb | line |
| bw | line | label | text | rd | line | z | text |
| c | fixed | ldb | line | rdb | line | zln | number |
| cc | line | ldd | line | rdd | line | zlt | text |
| ccd | line | line | line | rh | line | zn | number |
| cd | fixed | lrn | number | rm | transform | zrn | number |
| cz | text | lrt | text | rn | number | zrt | text |
| dash | line | mark | transform | rot | transform | zt | text |
| db | line | mir | transform | rrn | number | | |
| diff | transform | mov | transform | rrt | text | | |

```
AD                      [NEWMAN,AD]
AU                      [NEWMAN,AU]
B                       [NEWMAN,B]
BB                      [NEWMAN,BB]
BD                      [NEWMAN,BD]
BW                      [NEWMAN,BW]
C                       [NEWMAN,C]
CC                      [NEWMAN,CC1]
CCD                     [NEWMAN,CCD1]
CD                      [NEWMAN,CD]
CZ                      [NEWMAN,CZ] [A,B,C,D]
DB                      [NEWMAN,DB]
DR                      [NEWMAN,DR]
ED                      [NEWMAN,ED]
EP                      [NEWMAN,EP]
ES                      [NEWMAN,ES]
ET                      [NEWMAN,ET]
HB                      [NEWMAN,HB]
LDB                     [NEWMAN,LDB]
LDD                     [NEWMAN,LDD]
LINE                    [NEWMAN,LINE]
LRN                     [NEWMAN,LRN]
LRT                     [NEWMAN,LRT] [A,B,C,D]
MOV                     [NEWMAN,B,C,MOV1,rulecolor,NEWMAN,B,C] [red]
OE                      [NEWMAN,OE]
R                       [NEWMAN,R]
RB                      [NEWMAN,RB]
```

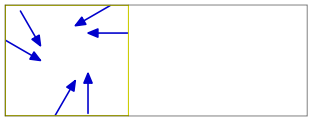
```
RBD                     [NEWMAN,RBD]
RD                      [NEWMAN,RD]
RDB                     [NEWMAN,RDB]
RDD                     [NEWMAN,RDD]
RH                      [NEWMAN,RH]
RN                      [NEWMAN,RN]
ROT                     [NEWMAN,B,C,ROT1,rulecolor,NEWMAN,B,C] [red]
RRN                     [NEWMAN,RRN]
RRT                     [NEWMAN,RRT] [A,B,C,D]
RT                      [NEWMAN,RT] [A,B,C,D]
RZ                      [NEWMAN,RZ] [A,B,C,D]
SB                      [NEWMAN,SB]
SD                      [NEWMAN,SD]
SR                      [NEWMAN,SR]
SUB                     [NEWMAN,B,C,SUB1,rulecolor,NEWMAN,B,C] [red]
TB                      [NEWMAN,TB]
Z                       [NEWMAN,Z] [A,B,C,D]
ZLN                     [NEWMAN,ZLN]
ZLT                     [NEWMAN,ZLT] [A,B,C,D]
ZN                      [NEWMAN,ZN]
ZRN                     [NEWMAN,ZRN]
ZRT                     [NEWMAN,ZRT] [A,B,C,D]
ZT                      [NEWMAN,ZT] [A,B,C,D]
```

# 29 Newman Eclipsed

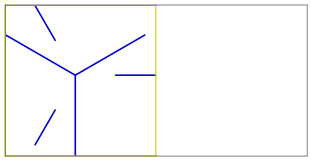
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ad | line | dir | transform | move | transform | rt | text |
| adj | transform | dr | line | number | number | rz | text |
| arrow | line | ed | line | oe | line | sb | line |
| au | line | ep | line | off | transform | sd | line |
| b | line | es | line | r | line | sr | line |
| bb | line | et | line | rb | line | sub | transform |
| bd | line | hb | line | rbd | line | tb | line |
| bw | line | label | text | rd | line | z | text |
| c | fixed | ldb | line | rdb | line | zln | number |
| cc | line | ldd | line | rdd | line | zlt | text |
| ccd | line | line | line | rh | line | zn | number |
| cd | fixed | lrn | number | rm | transform | zrn | number |
| cz | text | lrt | text | rn | number | zrt | text |
| dash | line | mark | transform | rot | transform | zt | text |
| db | line | mir | transform | rrn | number | | |
| diff | transform | mov | transform | rrt | text | | |

