

chemfig

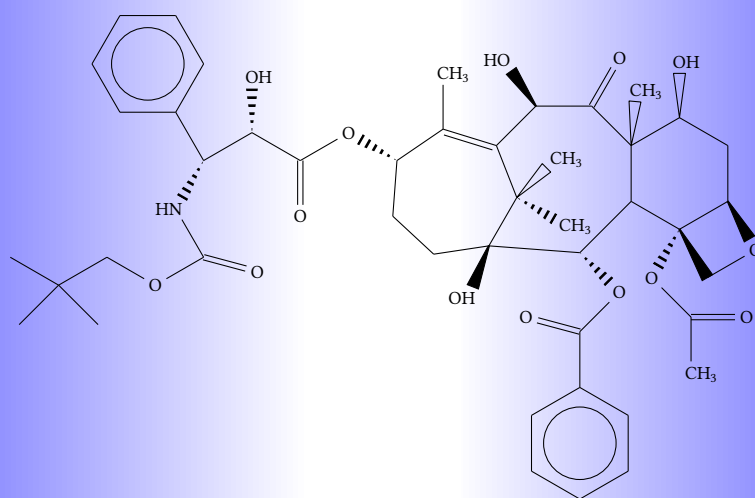
v1.71

2025 年 10 月 30 日

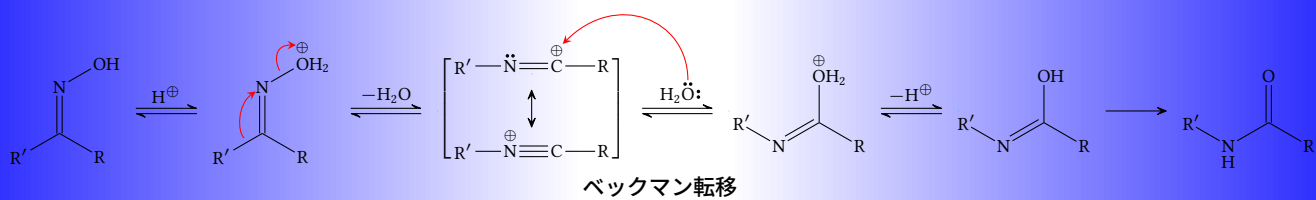
Christian Tellechea

unbonpetit@netc.fr

分子描画のための T_EX パッケージ



タキソテル (ドセタキセル)



目次

I	はじめに	5
1	バージョン 1.7 における変更点	5
1.1	水平方向の反応	5
1.2	内部の変更	5
2	chemfig を使う	5
3	謝辞	6
II	chemfig の機能	7
1	\chemfig マクロ	7
2	原子グループ	8
3	最初の原子の役割	8
4	色々な結合	9
5	結合の角度	10
5.1	定義済みの角度	11
5.2	角度の直接指定	11
5.3	相対角度	11
6	結合の長さ	12
7	出発原子と到着原子	13
8	結合のカスタマイズ	14
9	結合同士の接合	15
10	デフォルト値	15
11	側鎖	16
11.1	原理	16
11.2	入れ子	17
11.3	描き方のヒント	17
12	離れた原子を繋げる	18
13	環	20
13.1	構文	20
13.2	環の角度の調整	21
13.3	環の側鎖	22

13.4	入れ子の環	23
13.5	環と原子グループ	24
13.6	環の中心	24
14	電子の動きの表現	25
14.1	共鳴効果	25
14.2	反応機構	27
15	分子の下に分子名を書く	28
III	高度な使用方法	30
1	原子の分割	30
2	原子の表示	30
3	TikZ に渡される引数	31
4	二重結合をずらす	32
5	非局在化した二重結合	32
6	部分分子の保存	33
7	原子の配置	35
7.1	原子グループ	35
7.2	垂直方向の整列	35
7.3	原子間の結合	36
7.4	\chemskipalign マクロ	38
8	\charge マクロ	38
8.1	概要	38
8.2	パラメーター	39
8.3	ルイス構造式	40
8.4	chemfig の中で使う	41
9	積み重ね	41
10	\chemfig を tikzpicture 環境で使う	42
11	分子の記述の例——注釈を添えて	42
11.1	エタナール	43
11.2	2-アミノ-4-オキソヘキサン酸	43
11.3	グルコース	45
11.4	アドレナリン	48
11.5	グアニン	49
12	こんなときは……	51

12.1	色付きの原子を描くには	51
12.2	結合を変更せずに上付き文字を追加するには	51
12.3	結合を曲線で描くには	52
12.4	重合体の要素を描くには	52
12.5	分子の反転形を描くには	54
12.6	結合に沿ってテキストを追加したり、角度の弧を描くには	55
12.7	多重結合を描くには	56
IV	反応式	57
1	概要	57
2	矢印の種類	58
3	矢印の特徴付け	59
4	化合物名	60
5	アンカー	61
6	化合物のスタイル	63
7	分岐	63
8	部分反応式	64
9	矢印のオプション引数	67
10	矢印のカスタマイズ	69
10.1	はじめての矢印	70
10.2	曲がった矢印	71
11	\merge コマンド	72
12	プラス記号	74
V	水平方向の反応	77
1	文法	78
2	化合物の配置	78
3	化合物名	79
4	矢印のラベル	80
5	矢印の種類	80
VI	コマンドリスト	82

VII	ギャラリー	84
VIII	変更履歴	101

はじめに

1 バージョン 1.7 における変更点

変更履歴は今後 `chemfig.tex` ファイルの末尾ではなく、このマニュアルの末尾に記載されます。詳細は [101](#) ページを参照してください。これは英語に翻訳されません。

1.1 水平方向の反応

水平方向の反応を描くための新しい環境が実装されました。

```
\hreact[〈キーワード〉=<値〉のリスト]
      〈反応〉
\endhreact
```

この環境でできることは `\schemestart` 〈反応〉 `\schemestop` でできることよりかなり少ないですが、いくつかの高度な機能を維持しつつ、水平方向の反応式をより素早く描くことができます。[77](#) ページを参照してください。

1.2 内部の変更

内部的な変更がいくつか行われました。

- バージョン 1.7 以降、デフォルトでは直前の原子の支柱 (strut) は `\printatom` の引数に加えられなくなりました。これにより原子の配置がわずかに変わる場合があります。 `use atom strut = <true>` とすることで以前の動作に戻すことができます。[36](#) ページを参照してください。
- `use atom strut = <false>` と設定されたとき (これがデフォルトの動作です)、各原子は一度だけ組版されます。以前は 5 回以上組まれることがありました。
内部のマクロが修正されました。

重要: 一貫性を保つため、次期バージョンの `chemfig` では `\schemestart` の文法が変更されます。これまでの 2 つのオプション引数は削除され、〈キーワード〉=〈値〉形式のリストを取る単一のオプション引数に置き換えられます。これまで 2 つのオプション引数で指定されていた設定は、新しい〈キーワード〉で設定できるようになります。

2 chemfig を使う

`chemfig` パッケージを使うには、以下の行をプリアンプルに追加します。

- εTeX の場合は `\input chemfig.tex`

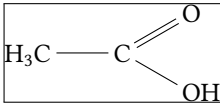
- \LaTeX の場合は `\usepackage{chemfig}`

いずれの場合も、もしまだ読み込まれていなければ、`chemfig` が `TikZ` パッケージを読み込みます。

分子描画のための最も重要なコマンドは `\chemfig{⟨コード⟩}` です。引数 `⟨コード⟩` は分子構造を記述する文字列であり、その規則についてはこのマニュアルで説明します。

シンプルで柔軟かつ直感的な構文を維持しながら、可能な限り多くの分子形状を描画できるように配慮されています。それでもなお、分子の二次元構造を記述する `⟨コード⟩` の複雑さは、描こうとする分子の複雑さに比例します。

`\chemfig` コマンドは `tikzpicture` 環境内に、`TikZ` パッケージが提供するコマンドを使って分子を描画します。`TikZ` を使用していることから、以下のような特徴を持ちます。

- ユーザーは自分の好きなエンジンを使用できます。`pdf \LaTeX` では `dvi` モード (`tex` \rightarrow `dvi` \rightarrow `ps` \rightarrow `pdf`) でも `pdf` モード (`tex` \rightarrow `pdf`) でも同じように使用できます。`TikZ` を使用することで、その基盤である `pgf` によって、いずれのモードでも同一の描画結果が得られます。
- バウンディングボックスは `TikZ` によって自動的に計算されるので、ユーザーはテキストと重なる心配をする必要がありません。しかし、分子を段落内に描く場合はその位置合わせに注意しなくてはなりません。分子のバウンディングボックスを表示させると、 のようになります。

3 謝辞

このパッケージが日の目を見ることができたのは、発案者である Christophe CASSEAU の協力のおかげです。コードを書く前に彼が提供してくれた支援と、彼が行ったテストに感謝します。

また、このマニュアルの英訳を申し出てくれた Theo HOPMAN に心から感謝します。

chemfig の機能

この部では chemfig の最も一般的な機能を説明します。これでたいの分子を描くことができるようになります。ここでは、理論的な観点から機能を説明します。この部の目的は実際の分子を描くことではなく、chemfig の機能を形式的に説明することです。第 III 部「高度な使用方法」(30 ページ) では、より実用的で、最も要求の厳しい用途のための高度な機能を説明します。また、具体的な分子を描画する方法にも着目します(42 ページ)。最後に、分子とその描画に使われるコードの例を示します。

1 \chemfig マクロ

\chemfig は次の構文を使用します。

`\chemfig[〈キーワード〉=〈値〉形式のリスト]{〈分子コード〉}`

角括弧内のオプション引数はこの分子の描画に使用するパラメーターを設定します。パラメーターは現在の分子に対してのみ変更され、マクロ実行後は以前の値に戻ることに注意が必要です。パラメーターを永続的に変更したい場合は `\setchemfig{〈キーワード〉=〈値〉}` マクロを使用してください。

以下に \chemfig マクロの全パラメーター、デフォルト値、および簡単な説明を示します。

〈キーワード〉	デフォルト 〈値〉	説明
<code>chemfig style</code>	〈空〉	TikZ に渡されるスタイル
<code>use atom strut</code>	false	\printatom の引数に、直前の原子と等しい支柱を挿入する
<code>atom style</code>	〈空〉	原子を含んでいる TikZ ノードのスタイル
<code>bond join</code>	false	結合間の接合に関する真偽値
<code>fixed length</code>	false	原子間間隔を固定するか (真偽値)
<code>cram rectangle</code>	false	true のとき、破線の結合を (くさび形ではなく) 矩形で描く
<code>cram width</code>	1.5ex	くさび形結合の底辺の長さ
<code>cram dash width</code>	1pt	くさび形結合の破線の太さ
<code>cram dash sep</code>	2pt	くさび形結合の破線間の間隔
<code>atom sep</code>	3em	原子間隔
<code>bond offset</code>	2pt	原子と結合の間隔
<code>double bond sep</code>	2pt	多重結合の線の間隔
<code>angle increment</code>	45	結合の角度の増分
<code>node style</code>	〈空〉	原子のスタイル
<code>bond style</code>	〈空〉	結合のスタイル
<code>baseline</code>	0pt	垂直位置を調整するための長さまたはノードの名前
<code>debug</code>	false	ノード、そのアンカーおよび名前を表示
<code>cycle radius coeff</code>	0.75	環の中に描かれる円・弧の縮小率
<code>stack sep</code>	1.5pt	\chemabove および \chembelow マクロの引数の垂直方向の間隔

〈キーワード〉	デフォルト 〈値〉	説明
<code>show cntcycle</code>	<code>false</code>	環の番号の表示
<code>autoreset cntcycle</code>	<code>true</code>	環の番号を <code>\chemfig</code> の実行ごとにリセット
<code>gchemname</code>	<code>true</code>	<code>true</code> の場合 <code>\chemnameinit</code> と <code>\chemname</code> による深さの保存をグローバルにする

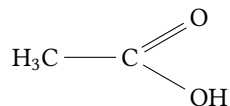
〈分子コード〉には、このドキュメントで説明する構文に従って、分子を描画するための指示を記述します。使用できる文字に制限はありません。

- 全ての 카테고리コード 11 と 12 の文字、つまり大文字、小文字、数字、算術演算子 (+ - * / =)、アクティビティかどうかによらず句読点類 (., ; : ! ? ' ' " " |)、括弧や角括弧
- “~” や “#”¹などのより特殊な文字や、“^” や “_” のような通常数式モードで使用される文字
- 空白。ただし、原子は数式モードで扱われるので、デフォルトでは無視されます。
- 波括弧 “{” および “}”。これはグルーピングやマクロ引数の区切りとしての通常の動作をします。
- マクロ

2 原子グループ

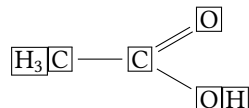
分子を描くということは、本質的には、原子グループを線で結ぶということです。つまり分子 $\text{O}=\text{O}$ の場合は、2つの原子グループがあり、それぞれが単一の原子 “O” で構成されています。

しかし、次の分子



は、“H₃C”、“C”、“O”、“OH” の4つの原子グループで構成されています。後で述べる理由から、`chemfig` は各グループを単一の原子に分解します。次に来る大文字または特殊文字 `☐ ☐ ☐ ☐ ! ☐ ☐ ☐ ☐` までを原子として扱います。`chemfig` はグループを原子に分割する際、波括弧内の全ての文字を無視します。

したがって、最初の原子グループ “H₃C” は `☐☐☐` と `☐` の2つの原子に分割されます。化学的には、これらはもちろん本当の原子ではありません。例えば H₃ は3つの水素原子から構成されています。以下では原子という単語を `chemfig` の定義に従って使用します。つまり `chemfig` は上の分子をこのように見えています。



原子グループの先頭にある空白は無視されます。

3 最初の原子の役割

分子全体の配置は、最初に配置される原子、すなわち最初の原子グループの最初の原子に依存します。この最初の原子の TikZ アンカー “base east” は、現在の行のベースライン上に配置されます。（このマニュアルの例では灰色で描かれています。）

¹`\chemfig` マクロがマクロの引数に含まれる場合に # の重複を避けるために、# のかわりに `\#` マクロや `\CFhash` マクロを使うことができます。

```
\chemfig{A-B}\qqquad
\chemfig{-B}\qqquad
\chemfig{A^1-B}
```



任意の垂直オフセットを設定したり、原子グループをベースライン上に配置するには、キーワード `<baseline>` を使用します (35 ページ参照)。

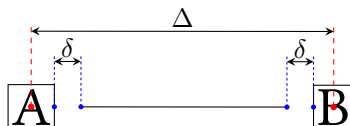
4 色々な結合

`chemfig` では、2つの原子間の結合として以下の9種類を使用することができ、それぞれ \square 、 \square 、 \square 、 \square 、 \square 、 \square 、 \square 、 \square 、 \square で記述します。

結合番号	記述	出力	結合の種類
1	<code>\chemfig{A-B}</code>	$A \text{ --- } B$	単結合
2	<code>\chemfig{A=B}</code>	$A \text{ == } B$	二重結合
3	<code>\chemfig{A\sim B}</code>	$A \text{ === } B$	三重結合
4	<code>\chemfig{A>B}</code>	$A \blacktriangleright B$	右実線くさび
5	<code>\chemfig{A<B}</code>	$A \blacktriangleleft B$	左実線くさび
6	<code>\chemfig{A>:B}</code>	$A \text{ } \cdots B$	右破線くさび
7	<code>\chemfig{A<:B}</code>	$A \cdots \text{ } B$	左破線くさび
8	<code>\chemfig{A> B}</code>	$A \triangleangleright B$	右中空くさび
9	<code>\chemfig{A< B}</code>	$A \triangleleft B$	左中空くさび

〈キーワード〉`double bond sep = <長さ>` を使うと二重結合と三重結合での線同士の距離を調整できます。この距離のデフォルトは 2pt です。

原子間に結合が作られるとき、それぞれの原子は見えない矩形に収められていることに注意してください。この2つの矩形の中心は「原子間距離」と呼ばれる調整可能な距離 Δ だけ離れています。さらに、結合は矩形の辺と直接繋がっている訳ではありません。矩形の辺と結合の始点・終点は、同じく調整可能な距離 δ だけ離れています。理解の補助となるように、次の図では矩形を表示しています。



〈キーワード〉`atom sep = <長さ>` は原子間距離 Δ を設定します。

原子間距離

```
\chemfig[atom sep=2em]{A-B}\par
\chemfig[atom sep=50pt]{A-B}
```



〈キーワード〉`bond offset = <長さ>` は結合と原子の距離 δ を設定します。デフォルト値は 2pt です。

結合の調整

```
\chemfig[bond offset=0pt]{A-B}\par
\chemfig[bond offset=5pt]{A-B}
```



ある結合が別の結合と直接繋がっている場合は、`chemfig` は空のグループ `{ }` を配置します。この空のグループの周りの距離 δ は 0 です。

空のグループ

```
\chemfig{A-B==C}
```



〈キーワード〉`bond style =`〈tikz コード〉は結合のスタイルを設定します。〈tikz コード〉はデフォルトでは空です。単一の結合のスタイルを変更したい場合については 14 ページを参照してください。

結合のスタイル

```
\chemfig[bond style={line width=1pt,red}]{A-B=C>|D<E>:F}
```

A — B = C > D < E : F

距離 δ を 1 つの結合のみに指定したい場合は、`#`、`\#` マクロ、あるいは `\CFhash` マクロを使用します。もし `\chemfig` マクロがあるマクロの引数の中にある場合は `#` を使ってはいけません。そのような場合は `\#` か `\CFhash` を使ってください。

トークン `#`、`\#` および `\CFhash` は結合記号の直後に書かねばならず、“(〈距離1〉,〈距離2〉)” の形式で括弧内に書かれた引数が必要です。ここで 〈距離1〉 は結合の始点での距離 δ であり、〈距離2〉 は結合の終点での距離です。〈距離2〉 が省略された場合、終点での距離にはその時点で有効な δ の値が使われます。以下に “B” に入る結合の距離、“B” から出る結合の距離、その両方を 0 にする方法をそれぞれ示します。なおこの例では見やすさのために、原子・結合間距離を 4pt にしてあります。

結合・原子間距離の微調整

```
\setchemfig{bond offset=4pt}
\chemfig{A-B-C}\par
\chemfig{A-#(,0pt)B-C}\par
\chemfig{A-B-#(0pt)C}\par
\chemfig{A-#(,0pt)B-#(0pt)C}
```

A — B — C

A — B — C

A — B — C

A — B — C

デフォルトでは、原子グループ内の全ての原子は数式モードで組まれます（空白は無視されます）。そのため、下付き文字や上付き文字のような数式モード特有のコマンドを書くことができます²。

数式モード

```
\chemfig{A_1B^2-C_3^4}
```

$A_1B^2 - C_3^4$

くさび形結合専用の設定もあります。

- `cram width =`〈長さ〉は三角形の底辺の長さで、デフォルトは 1.5pt です。
- `cram dash width =`〈長さ〉は破線の太さで、デフォルトは 1pt です。
- `cram dash sep =`〈長さ〉は破線間の距離で、デフォルトは 2pt です。

これら 3 つの値を変更した場合の例を示します。

くさび形結合の形の変更

```
\chemfig[cram width=10pt,
cram dash width=0.4pt,
cram dash sep=1pt]{A>B:C>|D}
```

A > B < C > D

5 結合の角度

それぞれの結合に角括弧でオプション引数を与えることができます。このオプション引数はカンマで区切られた 5 つのオプションフィールドで構成されており、これによって結合の全てを制御できます。最初のフィールドは、結合の角度を設定します。角度は水平を基準として、反時計回りに大きくなります。このフィールドが空の場合、角度はデフォルトの 0° になります。デフォルト値を変更する方法は後で説明します。

角度を指定するための方法はいくつかあります。

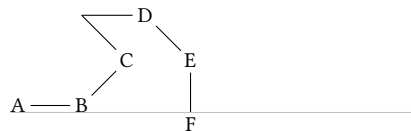
²指数や添え字を含む原子グループは配置の際に問題になることがあります。

5.1 定義済みの角度

角度フィールドが整数の場合、結合が水平に対してなす角度を 45° の倍数で表します。例えば、[0] は 0° 、[1] は 45° のようになります。

定義済みの角度

```
\chemfig{A-B-[1]C-[3]D-[7]E-[6]F}
```



これらの角度は原子が空の場合でも有効であり、このことは以下で紹介する全ての機能に当てはまります。

定義済みの角度と空のグループ

```
\chemfig{--[1]-[3]--[7]-[6]}
```

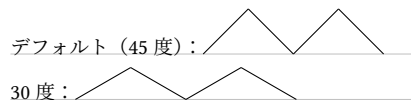


〈キーワード〉 `angle increment = 〈角度〉` は結合角度の計算に使用されるデフォルトの角度を設定します。

定義済みの角度の設定

デフォルト (45度): `\chemfig{-[1]-[1]-[1]-[1]}`

30度: `\chemfig[angle increment=30]{-[1]-[1]-[1]-[1]}`

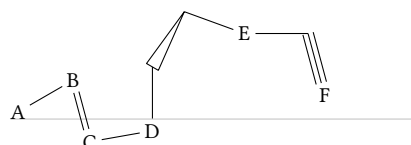


5.2 角度の直接指定

水平に対する角度を度単位で指定したい場合は、その角度を `[:<角度>]` 形式で、オプションの角度フィールドに書きます。〈角度〉は、必要な場合は [0, 360) の範囲におさまるように調整されます。

角度の直接指定

```
\chemfig{A-[:30]B-[:-75]C-[:10]D-[:90]>[:60]-[:-20]E-[:0]~[:-75]F}
```



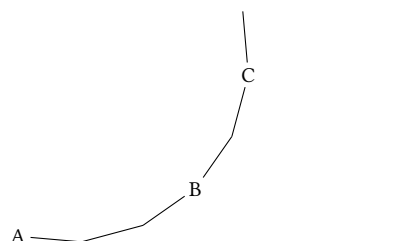
5.3 相対角度

直前の結合からの相対的な結合角度を指定できると便利な場合がよくあります。そのような場合は `[::<相対角度>]` 形式を使うことができます。〈相対角度〉の符号が + の場合は、その符号は省略できます。

次の例は、最初の結合の角度が -5° で、残りの結合角度が 20° ずつ増加する場合です。

相対角度の使用例

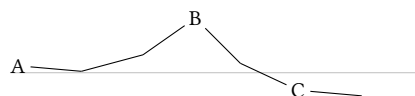
```
\chemfig{A-[:-5]-[::+20]-[:20]B-[:+20]-[:20]C-[:20]}
```



相対角度の連鎖は、角度を直接指定したり定義済みの角度を書くことで「断ち切る」ことができます。次の例では、原子“B”に 315° の結合が続いています。

相対角度の後に直接指定する場合の例

```
\chemfig{A-[:5]-[:20]-[:20]B-[7]-[:20]C-[:20]}
```



6 結合の長さ

実際には「結合の長さ」ではなく「原子間の距離」と言うべきでしょう。というのも、すでに 9 ページで述べたように、`atom sep` で設定される原子間の間隔のみが調節可能だからです。このパラメーターが設定されると、結合の長さは原子の種類に依存し、結合が水平となす角度にも幾分かは依存します。2つの「スリムな」原子の場合は、2つの太った原子の場合よりも、側面同士の間隔が大きくなります。次の例で明らかなように、“I”原子は“M”原子より幅が狭いので、“I”原子間の結合は“M”原子間の結合より長く描画されます。

原子の大きさの影響

```
\chemfig{I-I}\par
\chemfig{M-M}
```



原子の大きさの問題は、原子が下付き文字や上付き文字を含む場合に特に顕著になります。次の例では、結合が極端に短く、負号 $-$ と間違えてしまうかもしれません。

短すぎる結合

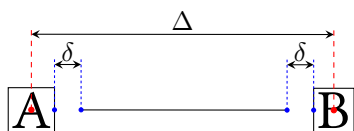
```
\chemfig{A^{++}_{2}-B^{-}_{3}}
```



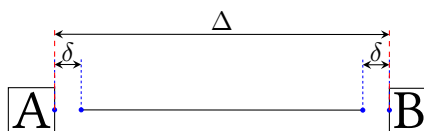
指数の $-$ は波括弧の中に書かれていることに注意してください。さもないと `chemfig` は原子をこの文字で区切って、結合の記号であると解釈します。そうすると原子は“B \wedge ”となり、予期しない結果となってしまいます。

`chemfig` の原子間間隔についての挙動を変更することができます。ブール値を取る〈キーワード〉`fixed length` を〈`true`〉に設定すると、〈キーワード〉`atom sep` すなわち Δ は、原子の中心間の距離ではなく**結合の長さ**として定義されます。これにより、結合は一定の長さを持つことになります。一方で原子の中心間の距離は原子の大きさに依存して変化することになります。9 ページの図において、`fixed length` のそれぞれの値で何が起こるかを示したのが次の図です。

`fixed length = <false>`



`fixed length = <true>`



環状化合物の場合は、環が正多角形となるように、`fixed length = <true>` であっても環の描画にはデフォルトの動作が使用されます。

固定長の結合

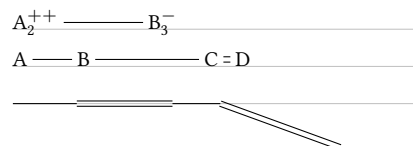
```
\chemfig{Cl-Cl}\par
\chemfig{fixed length=true}{Cl-Cl}
```



特にデフォルトの動作では、結合が短くなりすぎるのを避けるために、原子間距離を増やす（あるいは場合によっては減らす）ことが必要になる場合があります。そのために、結合のオプション引数は複数のフィールドを持っています。すでに見たように、最初のフィールドは角度を指定します。2つ目のフィールドは、もし空でなければ、デフォルトの原子間距離 Δ の倍率となります。したがって `-[,2]` と書くと、この結合は（最初のフィールドが空なので）デフォルトの角度のままで、結合する原子間の距離をデフォルトの2倍にする、という意味になります。

結合長の変更

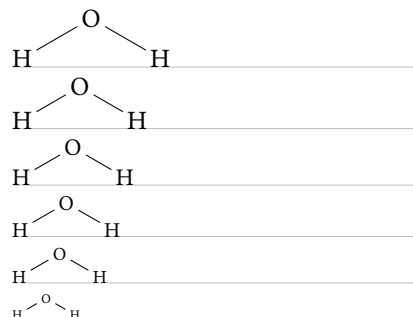
```
\chemfig{A^{++}_2-[,2]B^{-}_3}\par
\chemfig{A-B-[,2]C=[,0.5]D}\par
\chemfig{--[,1.5]-[,0.75]=[:-20,2]}
```