AD 

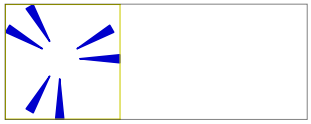
```
[NEWMAN,ECLIPSE,AD]
```

AU 

```
[NEWMAN,ECLIPSE,AU]
```

B 

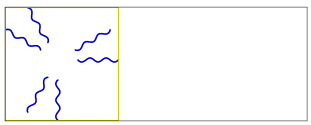
```
[NEWMAN,ECLIPSE,B]
```

BB 

```
[NEWMAN,ECLIPSE,BB]
```

BD 

```
[NEWMAN,ECLIPSE,BD]
```

BW 

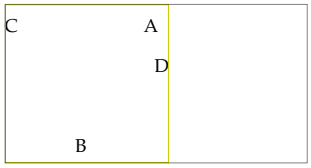
```
[NEWMAN,ECLIPSE,BW]
```

C  [NEWMAN,ECLIPSE,C]

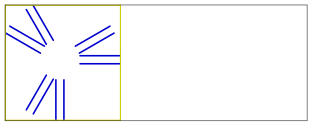
CC  [NEWMAN,ECLIPSE,CC1]

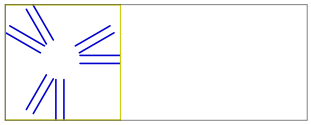
CCD  [NEWMAN,ECLIPSE,CCD1]

CD  [NEWMAN,ECLIPSE,CD]

CZ 

[NEWMAN,ECLIPSE,CZ] [A,B,C,D]

DB  [NEWMAN,ECLIPSE,DB]

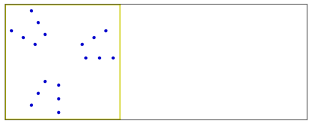
DR  [NEWMAN,ECLIPSE,DR]

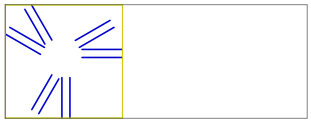
ED  [NEWMAN,ECLIPSE,ED]

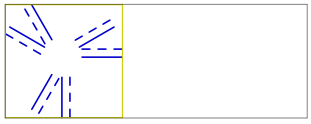
EP  [NEWMAN,ECLIPSE,EP]

ES  [NEWMAN,ECLIPSE,ES]

ET  [NEWMAN,ECLIPSE,ET]

HB  [NEWMAN,ECLIPSE,HB]

LDB  [NEWMAN,ECLIPSE,LDB]

LDD  [NEWMAN,ECLIPSE,LDD]

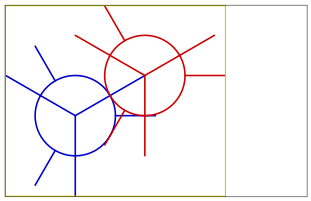
LINE [NEWMAN,ECLIPSE,LINE]

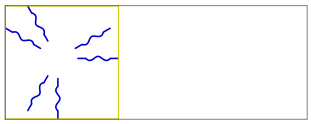
LRN 

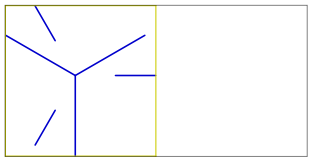
[NEWMAN,ECLIPSE,LRN]

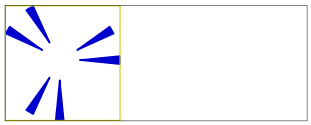
LRT 

[NEWMAN,ECLIPSE,LRT] [A,B,C,D]

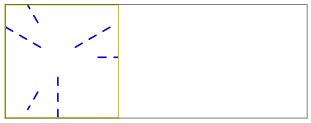
MOV 
[NEWMAN,ECLIPSE,B,C,MOV1,rulecolor,NEWMAN,ECLIPSE,B,C] [red]

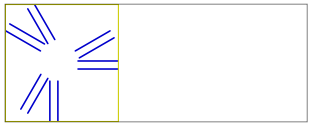
OE 
[NEWMAN,ECLIPSE,OE]

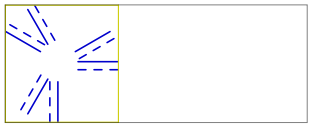
R 
[NEWMAN,ECLIPSE,R]

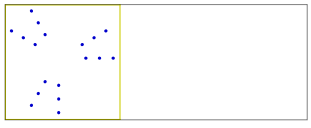
RB 
[NEWMAN,ECLIPSE,RB]

RBD 
[NEWMAN,ECLIPSE,RBD]

RD 
[NEWMAN,ECLIPSE,RD]

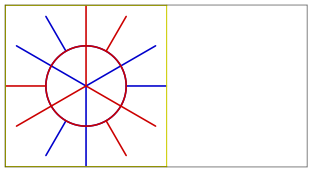
RDB 
[NEWMAN,ECLIPSE,RDB]

RDD 
[NEWMAN,ECLIPSE,RDD]

RH 
[NEWMAN,ECLIPSE,RH]

RN 

[NEWMAN,ECLIPSE,RN]

ROT 
[NEWMAN,ECLIPSE,B,C,ROT1,rulecolor,NEWMAN,ECLIPSE,B,C]
[red]

RRN 

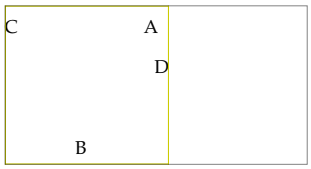
[NEWMAN,ECLIPSE,RRN]

RRT 

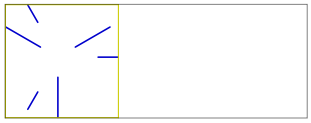
[NEWMAN,ECLIPSE,RRT] [A,B,C,D]

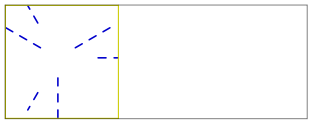
RT 

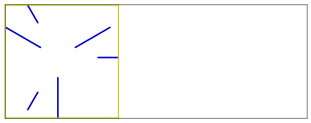
[NEWMAN,ECLIPSE,RT] [A,B,C,D]

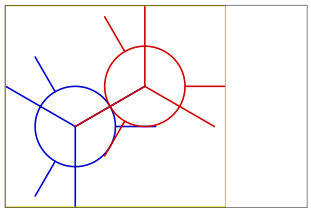
RZ 

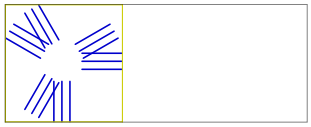
[NEWMAN,ECLIPSE,RZ] [A,B,C,D]

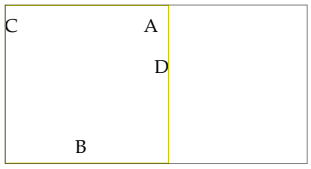
SB 
[NEWMAN,ECLIPSE,SB]

SD 
[NEWMAN,ECLIPSE,SD]

SR 
[NEWMAN,ECLIPSE,SR]

SUB 
[NEWMAN,ECLIPSE,B,C,SUB1,rulecolor,NEWMAN,ECLIPSE,B,C]
[red]

TB 
[NEWMAN,ECLIPSE,TB]

Z 

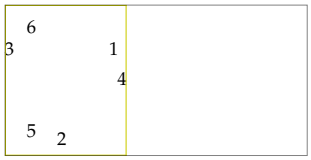
[NEWMAN,ECLIPSE,Z] [A,B,C,D]

ZLN 

[NEWMAN,ECLIPSE,ZLN]

ZLT 

[NEWMAN,ECLIPSE,ZLT] [A,B,C,D]

ZN 

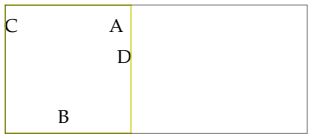
[NEWMAN,ECLIPSE,ZN]

ZRN 

[NEWMAN,ECLIPSE,ZRN]