分子の大きさはフォントサイズや、〈キーワード〉 `atom sep`（あるいはその両方）を設定することで変更することができます。もし影響範囲を限定したい場合は、これらの変更がグループ内に限定されるように気を付けてください。

分子の大きさの変更

```
\normalsize \chemfig{H-[:30]O-[:-30]H}\par
\setchemfig{atom sep=2.5em}
\chemfig{H-[:30]O-[:-30]H}\par
\small \chemfig{H-[:30]O-[:-30]H}\par
\footnotesize \chemfig{H-[:30]O-[:-30]H}\par
\scriptsize \chemfig{H-[:30]O-[:-30]H}\par
\tiny \chemfig{H-[:30]O-[:-30]H}
```



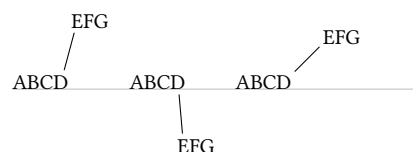
7 出発原子と到着原子

原子グループは複数の原子を含むことができます。いまグループ“ABCD”とグループ“EFG”を結合で繋ぎたいとしましょう。`chemfig` は最初のグループのどの原子と、2つ目のグループのどの原子を繋ぐかを、水平に対する結合の角度から計算します。もし角度が -90° より大きく 90° 未満 (360° の剰余) であれば、最初のグループの最後の原子と、2つ目のグループの最初の原子の間に結合を描きます。それ以外の場合は、最初のグループの最初の原子と、2つ目のグループの最後の原子の間に結合を描きます。

次の例はいずれも結合の角度が $(-90, 90)$ の範囲にある場合で、したがっていずれも D と E の間に結合が描かれます。

デフォルトの原子の結合

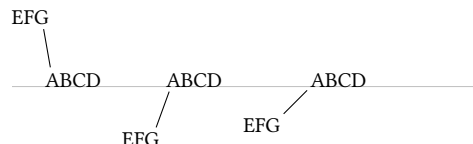
```
\chemfig{ABCD-[:75]EFG}\quad
\chemfig{ABCD-[:-85]EFG}\quad
\chemfig{ABCD-[1]EFG}
```



次の例は結合角度が $[90, 270]$ の範囲にある場合で、このとき結合は A と G の間に描かれます。

デフォルトの原子の結合

```
\chemfig{ABCD-[ :100]EFG}\quad
\chemfig{ABCD-[ :-110]EFG}\quad
\chemfig{ABCD-[ 5]EFG}
```



場合によっては、`chemfig` が計算した原子とは別の原子から結合を開始させたいことがあります。結合のオプション引数を使うことで、出発原子や到着原子を強制することができます。そのためには次のように記述します。

[, , <整数1> , <整数2>]

ここで <整数1> と <整数2> は繋ぎたい出発原子と到着原子の番号です。そのような原子が存在しない場合はエラーメッセージが出力されます。

結合する原子の指定

```
\chemfig{ABCD-[ :75,,2,3]EFG}\quad
\chemfig{ABCD-[ :75,,,2]EFG}\quad
\chemfig{ABCD-[ :75,,,3,2]EFG}
```



8 結合のカスタマイズ

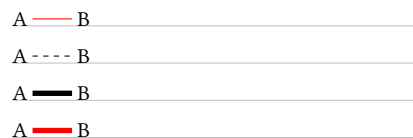
結合のオプション引数の 5 番目、すなわち最後のフィールドは、4 つ目のカンマの後にあります。

[,,, , <tikzコード>]

<tikz コード> は結合を描画する際に直接 TikZ に渡されます。ここには色 (red) や破線の種類 (dash pattern=on 2pt off 2pt)、太さ (line width=2pt) を指定したり、TikZ の decoration ライブラリーを読み込んでいれば、装飾を指定することもできます。“draw=none” とすれば結合は不可視になります。複数の属性を指定をするには、TikZ の構文と同様に、カンマで区切って記述します。

TikZ コードを渡す

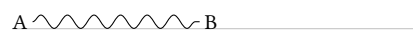
```
\chemfig{A-[,,, ,red]B}\par
\chemfig{A-[,,, ,dash pattern=on 2pt off 2pt]B}\par
\chemfig{A-[,,, ,line width=2pt]B}\par
\chemfig{A-[,,, ,red,line width=2pt]B}
```



TikZ では多数の decoration ライブラリーが利用可能です。例えば “pathmorphing” ライブラリーを読み込むと、結合に波線を使うことができます。 “pathmorphing” ライブラリーを読み込むには、プリアンブルに `\usetikzlibrary{decorations.pathmorphing}` と書きます。

波線の結合

```
\chemfig{A-[ ,3,,, ,decorate,decoration=snake]B}
```



くさび形結合では太さと破線の設定は無視されます。

9 結合同士の接合

デフォルトでは、単結合を表す線は接合されないため、線幅が大きいと目立ちすぎて見苦しい場合があります。

これを「醜い³」と感じる場合は、単結合同士を接合することができますが、この場合はコンパイル時間が少し増加します。これを有効にするには〈キーワード〉`bond join = 〈真偽値〉` マクロに〈`true`〉を設定します。デフォルトは、望ましい動作である〈`false`〉です。

結合同士の接合

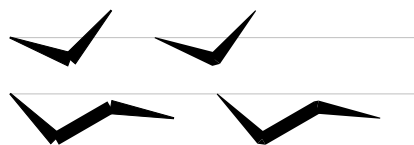
```
\setchemfig{bond style={line width=3pt}}
\chemfig{-[1]-[7]}と
\chemfig[bond join=true]{-[1]-[7]}
```



この問題がより顕著になるのは、くさび形結合が単結合や別のくさび形結合に繋がる場合です。この場合も同様に `bond join` を〈`true`〉に設定することで、より美しく接合されます。

くさび形結合の接合

```
\chemfig{<[: -20]>[:50]}\quad
\chemfig[bond join]{<[: -20]>[:50]}
\medbreak
\setchemfig{cram width=5pt}
\chemfig{<[: -45]-[:30,,,line width=5pt]>[: -10]}\quad
\chemfig[bond join]{<[: -45]-[:30,,,line width=5pt]>[: -10]}
```



以下の2点に注意してください。

- `bond join` が〈`true`〉のとき、くさび形結合はアウトライン無しで、すなわちほんの少し細く描画されます。(それが重要になるかどうかは〈`line width`〉パラメーターに依存します。)
- `bond join` は単結合とくさび形結合の間では、くさび形結合の三角形の底辺の2頂点が単結合の平行な2辺で区切られた空間の外側にあるときには効果がありません。数学的に言うと、〈`cram width`〉の値 d 、単結合の幅 w 、2つの結合の角度 α について、 $d \cos \alpha > w$ が成立する場合です。

くさび形結合の接合

```
\setchemfig{cram width=5pt, bond join}
\chemfig{[,,,,line width=3pt]>[:30]}\quad
\chemfig{[,,,,line width=3pt]>[:65]}
```



10 デフォルト値

各分子の最初で、以下のオプション引数のデフォルト値が初期化されます。

- 結合角度： 0°
- 長さの倍率：1
- 出発原子と到着原子の番号：〈`empty`〉。つまり `chemfig` は結合角に基づいてこれらを決定します。
- TikZ に渡されるパラメーター：〈`empty`〉

分子コードの最初に

[〈角度〉, 〈係数〉, 〈 n_1 〉, 〈 n_2 〉, 〈tikzコード〉]

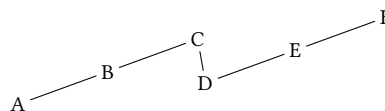
³<http://tex.stackexchange.com/questions/161796/ugly-bond-joints-in-chemfig> 参照。

と書くことで、デフォルト値を対象分子全体に対して変更することができます。

したがって、分子コードが[:20,1.5]で始まっている場合、全ての結合の角度のデフォルトは20°に、原子間距離はデフォルトの1.5倍の長さになります。これらのデフォルト値は、次の例の原子“C”に続く結合のように、オプションの引数を与えることでいつでも上書きすることができます。

デフォルト値の上書き

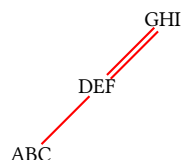
```
\chemfig{[:20,1.5]A-B-C-[:-80,0.7]D-E-F}
```



もし[1,1.5,2,2,red,thick]のような奇妙なことを書いた場合、特に指示がない限り、全ての結合の角度は45°になり、原子間距離はデフォルトの距離の1.5倍になり、結合は各グループの2番目の原子から始まって2番目の原子で終わり、結合は赤い太線になります。

デフォルト値

```
\chemfig{[1,1.5,2,2,red,thick]ABC-DEF=GHI}
```



11 側鎖

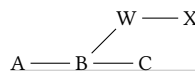
11.1 原理

ここまでの分子は全て直鎖状でしたが、そのようなことはめったにありません。部分分子を原子に繋げるには、対象の原子のうしろに括弧を書き、その中に〈コード〉を書きます。この〈コード〉はその原子に繋げたい部分分子の記述です。

次の例では、部分分子“-[1]W-X”が原子“B”に繋がっています。

側鎖

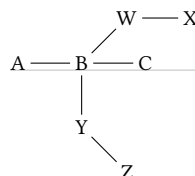
```
\chemfig{A-B(-[1]W-X)-C}
```



同じ原子に複数の部分分子が繋がることもあります。その場合、それぞれの部分分子のコードを括弧で囲んで書きます。

複数の側鎖

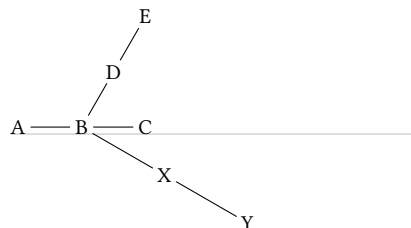
```
\chemfig{A-B(-[1]W-X)(-[6]Y-[-7]Z)-C}
```



各部分分子のコードには、その部分分子全体で有効なデフォルト値を定義することができます。次の例では部分分子[:60]-D-Eが原子“B”に繋がっており、そのデフォルトの角度は水平から60°です。さらに2つ目の部分分子[::-30,1.5]-X-Yが原子“B”に繋がっていて、デフォルトの結合角度はその直前の結合（この例では“A”と“B”の結合）に対し時計回りに30°で、原子間距離がデフォルト値の1.5倍になっています。

側鎖でのデフォルト値

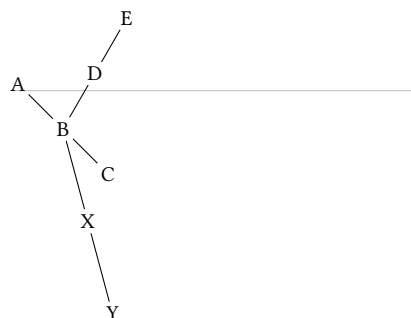
```
\chemfig{A-B([:60]-D-E)([:-30,1.5]-X-Y)-C}
```



主分子の最初に“[:-45]”と書くとうなるか見てみましょう。

デフォルトの結合角度の影響

```
\chemfig{[:-45]A-B([:60]-D-E)([:-30,1.5]-X-Y)-C}
```



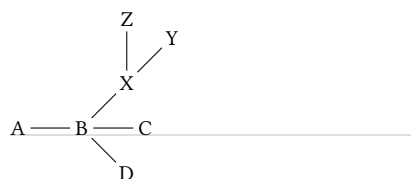
この例から、結合 B-C と結合 B-X の間の角度は 30° のままであることがわかります。これは部分分子“-X-Y”に相対角度が指定されているからです。これとは対照的に、側鎖“-D-E”は水平に対して 60° 傾いていて、最初に書かれている -45° の回転になっていません。これは予期した動作です。というのも、“-D-E”には水平に対する角度が直接指定されているからです。分子全体を回転させるためには、全ての角度を相対角度で指定する必要があります。

11.2 入れ子

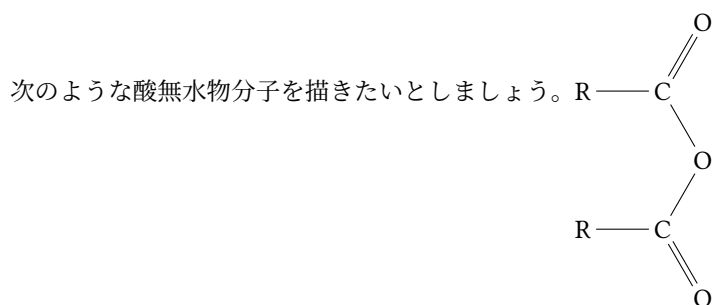
部分分子は入れ子にすることができ、上で見た規則がそのまま適用されます。

入れ子の側鎖

```
\chemfig{A-B([1]-X([2]-Z)-Y)(-[7]D)-C}
```



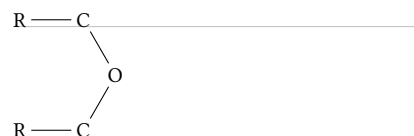
11.3 描き方のヒント



このとき、最も長い鎖を見つけるのが良いでしょう。ここでは例えば R-C-O-C-R 鎖を角度を考慮して描きます。ここでは相対角度だけを使用しています。

酸無水物構造

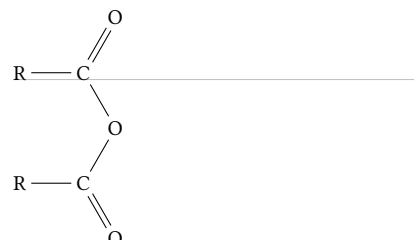
```
\chemfig{R-C-[::-60]O-[::-60]C-[::-60]R}
```



この構造に 2 つの部分分子 “=O” をそれぞれの炭素原子に付けます。

酸無水物

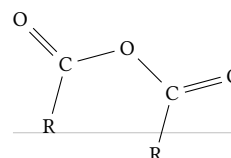
```
\chemfig{R-C(=[:+60]O)-[::-60]O-[::-60]C(=[:+60]O)-[::-60]R}
```



相対角度だけを使って書いたので、デフォルトの角度、例えば 75° を指定すると、分子全体を回転することができます。

分子の回転

```
\chemfig{[:75]R-C(=[:+60]O)-[::-60]O-[::-60]C(=[:+60]O)-[::-60]R}
```



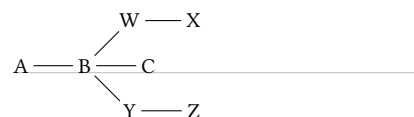
12 離れた原子を繋げる

ここまで、コード内で隣接している原子同士を繋げる方法を見てきました。しかし、コード内で隣接していない原子同士を繋ぎたい場合もあります。このような特殊な結合を「遠い結合」と呼ぶことにしましょう。

次の分子を例にとりましょう。

分岐構造

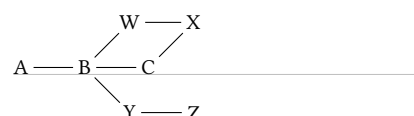
```
\chemfig{A-B(-[1]W-X)(-[7]Y-Z)-C}
```



そして X と C を繋ぎたいとしましょう。このような場合、chemfig では「フック」を目的の原子の直後に置くことができます。フックを表わす文字には、フックの形に似ているので “?” を使います。つまり、X? と書くと、原子 X はフックを持つことになります。そのコード内ではこれ以降、直後に? が続く原子は全て X に連結されます。

遠い結合

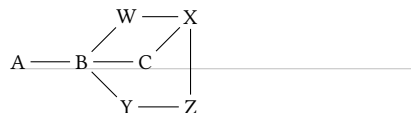
```
\chemfig{A-B(-[1]W-X?)(-[7]Y-Z)-C}
```



それ以外の原子も、それに続けて? を書けば X に繋げることができます。次の例では原子 C に加えて、Z も X に繋いでいます。

複数の遠い結合

```
\chemfig{A-B(-[1]W-X?)(-[7]Y-Z?)-C?}
```



もし遠い結合 X-C と X-Z をそのままにした上で、さらに別の遠い結合 A-W を描きたい場合はどうすればよいでしょうか。この場合は A と X のそれぞれに 2 つの異なるフックが必要になります。幸いなことに、文字 ? はオプション引数を取ることができます。

?[⟨名前⟩,⟨結合⟩,⟨tikz⟩]

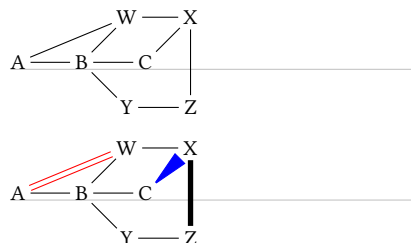
フィールドが空の場合は、それぞれの以下のデフォルト値を取ります。

- ⟨名前⟩ はフックの名前です。全ての英数字 (a...z、A...Z、0...9) を使うことができます⁴。名前のデフォルトは a です。指定された名前が初めて使われる場合のみ、このフィールドが使用されます。
- ⟨結合⟩ は、その原子とその名前のフックを持つ原子をどのように繋ぐかを指定します。それには 2 つの方法があります。1 つ目は、使いたい結合の種類を整数で指定する方法です。例えば単結合なら 1、二重結合なら 2 を指定します。(各結合の番号は 9 ページの表を参照してください)
2 つ目は、結合を表す文字のうちの 1 つを波括弧内に書く方法です。
- ⟨tikz⟩ は、すでに見た通常の結合の場合と同様に、TikZ に直接渡されます。

次の例は必要な遠い結合を加えた分子と、その上でさらに結合 A-W および X-C をカスタマイズした例です。

複数の遠い結合

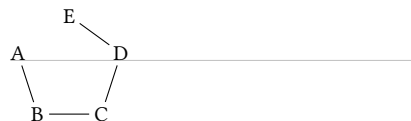
```
\chemfig{A?[a]-B(-[1]W?[a]-X?[b])(-[7]Y-Z?[b])-C?[b]}\par\medskip
\chemfig{A?[a]-B(-[1]W?[a,2,red]-X?[b])(-[7]Y-Z?[b,1,{line width=2pt}])-C?[b,>,blue]}
```



複数の異なるフックを 1 つの原子の後に書くこともできます。次の未完成の五角形を見てみましょう。いま、A-E と A-C、E-C を繋げたいものとします。

不完全な環

```
\chemfig{A-[:72]B-C-[:72]D-[:144]E}
```



そのためには、次のように書きます。

複数の遠い結合

```
\chemfig{A?[a]-[:72]B-C?[a]?[b]-[:72]D-[:144]E?[a]?[b]}
```



⁴これは正確ではありません。実際には \csname... \endcsname の間に書くことのできる全ての文字を使用することができます。

13 環

前の例では正多角形を描く方法を見ましたが、角度が多角形の辺の数に依存するのでこの方法は面倒です。

13.1 構文

chemfig では正多角形を簡単に描くことができます。そのために、この環の外側にある〈原子〉に環を付ける、と考えます。その構文は次の通りです。

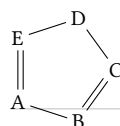
$$\langle \text{原子} \rangle * \langle n \rangle (\langle \text{コード} \rangle)$$

〈n〉は多角形の辺の数で、〈コード〉は環を構成する辺と頂点となる結合および原子グループの記述です。このコードは結合で始まらなくてはなりません。なぜなら、原子は環の外にあるからです。

次の例は原子“A”に付けられた五員環です。

五員環

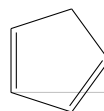
```
\chemfig{A*5(-B=C-D-E=)}
```



また、環以外の場合と同様に、1つまたは複数、あるいは全ての原子グループを空にした環を描くこともできます。

空の原子グループからなる五員環

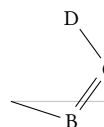
```
\chemfig{*5(-----)}
```



環は不完全でも構いません。

不完全な五員環

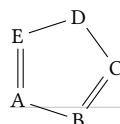
```
\chemfig{*5(-B=C-D)}
```



与えられた頂点数に対して結合や原子グループが多すぎる場合、過剰な結合と原子グループは全て無視されます。

切り詰められた五員環

```
\chemfig{A*5(-B=C-D-E=F-G=H-I)}
```



環の内側に円や弧を描くこともできます。そのためには次の構文を使います。

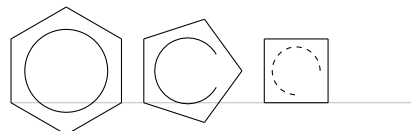
$$\langle \text{原子} \rangle ** [\langle \text{角度1} \rangle, \langle \text{角度2} \rangle, \langle \text{tikz} \rangle] \langle n \rangle (\langle \text{コード} \rangle)$$

ここで、オプション引数の各フィールドが空の場合は、以下のデフォルト値を取ります。

- ・〈角度1〉と〈角度2〉は弧の開始角度と終了角度です。デフォルトはそれぞれ 0° と 360° で、したがってデフォルトでは円が描かれます。
- ・〈tikz〉は弧を描く際に TikZ に渡されるコードです。

環と弧

```
\chemfig{**6(-----)}\quad
\chemfig{**[30,330]5(-----)}\quad
\chemfig{**[0,270,dash pattern=on 2pt off 2pt]4(-----)}
```



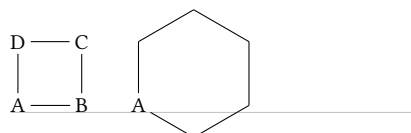
13.2 環の角度の調整

13.2.1 環の先頭での調整

次の例でわかるように、環の基準原子“A”は常に環の南西に置かれます。さらに、環は常に反時計回りに描かれ、最後の結合は基準原子に上から垂直に繋がります。

環の角度

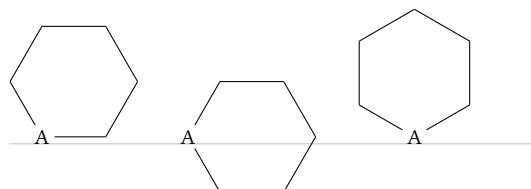
```
\chemfig{A*4(-B-C-D-)}\quad\chemfig{A*6(-----)}
```



この角度が望ましくない場合は、分子の先頭のオプション引数で別の角度を指定することができます。次の図は $+30^\circ$ 、 -30° および $+60^\circ$ 回転させた六員環の例です。

環の回転

```
\chemfig{[:30]A*6(-----)}\quad
\chemfig{[:-30]A*6(-----)}\quad
\chemfig{[:60]A*6(-----)}
```



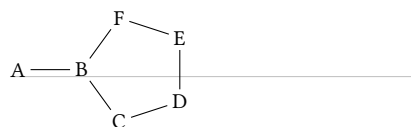
13.2.2 結合の後での調整

分子の記述が環以外で始まっていて 1 つ以上の結合がすでに描かれている場合は、デフォルトの角度が変更され、環の基準原子へ向かう結合が、環の最初と最後の辺がなす角度を二等分するように描かれます。

簡単な場合は次のようになります。

環に繋がる結合

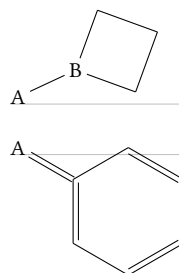
```
\chemfig{A-B*5(-C-D-E-F-)}
```



この規則は、環の直前の結合の角度によらず有効です。

環に繋がる結合

```
\chemfig{A-[ :25]B*4(-----)}\vskip5pt
\chemfig{A=[ :-30]*6(=---=)}
```



13.3 環の側鎖

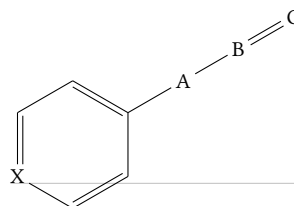
環の頂点につく側鎖を描くには、すでに見た構文を使用します。

〈原子〉(〈コード〉)

ここで〈コード〉は部分分子の記述で、〈原子〉は頂点となる原子です。環に特有なこととして、部分分子のデフォルトの角度は 0° ではなく、頂点を出る結合が、その頂点に繋がる環の 2 辺がなす角度を二等分するように計算されます。

環の側鎖

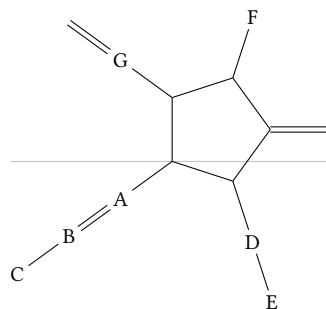
```
\chemfig{X*6(--(-A-B=C)---)}
```



部分分子を、他の頂点と同様に、環の最初の頂点に繋げることもできます。

環と側鎖

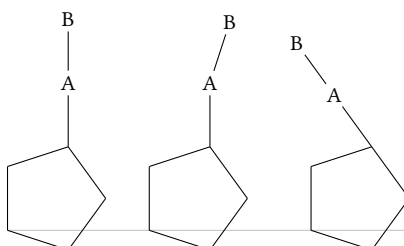
```
\chemfig{*5((-A=B-C)-(-D-E)-(=)-(-F)-(-G=)-)}
```



もし頂点から出る結合が、環の 2 辺がなす角を二等分しないようにしたい場合は、オプションのグローバルパラメーターか結合のオプションパラメーターで調整できます。

指定された角度での側鎖

```
\chemfig{*5(---([ :90]-A-B)---)}\quad
\chemfig{*5(---(-[ :90]A-B)---)}\quad
\chemfig{*5(---([ :+0]-A-B)---)}
```

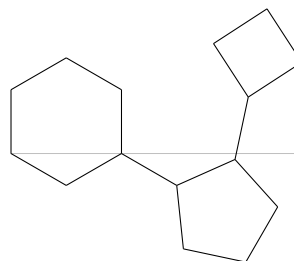


上の例の 3 番目の分子では側鎖の結合は相対角度 0° と与えられていて、側鎖の結合は直前の環の結合と同一直線上に描かれていることに注目してください。これは、11 ページの規則により、最後に描かれた結合の角度を基準としているからです。

環同士を結合で繋ぐこともできます。

繋った環

```
\chemfig{*6(--(*5(----(*4(----))--))----)}
```

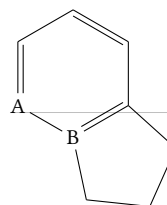


13.4 入れ子の環

2 つの環を「接着」する場合は、ほんの少し異なる構文を使用します。まず、もう一方の環が始まる頂点を特定します。そしてこの頂点の後に、通常の環の構文を続けるだけです。例えば、六員環の 2 番目の頂点から始まる五員環は次のようになります。

入れ子の環

```
\chemfig{A*6(-B*5(----)=--=)}
```

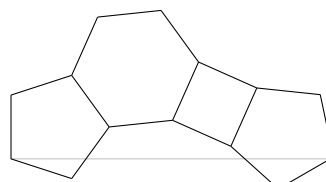


2 つ目の環は 1 つ目の環に対して、2 つの側面が一致するような角度で描かれていることに注意してください。さらに、五員環には 4 つの結合 “----” しかありません。実際 5 つ目の結合は、すでに描かれている六員環の 2 番目の辺なので必要ありません。