ZRT 

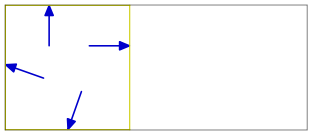
[NEWMAN,ECLIPSE,ZRT] [A,B,C,D]

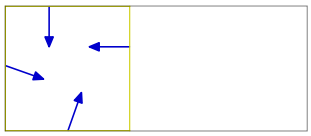
ZT 

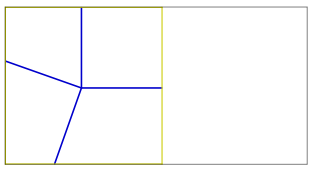
[NEWMAN,ECLIPSE,ZT] [A,B,C,D]

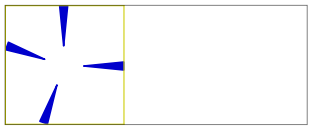
# 30 Alkyl

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ad | line | dir | transform | move | transform | rt | text |
| adj | transform | dr | line | number | number | rz | text |
| arrow | line | ed | line | oe | line | sb | line |
| au | line | ep | line | off | transform | sd | line |
| b | line | es | line | r | line | sr | line |
| bb | line | et | line | rb | line | sub | transform |
| bd | line | hb | line | rbd | line | tb | line |
| bw | line | label | text | rd | line | z | text |
| c | fixed | ldb | line | rdb | line | zln | number |
| cc | line | ldd | line | rdd | line | zlt | text |
| ccd | line | line | line | rh | line | zn | number |
| cd | fixed | lrn | number | rm | transform | zrn | number |
| cz | text | lrt | text | rn | number | zrt | text |
| dash | line | mark | transform | rot | transform | zt | text |
| db | line | mir | transform | rrn | number | | |
| diff | transform | mov | transform | rrt | text | | |

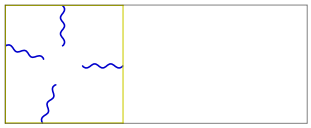
AD  [ALKYL,AD]

AU  [ALKYL,AU]

B  [ALKYL,B]

BB  [ALKYL,BB]

BD  [ALKYL,BD]

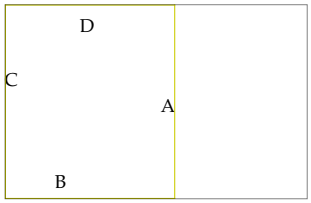
BW  [ALKYL,BW]

C  [ALKYL,C]

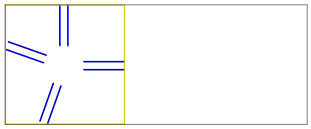
CC  [ALKYL,CC1]

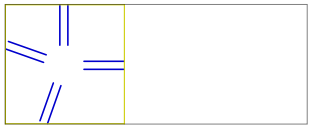
CCD  [ALKYL,CCD1]

CD  [ALKYL,CD]

CZ 

[ALKYL,CZ] [A,B,C,D]

DB  [ALKYL,DB]

DR  [ALKYL,DR]

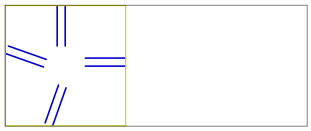
ED  [ALKYL,ED]

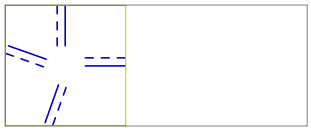
EP  [ALKYL,EP]

ES  [ALKYL,ES]

ET  [ALKYL,ET]

HB  [ALKYL,HB]

LDB  [ALKYL,LDB]

LDD  [ALKYL,LDD]

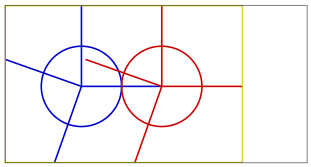
LINE [ALKYL,LINE]

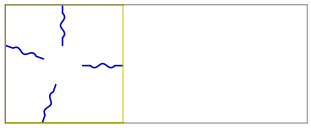
LRN 

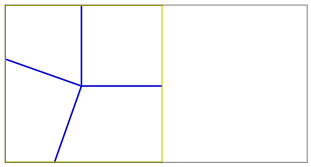
[ALKYL,LRN]