複数の環を接着することもできます。

複数の入れ子の環

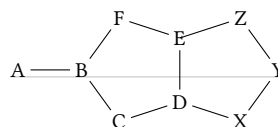
```
\chemfig{*5(--*6(*4(*5(----)--))----)}\chemfig{*5(--*6(*4(*5(----)--))----)}
```



場合によっては小技を使わなければなりません。次の例では、2 番目の五員環の 4 つ目の辺が原子 “E” の中心に達しています。

欠陥のある図

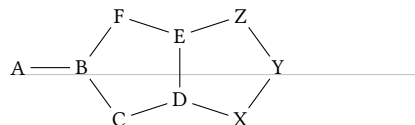
```
\chemfig{A-B*5(-C-D*5(-X-Y-Z)-E-F-)}
```



これは正常です。なぜなら（“D” に付加されている）2 番目の五員環は、`chemfig` が原子 “E” について知る前に描かれているからです。この場合、結合 Z-E を 2 つのフックを使用して描く必要があります。

遠い結合と環

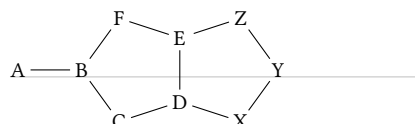
```
\chemfig{A-B*5(-C-D*5(-X-Y-Z?)-E?-F-)}
```



五員環の最後の頂点に `` を使う方法もあります。

`\phantom` を使う方法

```
\chemfig{A-B*5(-C-D*5(-X-Y-Z-\phantom{E})-E-F-)}
```

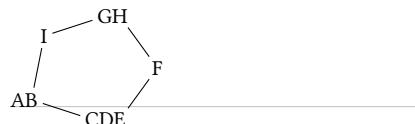


13.5 環と原子グループ

環の頂点が原子グループで構成されている場合は注意が必要です。

環と原子グループ

```
\chemfig{AB*5(-CDE-F-GH-I-)}
```



この環を正多角形にするには、`chemfig` によって決定された結合の出発原子と到着原子を上書きする必要があります。ここでは C-F と F-G を繋ぐ必要があります、そのことをこれらの結合のオプション引数で指定します。

出発原子と到着原子の指定

```
\chemfig{AB*5(-CDE-[,,,1]F-[,,,1]GH-I-)}
```



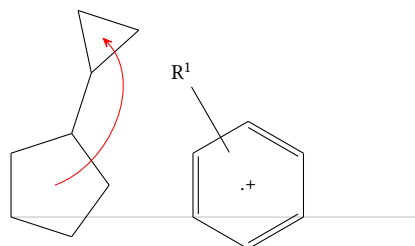
13.6 環の中心

各環にはその中心に、`centrecycle<n>` という名前の大きさを持たないノードが配置されます。ここで `<n>` は環の番号です。（環は描画された順に番号が付けられます。）ブール値 `show cntcycle` を `true` に設定することで、各環の番号を表示することができます。

ブール値を取る `autoreset cntcycle`（デフォルトは `true`）は、各分子の最初に（つまり各 `\chemfig` の実行時に）環のカウンターを 0 にリセットするかどうかを設定します。

環の中心

```
\chemfig{*5(---(-*3(---))---)}
\chemmove{\draw[red](cyclecenter1)to[out=20,in=-45](cyclecenter2);}
\quad
\chemfig{*6(==--=)}
\chemmove{%
  \node[at=(cyclecenter1)]{.+}
  \node[at=(cyclecenter1),shift=(120:1.75cm)](end){\printatom{R^1}};
  \draw[-,shorten <=.5cm](cyclecenter1)--(end);
}
```



14 電子の動きの表現

chemfig バージョン 0.3 以降では、共鳴効果や反応機構における電子の動きを描くことができます。そのためには構文 “@{<引数>}” を使用して、電子の動きを表す矢印の出発点と到着点にマークを付けます。この記述により TikZ ノードを配置し、\chemfig コマンドの引数の外部からこのノードを参照できるようになります。これは、全ての “tikzpicture” 環境に “remember picture” オプションが渡されているからです。そのため、コンパイルが 2 回行われることが前提となります。

具体的には以下の 2 つのケースが考えられます。

- 対象の**結合**に大きさのないノードを配置します。そのためには、その結合のオプション引数の最初に “@{<名前>,<係数>}” と書きます。1 つ目のオプション引数がある場合は、カンマを付けずに “@{<名前>,<係数>}” の後に続けます。このノードは “<名前>” という名前と、ノードの結合上の位置を指定する 0 と 1 の間の値 <係数> を取ります。単に “@{<名前>}” と書かれた場合、<係数> はデフォルトの 0.5 に設定され、ノードは結合の中央に配置されます。
- 対象の**原子**上にノードを配置します。そのためには、原子の直前に “@{<名前>}” と書きます。このノードは原子と全く同じ大きさですが、原子が空の場合は大きさを持ちません。

\chemfig コマンドが分子を描画し、上記の構文によるノードを配置した後は、TikZ の機能を使ってこれらのノードを相互に繋げることができます。そのための指示は、(例えば) ノード “<名前1>” とノード “<名前2>” を繋ぎたい場合、次の構文で \chemmove コマンドの引数に書きます⁵。

```
\chemmove[<オプション>]{\draw[<tikzオプション>](<名前1>)<tikzリンク>(<名前2>);}
```

\chemmove のオプション引数 <オプション> は、ノード間のリンクが描画される tikzpicture 環境の引数に追加されます。<tikzオプション> および <tikzリンク> は、TikZ パッケージのドキュメントで詳細に説明されています。

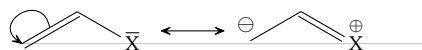
14.1 共鳴効果

上記の考え方をより具体化するために、二重結合と孤立電子対の共役による共鳴効果の例を取り上げます。まず二重結合の電子の非局在化を考えます。二重結合の中央に “db” (double bond) という名前のノードを、二重結合の終端に “a1” という名前のノードをそれぞれ配置します。

\schemestart、\schemestop、\arrow および \+ マクロについては、57 ページから始まる第 IV 部で説明します。

共鳴効果 1

```
\schemestart
\chemfig{@{a1}=_[@{db}::30]-[::-60]\charge{90=|}{X}}
\arrow{<->}
\chemfig{\chemabove{\vphantom{X}}{\ominus}-[::30]=_[[::-60]
\chemabove{X}{\scriptstyle\oplus}}
\schemestop
\chemmove{\draw(db).. controls +(100:5mm) and +(145:5mm).. (a1);}
```



上で述べた通り、結合のオプション引数に書かれたノードの後にカンマがないことに注意してください。つまり “=_[@{db}::30]” と書きます。“=_[@{db},::30]” と書きたくなるかもしれませんがこのようには書けません。

ノード “db” と “a1” を繋ぐために、次の構文を使用しました。

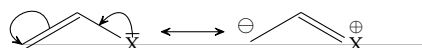
⁵実際には \chemmove コマンドは、その引数を “remember picture, overlay” オプションと共に “tikzpicture” 環境に渡します。

```
\chemmove{\draw(db).. controls +(100:5mm) and +(145:5mm).. (a1);}
```

\chemmove で使われる矢尻のデフォルトは“CF”です。この例では、2つの制御点を使用しています⁶。この制御点はいずれも極座標表示で、最初のものが“db”から 100° 方向に 5 mm の位置に、2 番目のものが“a1”から 145° 方向に 5 mm の位置に配置されています。この構文は最初は複雑に見えるかもしれませんが、たいいてい場合は単にコピー&ペーストすれば良いだけなので、警戒する必要はありません。次の例で分かるように、変更する必要があるのは制御点の名前と座標だけです。この例では孤立電子対（ノード“dnl”）から単結合（ノード“sb”）への矢印が追加されています。

共鳴効果 2

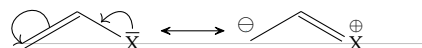
```
\schemestart
\chemfig{@{a1}=_[@{db}::30]-[@{sb}::-60]@{dnl}\charge{90=\!}{X}}
\arrow{<->}
\chemfig{\chemabove{\vphantom{X}}{\ominus}-[:30]=_[::-60]}
\chemabove{X}{\scriptstyle\oplus}
\schemestop
\chemmove{
\draw(db)..controls +(100:5mm) and +(145:5mm).. (a1);
\draw(dnl)..controls +(90:4mm) and +(45:4mm).. (sb);}
```



2 回目の矢印のために、“dnl”から 90° 方向に 4 mm、“sb”から 45° 方向に 4 mm の 2 つの制御点を設定しました。しかし、矢印が孤立電子対を表す線に触れてしまっており、まだ満足な結果になっていません。これを改善するために、矢印にいくつかのオプションを追加します。

共鳴効果 3

```
\schemestart
\chemfig{@{a1}=_[@{db}::30]-[@{sb}::-60]@{dnl}\charge{90=\!}{X}}
\arrow{<->}
\chemfig{\chemabove{\vphantom{X}}{\ominus}-[:30]=_[::-60]}
\chemabove{X}{\scriptstyle\oplus}
\schemestop
\chemmove[>]{% 矢尻のスタイルを変更
\draw(db).. controls +(100:5mm) and +(145:5mm).. (a1);
\draw[shorten <=3pt,shorten >=1pt](dnl) .. controls +(90:4mm)
and +(45:4mm) .. (sb);}
```

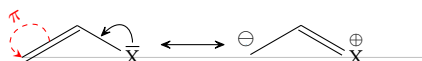


“shorten <=3pt” オプションは矢印の尾部を 3 pt 短く、同様に“shorten >=1pt”は矢印の先端を 1 pt 短くすることを意味します。

TikZ の全ての力を駆使して、矢印のスタイルを変更できます。次の例では、二重結合から出る矢印の矢尻を“-stealth”に変更し、矢印を細い破線の赤線で描画しています。また、矢印の中央の上部に文字 π を追加しています。

共鳴効果 4

```
\schemestart
\chemfig{@{a1}=_[@{db}::30]-[@{sb}::-60]@{dnl}\charge{90=\!}{X}}
\arrow{<->}
\chemfig{\chemabove{\vphantom{X}}{\ominus}-[:30]=_[::-60]}
\chemabove{X}{\scriptstyle\oplus}
\schemestop
\chemmove{
\draw[-stealth,thin,dash pattern= on 2pt off 2pt,red]
(db).. controls +(100:5mm) and +(145:5mm)..
node[sloped,above] {$\pi$} (a1);
\draw[shorten <=3pt, shorten >= 1pt]
(dnl).. controls +(90:4mm) and +(45:4mm).. (sb);}
```

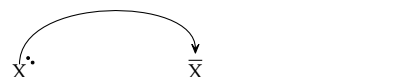


⁶2 つのノードを TikZ で繋ぐ他の方法については、TikZ のドキュメントを参照してください。

次に、矢印の出発点や到着点の位置を指定する方法を見ていきましょう。

出発・到着点のアンカーポイント 1

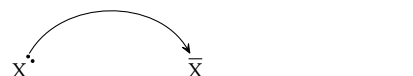
```
\chemfig{@{x1}\charge{45=\:}{X}}
\hspace{2cm}
\chemfig{@{x2}\charge{90=\|}{X}}
\chemmove{
  \draw[shorten >=4pt](x1).. controls +(90:1cm) and +(90:1cm).. (x2);}
```



この例では、電子から出る矢印の尾部が、正しい位置から出発していないことに注意してください。つまりノードの上辺の中央から出発してしまっています。実際、出発角として 90° を指定するので、TikZ はこの矢印をアンカー “x1.90” から出発させています。これは、ノード “x1” の中心から水平に対して 90° の角度で発する光線と、矩形ノードの辺との交点に対応しています。望む矢印の出発角度を得るためには、その位置を指定する必要があります。何度かの試行錯誤の結果、それは “x1.57” と分かりました。

出発・到着点のアンカーポイント 2

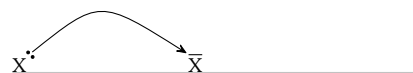
```
\chemfig{@{x1}\charge{45=\:}{X}}
\hspace{2cm}
\chemfig{@{x2}\charge{90=\|}{X}}
\chemmove[shorten <=4pt,shorten >=4pt]{
  \draw(x1.57).. controls +(60:1cm) and +(120:1cm).. (x2.90);}
```



場合によっては、制御点をデカルト座標で指定する方が簡単かもしれません。次の例では、“x1” から右に 1 cm、上に 0.8 cm の位置に配置された単一の制御点を使用しています。

単一の制御点

```
\chemfig{@{x1}\charge{45=\:}{X}}
\hspace{2cm}
\chemfig{@{x2}\charge{90=\|}{X}}
\chemmove[shorten <=4pt,shorten >=4pt]{
  \draw(x1.57).. controls +(1cm,.8cm).. (x2.90);}
```



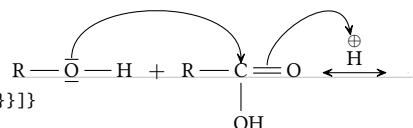
\chemmove コマンドによる描画は全て重ね合わされ、バウンディングボックスには含まれません。

14.2 反応機構

全ての “tikzpicture” 環境に remember picture オプションが渡されるので、反応機構を示す矢印を簡単に描くことができます。エステル化反応の最初のステップを例に取りましょう。

エステル化反応 ステップ 1

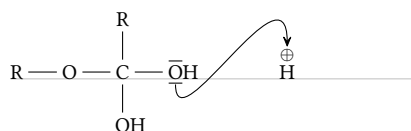
```
\setchemfig{atom sep=7mm}
\schemestart
  \chemfig{R-@{dn1}\charge{90=\|,-90=\|}{O}-H}
  \+
  \chemfig{R-@{atoc}C([6]-OH)=[@{db}]O}
  \arrow(.mid east--){<->[\chemfig{@{atoh}\chemabove{H}{\scriptstyle\oplus}}]}
\schemestop
\chemmove[shorten <=2pt]{
  \draw(dn1)..controls +(90:1cm)and+(north:1cm)..(atoc);
  \draw[shorten >=6pt](db)..controls +(north:5mm)and+(100:1cm)..(atoh);}
```



\chemabove{<コード>}{<物質>} コマンドは、<コード> のバウンディングボックスの寸法を変更しません。そのため、電荷を表す記号 (\oplus あるいは \ominus) への矢印を描くのはいくらか苦労することがあります。上の例での解決法は “atoh” から 110° 方向に 1 cm の位置に制御点を作成し、矢印を 6 pt 短くすることです。次のエステル化反応の第 2 ステップを描いた例では、矢印がもっと複雑な形を取り得ることがわかります。コード自体は複雑になっていないことに注意してください。

エステル化反応 ステップ 2

```
\setchemfig{atom sep=7mm}
\chemfig{R-O-C(-[2]R)(-[6]OH)-@{\dnl}\charge{90=\|,-90=\|}{O}H}\hspace{1cm}
\chemfig{@{atoh}\chemabove{H}{\scriptstyle\oplus}}
\chemmove{
  \draw[shorten <=2pt, shorten >=7pt]
    (dnl).. controls +(south:1cm) and +(north:1.5cm).. (atoh);}
```



残りは読者の課題としておきます。

15 分子の下に分子名を書く

便利な機能として chemfig では、次のコマンドで分子の下に名前を書くことができます。

```
\chemname[⟨長さ⟩]{\chemfig{⟨分子コード⟩}}{⟨名前⟩}
```

⟨長さ⟩ は分子のベースラインと ⟨名前⟩ の文字の上部との間隔で、デフォルトは 1.5 ex です。⟨名前⟩ は分子に対して中央揃えとなります。複数の段落を含めることはできません。このように書いてみると分かるように H—O—H、分子の下に表示される ⟨名前⟩ は、バウンディングボックスの垂直サイズのみが反映されま

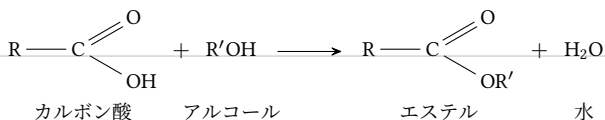
The water molecule (水分子) : H₂O

す。⟨名前⟩ の水平サイズは常にゼロです。

以下は反応に出てくる各分子の下に名前を表示する例です。

分子名の表示

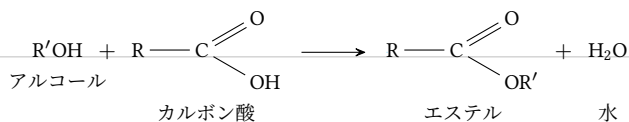
```
\schemestart
  \chemname{\chemfig{R-C(-[:30]OH)=[:30]O}}{カルボン酸}
  \+
  \chemname{\chemfig{R'OH}}{アルコール}
  \arrow(.mid east--.mid west)
  \chemname{\chemfig{R-C(-[:30]OR')=[:30]O}}{エステル}
  \+
  \chemname{\chemfig{H_2O}}{水}
\schemestop
\chemnameinit{}
```



このコマンドにはいくつかの制約があります。上の例で左辺の酸とアルコールを入れ替えてみましょう。

名前の整列 1

```
\schemestart
  \chemname{\chemfig{R'OH}}{アルコール}
  \+
  \chemname{\chemfig{R-C(-[:30]OH)=[:30]O}}{カルボン酸}
  \arrow(.mid east--.mid west)
  \chemname{\chemfig{R-C(-[:30]OR')=[:30]O}}{エステル}
  \+
  \chemname{\chemfig{H_2O}}{水}
\schemestop
\chemnameinit{}
```



コマンド `\chemname` は $\langle\text{name}\rangle$ を描画するために、各分子のベースライン（このマニュアルの例では薄い灰色）の下に $1.5\text{ex} + \text{これまでの分子の深さの最大値}^7$ の空きを挿入します。`\chemnameinit{要素}` コマンドはこの最大深さを $\langle\text{要素}\rangle$ で初期化します。したがって、望ましい出力を得るには、次のようにしなければなりません。

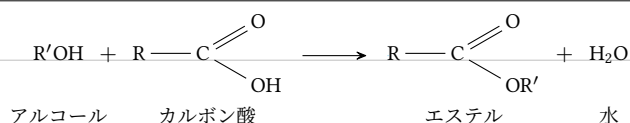
- 反応が最も深い分子で始まらない場合は、反応内で `\chemname` コマンドを使う前に `\chemnameinit{最も深い分子}` を書きます。
- 全ての分子の名前を書いた後に `\chemnameinit{}` でこの深さを初期化して、この反応での最大深さが、以降の別の反応に影響を与えないようにします。

最大深さは、`gchemname = true` の場合はグローバルであり、これはデフォルトの動作です。それ以外の場合はローカルになります。

したがって、正しいコードでは反応の前後に `\chemnameinit` を使用します。

名前の整列 2

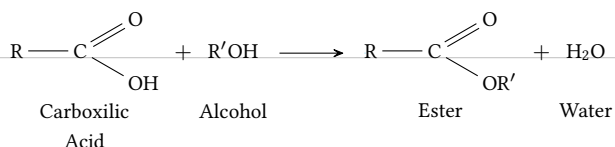
```
\chemnameinit{\chemfig{R-C(-[:30]OH)=[:30]O}}
\schemestart
  \chemname{\chemfig{R'OH}}{アルコール}
  \+
  \chemname{\chemfig{R-C(-[:30]OH)=[:30]O}}{カルボン酸}
  \arrow(.mid east--.mid west)
  \chemname{\chemfig{R-C(-[:30]OR')=[:30]O}}{エステル}
  \+
  \chemname{\chemfig{H_2O}}{水}
\schemestop
\chemnameinit{}
```



最後に、名前を複数行にわたって書きたい場合は、 $\langle\text{名前}\rangle$ 内で `\\` コマンドを使用して改行します⁸。

2 行に渡る名前

```
\schemestart
  \chemname{\chemfig{R-C(-[:30]OH)=[:30]O}}{Carboxilic\\Acid}
  \+
  \chemname{\chemfig{R'OH}}{Alcohol}
  \arrow(.mid east--.mid west)
  \chemname{\chemfig{R-C(-[:30]OR')=[:30]O}}{Ester}
  \+
  \chemname{\chemfig{H_2O}}{Water}
\schemestop
\chemnameinit{}
```



`\chemname*{名前}` マクロは、それ以前の名前を考慮しません。

⁷TeX 用語では、深さはベースラインの下に垂直に拡張される寸法です。

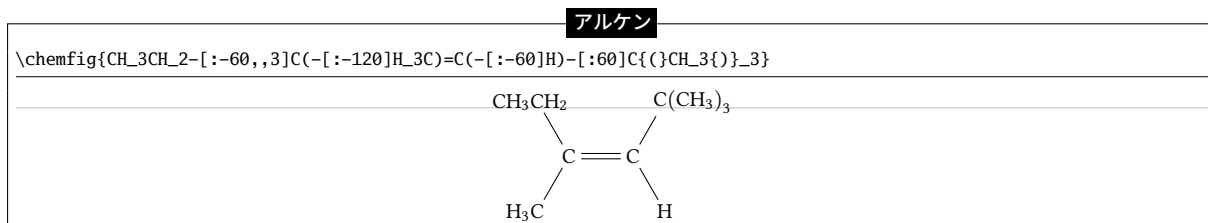
⁸`\par` コマンドは使えません。使用した場合はコンパイルエラーとなります。

高度な使用方法

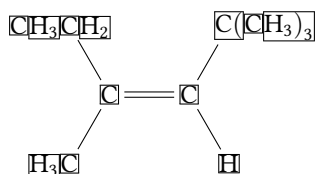
1 原子の分割

すでに述べた原子分割機構は、次に出現する大文字または日 目 罫 罫 罫 罫 罫 のいずれかの直前までを 1 つの原子として扱います。

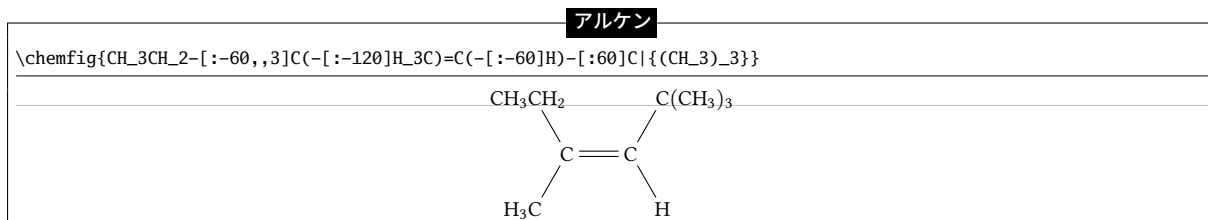
特定のケースではこの自動分割が不正確な原子を生成し、不完全な図が生成されることがあります。次の分子を例に見てみましょう。chemfig が間違っ分岐を作らないようにするために“(”が波括弧内に書かれていることに注意してください。



この例では右上の炭素原子への結合が短すぎることがわかります。これは、chemfig のルールを適用して右上のグループを原子に分割すると、それぞれ“C{”、“C”、“H_3{}}_3”となるためです。つまり最初の原子が括弧を含んでおり、その数式モードでの深さが大きすぎることが原因です。バウンディングボックスを描くとこれを確認できます。



“|”を使用すると、chemfig にこの位置での原子の分割を強制できます。したがって、C|{(CH_3)_3}と書くことで、chemfig が“C”と“{(CH_3)_3}”の間で原子を分割することを保証できます。これで短すぎる結合の問題が解決されます。



2 原子の表示

分子が原子に分割されると、chemfig は各原子を表示するために内部的に \printatom マクロを呼び出します。このマクロの唯一の引数は、分子コードから読み取られた現在の原子を表すトークン（例えば“H_3”）で構成

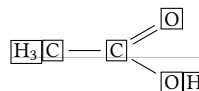
されています。キーワード `use atom strut` が `<true>` に設定されている場合は、この引数に `\vphantom` が追加されます (36 ページ参照)。このマクロは数式モードに切り替え、その引数を数式フォントのローマン体 “rm” を用いて表示します。これは次のように定義されています。

- \LaTeX でコンパイルする場合： `\newcommand*\printatom[1]{\ensuremath{\mathrm{\mathstrut#1}}}`
- \LaTeX または \ConTeXt でコンパイルする場合： `\def\printatom#1{\ifmmode\rm#1\else$\rm#1$\fi}`

このマクロを変更して、原子の表示方法をカスタマイズすることができます。以下の例では `\printatom` を再定義して、各原子を矩形で囲んでいます。

`\printatom` の再定義

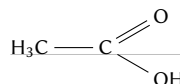
```
\fboxsep=1pt
\renewcommand*\printatom[1]{\fbox{\ensuremath{\mathrm{\mathstrut#1}}}}
\chemfig{H_3C-C(=[:30]O)(-[:30]OH)}
```



次の例では、数式フォントのサンセリフ体 “sf” を使用するように再定義しています。

原子を “sf” フォントファミリーで表示

```
\renewcommand*\printatom[1]{\ensuremath{\mathsf{\mathstrut#1}}}}
\chemfig{H_3C-C(=[:30]O)(-[:30]OH)}
```



3 TikZ に渡される引数

〈キーワード〉 `chemfig style` には、その分子が描画される `tikzpicture` 環境に渡される TikZ 命令を記述します。一方 〈キーワード〉 `atom style` には、各ノードが描画される時に実行される TikZ 命令を記述します。これらの命令は、`every node/.style{<引数>}` の末尾に追加されます。つまり、“`anchor=base,inner sep=0pt,outer sep=0pt,minimum size=0pt`” の後に追加されます。

最初のオプション引数を使用することで、例えば線のグローバルな色や太さを設定できます。

スタイルの設定

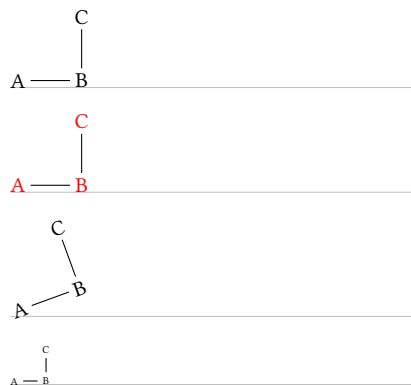
```
\chemfig{A-B-[2]C}\par\medskip
\setchemfig{chemfig style={line width=1.5pt}}\chemfig{A-B-[2]C}\par\medskip
\setchemfig{chemfig style=red}\chemfig{A-B-[2]C}
```



`atom style` を使用することで、TikZ によって描画されるノードの色を変更したり、描画の際の角度や拡大・縮小率を変更することができます。

スタイルの設定

```
\chemfig{A-B-[2]C}\par\medskip
\setchemfig{atom style=red}\chemfig{A-B-[2]C}\par\medskip
\setchemfig{atom style={rotate=20}}\chemfig{A-B-[2]C}\par\medskip
\setchemfig{atom style={scale=0.5}}\chemfig{A-B-[2]C}
```



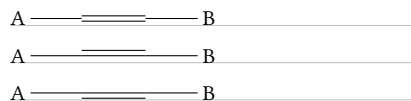
4 二重結合をずらす

二重結合は2つの線分で構成されており、これらの線分は、単結合の場合に描かれる線を仮想的な線として、その両側に描かれます。一方の線分がこの仮想線上になるように二重結合をずらすことができます。もう一方の線分は、結合の上または下に移動します。実際には、結合は描画の方向に沿って進むため、仮想線の「左」または「右」と言う方が正確です。

結合を左にずらすには“=^”と、右にずらすには“=_”と書きます。

二重結合をずらす

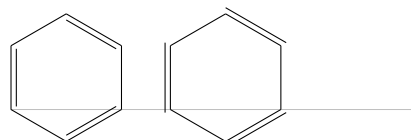
```
\chemfig{A==B}\par
\chemfig{A=^B}\par
\chemfig{A=_B}
```



環式化合物では、二重結合は自動的に左にずらされます。しかし、“=_”を指定することで右にずらすこともできます。

環式化合物の二重結合をずらす

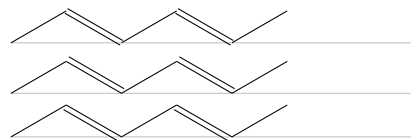
```
\chemfig{*6(==)}\quad
\chemfig{*6(=_)}\quad
```



この機能は、二重結合を持つ炭素鎖からなる分子の骨格構造式を描く場合に特に便利です。通常の二重結合では経路が途切れてしましますが、この方法を使えば連続したジグザグで描くことができます。

ずらされた二重結合と骨格構造式

```
\chemfig{-[:30]=[:30]-[:30]=[:30]-[:30]}\par
\chemfig{-[:30]=^[:30]-[:30]=^[:30]-[:30]}\par
\chemfig{-[:30]=_[:30]-[:30]=_[:30]-[:30]}
```



5 非局在化した二重結合

1本の線が実線で、もう1本が破線であるような二重結合を描きたい場合があります。この機能は chemfig 自体には実装されていません。というのは、TikZ の “decorations.markings” ライブラリーを使うことで実現できるからです。

```

\catcode'\_ =11
\tikzset{
  ddbond/.style args={#1}{
    draw=none,
    decoration={%
      markings,
      mark=at position 0 with {
        \coordinate (CF@startdeloc) at (0,\dimexpr#1\CF_doublesep/2)
        coordinate (CF@startaxis) at (0,\dimexpr-#1\CF_doublesep/2);
      },
      mark=at position 1 with {
        \coordinate (CF@enddeloc) at (0,\dimexpr#1\CF_doublesep/2)
        coordinate (CF@endaxis) at (0,\dimexpr-#1\CF_doublesep/2);
        \draw[dash pattern=on 2pt off 1.5pt] (CF@startdeloc)--(CF@enddeloc);
        \draw (CF@startaxis)--(CF@endaxis);
      }
    },
    postaction={decorate}
  }
}

```

```

\catcode'\_ =8
\chemfig{A-[,,,ddbond={+}]B-[,,,ddbond={-}]C}

```

A ----- B ----- C

6 部分分子の保存

chemfig では部分分子の〈コード〉を、分子コード内で再利用するために別名をつけて保存することができます。これは、〈コード〉が何度も現れる場合に特に便利です。

そのためには次のコマンドを使います。

```
\definesubmol{〈名前〉}{〈コード〉}
```

これにより、分子コード中で “! $\{$ 名前 $\}$ ” という別名を使って、この〈コード〉を呼び出すことができます。この〈名前〉には、

- `\csname` と `\endcsname` の間に書ける文字からなる文字列
- コントロールシーケンス

を使用することができます。

どのような場合においても、別名がすでに定義されている場合は、`\definesubmol` でそれを別の定義で上書きしてはいけません。その場合は、古い別名が新しい別名で上書きされるという警告が表示されます。以前に作成された別名の定義を上書きするには、次のコマンドを使用してください。

```
\redefinesubmol{〈名前〉}{〈コード〉}
```

次の例はペンタン分子を描くコードです。別名 “xy” が、コード CH₂ として定義されています。

ペンタン

```

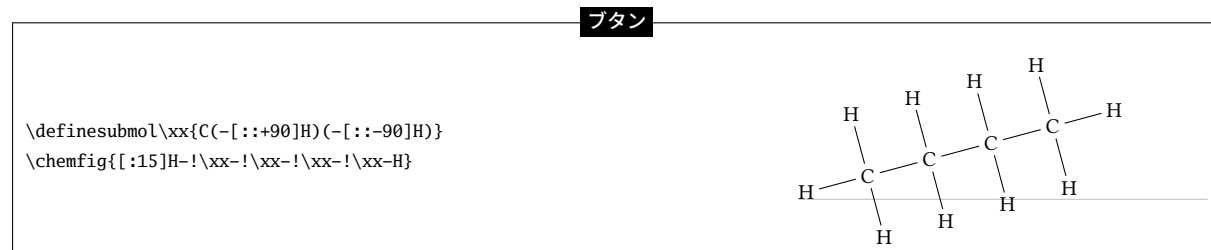
\definesubmol{xy}{CH_2}
\chemfig{H_3C-!{xy}-!{xy}-!{xy}-CH_3}

```

H₃C — CH₂ — CH₂ — CH₂ — CH₃ —

“! $\{xy\}$ ”は置き換えようとしているコードと同じくらいの長さなので、この例はあまり面白くありません。

しかしこの機能によって、分子コードを短くすることができ、それによって可読性が向上する場合があります。次の例ではブタンの完全な構造図を描いています。ここでは部分分子 CH_2 の別名を、コントロールシーケンス “ $\backslash\text{xx}$ ” として定義しています。相対角度のみを使用すれば、主分子のデフォルトの結合角を指定するオプションのグローバルパラメーターを使うことで、分子全体を任意の角度に回転させることができます。ここではその角度を 15° としています。

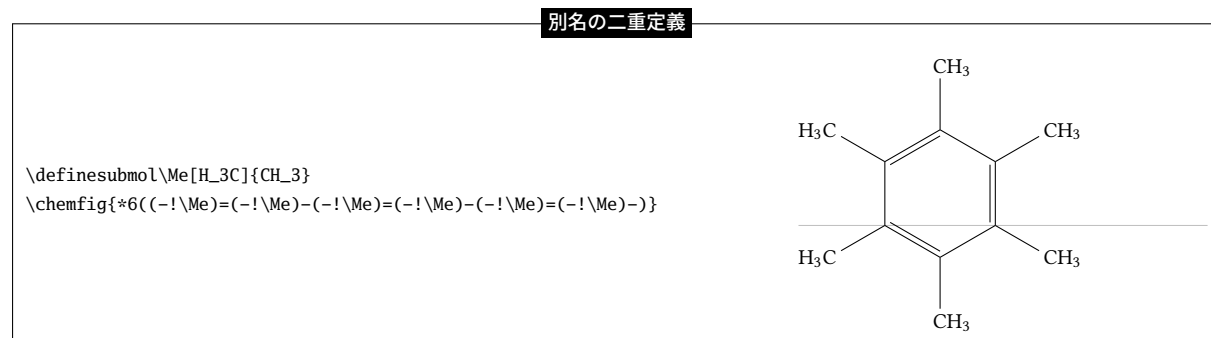


$\backslash\text{definesubmol}$ コマンドはオプション引数を 1 つ取ります。その構文は次の通りです。

$\backslash\text{definesubmol}\{\langle\text{名前}\rangle\}[\langle\text{コード1}\rangle]\{\langle\text{コード2}\rangle\}$

このオプション引数が指定された場合、別名への結合が右から到着する場合、つまり結合が到着する角度が -90° より大きく 90° 未満である場合は、別名 “! $\langle\text{名前}\rangle$ ” には $\langle\text{コード1}\rangle$ が使われます。それ以外の場合、つまり結合が左からあるいは垂直に到着する場合は、この別名は $\langle\text{コード2}\rangle$ に置き換えられます。

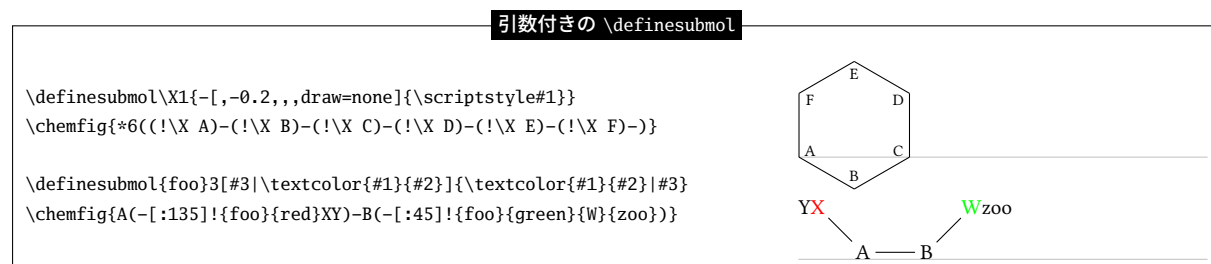
次の例ではメチル基の別名をコントロールシーケンス $\backslash\text{Me}$ として定義し、右から結合が到達する場合には “ H_3C ” に、左から到達する場合には “ CH_3 ” に置き換えられるようにしています。この例からわかるように、この別名を使えばもはや結合がどの角度から入るかを気にしなくてよくなります。



$\langle\text{名前}\rangle$ が “!” によって呼び出されるとき、保存された部分分子は引数を取りません。1 つ以上の引数を取る部分分子を定義するには、 $\langle\text{引数の数}\rangle$ を $\langle\text{名前}\rangle$ の直後に置きます。 $\backslash\text{definesubmol}$ の完全な構文は次のとおりです。

$\backslash\text{definesubmol}\{\langle\text{名前}\rangle\}\langle\text{引数の数}\rangle[\langle\text{コード1}\rangle]\{\langle\text{コード2}\rangle\}$

この引数は、 $\langle\text{コード}\rangle$ 内で通常の “ $\#n$ ” の形式で記述する必要があります。ここで n は引数の番号です。



〈引数の数〉が不正である（負または9より大きい数が指定された）場合、エラーメッセージが表示され、chemfigは部分分子が引数を取らないものと見なします。

文字“#”に1から〈引数の数〉の数が続く、すなわち引数を指す場合を除き、分子コード内で“#”を使用することができます。

#の使用

```
\definesubmol\X2{#1-#2-#3-#(3pt,3pt)#4}  
\chemfig{A-!\X{M}{N}-B}
```

A — M — N — #3 — #4 — B

この例では#1と#2のみが部分分子\Xの引数として扱われます。それ以外の“#”は分子内でそのまま表示される（この例では#3と#4）か、原子と結合の距離の微調整を指示するための文字として扱われます。

7 原子の配置

7.1 原子グループ

原子グループ内では、各原子は次のルールによって順に配置されます。

- 最初に配置される原子（「基準原子」と呼びます）は、結合が到達する原子です。分子の先頭の場合は、左側の原子が基準原子です。
- 次に、基準原子の右側にある原子が、左から右に配置されます。
- 最後に、基準原子の左側にある原子が、右から左に配置されます。

このように形成された原子グループでは、各原子のベースラインは**同一の水平線上**にあり、言い換えると、全ての原子が同じ水平線上に整列しています。

コード“\chemfig{A[: -60,,,3]BCDEF}”の場合、2つ目の原子グループの基準原子は“D”です。これは結合が3つ目の原子に到着するように指示されているからです。下図は、このグループの各原子に割り当てられた連番を示しています。



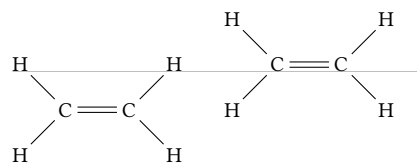
7.2 垂直方向の整列

現在の段落のベースラインに対して分子の垂直方向の位置を微調整するために **baseline** を使用します。

これはデフォルトでは **<opt>** に設定されており、この場合、最初に遭遇する原子（空かどうかによらず）が現在の段落のベースラインに配置されます。このマニュアルではベースラインは灰色の線で示されています。どの原子を最初に描くかによって他の原子の配置が決まり、しばしば分子全体の配置にも影響を与えます。

最初の原子

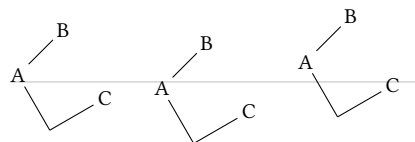
```
\chemfig{H-[7]C(-[5]H)=C(-[1]H)-[7]H}\quad  
\chemfig{C(-[3]H)-(-[5]H)=C(-[1]H)-[7]H}
```



構文 **baseline = <長さ>** によって距離を指定することで、分子を垂直方向に移動させることができます。

垂直方向の移動

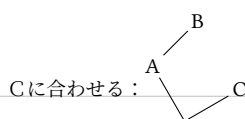
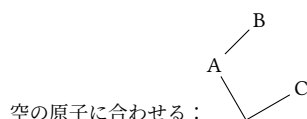
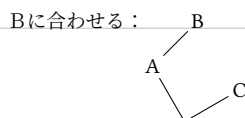
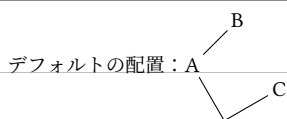
```
\chemfig{A(-[: -60]-[:30]C)-[:45]B}\qquad
\chemfig[baseline=5pt]{A(-[: -60]-[:30]C)-[:45]B}\qquad
\chemfig[baseline=-5pt]{A(-[: -60]-[:30]C)-[:45]B}
```



構文 `baseline = <(名前)>` (名前は括弧内に書く必要があります) を使用すると、ノード <名前> で分子のベースラインを指定できます。原子の名前には、`chemfig` によって自動的に割り当てられた (n(a)-) 形式の名前、またはユーザーが構文 `@{名前}` で付けた名前を使うことができます (25 ページ参照)。

原子の配置

```
デフォルトの配置： \chemfig{A(-[: -60]-[:30]C)-[:45]B}\medbreak
Bに合わせる： \chemfig[baseline=(b.base)]{A(-[: -60]-[:30]C)-[:45]@{b}B}\medbreak
空の原子に合わせる： \chemfig[baseline=(vide)]{A(-[: -60]@{vide}-[:30]C)-[:45]B}\medbreak
Cに合わせる： \chemfig[baseline=(c.base)]{A(-[: -60]-[:30]@{c}C)-[:45]B}
```



TikZ ノード名を使うこともできます。例えば、現在のベースラインに対して複数の分子を垂直に中央揃えしたい場合は、`baseline` に `<(current bounding box.center)>` を設定します。

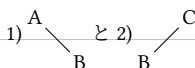
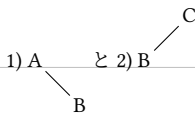
中央揃え

1) `\chemfig{A-[: -45]B}` と 2) `\chemfig{B-[:45]C}\bigbreak`

`\setchemfig{baseline=(current bounding box.center)}%` 以降の分子を垂直方向にセンタリング

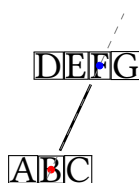
1) `\chemfig{A-[: -45]B}` と 2) `\chemfig{B-[:45]C}`

`\setchemfig{baseline=0pt}%` デフォルト値に戻す



7.3 原子間の結合

ある原子から始まる結合は、延長するとその原子のバウンディングボックスの中心を通過します。同様に、結合の終端に配置された原子のバウンディングボックスの中心は、結合の延長線上にあります。つまり、結合で結ばれている2つの原子のバウンディングボックスの中心はいずれも、この2つの原子間の結合の延長線上にあります。



この方法で位置決めをすると原子グループの整列に乱れが生じる場合があります、特に結合が水平の場合に非常に目立つことがあります。原子が同じ垂直方向の長さを持っている場合は全てがうまく機能しますが、出発原子が高い（上付き文字がある）または深い（下付き文字がある）場合に、到着原子がこれと異なる垂直方向の長さを持っていると、整列が崩れます。

水平整列

```
\large\setchemfig{atom sep=2em}
\chemfig{A^1-B-C-D}\par
\chemfig{E_1-F-G-H}
```

$A^1 - B - C - D$

$E_1 - F - G - H$

この現象は、原子のバウンディングボックスを表示させてみるとよくわかります。次の例では、原子“B”と“F”のバウンディングボックスの高さが、直前の原子の高さを考慮したものになっていることがわかります。

水平配置とバウンディングボックス

```
\large\setchemfig{atom sep=2em}
\fbboxsep=-0.2pt \fbboxrule0.2pt
\renewcommand\printatom[1]{\fbbox{\ensuremath{\mathrm{#1}}}}
\chemfig{A^1-B-C-D}\par
\chemfig{E_1-F-G-H}
```

$A^1 - B - C - D$

$E_1 - F - G - H$

最初の原子のバウンディングボックスの垂直方向の高さがその次の原子のものと異なるため、後続のバウンディングボックスに垂直方向のずれが生じます。結合が水平方向のとき、このずれは明確に見て取れます。

chemfig の 1.7 より前のバージョンでは直前の原子の `\vphantom` をそれぞれの `\printatom` の引数に追加することで、問題のあるバウンディングボックスに続く最初の原子に対する影響を隠していました。この動作に戻したい場合は、キーワード `use atom strut` を `<true>` に設定してください。

垂直配置とバウンディングボックス

```
\large\setchemfig{atom sep=2em}
\fbboxsep=-0.2pt \fbboxrule0.2pt
\renewcommand\printatom[1]{\fbbox{\ensuremath{\mathrm{#1}}}}
\chemfig{A^1-B-C-D}\quad
\chemfig[use atom strut=true]{A^1-B-C-D}

\chemfig{E_1-F-G-H}\quad
\chemfig[use atom strut=true]{E_1-F-G-H}
```

$A^1 - B - C - D$ $A^1 - B - C - D$

$E_1 - F - G - H$ $E_1 - F - G - H$

どのような自動的な解決策も満足のいくものではないので、出発原子の終端に `\vphantom{X}` と等しい「支柱 (strut)」を作ることで、この問題を手動で回避できます。これにより出発原子は「通常の」高さを持ち、ずれはその後の原子グループに反映されません。次の例ではより簡潔にするために部分分子を使用しています。

垂直配置の回避

```
\large\setchemfig{atom sep=2em}
\definesubmol\I{\vphantom{X}}
\chemfig{A^1|!\I-B-C-D}\par
\chemfig{E_1|!\I-F-G-H}
```

$A^1 - B - C - D$

$E_1 - F - G - H$

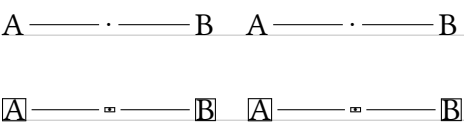
この方法の欠点は、出発原子の幅が 0 であるため、最初の結合が長すぎることです。

7.4 \chemskipalign マクロ

任意の原子グループに対して、`chemfig` の整列調整機構を一時的に無効にし、`\vphantom` を「中和」することができます。そのためには、対象の原子グループ内に `\chemskipalign` コマンドを書くだけです。すると次の原子グループから整列が再開され、`\chemskipalign` を含む原子グループが存在しなかったかのようになります。次の例で、この命令の効果を確認できます。この命令により、最初の原子を含むボックスの基準点は、左から来る結合の高さに位置しています。2 行目では原子を含むバウンディングボックスが描かれています。

整列機構の無効化


```
\large
\chemfig{A-.-B}\quad
\chemfig{A-\chemskipalign.-B}\par\bigskip
\fbboxsep=0pt
\renewcommand\printatom[1]{\fbox{\ensuremath{\mathrm{#1}}}}
\chemfig{A-.-B}\quad
\chemfig{A-\chemskipalign.-B}
```



このコマンドは注意して使用しないと、次の原子グループの整列を崩してしまいます。`\chemskipalign` を含む原子グループが**単一の原子**で構成され、その高さと深さが前後の原子より**小さい**ときは、前後の原子の高さと深さが同じであれば問題はありません。次の例では、原子グループが2つの原子、ここでは“`\chemskipalign.`”と“`B`”を含んでいるときに、整列がうまくいかない場合を示しています。

\chemskipalign の結果

```
\large
\fbboxsep=0pt
\renewcommand\printatom[1]{\fbox{\ensuremath{\mathrm{#1}}}}
\chemfig{A-\chemskipalign.B-C}
```



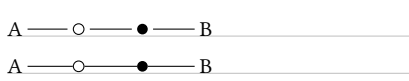
この機能が便利なときもあります。次の分子を描画したいとしましょう。



中空の円と塗り潰された円を TikZ で定義します。これらの円を正しい高さ、すなわち到達する結合の高さに配置するために `\chemskipalign` コマンドを使用します。2 行目で円と結合が「くっついて」いますが、これには 10 ページで見た“`#`”の機能、すなわち結合と原子の間の距離を変更する機能を使っています。

\chemskipalign コマンドと # の使用

```
\def\emptydisk{\chemskipalign\tikz\draw(0,0)circle(2pt);}
\def\fulldisk{\chemskipalign\tikz\fill(0,0)circle(2pt);}
\chemfig{A-\emptydisk-\fulldisk-B}\par
\chemfig{A-#(,0pt)\emptydisk-#(0pt,0pt)\fulldisk-#(0pt)B}
```



8 \charge マクロ

8.1 概要

`\charge` マクロは必須の引数を 2 つ取り、〈原子〉の周りに (〈電荷〉と呼ばれる) 要素を配置します。

完全な構文は次の通りです。

`\charge{[〈一般的なパラメーター〉][〈位置〉][〈tikzコード〉]=〈電荷〉}{〈原子〉}`

ここで、

- ・〈原子〉は通常 1 文字または 2 文字ですが、空でも構いません。
- ・〈電荷〉は原子の周りに配置したい任意の要素です。この内容に関する制約は少なく、(必要なら数式モードでの) テキストや、TikZ コード、あるいは `\chemfig` で描かれた分子でも構いません。
- ・〈一般的なパラメーター〉(非必須) は、キーワードと値のリストであり、このマクロの実行時に満たすべきオプションを指定します。具体的なキーワードと値は下に示します。
- ・〈位置〉は“〈角度〉:〈距離〉”ですが、〈角度〉のみを指定することも可能です。その場合 〈距離〉は `0pt` となります。
- ・〈tikzコード〉(非必須) は、〈電荷〉を配置する TikZ の `\node` マクロに与えられるオプションを設定します。

8.2 パラメーター

〈一般的なパラメーター〉で利用可能な〈キーワード〉=〈値〉は次表の通りです。

〈キーワード〉	デフォルト 〈値〉	説明
<code>debug</code>	<code>false</code>	真偽値を取り、〈 <code>true</code> 〉のとき 〈原子〉 (緑)、〈電荷の配置位置〉 (青)、および 〈電荷〉 (赤) の各ノードのアウトラインを描画します。
<code>macro atom</code>	<code>\printatom</code>	引数として 〈原子〉 を受け取るマクロ
<code>circle</code>	<code>false</code>	真偽値を取り、〈 <code>true</code> 〉のときは 〈原子〉 を円形ノードに配置します。そうでない場合はノードは矩形です。
<code>macro charge</code>	〈空〉	それぞれの 〈電荷〉 を引数として受け取るマクロ (例えば <code>\printatom</code> や <code>\ensuremath</code>)
<code>extra sep</code>	<code>1.5pt</code>	〈電荷〉 を配置する際の、〈原子〉 ノードサイズの増分。これは TikZ の <code>inner sep</code> に渡される値です。
<code>overlay</code>	<code>true</code>	真偽値を取り、〈 <code>true</code> 〉のとき 〈電荷〉 を「オーバーレイ」として、つまり最終的なバウンディングボックスの外側に描画します。
<code>shortcuts</code>	<code>true</code>	真偽値を取り、〈 <code>true</code> 〉のとき Lewis 式を描画するためのショートカット “ <code>\.</code> ”、“ <code>\:</code> ”、“ <code>\ </code> ” および “ <code>\</code> ” を有効にします。
<code>lewisautorot</code>	<code>true</code>	真偽値を取り、〈 <code>true</code> 〉のとき “ <code>\:</code> ”、“ <code>\ </code> ” および “ <code>\</code> ” を自動的に回転させます。
<code>.radius</code>	<code>0.15ex</code>	“ <code>\.</code> ” および “ <code>\:</code> ” での点の半径
<code>:sep</code>	<code>0.3em</code>	“ <code>\:</code> ” の 2 つの点の間の距離
<code>.style</code>	<code>fill=black</code>	“ <code>\.</code> ” および “ <code>\:</code> ” の点を描画で使用する TikZ スタイル
<code>"length</code>	<code>1.5ex</code>	矩形 “ <code>\</code> ” および線 “ <code>\ </code> ” の長さ
<code>"width</code>	<code>.3ex</code>	矩形 “ <code>\</code> ” の幅
<code>"style</code>	<code>black,line width=0.4pt</code>	矩形 “ <code>\</code> ” の描画で使用する TikZ スタイル
<code> style</code>	<code>black,line width=0.4pt</code>	線 “ <code>\ </code> ” の描画で使用する TikZ スタイル

これらのパラメーターの一部 (または全て) を設定するには、マクロ

```
\setcharge{〈キーワード〉=〈値〉形式のリスト}
```

を実行し、全てのパラメーターをデフォルト値にリセットするには

```
\resetcharge
```


を実行します。`\charge` マクロは (〈パラメーター〉で指定されていない限り) 〈電荷〉をバウンディングボックスの外に配置します。一方 `\Charge` では 〈電荷〉はバウンディングボックスの内に配置されます。

〈角度〉は、〈電荷〉が配置されるノードの境界上の位置です。この 〈角度〉は度で与えることも、“south east” などの TikZ アンカーで与えることもできます。〈距離〉は \TeX の長さであり、〈原子〉ノードの境界と 〈電荷〉が配置される位置との間に追加される長さを表します。〈tikzコード〉内に指定されていない限り、〈電荷〉のアンカーは *center* です。

パラメーターの変更によって起きる変化を分かりやすくするために、次の 2 つの例では `debug` を 〈true〉に設定しています。さらに、マクロ `\Charge` が使用され、バウンディングボックスが電荷を含むようにしています。ここでは、ノードの形状が電荷の配置に与える影響を見てみましょう。

一般的な例

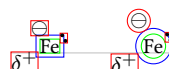
```
\setcharge{debug}
デフォルトおよび円形：
\Charge{30=\:,120=\ominus$,210=\delta^+${Fe}}\quad
\Charge{[circle]30=\:,120=\ominus$,210=\delta^+${Fe}}
```