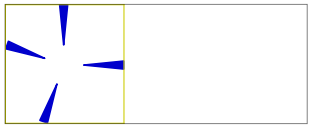
LRT 

[ALKYL,LRT] [A,B,C,D]

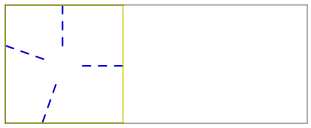
MOV  [ALKYL,B,C,MOV1,rulecolor,ALKYL,B,C] [red]

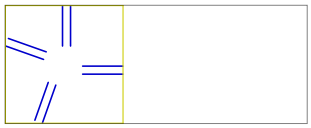
OE  [ALKYL,OE]

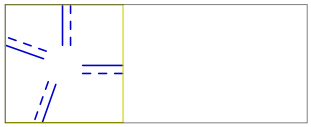
R  [ALKYL,R]

RB  [ALKYL,RB]

RBD  [ALKYL,RBD]

RD  [ALKYL,RD]

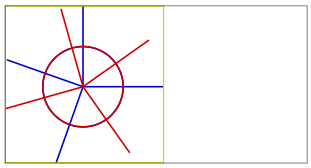
RDB  [ALKYL,RDB]

RDD  [ALKYL,RDD]

RH  [ALKYL,RH]

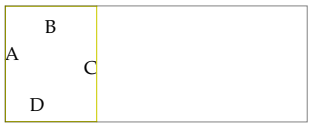
RN 

[ALKYL,RN]

ROT  [ALKYL,B,C,ROT1,rulecolor,ALKYL,B,C] [red]

RRN 

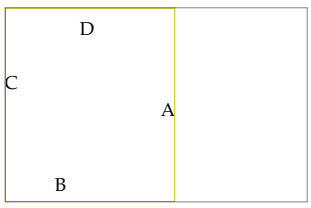
[ALKYL,RRN]

RRT 

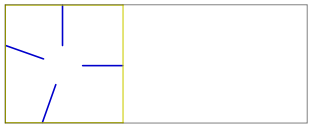
[ALKYL,RRT] [A,B,C,D]

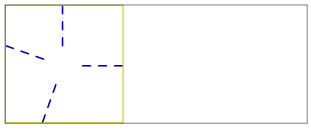
RT 

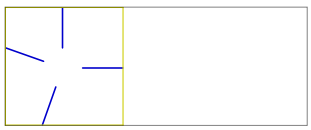
[ALKYL,RT] [A,B,C,D]

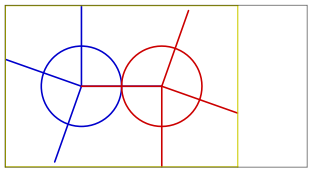
RZ 

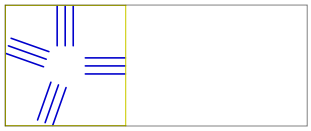
[ALKYL,RZ] [A,B,C,D]

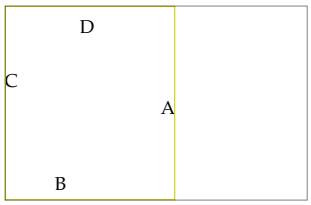
SB  [ALKYL,SB]

SD  [ALKYL,SD]

SR  [ALKYL,SR]

SUB  [ALKYL,B,C,SUB1,rulecolor,ALKYL,B,C] [red]

TB  [ALKYL,TB]

Z 

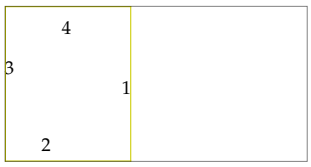
[ALKYL,Z] [A,B,C,D]

ZLN 

[ALKYL,ZLN]

ZLT 

[ALKYL,ZLT] [A,B,C,D]

ZN 

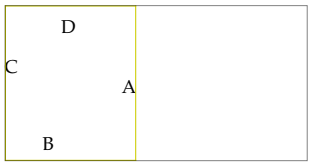
[ALKYL,ZN]

ZRN 

[ALKYL,ZRN]

ZRT 

[ALKYL,ZRT] [A,B,C,D]