デフォルトおよび円形：

電荷 \ominus と δ^+ をもっと遠くに配置するには、〈距離〉や、より良い方法としてアンカーで調整できます。電荷が配置される 〈角度〉は `\chargeangle` マクロに保存されているので、アンカーを `180+\chargeangle` とするのが賢いやりかたです。また、電荷を配置するために円形ノードを指定することも可能です。

位置の微調整

```
\setcharge{debug}
\Charge{30=\:,120:3pt=\ominus$,210:5pt=\delta^+${Fe}}\quad
\Charge{[circle]30=\:,
120[circle,anchor=180+\chargeangle]=\ominus$,
210[anchor=180+\chargeangle]=\delta^+${Fe}}
```



円形ノードは「古典的」な矩形ノードとはときに非常に異なる長さ、特に水平方向および垂直方向の広がりを持つことに注意が必要です。したがって、キーワード `circle` を 〈true〉に設定しようとするときは、そのことを十分意識したうえで設定することをお勧めします。

円形のノード

```
\chemfig{\charge{90=\.}{N}H_3}：矩形ノード\smallbreak
\chemfig{\charge{[circle]90=\.}{N}H_3}：円形ノード
```

$\dot{\text{N}}\text{H}_3$ ：矩形ノード

$\dot{\text{N}}\text{H}_3$ ：円形ノード

8.3 ルイス構造式

`shortcut` が 〈true〉のとき、ルイス構造式 “.”、“:”、“|” および “|” を描画するためのショートカット “\.”、“\:”、“\|” および “\|” が有効になります。`\disables shortcuts` マクロを使えばいつでもこれを無効にすることができ、`\enables shortcuts` で再度有効にできます。

`shortcut` が 〈false〉であるか、ショートカットが `\disables shortcuts` で無効にされている場合、ショートカット “\.”、“\:”、“\|” および “\|” はルイス構造式を描画するためには使えないので、かわりに `\chargedot`、`\chargeddot`、`\chargeline` および `\chargerect` マクロを使用する必要があります。

キーワード `lewisautorot` はデフォルトで 〈true〉であり、描画の際に “:”、“|” および “|” を回転させます。

自動回転

```
\Charge{60=\:,150=\}{{A}} と
\Charge{[lewisautorot=false]60=\:,150=\}{{A}}
```



ルイス構造式のカスタマイズは `\setcharge` マクロまたは `\charge` のオプション引数を使って、キーワード `.radius`、`:sep`、`.style`、`|style`、`"length`、`"width` および `"style` を設定することで行います。また、各構造式のオプション引数に `キーワード = <値>` のリストを与えることで、これらのキーワードを変更することも可能です。

カスタマイズ

```
\Charge{[.radius=1.5pt,.style={draw=gray}]
  45 =\.[{.style={draw=none,fill=red}}],
  135 =\.[{.style={draw=none,fill=blue}}],
  -45 =\.[{.style={draw=none,fill=green}}],
  -135=\.[{A}\quad
\Charge{
  45 =\["style={draw=red,fill=gray}}],
  135=\["width=3pt,"style={line width=.8pt,draw=blue,fill=cyan}}]{A}
```

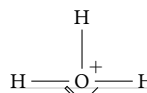


8.4 chemfig の中で使う

`\charge` マクロは原子の代わりに使用できます。

chemfig 中の電荷

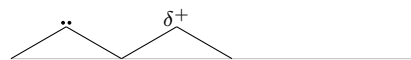
```
\chemfig{H-\charge{45:1.5pt=$\scriptstyle+$,-45=\|,-135=\"}{O}(-[2]H)-H}
```



なお `chemfig` の現在のバージョンでは、原子のサイズが 0、すなわち幅、高さ、深さが全て 0pt である場合には結合がくっつくように変更されています。以前のバージョンでは、そのようになるのは原子が空のときだけでした。この新機能により、炭素鎖に電荷を配置するのが容易になりました。

炭素鎖上の電荷

```
\chemfig{[:30]-\charge{90=\:}{}}
-[:30]\charge{-90=\:}{}-\charge{90:2pt=$\delta^+$}{-[:30]}
```



9 積み重ね

次のマクロ

```
\chemabove[<長さ>]{<コード>}{<内容>}
```

および

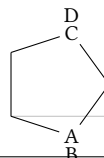
```
\chembelow[<長さ>]{<コード>}{<内容>}
```

はそれぞれ `<コード>` の上および下の垂直距離 `<長さ>` の位置に `<内容>` を配置します。その際 `<コード>` のバウンディングボックスは変化しません。オプション引数を指定すると、それぞれの呼び出しでこの長さを指定できます。オプション引数を使用されない場合、デフォルト値 `<1.5pt>` が使用されます。この値は `<キーワード>` `stack sep = <長さ>` で変更できます。

これらのコマンドは、`\chemfig` マクロとは独立しており、その引数の内でも外でも使用できます。

これらは環で特に便利ですが、`chemfig` が文字 A、B、C、D で新しい原子を開始しないように、これらを波括弧の中を書く必要があることに注意してください。

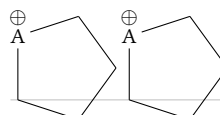
```
\chemfig{*5(-\chembelow{A}{B}--\chemabove{C}{D}--)}
```



`\Chemabove` および `\Chembelow` コマンドも同様の機能を持ちますが、バウンディングボックスは上または下に配置された〈内容〉を含みます。

1 つのアイテムを別のアイテムの上または下に配置する場合に、`\chemabove` と `\charge` の違いはどのようなか。

```
\chemfig{*5(----\chemabove{A}{\oplus}--)}
\chemfig{*5(----\charge{90[anchor=-90]=$\oplus$}{A}--)}
```

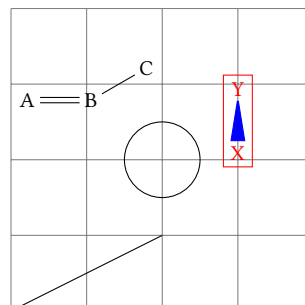


デフォルトではこれら 2 つのマクロは非常に似た結果を与えます。ただし、`\charge` マクロは TikZ を必要としますが、`\chemabove` および `\chembelow` は低レベルの TeX プリミティブを使用しているため、高速であり、パッケージに依存しないという違いがあります。

10 `\chemfig` を tikzpicture 環境で使う

tikzpicture 環境内で `\chemfig` を呼ぶこともできます。

```
\begin{tikzpicture}[help lines/.style={thin,draw=black!50}]
  \draw[help lines] (0,0) grid (4,4);
  \draw(0,0) -- (2,1);
  \draw(2,2) circle (0.5);
  \node at (1,3) {\chemfig{A=B-[:30]C}};
  \node[draw,red,anchor=base] at(3,2){\chemfig{X>[2,,,blue]Y}};
\end{tikzpicture}
```

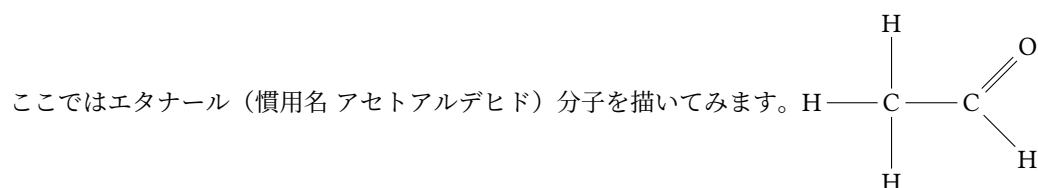


11 分子の記述の例——注釈を添えて

この節ではこれまで説明した方法を使用していくつかの分子を描いてみます。ここでの目的は、`chemfig` に不慣れたユーザーが複雑な分子の描き方を学べるように、分子を組み立てるための順序を論理的に示すことです。学習を助けるために、分子を構築する手順を段階を追って説明します。

加えて、同じ結果が得られる異なった書き方を示します。これには直感的なものもそうではないものもありますが、`chemfig` が分子の記述において一定の柔軟性を持っていることを示します。それぞれがどのように組み合わせられているかを確認し、自分に最も使いやすい方法を採用することができます。

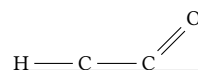
11.1 エタナール



非環状分子を描く最適な方法は、最長の鎖を選択することです。ここでは例えば“H-C-C=O”を選ぶことができます。結合 C=O は、事前定義された角度 “[1]” を使用して 45° で描かれています。これにより、分子の「主鎖」が得られ、わずかな側鎖を追加するだけで済みます。

エタナールの主鎖

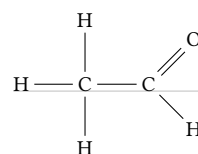
```
\chemfig{H-C-C=[1]O}
```



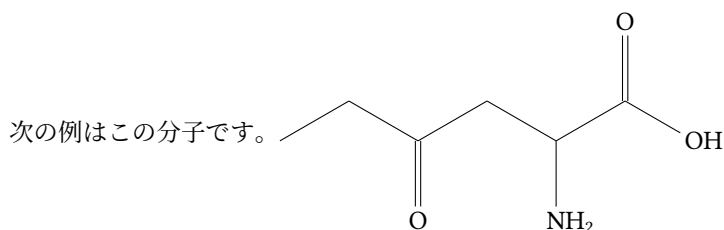
3つの水素原子は、事前定義された角度の助けを借りて正しい方向に配置する必要があります。最初のものは側鎖“(−[2]H)”として90°に、2番目は“(−[6]H)”を270°に、右側のものは“(−[7]H)”を315°に配置します。

エタナール

```
\chemfig{H-C(-[2]H)(-[6]H)-C(-[7]H)=[1]O}
```



11.2 2-アミノ-4-オキソヘキサン酸



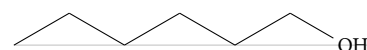
ほとんどの分子においてそうであるように、この分子を描く方法はいくつかあり、それぞれに異なるやり方があります。ここでは4つの異なる方法を見ていきます。

11.2.1 角度の直接指定

まず角度を直接指定して中心となる鎖を描きます。オプション引数でデフォルトの角度を +30° に設定しておけば、下降する結合に対してのみ角度を −30° に指定するだけで済みます。

主鎖（角度の直接指定）

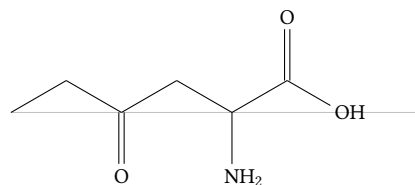
```
\chemfig{[:30]---[:−30]---[:−30]---[:−30]OH}
```



その後、側鎖“(= [6]O)”、“(− [6]NH₂)”および“(= [2]O)”を、正しい頂点に追加します。

分子（角度の直接指定）

\chemfig{[:30]--[: -30](=[6]O)--[: -30](-[6]NH_2)-([2]O)-[: -30]OH}



11.2.2 相対角度

より一般的な方法は、次のような相対角度のみを使用する方法です。

構造（相対角度）

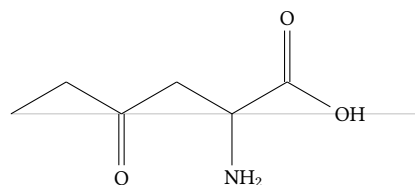
\chemfig{[:30]--[: -60]--[: -60]--[: -60]OH}



そして、

分子（相対角度）

\chemfig{[:30]--[: -60](=[: -60]O)--[: -60](-[: -60]NH_2)
-([:60]O)-[: -60]OH}



11.2.3 環

この方法はあまり自然ではありませんが、結合間の角度が 120° なので、六員環を使って描くことができます。ここで、環は未完成のままにすることができるという特徴を利用します。最初の頂点が環の南東に来るようにするために、環を 120° 回転させる必要があります。

主鎖（環）

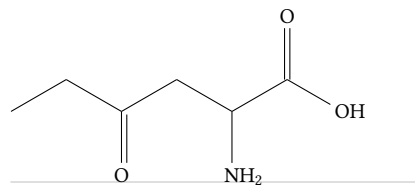
\chemfig{[:120]NH_2*6(---=O)}



次に、正しい頂点に側鎖を追加します。

分子（環）

\chemfig{[:120]NH_2*6(-([:60]O)-[: -60]OH)--([:60])=O)}

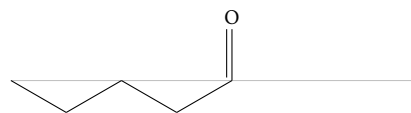


11.2.4 入れ子の環

環の方法をさらに掘り下げると、不完全な六員環を入れ子にする方法も考えられます。次の主鎖から始めましょう。

主鎖（入れ子の環）

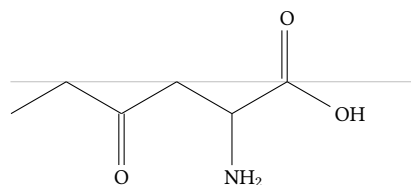
```
\chemfig{*6(--*6(==O))}
```



次に、これらの環の頂点から出る結合を追加します。環から出る結合は環の2つの辺の二等分線であり、まさに求めているものなので、角度について悩む必要はありません。

分子（入れ子の環）

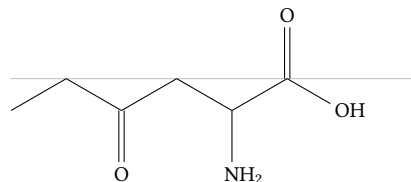
```
\chemfig{*6((-)(=O)-*6(-(-NH_2)-(-OH)=O))}
```



よく見ると、酸素原子への二重結合の2つ目の線分は不完全な六員環の内部にあります⁹。このコードは簡潔ではありますが、完璧な結果を与えてくれません。もちろん、コードを少し追加することで修正できます。

分子（修正された入れ子の環）

```
\chemfig{*6((-)(=O)-*6(-(-NH_2)-(-OH)(=[:60]O)))}
```



11.3 グルコース

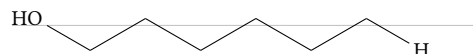
ここでの目標は、グルコース分子をいくつかの異なる形式で描くことです。

11.3.1 破線・くさび形表記

このコードは、2-アミノ-4-オキソヘキサン酸のものに似ています。そのときと同様に角度を直接指定してほぼ同じ構造が得られますが、ここではデフォルトの角度を -30° とします。

主鎖

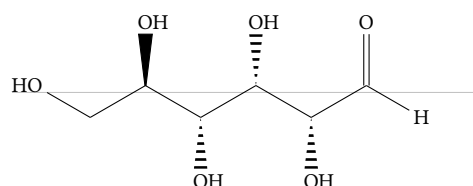
```
\chemfig{[: -30]HO--[:30]--[:30]--[:30]-H}
```



側鎖の追加は何の問題もありません。事前定義された角度を使用します。

破線・くさび形表記のグルコース

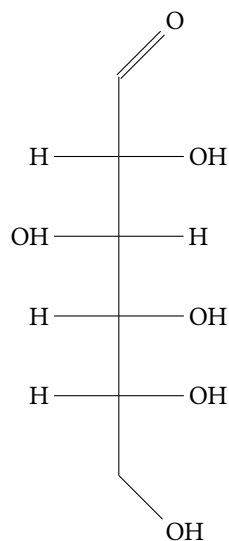
```
\chemfig{[: -30]HO--[:30](<[2]OH)-(<:[6]OH)-[:30](<:[2]OH)-(<:[6]OH)-[:30](=[2]O)-H}
```



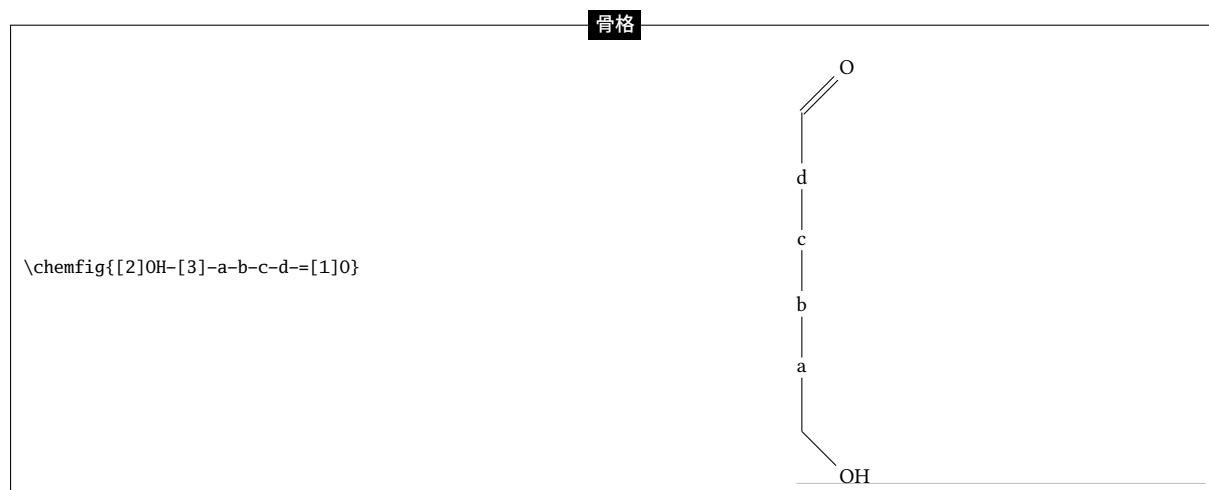
⁹このことは、1つの環で描いた方法にも当てはまります。

11.3.2 フィッシャー投影式

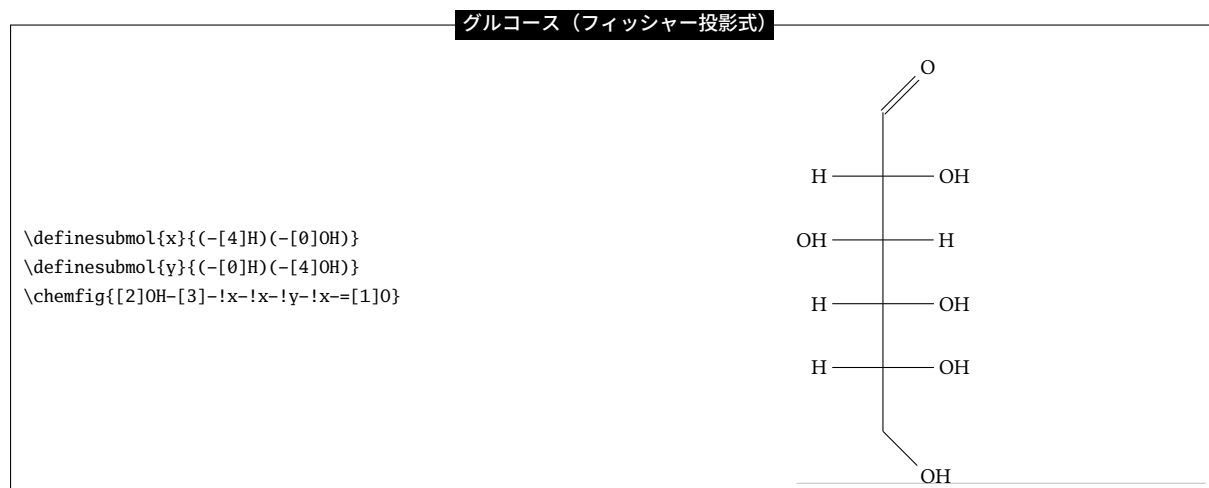
ここでの目標は、以下の分子を描くことです。



デフォルトの角度を “[2]” に設定し、最長の鎖を垂直に描きます。骨格は以下ようになります。ここでは、それぞれの垂直な結合の後に小文字で文字を追加してあります。

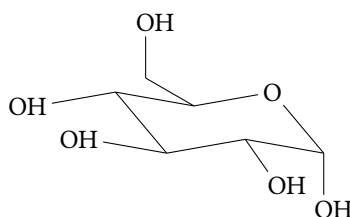


次に、水平にのびている結合とその端にある原子を描くために 2 つの別名を定義します。“x” を小文字の a、b、d と置き換え、“y” を文字 c と置き換えます。これらの別名は 1 文字だけなので、“!{x}”ではなく、波括弧なしで “!x” と書くことができます：



11.3.3 「いす形」表現

次は α -D-グルコース分子を描きましょう。



そのためにまず、いすの五つの辺を描き、最初の頂点と最後の頂点をフック“?”によって繋ぎます。角度は -50° 、 10° 、 -10° 、 130° 、 190° を反時計回りに指定します。

構造

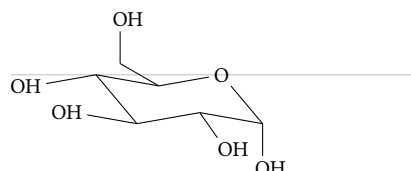
```
\chemfig{?-[:-50]-[:10]-[:-10]-[:130]O-[:190]?}
```



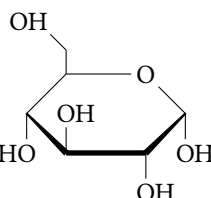
あとは単に側鎖を括弧内に書いて追加するだけです。次の例では、角度は最も遠近感が出るように選ばれ、一部の結合は 0.7 倍に短縮されています。

「いす形」表現

```
\chemfig{?(-[:190]OH)-[:50](-[:170]OH)-[:10](-[:55,0.7]OH)-[:-10](-[:6,0.7]OH)-[:130]O-[:190]?(-[:150,0.7]-[:2,0.7]OH)}
```



11.3.4 ハース投影式



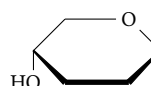
次の目標は D-グルコピラノース分子を描くことです。

まず最長の鎖を選ぶと、左の“HO”グループから始まり、5つの辺を通る環になります。環はフックを使って閉じられます。最初の“HO”グループから出る垂直な結合は、オプション引数を使用して2番目の原子から出ることを指示する必要があります。さらにこの結合を 0.5 倍に短縮します。したがってオプション引数は “[2,0.5,2]” となります。

次に、環に遠近感を与えるために、斜めの結合は 0.7 倍に短縮します。太い斜線には底辺の長さを 2pt に再定義した三角形によるくさび形結合を使用します。太い水平の結合は太さ 2pt で描く必要があるため、オプション引数 “[0,,,line width=2pt]” を指定します。分子の骨格を描くと次のようになります。

構造

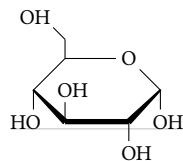
```
\chemfig[cram width=2pt]{HO-[2,0.5,2]?<[7,0.7]-[,,,line width=2pt]>[1,0.7]-[3,0.7]O-[4]?}
```



最後に、側鎖を正しい場所に正しい角度で、必要に応じて長さを短くして追加し、より良い遠近感を与えるようにします。

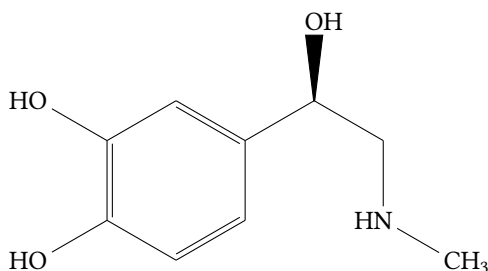
ハース投影式

```
\chemfig[cram width=2pt]{HO-[2,0.5,2]?<[7,0.7](-[2,0.5]OH)-[,,,,
line width=2pt](-[6,0.5]OH)>[1,0.7](-[6,0.5]OH)-[3,0.7]
O-[4]?(-[2,0.3]-[3,0.5]OH)}
```



11.4 アドレナリン

アドレナリン分子を描きましょう。



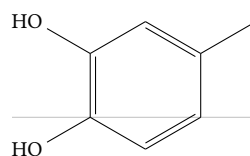
2 種類の方法を使用します。

11.4.1 1つの環を使う

まず、六員環から始め、そこから出る側鎖の始まりを描きます。

アドレナリンの骨格

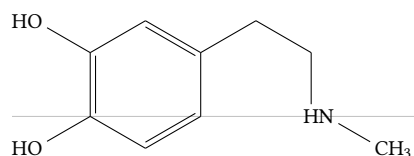
```
\chemfig{*6((-HO)--(-)--(-HO)=)}
```



右の側鎖は、例えば相対角度を使用して完成させます。

アドレナリンステップ 2

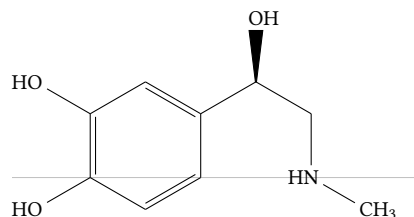
```
\chemfig{*6((-HO)--(--[: -60]-[: -60]
HN-[: +60]CH_3)=(-HO)=)}
```



次にくさび形結合で繋いだ OH を追加し、さらに“HN”への結合が第 2 の原子、すなわち“N”に到着するように指示する必要があります。そのために結合の 4 つ目のオプション引数を使用します。

アドレナリン

```
\chemfig{*6((-HO)--(-(<[: 60]OH)-[: -60]-[: -60,,,2]
HN-[: +60]CH_3)=(-HO)=)}
```



11.4.2 2つの環を使う

この方法はあまり自然ではありませんが、ここでは結合を不可視にする方法を示すことが目標です。

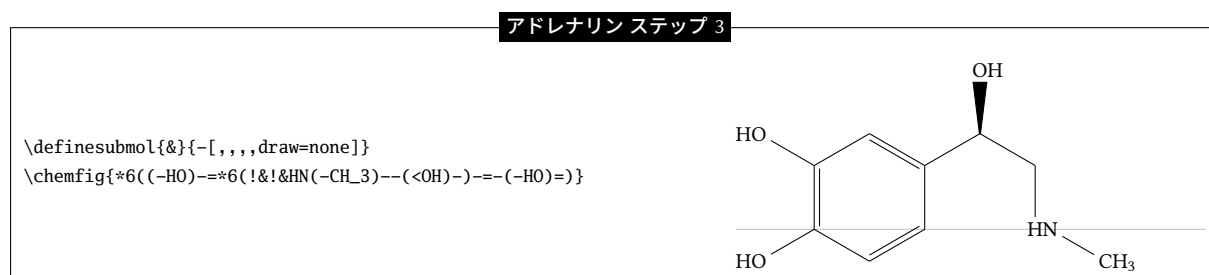
アドレナリン分子が隣接する2つの六員環で構成されていると考えると、上のコードは次のように改善できます。



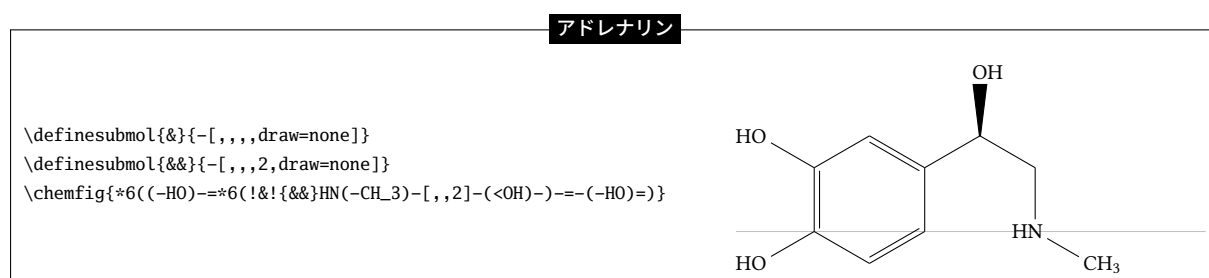
右の環の最初の2つの結合を見えなくする必要があります。これを行うために、TikZに渡される引数“draw=none”を使用します。したがってこれらの結合のコードは“-[,,,,draw=none]”となります。コードを読みやすく保つために、この結合に“&”という別名を定義します。



残りは簡単です。正しい頂点に側鎖を追加するだけです。

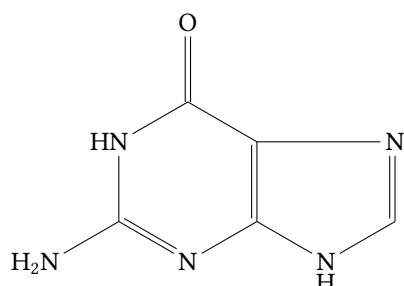


最後に、“HN”において、結合が2番目の原子に到達し、ここから出発することを指定します。そのために“HN”に到達する見えない結合に、新しい別名を定義します。



11.5 グアニン

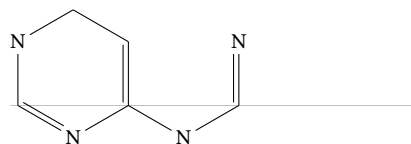
グアニン分子を描きましょう。



まず、入れ子の環を描いて、窒素原子だけを頂点に置くことから始めましょう。

グアニン骨格

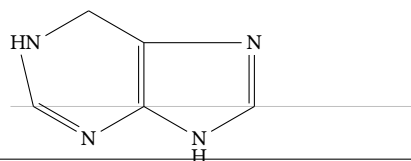
```
\chemfig{*6(=N-*6(-N=N)=--N-)}
```



次に、右の環の水平な結合をフックで描きます。また、`\chembelow{N}{H}` コマンドを使用して、五員環の窒素原子の下に水素原子を配置します。分子の左上の頂点は“N”ではなく“HN”と書く必要があります。

グアニン ステップ 2

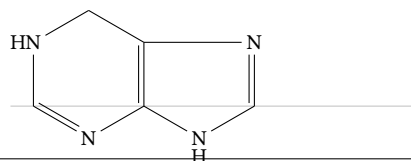
```
\chemfig{*6(=N-*6(-\chembelow{N}{H}-N?)=?--HN-)}
```



1 つの結合が間違った原子から出発してしまっています¹⁰。この結合が 1 番目の原子ではなく 2 番目の原子“N”から出発するように、自動的に計算された結果を修正する必要があります。そのために、最初の六員環の最後の結合にオプション引数“[,2]”を与えます。

グアニン ステップ 3

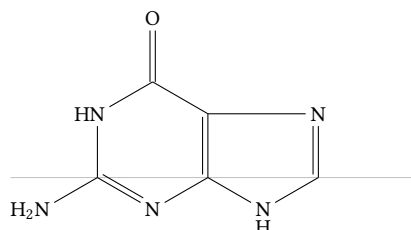
```
\chemfig{*6(=N-*6(-\chembelow{N}{H}-N?)=?--HN-[,,2])}
```



あとは、六員環の頂点に側鎖を追加するだけです。特に、1 つ目の六員環の 1 つ目の頂点から出発する側鎖“(H_2N)”に注意してください。

グアニン

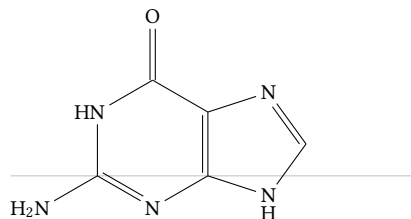
```
\chemfig{*6((H_2N)=N-*6(-\chembelow{N}{H}-N?)=?-(=O)-HN-[,,2])}
```



よく見る正五角形の環を使った描き方もできます。

五員環を使ったグアニン

```
\chemfig{*6((H_2N)=N-*5(-\chembelow{N}{H}-N)-=(O)-HN-[,,2])}
```



¹⁰HN グループから最初の頂点に向う結合の角度は -90° と 90° の間にあるので、非論理的であるように思われます。したがって、`chemfig` は結合を 2 番目の原子から出発させるべきです。この矛盾を説明するには、環では最後の結合は常に最後の頂点を最初の頂点に繋げることを知っておく必要があります。その際、この結合の計算された理論的角度（ここでは -90° ）を無視します。`chemfig` はこの理論的な角度を使用して出発原子と到着原子を決定しますが、結合を描くためには使用しません。なぜなら、両端はすでに定義されているからです。したがって、最後の結合の出発原子は 1 番目の原子となります。

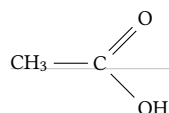
12 こんなときは……

12.1 色付きの原子を描くには

xcolor パッケージが TikZ によって読み込まれ、TikZ は chemfig によって読み込まれるため、分子コード内に色を変更するためのコマンドを書くことができます。主に `\color` と `\textcolor` を使います。原子は TikZ ノード内に表示されますが、これは \TeX のボックスのように動作するので、これらの原子はグループ内に置かれているのと同様になります。したがって、色の変更は当該原子に対してのみ有効です。

色付け

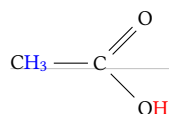
```
\chemfig{C\color{blue}H_3-C(=[1]O)-[7]O\color{red}H}
```



原子を分割するルールが原因で、このコードはうまく動きません。ここでは、最初の原子は“c”から始まり、次の大文字まで続くことになります。するとこの原子は“`C\color{blue}`”となり、色の変更は原子の終わりで発生し、結局のところ何の効果もありません。したがって“|”によって chemfig に最初の原子を“c”の直後での分割を強制し、`\color{blue}H_3` を波括弧で囲むことで、chemfig が大文字の“H”の前で2つ目の原子を分割しないようにする必要があります。さもないと、色を変化させる命令だけが1つの原子として扱われ、やはり効果が無効になってしまいます。

色付け

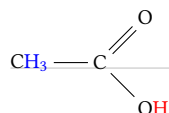
```
\chemfig{C|{\color{blue}H_3}-C(=[1]O)-[7]O|{\color{red}H}}
```



同じ効果を得るために `\textcolor` を使うこともできます。

色付け

```
\chemfig{C|\textcolor{blue}{H_3}-C(=[1]O)-[7]O|\textcolor{red}{H}}
```



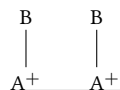
主な欠点は、色付けの必要がある全ての原子に対して、たとえそれらが隣接していたとしても、同じことをしなければならないことです。

12.2 結合を変更せずに上付き文字を追加するには

数学的な指数の形で原子に電荷を加えると、その原子を含むボックス（したがって TikZ ノード）の寸法が変更されます。このことはその原子が鎖の最後にある場合は重要ではありませんが、その原子から結合が出る場合は整列が崩れる場合があります。これを解決する1つ目の方法は、電荷を幅のないボックスに入れる、すなわち `\rlap` コマンドを使用することです¹¹。多くの場合、この方法で良好な結果が得られます。次の例を見ると、`\rlap` を使用することで、原子の水平整列が保持されていることがわかります。

電荷と結合

```
\chemfig{A^{+}[2]B}  
\quad  
\chemfig{A\rlap{$^{+}$}[2]B}
```

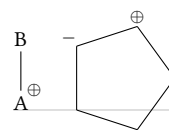


¹¹ 原子の左側に電荷を置く必要がある場合は `\llap` コマンドを使用しなければなりません。

`\charge` マクロを使えば、この処理を簡単かつ正確に実行することができます。

電荷の配置

```
\chemfig{\charge[extra sep=0pt]45[anchor=180+\chargeangle]=%
$\scriptstyle\oplus$\{A\}-[2]B}
\qquad
\chemfig{*5(---\charge{90:2pt=$\scriptstyle\oplus$\}{-}%
\charge{135:2pt=$\scriptstyle-$\}{-})}
```



12.3 結合を曲線で描くには

TikZ の “`decorations.pathmorphing`” ライブラリーを使用すると、波線を使って結合を描くことができます。

波線の結合

```
\chemfig{A-[3,,,decorate,decoration=snake]B}

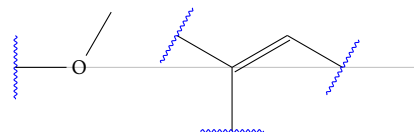
\chemfig{A-[3,,,decorate,decoration={snake,amplitude=1.5mm,
segment length=2.5mm}]B}
```



部分分子 “`wb`” を定義して、直前の結合に対して垂直な波線の結合を描くこともできます。

波線の結合

```
\definesubmol{wb}{%
(-[:90,.5,,,draw=none]%
-[:180,,,blue,decorate,decoration={%
snake,amplitude=0.2mm,segment length=0.75mm
}%
)}}
\chemfig{!{wb}-O-[:60]}
\qquad
\chemfig{([:150]-!{wb})([:90]-!{wb})=[:30]-[:30]!{wb}}
```

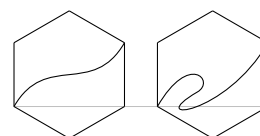


より柔軟性を持たせるために、文字 “`@`” を使用してノードを定義し、分子が描かれた後に `\chemmove` を使ってこれらのノードを曲線で繋げることもできます。

曲った結合

```
\chemfig{@{a}A-[,,,draw=none]@{b}B}
\chemmove{\draw[-](a)..controls +(45:7mm) and +(225:7mm)..(b);}
\bigskip

\chemfig{*6(@{a}---@{b}---)}
\chemmove{\draw[-](a)..controls +(60:3em) and +(240:3em)..(b);}
\quad
\chemfig{*6(@{a}---@{b}---)}
\chemmove{\draw[-](a)..controls +(60:3em) and +(30:1em)..
++(20:2em) ..controls +(210:3em) and +(-120:4em) ..(b);}
```



12.4 重合体の要素を描くには

`\polymerdelim` マクロはこれまで文書化されておらずテスト段階にありましたが、`chemfig` バージョン 1.33 で正式に採用されました。その構文は次のとおりです。

```
\polymerdelim[⟨キーワード⟩=⟨値⟩形式のリスト]{⟨ノード1⟩}{⟨ノード2⟩}
```

おそらく 2 回のコンパイルの後、指定されたノードに垂直の区切り線が配置されます。パラメーターは〈キーワード〉と〈値〉で指定されます。具体的なキーワード、デフォルト値および動作を次表に示します。

〈キーワード〉	デフォルト 〈値〉	動作
<code>delimiters</code>	<code>()</code>	区切り記号を指定します。角括弧を使用する場合は <code>delimiters={[]}</code> と記述します。
<code>height</code>	<code>10pt</code>	区切り記号の（ノードより上の）高さを定義します。
<code>depth</code>	〈空〉	区切り記号の（ノードより下の）深さを定義します。〈値〉が空の場合は、深さは高さと同じ値になります。
<code>h align</code>	<code>true</code>	ブール値を取り、〈 <code>false</code> 〉の場合、2 番目の区切り記号を 2 番目のノードに配置します。このとき、区切り記号が同じ水平線上に配置されない可能性があります。
<code>auto rotate</code>	<code>false</code>	ブール値を取り、〈 <code>true</code> 〉かつ <code>h align = <false></code> の場合に、区切り記号を 2 つのノードを結ぶ線に対して垂直になるように回転させて配置します。
<code>rotate</code>	<code>0</code>	2 つの区切り記号の回転角度を設定します。 <code>h align = <false></code> かつ <code>auto rotate = <false></code> の場合のみ有効です。
<code>open xshift</code>	<code>0pt</code>	開き区切り記号の水平オフセットを設定します。
<code>close xshift</code>	〈空〉	閉じ区切り記号の水平オフセットを設定します。〈値〉が空の場合、このオフセットは開き区切り記号の水平オフセットの逆になります。
<code>indice</code>	<code>n</code>	閉じ区切り記号の右下に配置する添え字を定義します。

重合体

ポリエチレン:

```
\chemfig{\vphantom{CH_2}-[@{op,.75}]CH_2-CH_2-[@{cl,0.25}]}
\polymerdelim[height = 5pt, indice = \!\!n]{op}{cl}
\bigskip
```

ポリ塩化ビニル:

```
\chemfig{\vphantom{CH_2}-[@{op,1}]CH_2-CH(-[6]Cl)-[@{cl,0}]}
\polymerdelim[height = 5pt, depth = 25pt, open xshift = -10pt, indice = \!\!n]{op}{cl}
\bigskip
```

ナイロン6:

```
\chemfig{\phantom{N}-[@{op,.75}]{N}(-[2]H)-C(=[2]O)-({CH_2})_5-[@{cl,0.25}]}
\polymerdelim[height = 30pt, depth = 5pt, indice = {}]{op}{cl}
\bigskip
```

ポリカブロラクタム:

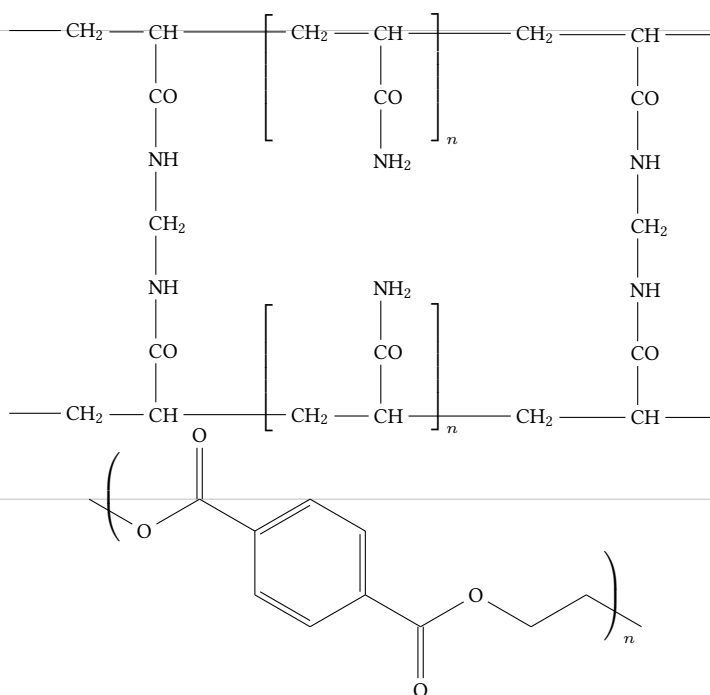
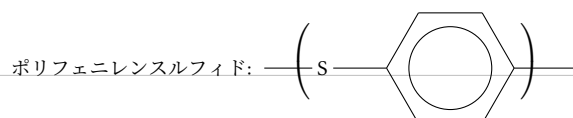
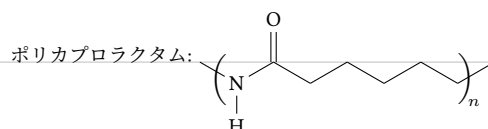
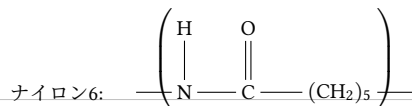
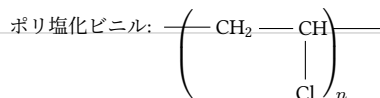
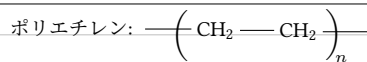
```
\chemfig{atom sep = 2em}{[: -30]-[@{left,.75}]N(-[6]H)-[:30](=[2]O)--[:30]--[:30]--[@{right,0.25}:30]}
\polymerdelim[height = 5pt, indice = \!\!n]{left}{right}
\bigskip
```

ポリフェニレンスルフィド:

```
\chemfig{\vphantom{S}-[@{op,.75}]S-(**6(---(-[@{cl,0.25}])---))}
\polymerdelim[delimiters = (), height = 15pt, indice = {}]{op}{cl}
\bigskip
```

```
\chemfig{CH_2-CH([6]-CO-NH-CH_2-NH-CO-CH([4]-CH_2-)([0]-[@{downleft,0.8},2]CH_2
-CH([2]-CO-NH_2)-[@{downright,0.3},2]CH_2-[1.5]C?H-)-[@{upleft,0.8},2]CH_2
-CH([6]-CO-NH_2)-[@{upright,0.3},2]CH_2-[1.5]CH([6]-CO-NH-CH_2-NH-C?O)-}
\polymerdelim[delimiters={[]}, height = 5pt, depth = 40pt, indice = n]{upleft}{upright}
\polymerdelim[delimiters={[]}, height = 40pt, depth = 5pt, indice = n]{downleft}{downright}

\chemfig{ -[@{op,.5}: -30]O-[::60](=[::60]O)-[: -60]*6(==(-([::60]O)-[:60]O-[: -60]-[:60]-[@{cl,.5}: -60])=))}
```

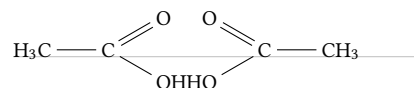
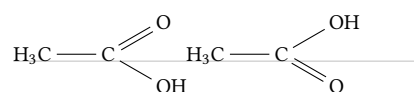
$$\backslash\mathrm{polymerdelim}[height=6ex, indice=n, h align=false]\{op\}\{cl\}$$


12.5 分子の反転形を描くには

`\hflipnext` コマンドと `\vflipnext` コマンドはそれぞれ、次に書かれた分子を水平または垂直軸に対して対称に描きます。複数の反転形を描きたい場合は、それぞれの対象分子の前にこれらのコマンドを書かなければなりません。

反転形

```
\chemfig{H_3C-C(=[:30]O)-[:-30]OH}% オリジナル
\vflipnext
\chemfig{H_3C-C(=[:30]O)-[:-30]OH}\medskip
```



12.6 結合に沿ってテキストを追加したり、角度の弧を描くには

“@”が「グローバルな」TikZ ノード、つまり分子が描かれた後にアクセス可能なノードを配置することを理解したならば、TikZ がノードに対してできること（つまり多くのこと）が可能になります。

結合の上または下に何かを書くには、この結合の両端にある原子それぞれに「グローバル」ノードを置き、それらの間にテキストを配置した上で、結合のちょうど上または下に来るように位置を上げ下げします。これを実現するのが、以下のコードの `\namebond` マクロです。

2 つの結合の間に弧を描くには、3 つの原子の上にそれぞれ「グローバル」ノードを置く必要があります。`\arcbetweennodes` マクロは、あるノードから引かれた 2 つの線の間の角度を計算します。次に、`\arclabel` が 2 つの結合の間に弧を描き、その弧のそばにテキストを書きます。このマクロの最初のオプション引数は、弧をカスタマイズするために使用される TikZ コードです。2 番目の引数は弧の半径で、次の 3 つの引数は弧を描くべきグローバルノードの名前であり、2 つ目に頂点になるノードを指定します。最後の引数は、弧のそばに書かれるテキストです。

弧と結合に沿ったテキスト

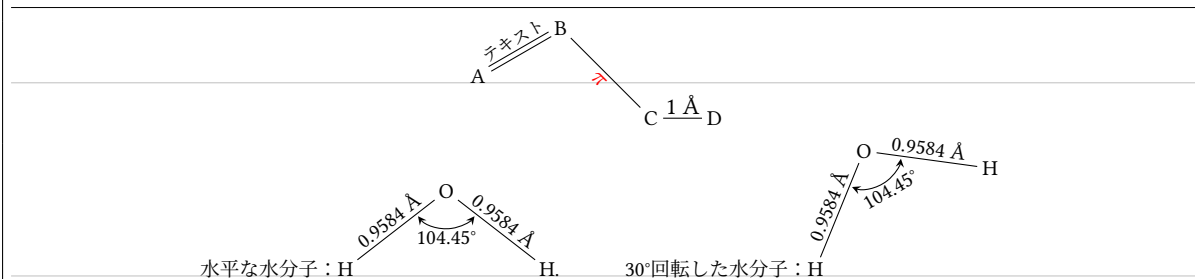
```
\newcommand\angstrom{\mbox{\normalfont\AA}}
\newcommand\namebond[4][5pt]{\chemmove{\path(#2)--(#3)node[midway,sloped,yshift=#1]{#4}};}

\newcommand\arcbetweennodes[3]{%
  \pgfmathanglebetweenpoints{\pgfpointanchor{#1}{center}}{\pgfpointanchor{#2}{center}}%
  \let#3\pgfmathresult}

\newcommand\arclabel[6][stealth-stealth,shorten <=1pt,shorten >=1pt]{%
  \chemmove{%
    \arcbetweennodes{#4}{#3}\anglestart \arcbetweennodes{#4}{#5}\angleend
    \draw[#1]([shift=(\anglestart:#2)]#4)arc(\anglestart:\angleend:#2);
    \pgfmathparse{(\anglestart+\angleend)/2}\let\anglestart\pgfmathresult
    \node[shift=(\anglestart:#2+1pt)#4,anchor=\anglestart+180,rotate=\anglestart+90,inner sep=0pt,
      outer sep=0pt]at(#4){#6};}}

\chemfig{@{a}A=[:30,1.5]@{b}B-[7,2]@{c}C-@{d}D}
\namebond{a}{b}{\scriptsize テキスト}
\namebond[-3.5pt]{b}{c}{\small\color{red}$\pi$}
\namebond{c}{d}{\small1 \angstrom}
\medskip

水平な水分子:\chemfig{@{1}H-[:37.775,2]@{2}O-[::-75.55,2]@{3}H}.
\namebond{1}{2}{\footnotesize0.9584 \angstrom}
\namebond{2}{3}{\footnotesize0.9584 \angstrom}
\arclabel{0.5cm}{1}{2}{3}{\footnotesize104.45\textdegree}
\qquad
30°回転した水分子:\chemfig{[:30]@1H-[:37.775,2]@2O-[::-75.55,2]@3H}
\namebond12{\footnotesize0.9584 \angstrom}
\namebond23{\footnotesize0.9584 \angstrom}
\arclabel{0.5cm}{1}{2}{3}{\footnotesize104.45\textdegree}
```

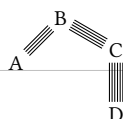


12.7 多重結合を描くには

TikZ の “decorations.markings” ライブラリーを使うと、多重結合を書くことができます。

多重結合

```
\catcode'\_ =11
\tikzset{nbond/.style args={#1}{%
  draw=none,%
  decoration={%
    markings,%
    mark=at position 0 with {\coordinate (CFstart@) at (0,0);},
    mark=at position 1 with {%
      \foreach\CF_i in{0,1,...,\number\numexpr#1-1}{%
        \pgfmathsetmacro\CF_nbondcoeff{\CF_i-0.5*(#1-1)}%
        \draw ([yshift=\CF_nbondcoeff\CF_doublesep]CFstart@)--(0,\CF_nbondcoeff\CF_doublesep);
      }%
    },
  },
  postaction={decorate}
}
}
\catcode'\_ =8
\chemfig{A-[1,,,nbond=4]B-[:30,,,nbond=5]C-[6,,,nbond=6]D}
```



反応式

ユーザーからの複数のリクエストにより、chemfig が反応式の描画に関して弱点を抱えていることが明らかになりました。この欠点は今や解消され、chemfig はバージョン 1.0 になりました。これは、そのための主要な機能がすでに利用可能になっていると考えているためです。

この部で紹介される新機能に関するテストと支援をしてくれた Clemens NIEDERBERGER に感謝します。

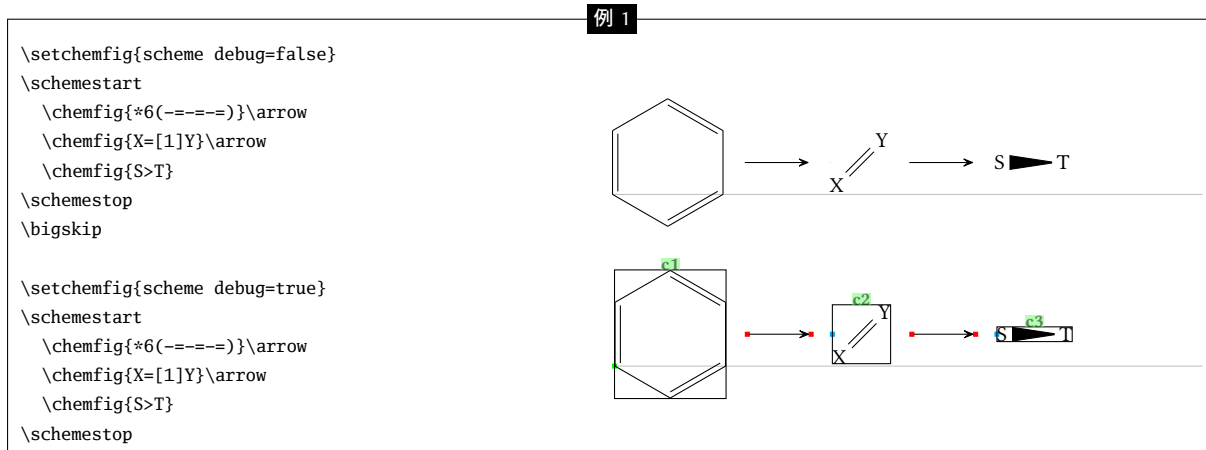
1 概要

反応式は次の文法で記述されます。

`\schemestart[角度,係数,スタイル][位置] <反応> \schemestop`

重要： 今後のいずれかのバージョンにおいてオプション引数の文法が変更されます。2 番目のオプション引数は削除され、1 つ目のオプション引数で **キーワード = <値>** 形式のリストを与えるようになります。したがって、現在のオプション引数で許可されている設定を行うには、以下の <キーワード> を使うことをおすすめします。 `arrow angle`、`arrow coeff`、`arrow style` および `init anchor`。

次の例に示すように、<キーワード> `scheme debug` に値 `<true>` または `<false>` を設定することで、デバッグ情報の表示・非表示を設定できます。



以下の点に注意してください。

- `\arrow` コマンドは矢印を描画します。
- 2 つの `\arrow` コマンドの間にあるものは全て化合物と見なされます。矢印か化合物かによらず全ての設定は、`\arrow` コマンドの引数によって制御されます。その構文はかなり複雑になる場合があります。
- 矢印は水平方向に描画されますが、これはもちろん変更可能です。
- 矢印は化合物のバウンディングボックスの中心を結ぶ仮想的な線の上に描画されます。(赤い四角と青い四角は矢印のアンカーポイントです。) この動作も変更可能です。