ZT 

[ALKYL,ZT] [A,B,C,D]

# 31 Chair

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| adj | transform | lrb | line | r | line | rsr | line |
| arrow | line | lrbd | line | rb | line | rsub | transform |
| b | line | lrd | line | rbd | line | rt | text |
| bb | line | lrh | line | rd | line | rz | text |
| c | fixed | lrn | number | rh | line | sb | line |
| cc | line | lrt | text | rn | number | sr | line |
| ccd | line | lrz | text | rr | line | sub | transform |
| cd | fixed | lsr | line | rrb | line | z | text |
| dash | line | lsub | transform | rrbd | line | zln | number |
| diff | transform | mark | transform | rrd | line | zlt | text |
| eb | line | mov | transform | rrh | line | zn | number |
| label | text | move | transform | rrn | number | zrn | number |
| line | line | number | number | rrt | text | zrt | text |
| lr | line | off | transform | rrz | text | zt | text |

AD [CHAIR,AD]
AU [CHAIR,AU]
B  [CHAIR,B]

BB  [CHAIR,BB]

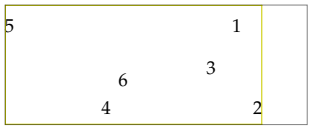
BD [CHAIR,BD]
BW [CHAIR,BW]
C  [CHAIR,C]

CC  [CHAIR,CC1]

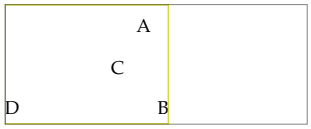
CCD  [CHAIR,CCD1]

CD  [CHAIR,CD]

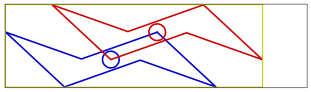
CZ [CHAIR,CZ] [A,B,C,D]
DB [CHAIR,DB]
DR [CHAIR,DR]
ED [CHAIR,ED]
EP [CHAIR,EP]
ES [CHAIR,ES]
ET [CHAIR,ET]
HB [CHAIR,HB]
LDB [CHAIR,LDB]
LDD  [CHAIR,LDD]
LINE [CHAIR,LINE]

LRN 

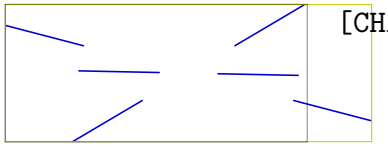
[CHAIR,LRN]

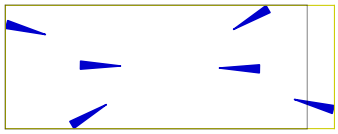
LRT 

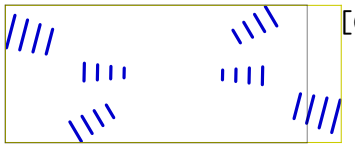
[CHAIR,LRT] [A,B,C,D]

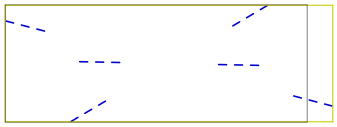
MOV 
[CHAIR,B,C,MOV1,rulecolor,CHAIR,B,C] [red]

OE [CHAIR,OE]

R 
[CHAIR,R]

RB 
[CHAIR,RB]

RBD 
[CHAIR,RBD]

RD 
[CHAIR,RD]

RDB [CHAIR,RDB]

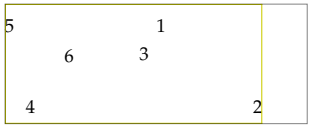
RDD [CHAIR,RDD]

RH 
[CHAIR,RH]

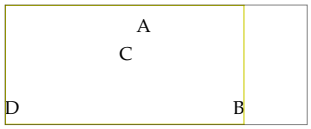
RN 

[CHAIR,RN]

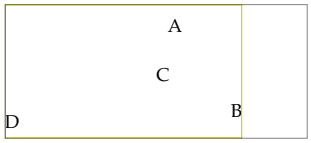
ROT 
[CHAIR,B,C,ROT1,rulecolor,CHAIR,B,C] [red]