- デバッグ情報は〈キーワード〉 `scheme debug` で表示・非表示が決まります。true の場合は以下のものが表示されます。
 - バウンディングボックスの上にかかれる緑のラベル。これは、chemfig によって化合物に割り当てられたデフォルト名であり、“c1”、“c2” のように連続した番号が振られます。番号は各反応式ごとに 1 にリセットされます。
 - 化合物のバウンディングボックス
 - 赤い点で示される矢印の始点と終点、および青い点で示されるアンカー
- 2 つの化合物の端から端までの距離は、〈キーワード〉 `compound sep = 〈長さ〉` で定義されます。この〈長さ〉のデフォルトは 5em です。
- 最後に、化合物の端と矢印の始点および終点の間の距離は、〈キーワード〉 `arrow offset = 〈長さ〉` で定義されます。この〈長さ〉のデフォルトは 4pt です。

以下に \schemestart マクロの全パラメーター、デフォルト値、および簡単な説明を示します。これらのパラメーターは \setchemfig{〈キーワード〉=〈値〉} マクロを使用して設定する必要があります。

〈キーワード〉	デフォルト 〈値〉	説明
<code>schemestart code</code>	〈空〉	入れ子ではない反応式の最初に実行されるコード
<code>schemestop code</code>	〈空〉	入れ子ではない反応式の最後に実行されるコード
<code>scheme debug</code>	false	この値が <code><true></code> のとき、ノード、名前およびアンカーを表示
<code>compound style</code>	〈空〉	化合物のスタイル
<code>compound sep</code>	5em	化合物間の間隔
<code>name sep</code>	1.5ex	化合物のボックスとその名前の垂直方向の最小距離
<code>arrow offset</code>	1em	化合物と矢印の間隔
<code>arrow angle</code>	0	反応の矢印の角度
<code>arrow coeff</code>	1	矢印の長さを与える compound sep に対する係数
<code>arrow style</code>	〈空〉	矢印のスタイル
<code>arrow double sep</code>	2pt	両矢印の間隔
<code>arrow double coeff</code>	0.6	両矢印 “<->” および “<<->” の短い矢印の縮小率
<code>arrow double harpoon</code>	true	両矢印の矢尻を片羽で描画
<code>arrow label sep</code>	3pt	矢印とそのラベルの間隔
<code>arrow head</code>	-CF	矢印の矢尻のスタイル
<code>+ sep left</code>	0.5em	+ 記号の前の空き
<code>+ sep right</code>	0.5em	+ 記号の後の空き
<code>+ vshift</code>	0pt	+ 記号の垂直方向の移動距離

重要：一貫性を保つため、次期バージョンの chemfig では \schemestart の文法が変更されます。これまでの 2 つのオプション引数は削除され、〈キーワード〉=〈値〉形式のリストを取るオプション引数に置き換えられます。これまで 2 つのオプション引数で指定されていた設定は、新しい〈キーワード〉で設定できるようになります。

2 矢印の種類

\arrow コマンドの後に波括弧内に書かれたオプション引数（これは必須ではありません）が続く場合、その引

矢印の種類	
<code>\schemestart A\arrow{->}B\schemestop\par % デフォルト</code>	$A \longrightarrow B$
<code>\schemestart A\arrow{->}B \schemestop\par</code>	$A \longrightarrow B$
<code>\schemestart A\arrow{<-}B \schemestop\par</code>	$A \longleftarrow B$
<code>\schemestart A\arrow{<->}B \schemestop\par</code>	$A \longleftrightarrow B$
<code>\schemestart A\arrow{<=>}B \schemestop\par</code>	$A \rightrightarrows B$
<code>\schemestart A\arrow{<->}B \schemestop\par</code>	$A \rightleftharpoons B$
<code>\schemestart A\arrow{<<->}B \schemestop\par</code>	$A \rightleftarrows B$
<code>\schemestart A\arrow{0}B \schemestop\par</code>	$A \longrightarrow B$
<code>\schemestart A\arrow{-U>}B \schemestop</code>	$A \longrightarrow B$

以降では分かりやすさのために、いくつかの例を除き、`\chemfig` コマンドによる化学式の代わりに大文字を使用します。もちろん、反応式は文字でも分子でも同様に機能します。ギャラリーには反応式の例が示されています。

各矢印は以下により特徴付けられます。

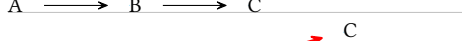
- これらの特徴は以下の〈キーワード〉で定義されます。

- デフォルト値の定義**

```

\setchemfig{arrow angle=15,arrow coeff=1.5,
arrow style={red, thick}}
\schemestart A\arrow B\arrow C\schemestop


```



```

\setchemfig{arrow coeff=2.5,arrow style=dashed}
\schemestart A\arrow B\arrow C\schemestop


```



```

\setchemfig{arrow angle={},arrow coeff={},arrow style={}}
\schemestart A\arrow B\arrow C\schemestop

```



コンパイル日：2025 年 11 月 3 日

オプション引数

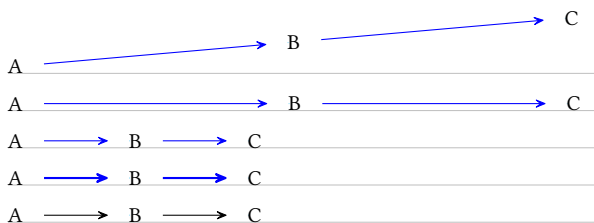
```
\setchemfig{arrow angle=5,arrow coeff=2.5,arrow style=blue}
\schemestart A\arrow B\arrow C\schemestop

\schemestart[0] A\arrow B\arrow C\schemestop

\schemestart[0,1] A\arrow B\arrow C\schemestop

\schemestart[0,1,thick] A\arrow B\arrow C\schemestop

\schemestart[0,1,black] A\arrow B\arrow C\schemestop
```

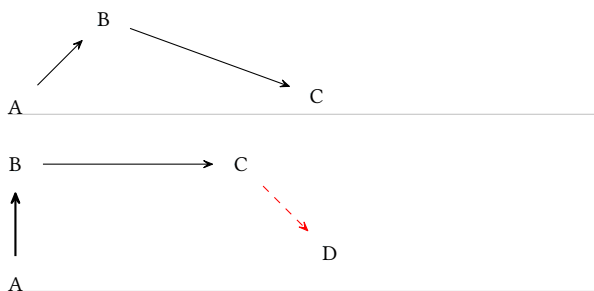


スタイルに関するルールは次のとおりです。角括弧内の引数で指定されたスタイルはデフォルトスタイルの後に適用され、指定したスタイルがデフォルトで上書きされることはありません。このため上の例では、“black”と書いた場合のみ、デフォルトスタイル「青」を上書きできます。

最後に、`\arrow` コマンドは与えられた矢印の特徴を変更するために、角括弧内のオプション引数 [角度, 係数, スタイル] を取ります。前述の場合と同様に、このスタイルはデフォルトスタイルと `\schemestart` コマンドのオプション引数で指定されたスタイルの後に適用されるので、デフォルトスタイルが角括弧内のオプション引数を上書きすることはありません。

矢印の特徴付け

```
\schemestart
  A\arrow[45]B\arrow[-20,2]C
\schemestop
\bigskip
\schemestart
  A\arrow[90,,thick]B\arrow[,-2]C
  \arrow[-45,,dashed,red]D
\schemestop
```



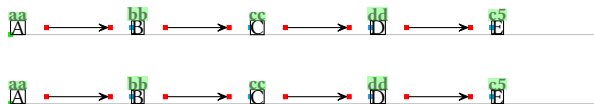
4 化合物名

化合物に自動的に付けられた名前 (“c1”、“c2” など) は上書きできます。そのためには `\arrow` コマンドの直後に、(n1--n2) の形式の引数を書きます。矢印の始まりと終わりに位置する化合物は、それぞれ “n1” と “n2” と名付けられます。任意の英数字の文字列を使用できます。名前 “c<n>” の番号付けは内部的に続行されるため、化合物の名前がデフォルトから変更されていても、以降の化合物のデフォルト名に影響しません。

名前はオプションであり、引数は (n1--) や (--n2) でも構いません。

化合物名

```
\setchemfig{scheme debug=true}
\schemestart
  A\arrow(aa--bb)B\arrow(--cc)C\arrow(--dd)D\arrow E
\schemestop
\bigskip
\schemestart
  A\arrow(aa--)B\arrow(bb--)C\arrow(cc--dd)D\arrow E
\schemestop
```



この例のように、2つの方法は同等であり、入ってくる矢印によっても、出ていく矢印によっても化合物に名前を付けることができることに注意してください。ただし、化合物がその名前を指定する2つの矢印に挟まっている場合、最初の名前は無視され、その旨の警告メッセージが表示されます。

過剰な命名

```
\setchemfig{scheme debug=true}
\schemestart
  A\arrow{--foo}B\arrow{bar--}C
\schemestop
```



ここで化合物“B”は、それを指す矢印によって“foo”と名付けられ、そこから出る矢印によって“bar”と名付けられています。したがって chemfig は、名前“foo”が無視されることを警告するメッセージを表示します。

Package chemfig Warning: two names for the same node, first name "foo" ignored
(パッケージ chemfig 警告: 同じノードに対する 2 つの名前。最初の名前 "foo" が無視されました。)

5 アンカー

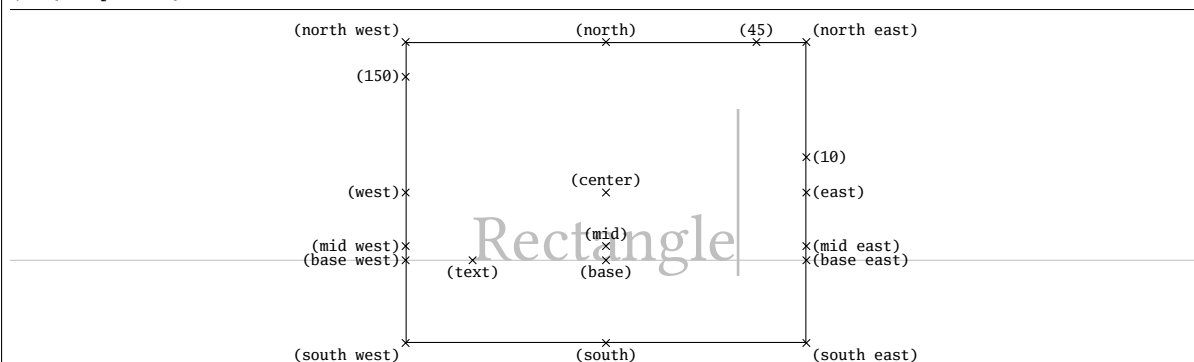
上述のように、矢印は化合物のバウンディングボックスの中心を結ぶ線上にあります。このデフォルトのアンカーは TikZ 用語では“center”と呼ばれます。デフォルトではないアンカーポイントを使いたい場合は、括弧内に引数を書いて指定することもできます。

(n1.a1--n2.a2)

ここで“a1”や“a2”がアンカーであり、north west、north、north east、west、center、east、mid west、mid、mid east、base west、base、base east、south west、south、south east、text および任意の角度を指定できます。次の図は TikZ のマニュアル内の例で、バウンディングボックスに配置されているアンカーを示しています。

TikZ のアンカー

```
\Huge
\begin{tikzpicture}[baseline]
\node[anchor=base west,name=x,draw,inner sep=25pt] {\color{lightgray}Rectangle\vrule width 1pt height 2cm};
\foreach \anchor/\placement in
{north west/above left, north/above, north east/above right,west/left, center/above, east/right,
mid west/left, mid/above, mid east/right,base west/left, base/below, base east/right,
south west/below left, south/below, south east/below right,text/below,10/right,45/above,150/left}
\draw[shift=(x.\anchor)] plot[mark=x] coordinates{(0,0)}
\node[placement,inner sep=0pt,outer sep=2pt] {\scriptsize\texttt{(\anchor)}};
\end{tikzpicture}
```

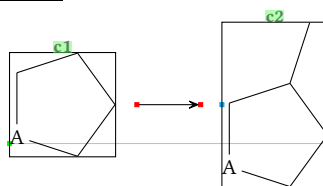


名前と同様に、到着および出発アンカーポイントは独立であり、オプションです。

次の例は、2つの“A”が異なる垂直位置にあり、デフォルトの整列がうまくいってない場合です。デバッグ情報から、デフォルトの“center”アンカーは適切でないことが分かります。

整列の問題

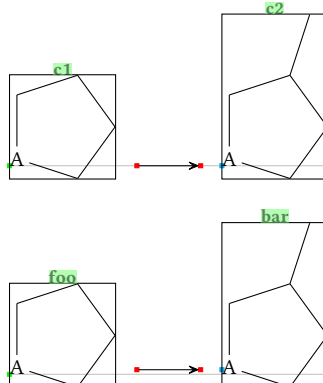
```
\setchemfig{scheme debug=true}
\schemestart
\chemfig{A*5(-----)}
\arrow
\chemfig{A*5(---(-)--)}
\schemestop
```



整列が正しくなるためには、矢印は“base east”から“base west”へ、あるいは“mid east”から“mid west”へ出発・到着する必要があります。

整列の問題

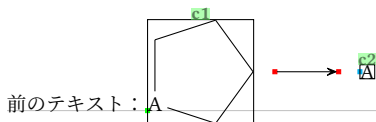
```
\setchemfig{scheme debug=true}
\schemestart
\chemfig{A*5(-----)}
\arrow(.base east--.base west)
\chemfig{A*5(---(-)--)}
\schemestop
\bigskip
\schemestart
\chemfig{A*5(-----)}
\arrow(foo.mid east--bar.mid west)
\chemfig{A*5(---(-)--)}
\schemestop
```



最後に、次の図の左側の化合物に緑の点として示しているアンカーは、直前のテキストのベースラインに対する最初の化合物のアンカーです。

開始アンカー

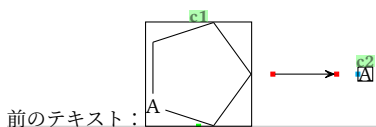
```
\setchemfig{scheme debug=true}
前のテキスト：
\schemestart
\chemfig{A*5(-----)}\arrow A
\schemestop
```



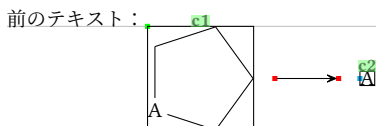
最初の化合物のバウンディングボックスにおけるこのアンカーのデフォルトは“text”です。この位置は、`\schemestart` コマンドの 2 番目のオプション引数で制御できます。

初期アンカーの調整

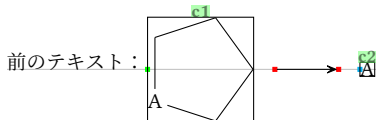
```
\setchemfig{scheme debug=true}
前のテキスト：
\schemestart[][south]
\chemfig{A*5(-----)}\arrow A
\schemestop
\bigskip
```



```
前のテキスト：
\schemestart[][north west]
\chemfig{A*5(-----)}\arrow A
\schemestop
\bigskip
```



```
前のテキスト：
\schemestart[][west]
\chemfig{A*5(-----)}\arrow A
\schemestop
```

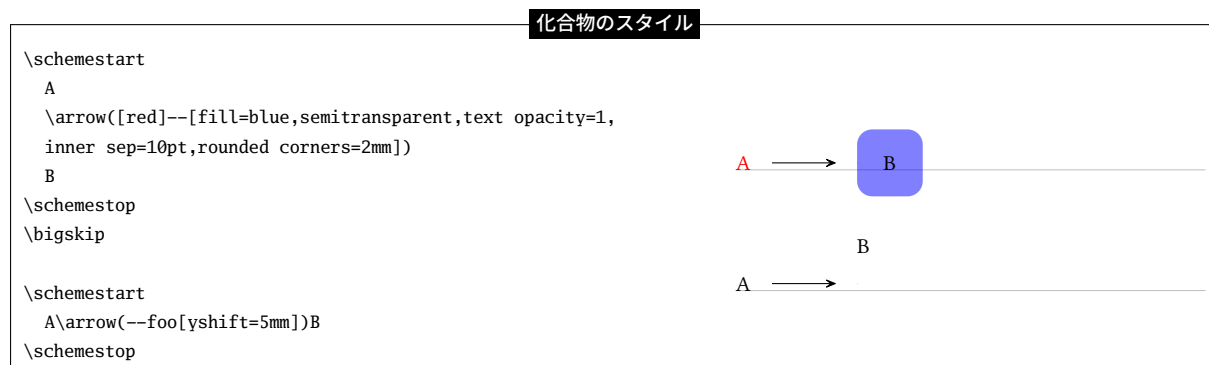


6 化合物のスタイル

`\arrow` コマンドに括弧内に書かれた引数を与えると、反応における反応物と生成物のバウンディングボックスのスタイル“s”を指定することができます。出発・到着化合物のスタイル“s”は TikZ コードを使用して記述し、それぞれ角括弧の引数として与えます。したがって `\arrow` コマンドの完全な構文は次のようになります。ここで、各指定はオプションであり省略可能です。

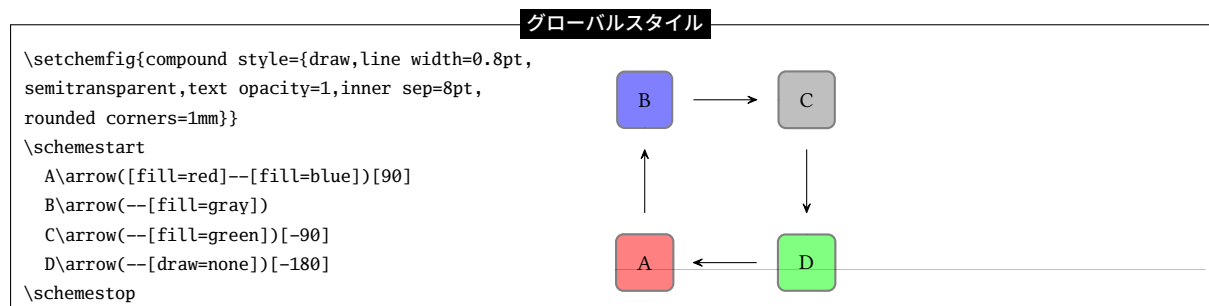
`\arrow(n1.a1[s1]--n2.a2[s2]){矢印の種類}[角度,係数,矢印のスタイル]`

名前の場合と同様に、同一の化合物に対して到着する矢印と出発する矢印の両方にスタイルが指定されている場合、最初のスタイルは警告とともに無視されます。



〈キーワード〉 `compound style =` 〈tikz コード〉を設定すると、それ以降に描画される化合物のスタイルをグローバルに定義することができます。空の引数を与えるとスタイルなし、つまりデフォルトになります。

次の例では、角が丸まった矩形と半透明の背景を持つスタイルが指定されています。

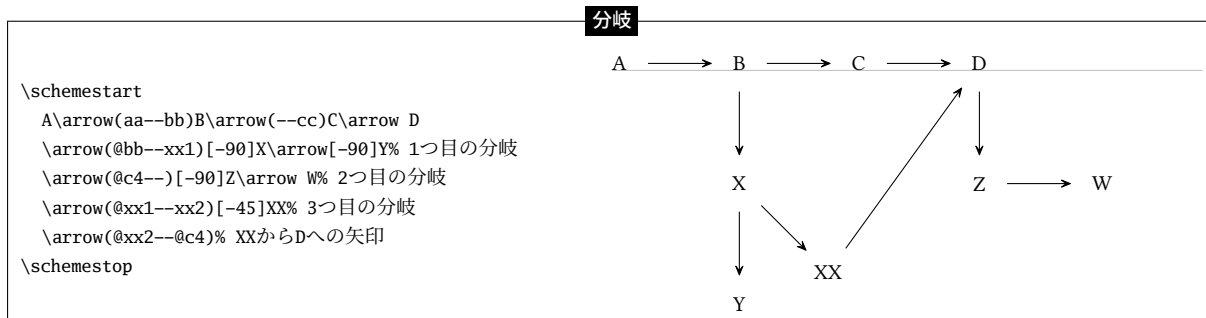


7 分岐

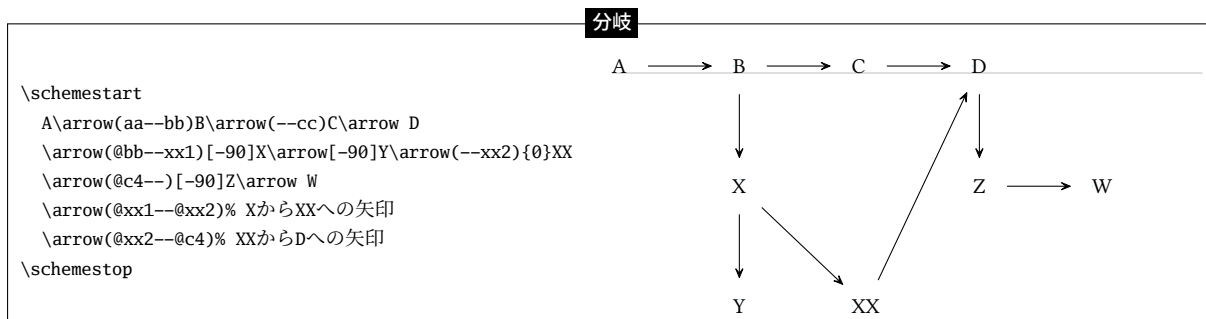
これまでは一直線の反応式のみを扱ってきました。分岐する反応式も描くことができますが、ここで化合物名が重要な役割を果たします。`\arrow` コマンドの括弧の引数内で名前の前に“@”が付いている場合は、その化合物がすでに存在することを意味します。いくつかのケースが考えられます。

- (`@n1--n2`)：矢印は既存の化合物“n1”から出発し、その矢印の後に反応図が続くため、分岐が作成されます。
- (`@n1--@n2`)：それらがすでに定義されているか反応式内のこれより後で定義されるかにかかわらず、矢印は2つの既存の化合物の間に描かれます。したがってこの構文は反応式のコード内のどこにでも書くことができます。
- (`n1--@n2`)：この構文は許されません。

次図は、まず“B”から、次に“D”から、最後は“X”からの3つの分岐が作成される例です。最後に2つの既存の化合物“XX”と“D”の間に矢印が描かれます。

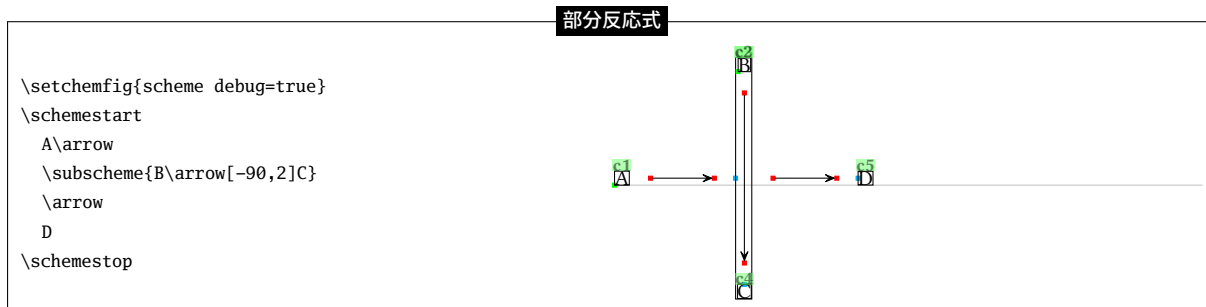


“Y”と“XX”を同じ水平線上の高さに配置したいかもしれません。そのためには“Y”と“XX”の間に水平の見えない結合を描きます。そして、2つの既存の化合物“XX”と“D”の間に最後の矢印を描いて、この反応式を完成させます。



8 部分反応式

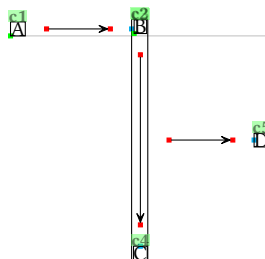
反応式の一部を単一のバウンディングボックスに配置し、chemfig に1つの化合物として扱わせることができます。部分反応式の定義は `\subscheme` コマンドの必須の引数として波括弧内に記述され、それ以降は単一の要素として扱われます。`\subscheme` が矢印の後に置かれた場合は、このコマンドはこの部分反応式に化合物としての名前“`c<n+1>`”を付けます。



ラベルが重なっているために明確には見えませんが、部分反応式の周りのボックスの名前は“`c2`”であり、部分反応式内ではBに“`c3`”、Cに“`c4`”という名前が付いています。部分反応式内の最初の化合物は“B”なので、部分反応式のベースラインは“B”のベースラインとなります。アンカーを指定してみると、それが明確になります。

部分反応式

```
\setchemfig{scheme debug=true}
\schemestart
  A\arrow{--mid west}
  \subscheme{B\arrow[-90,2]C}
  \arrow
  D
\schemestop
```



“`\subscheme{<反応式>}`” は単なる

`\schemestart<反応式>\schemestop`

の便利なショートカットにすぎないので、`\schemestart` と同じオプション引数を使用することができます。

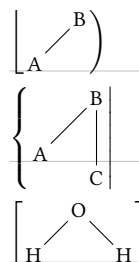
`chemfig` では `\chemleft` と `\chemright` のコマンド対を使用できます。これにより、要素の両側に伸縮可能な区切り記号を設定することができます。T_EX のプリミティブコマンド `\left` と `\right` の場合と同様に、コマンドの後には区切り記号が続かなければなりません。

`\chemleft<car1><要素>\chemright<car2>`

ここで `<car1>` と `<car2>` は、“(” と “)” あるいは “[” と “]” など、`\left`、`\right` コマンドで使用可能な任意の伸縮可能な区切り記号です。

`\chemleft`、`\chemright` マクロ

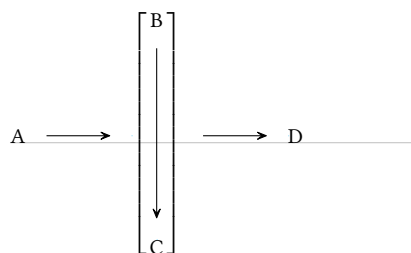
```
\chemleft\lfloor\chemfig{A-[1]B}\chemright)
\chemleft{\chemfig{A-[1,1.25]B-[6,1.25]C}\chemright|
\chemleft[\chemfig{H-[1]O-[7]H}\chemright]
```



上述の反応式のコードに `\chemleft` と `\chemright` を加えると、次のような出力が得られます。

`\chemleft` と `\chemright` を使った反応式

```
\schemestart
  A\arrow
  \chemleft[\subscheme{B\arrow[-90,2]C}\chemright]
  \arrow
  D
\schemestop
```