RRN 

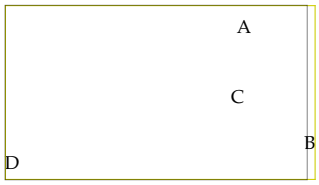
[CHAIR,RRN]

RRT 

[CHAIR,RRT] [A,B,C,D]

RT 

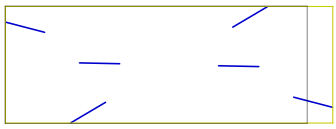
[CHAIR,RT] [A,B,C,D]

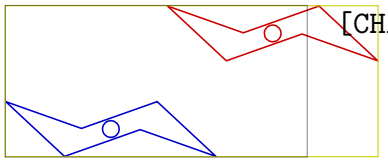
RZ 

[CHAIR,RZ] [A,B,C,D]

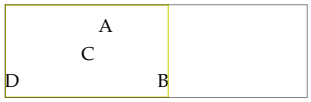
SB 
[CHAIR,SB]

SD
[CHAIR,SD]

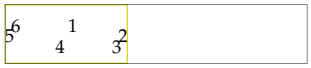
SR 
[CHAIR,SR]

SUB 
[CHAIR,B,C,SUB1,rulecolor,CHAIR,B,C] [red]

TB
[CHAIR,TB]

Z 

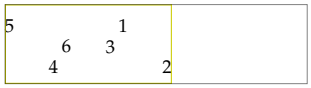
[CHAIR,Z] [A,B,C,D]

ZLN 

[CHAIR,ZLN]

ZLT 

[CHAIR,ZLT] [A,B,C,D]

ZN 

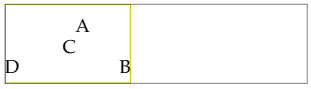
[CHAIR,ZN]

ZRN 

[CHAIR,ZRN]

ZRT 

[CHAIR,ZRT] [A,B,C,D]

ZT 

[CHAIR,ZT] [A,B,C,D]

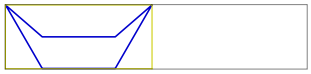
# 32 Boat

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| adj | transform | lrb | line | r | line | rsr | line |
| arrow | line | lrbd | line | rb | line | rsub | transform |
| b | line | lrd | line | rbd | line | rt | text |
| bb | line | lrh | line | rd | line | rz | text |
| c | fixed | lrn | number | rh | line | sb | line |
| cc | line | lrt | text | rn | number | sr | line |
| ccd | line | lrz | text | rr | line | sub | transform |
| cd | fixed | lsr | line | rrb | line | z | text |
| dash | line | lsub | transform | rrbd | line | zln | number |
| diff | transform | mark | transform | rrd | line | zlt | text |
| eb | line | mov | transform | rrh | line | zn | number |
| label | text | move | transform | rrn | number | zrn | number |
| line | line | number | number | rrt | text | zrt | text |
| lr | line | off | transform | rrz | text | zt | text |

AD [BOAT,AD]

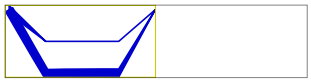
AU [BOAT,AU]

B



[BOAT,B]

BB



[BOAT,BB]

BD [BOAT,BD]

BW [BOAT,BW]

C

[BOAT,C]

CC

[BOAT,CC1]

CCD

[BOAT,CCD1]

CD

[BOAT,CD]

CZ [BOAT,CZ] [A,B,C,D]

DB [BOAT,DB]

DR [BOAT,DR]

ED [BOAT,ED]

EP [BOAT,EP]

ES [BOAT,ES]

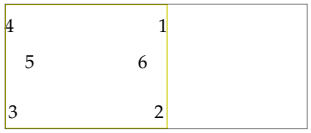
ET [BOAT,ET]

HB [BOAT,HB]

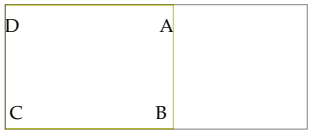
LDB [BOAT,LDB]

LDD [BOAT,LDD]

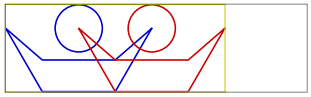
LINE [BOAT,LINE]

LRN 

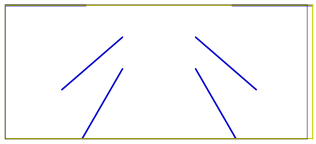
[BOAT,LRN]

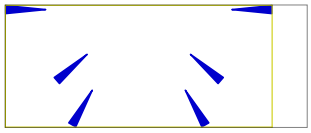
LRT 

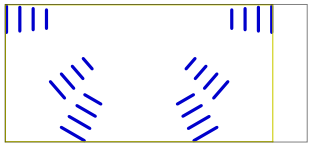
[BOAT,LRT] [A,B,C,D]

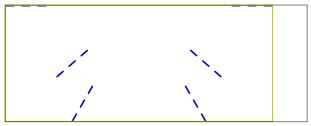
MOV 
[BOAT,B,C,MOV1,rulecolor,BOAT,B,C] [red]

OE [BOAT,OE]

R 
[BOAT,R]

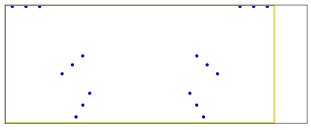
RB 
[BOAT,RB]

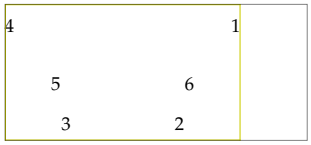
RBD 
[BOAT,RBD]

RD 
[BOAT,RD]

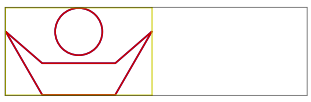
RDB [BOAT,RDB]

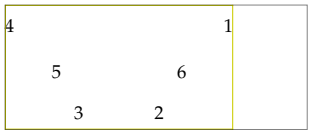
RDD [BOAT,RDD]

RH 
[BOAT,RH]

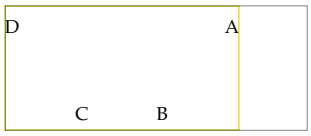
RN 

[BOAT,RN]

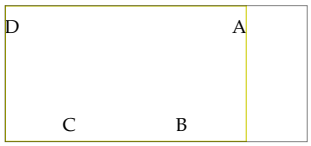
ROT 
[BOAT,B,C,ROT1,rulecolor,BOAT,B,C] [red]

RRN 

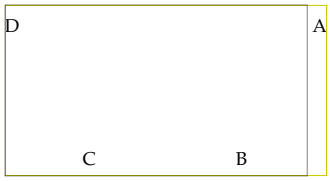
[BOAT,RRN]

RRT 

[BOAT,RRT] [A,B,C,D]

RT 

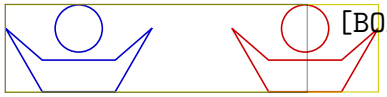
[BOAT,RT] [A,B,C,D]

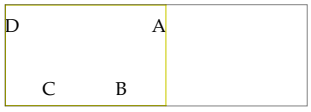
RZ 

[BOAT,RZ] [A,B,C,D]

SB 
[BOAT,SB]

SD
[BOAT,SD]
SR
[BOAT,SR]

SUB 
[BOAT,B,C,SUB1,rulecolor,BOAT,B,C] [red]

TB
[BOAT,TB]
Z 

[BOAT,Z] [A,B,C,D]

ZLN 

[BOAT,ZLN]

ZLT 

[BOAT,ZLT] [A,B,C,D]

ZN 

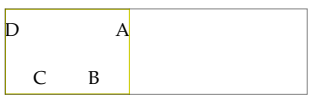
[BOAT,ZN]

ZRN 

[BOAT,ZRN]

ZRT 

[BOAT,ZRT] [A,B,C,D]

ZT 

[BOAT,ZT] [A,B,C,D]

# 33 Symbol

CHEM [CHEM]

SPACE [SPACE]

PLUS [PLUS]

MINUS [MINUS]

EQUALS [EQUALS]

GIVES [GIVES] [A]

EQUILIBRIUM [EQUILIBRIUM] [A]

MESOMERIC [MESOMERIC] [A]

OPENCOMPLEX [OPENCOMPLEX]

CLOSECOMPLEX [CLOSECOMPLEX]