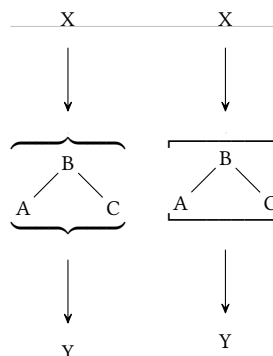
同様に、`\chemup` マクロおよび `\chemdown` マクロを使用すると、それぞれ要素の上と下に伸縮可能な区切り記号を描くことができます。

`\chemup<car1><要素>\chemdown<car2>`

例えば、次のような図を描くことができます。

\chemup マクロと \chemdown マクロ

```
\schemestart[-90]
X\arrow
\chemup{\chemfig{A-[1]B-[7]C}\chemdown\}
\arrow Y
\schemestop
\qqquad
\schemestart[-90]
X\arrow
\chemup{\chemfig{A-[1]B-[7]C}\chemdown}
\arrow Y
\schemestop
```



区切り記号は化合物のスタイル指定を使って描くこともでき、任意の化合物（したがって部分反応式を含みます）に適用できます。このとき伸縮可能な区切り記号（括弧、角括弧、波括弧）は、文書のプリアンプルに次のように書いて TikZ の “matrix” ライブラリーを読み込むことで使用できます。

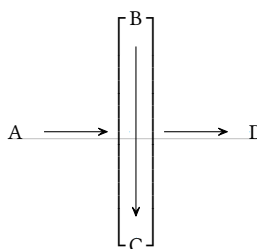
\usetikzlibrary{matrix}

chemfig パッケージでは \chemleft、\chemright、\chemup および \chemdown コマンドが利用可能なので、chemfig はこのライブラリーを自動的には読み込みません。したがって “matrix” ライブラリーが提供する区切り記号を使用したい場合は、ユーザーが明示的にこのライブラリーを読み込む必要があります。

“matrix” ライブラリーを使って前述の反応式を書き直すと、次のようになります。

“matrix” ライブラリーによる区切り記号

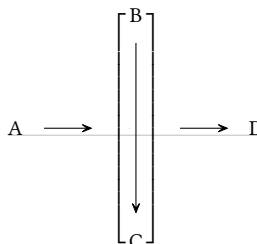
```
\schemestart
A\arrow(--[left delimiter={[]}, right delimiter={[]}])
\subscheme{B\arrow[-90,2]C}
\arrow
D
\schemestop
```



区切り記号がバウンディングボックスの外に描かれるため、入ってくる矢印と出ていく矢印を少し短くすると良いでしょう。

部分反応式

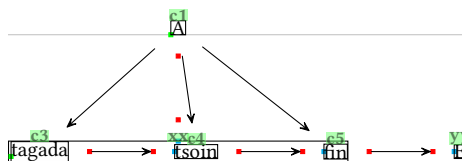
```
\schemestart
A\arrow(--[left delimiter={[]},
right delimiter={[]}][,shorten >=6pt]
\subscheme{B\arrow[-90,2]C}
\arrow[,shorten <=6pt]
D
\schemestop
```



部分反応式は注意して使用しないと、望ましくない結果をもたらすことがあります。次の例では、部分反応式を使用して 3 つの化合物を水平方向に整列させています。

部分反応式

```
\setchemfig{scheme debug=true}
\schemestart
A
\arrow{0}{-90}
\subscheme{%
tagada\arrow{}
tsoin\arrow{}
fin}
\arrow{xx--yy}{E}
\arrow{@c1--@c3}{E}
\arrow{@c1--@c5}{E}
\arrow{@c1--@c4}{E}
\schemestop
```

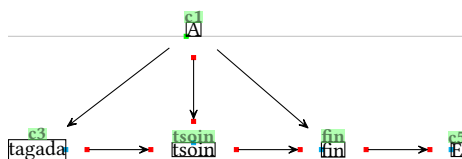


部分反応式の中心と化合物“A”の中心は、完全に同じ垂直線上に位置しています。これは、2つの要素が -90° の見えない矢印で接続されているからです。しかし、もともとあった化合物“A”と“tsoin”の間の矢印は**垂直ではありません**。“tagada”は“fin”よりも幅が広いので、“tsoin”が部分反応式の中央にならないからです。この矢印を `\subscheme` コマンドを使用して垂直にするには、見えない矢印の到着アンカーの正しい角度を試行錯誤で見つけないければなりません。

この場合は、部分反応式を使用せずに分岐を利用する方がはるかに簡単です。“A”と“tsoin”の間に**見える**矢印を引き、“tsoin”から左右に水平な矢印を引いた上で、右側への矢印のための分岐を作成します。

部分反応式

```
\setchemfig{scheme debug=true}
\schemestart
A
\arrow{--tsoin}{->}{-90}
tsoin
\arrow{<-}{180}
tagada
\arrow{@tsoin--fin}{E}
fin
\arrow{}
E
\arrow{@c1--@c3}{E}
\arrow{@c1--@fin}{E}
\schemestop
```



9 矢印のオプション引数

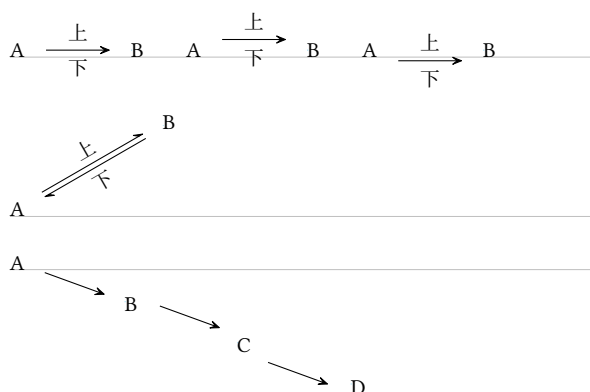
`\arrow` コマンドの波括弧内に書かれた引数の中では、矢印の名前の後に角括弧を使ったオプション引数を続けることができます。矢印のオプション引数が取り得る値とその意味は、`chemfig` において次のように定義されています。

- 矢印“ \rightarrow ”、“ \leftarrow ”、“ \leftrightarrow ”、“ \rightleftharpoons ”、“ \longleftrightarrow ”、“ \rightleftarrows ”、“ \rightleftharpoons ”には、3つのオプション引数があります。
 - 最初の引数は、矢印の上に配置される「ラベル」です。
 - 2番目の引数は、矢印の下に配置される「ラベル」です。これら2つのラベルの向きは、矢印と同じ角度になります。矢印とラベルのアンカーの間の、矢印に対して垂直方向の間隔は、〈キーワード〉 `arrow label sep = 〈長さ〉` で調整でき、その値はデフォルトで 3pt です。2つのオプション引数に含まれるラベルは**数式モードではありません**。
 - 3番目の引数は、矢印に適用される、矢印に対して垂直方向の移動の距離です。この距離が正の場合は矢印（およびラベルがある場合はそのラベル）が上に、負の場合は下に移動します。

- 矢印“-U>”には5つのオプション引数があります。
 - 最初の3つは、他の矢印と同様です。
 - 4番目は、弧の半径を与える、矢印の長さに対する倍率（デフォルトは0.333）です。
 - 5番目は、弧の中心角の半分で、デフォルトは60度です。
- 見えない矢印“0”は2つのオプション引数を取り、それぞれ他の矢印の最初の2つのオプション引数と同様です。

矢印のオプション引数

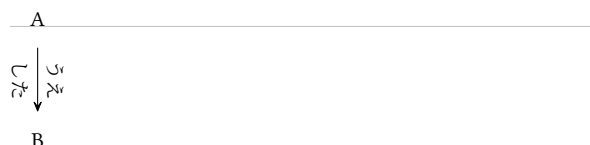
```
\setchemfig{scheme debug=false}
\schemestart A\arrow{>[上][下]}B \schemestop
\qqquad
\schemestart A\arrow{>[上][下][4pt]}B \schemestop
\qqquad
\schemestart A\arrow{>[上][下][-4pt]}B \schemestop
\medskip
\schemestart A\arrow{<=>[上][下]}[30,1.5]B \schemestop
\medskip
\schemestart[-20]
  A\arrow{>}B\arrow{>}[3pt]C\arrow{>}[-3pt]D
\schemestop
```



矢印が鉛直方向の場合は問題が発生します。

鉛直方向の矢印

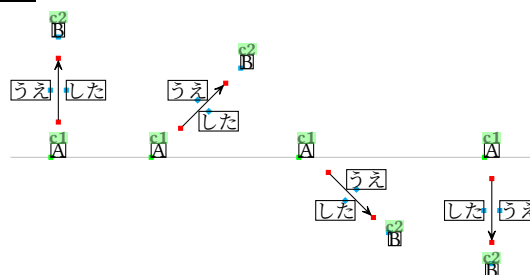
```
\schemestart
  A\arrow{>[うえ][した]}[-90]B
\schemestop
```



読みやすさのために、「うえ」と「した」のラベルを水平に書きたいと思うかもしれません。ラベルの角度は指定することができ、デフォルトは矢印と同じ角度です。別の角度を指定するには、オプション引数の最初に*{角度}と書きます。

ラベルの角度指定

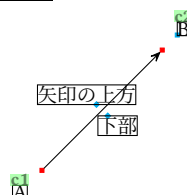
```
\setchemfig{scheme debug=true}
\schemestart A\arrow{>[*{0}うえ][*{0}した]}[90]B\schemestop
\qqquad
\schemestart A\arrow{>[*{0}うえ][*{0}した]}[45]B\schemestop
\qqquad
\schemestart A\arrow{>[*{0}うえ][*{0}した]}[-45]B\schemestop
\qqquad
\schemestart A\arrow{>[*{0}うえ][*{0}した]}[-90]B\schemestop
```



ラベルのアンカーの位置がデフォルトのままの場合は、望ましくない結果になる可能性があります。

Anchors

```
\setchemfig{scheme debug=true}
\schemestart
  A\arrow{>[*{0}矢印の上方][*{0}下部]}[45,2]B
\schemestop
```



このような問題を避けるために、`*{角度}.{アンカー}` の構文でアンカーを指定して、`chemfig` のデフォルトを上書きすることができます。

アンカー

```
\setchemfig{scheme debug=true}
\schemestart
  A\arrow{->[*{0.0}矢印の上方][*{0.180}下部]}[45,2]B
\schemestop
\qquad
\schemestart
  A\arrow{->[*{0.south east}矢印の上方][*{0.north west}下部]}[45,2]B
\schemestop
```

特別な矢印 “-U>” は、最初の 2 つのオプション引数の少なくとも一方にラベルが含まれている場合、対応する弧が描画されます。

The -U> arrow

```
\schemestart A\arrow{-U>[123][456]}B\schemestop
\qquad
\schemestart A\arrow{-U>[123]}[30]B\schemestop
\qquad
\schemestart A\arrow{-U>[]}[456][-30]B\schemestop
```

4 番目と 5 番目のオプション引数は、弧の外観を変更します。それぞれ、矢印の長さを単位とする弧の半径と、弧の中心角の半分です。

矢印-U>

```
\schemestart A\arrow{-U>[123][456][][0.25]}B\schemestop
\qquad
\schemestart A\arrow{-U>[123][456][][90]}B\schemestop
\qquad
\schemestart A\arrow{-U>[123][456][1][45]}B\schemestop
```

半径と角度に負の値を指定すると、弧は矢印の下に描画されます。

矢印-U>

```
\schemestart
  A\arrow{-U>[123][456][][-0.333][-60]}B
\schemestop
```

他の矢印と同様に、ラベルの角度とアンカーは最初の 2 つの引数でカスタマイズできます。

矢印-U>

```
\schemestart
  A\arrow{-U>[123][456]}[-90]B
\schemestop
\qquad
\schemestart
  A\arrow{-U>[*{0.180}123][*{0.180}456]}[-90]B
\schemestop
```

10 矢印のカスタマイズ

この節の内容は非常に技術的であり、TikZ の知識が必要となります。つまりこの節は、独自の矢印を定義する必要のある上級ユーザーのみを対象としています。

`\definearrow` コマンドを使うと、カスタム矢印を作成できます。その構文は次のとおりです。

`\definearrow{<数>}{<矢印の名前>}{<コード>}`

ここで、<数> は <コード> で使用されるオプション引数の数であり、コードの中では #1、#2 などの通常の構文で使われます。これらのオプション引数にデフォルト値を与えることはできません。`\arrow` マクロを使用する際に値が指定されていない場合は、引数は空のままになります。

先に進む前に、矢印を描画する際に利用できる内部マクロを確認しましょう。これらのマクロは名前に文字 “_” を含むため、`\catcode‘_ =11` コマンドと `\catcode‘_ =8` コマンドの間でのみ使用できます。

- `\CF_arrowstartname` と `\CF_arrowendname` は (TikZ がノードと見なす) 化合物の名前であり、これらの化合物の間に矢印が描かれます。
- `\CF_arrowstartnode` と `\CF_arrowendnode` は矢印の両端に位置するノードの名前です。`\arrow` コマンドの括弧内にユーザー定義のアンカーが指定されている場合は、そのアンカーはこれらの名前の後に付加されます。
- `\CF_arrowcurrentstyle` と `\CF_arrowcurrentangle` はそれぞれ、描画される矢印のスタイルと角度です。
- `\CF_arrowshiftnodes{<長さ>}` は、“`\CF_arrowstartnode`” と “`\CF_arrowendnode`” を矢印に対して垂直方向に、その引数に指定された距離だけ移動させます。
- 最も複雑な `\CF_arrowdisplaylabel{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#8}` は、以下の引数でラベルの位置を指定します。
 - #1 と #5 は表示されるラベルです。
 - #2 と #6 は 0 と 1 の間の実数で、矢印上でのラベルの位置を指定します。0 は矢印の始点を、1 は終点を示します。**矢印は直線であると仮定されています。**
 - #3 と #7 は文字 “+” または “-” です。“+” の場合はラベルを矢印の上に表示し、“-” の場合は下に表示します。
 - #4 と #8 は矢印の始点と終点に対応するノードの名前です。
- 矢印の矢尻は、両羽矢印の場合は “CF” で、片羽矢印の場合のために “harpoon” オプションが付いています。

10.1 はじめての矢印

例として、中心に点が打たれた矢印を作成したいと仮定しましょう。これを “`-.>`” と呼びます。この矢印は 4 つのオプション引数を取ります。事前定義されている矢印と同様に、最初と 2 番目の引数で矢印の上と下に配置されるラベルを指定します。3 番目の引数は矢印の方向に対する垂直方向の移動距離を定義します。最後に、4 番目の引数は点の大きさを定義します。この最後の引数がない場合、デフォルトの点の大きさは 2pt になります。

まず `\definearrow{4}{-.>}` を使用して、これから定義する矢印が 4 つのオプション引数を持ち、その名前が `-.>` であることを宣言します。最初に、3 番目の引数による移動を考慮するために、矢印が描かれるノードの位置を変更する必要があります。これは `\CF_arrowshiftnodes` マクロを使用して行うので、矢印のコードは `\CF_arrowshiftnodes{#3}%` で始めれば十分です。次に、矢印自体を描画し、その際にその中央に “`mid@point`” という名前のノードを置き、そのノードを中心とした円を描きます。全体の TikZ コードは次のとおりです。

```
\edef\pt_radius{\ifx\empty#4\empty 2pt\else #4\fi}% 点の半径
\expandafter\draw\expandafter[\CF_arrowcurrentstyle,-CF]
```

```
(\CF_arrowstartnode)--(\CF_arrowendnode)coordinate[midway](mid@point);
\filldraw(mid@point)circle(\pt_radius);%
```

最後に、ラベルが指定されている場合は、次の行によってそのラベルを表示します。

```
\CF_arrowdisplaylabel{#1}{0.5}{+}{\CF_arrowstartnode}{#2}{0.5}{-}{\CF_arrowendnode}
```

こうして完成した矢印は次のようになります。

矢印 “-.->”

```
\catcode'\_11
\definearrow4{-.->}{%
  \edef\pt_radius{\ifx\empty#4\empty 2pt\else #4\fi}% 点の半径
  \CF_arrowshiftnodes{#3}%
  \expandafter\draw\expandafter[\CF_arrowcurrentstyle,-CF](\CF_arrowstartnode)--(\CF_arrowendnode)
    coordinate[midway](mid@point);
  \filldraw(mid@point)circle(\pt_radius);%
  \CF_arrowdisplaylabel{#1}{0.5}{+}{\CF_arrowstartnode}{#2}{0.5}{-}{\CF_arrowendnode}
}
\catcode'\_8
\schemestart
A \arrow{-.->} B \arrow{-.->[うえ][した][1pt]} C \arrow{-.->[した][30]} D \arrow{-.->[うえ][5pt][1.5pt]} E
\schemestop
```

10.2 曲がった矢印

次は曲がった矢印を作ってみましょう。できるだけシンプルにするために、オプション引数を 1 つだけ取るものとし、その引数には制御点を指定する TikZ コードを書くものとします。この引数が空である場合は “-CF” タイプの矢印が描画されます。

引数 #1 が空でない場合、矢印の端の位置のノード名である “\CF_arrowstartnode” と “\CF_arrowendnode” を使いたくなるかもしれませんが、そうではありません。なぜなら、これらのノードの位置は**矢印が直線であると仮定して計算されたアンカー**をもとに決定されているからです。その代わりに、描画されるべき矢印の始点・終点に位置する化合物名である \CF_arrowstartname と \CF_arrowendname（これらは TikZ ノードです）を使用します。2 つの化合物の間に曲がった矢印を描くための TikZ コードは次のようになります。

```
\draw[shorten <=\CF_arrowoffset,shorten >=\CF_arrowoffset,\CF_arrowcurrentstyle,-CF,
(\CF_arrowstartname).. controls #1 ..(\CF_arrowendname)];%
```

\setarrowoffset で定義された \CF_arrowoffset の分だけ矢印を短くするための TikZ コードを追加しなければなりません。これは、矢印が出発・到着するノードが、直線の矢印のためのノード (\CF_arrowstartnode と \CF_arrowendnode) ではないからです。したがって \CF_arrowcurrentstyle の前に以下のコードを追加する必要があります。

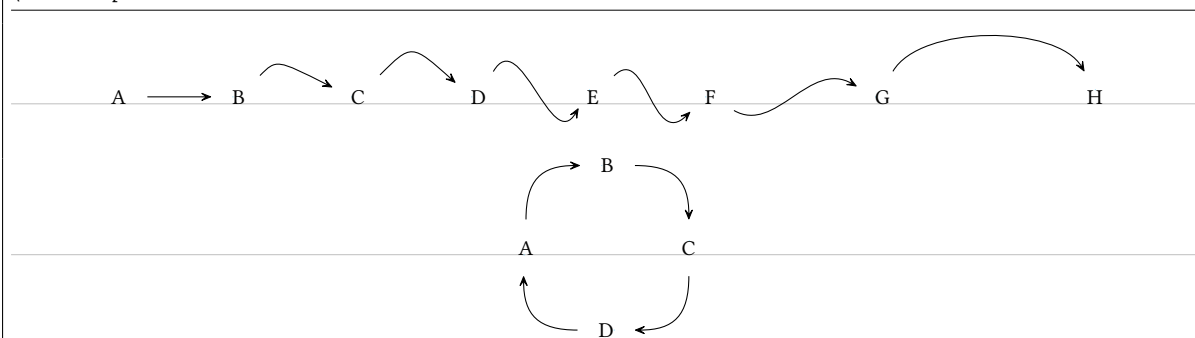
```
shorten <=\CF_arrowoffset, shorten >=\CF_arrowoffset
```

これが \else の後の 2 行の役割です。

最終的な曲がった矢印の定義は以下のようになります。


```
\catcode'\_11
\definearrow1{s>}{%
\ifx\empty#1\empty
\expandafter\draw\expandafter[\CF_arrowcurrentstyle,-CF](\CF_arrowstartnode)--(\CF_arrowendnode);%
\else
\def\curvedarrow_style{shorten <=\CF_arrowoffset,shorten >=\CF_arrowoffset,%
\CF_eaddtomacro\curvedarrow_style\CF_arrowcurrentstyle
\expandafter\draw\expandafter[\curvedarrow_style,-CF](\CF_arrowstartname)..controls#1..(\CF_arrowendname);
\fi
}
\catcode'\_8
\schemestart
A\arrow{s>}
B\arrow{s>[+(0.5cm,0.5cm)]}
C\arrow{s>[+(45:1cm)]}
D\arrow(.60--.120){s>[+(60:1cm) and +(-120:1cm)]}
E\arrow{s>[+(45:1) and +(-135:1)]}
F\arrow{s>[+(-30:1) and +(150:1)]}[1.5]
G\arrow(.90--.90){s>[+(60:1)and+(120:1)]}[2]
H
\schemestop

\schemestart
A\arrow(.90--.180){s>[+(90:0.8) and +(180:0.8)]}[45]B
\arrow(.0--.90){s>[+(0:0.8) and +(90:0.8)]}[-45]C
\arrow(.90--.0){s>[+(-90:0.8) and +(0:0.8)]}[-135]D
\arrow(.180--.90){s>[+(180:0.8) and +(-90:0.8)]}[135]
\schemestop
```



11 \merge コマンド

\merge コマンドを使うことで、複数の既存の化合物から出ている矢印が1つの矢印に合流し、1つの化合物に到達するような図を描くことができます。

\merge コマンドの直後には、続く方向を指定する必要があります。これには、“>”（文字が入力されていない場合のデフォルトの方向）、“<”、“^”または“v”のいずれかの文字を使用します。

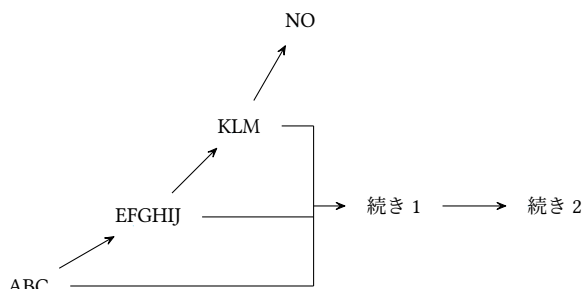
構文は次の通りです。

```
\merge{方向}(n1.a1)(n2.a2)(...)(ni.ai)--(n.a[s])
```

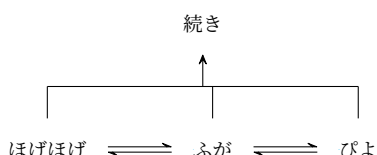
ここで二重ハイフン--の前にある“ni”は、合流させたい矢印が出発している、すでに定義されている化合物の名前です。また、デフォルトのアンカーでは不便な場合は、アンカー“ai”を指定することもできます。 \arrow コマンドと同様に、“n.a[s]”コマンドで到着化合物の名前、アンカー、およびスタイルを指定します。

\merge コマンド

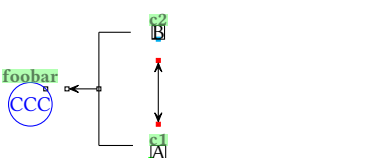
```
\schemestart
ABC\arrow[30]EFGHIJ\arrow[45]KLM\arrow[60]NO
\merge>(c1)(c2)(c3)--()続き1
\arrow 続き2
\schemestop
\bigskip
```



```
\schemestart
ほげほげ\arrow(foo--bar){<=>}ふが\arrow(--baz){<=>}びよ
\merge^(foo)(bar)(baz)--()続き
\schemestop
\bigskip
```



```
\setchemfig{scheme debug=true}
\schemestart
A\arrow{<->}[90]B
\merge<(c1.120)(c2)--(foobar.45[circle,blue])CCC
\schemestop
```



\merge の矢印の配置に関して、 n 個の化合物から出ている n 個の線分が、垂直な接続線まで続きます。これらの線分の最小長のデフォルトは、\setcompoundsep で定義された化合物間の距離の半分に相当します。接続線から生成物への矢印は、同じデフォルトの長さを持ち、接続線の中央から出発します。これらの 3 つの幾何学的特徴は、到着化合物の直後のオプション引数でカスタマイズできます。

\merge{方向}(n1.a1)(n2.a2)(...)(ni.ai)--(n.a[s])[c1,c2,c,スタイル]

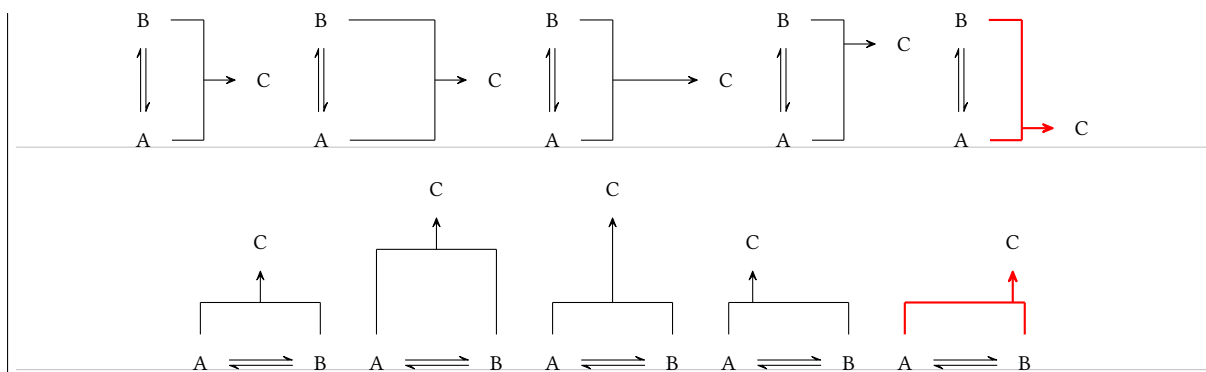
ここで、

- 反応物と接続線の間の距離の最小値は、距離 \setcompoundsep に係数 $c1$ を掛けた値であり、 $c1$ のデフォルト値は 0.5 です。
- 接続線と生成物との間の矢印の長さは、距離 \setcompoundsep に係数 $c2$ を掛けた値であり、 $c2$ のデフォルト値は 0.5 です。
- 接続線と生成物との間の矢印の起点は c によって決定され、0 の場合は接続線の左端（方向が v または \wedge の場合は上端）から出発することを意味します。
- \merge の矢印のスタイルは、最後のスタイル引数で定義されます。

\merge の配置パラメーター

```
\schemestart A\arrow{<=>}[90]B\merge(c1)(c2)--()C\schemestop\quad
\schemestart A\arrow{<=>}[90]B\merge(c1)(c2)--()[1]C\schemestop\quad
\schemestart A\arrow{<=>}[90]B\merge(c1)(c2)--()[,1]C\schemestop\quad
\schemestart A\arrow{<=>}[90]B\merge(c1)(c2)--()[,0.2]C\schemestop\quad
\schemestart A\arrow{<=>}[90]B\merge(c1)(c2)--()[,0.9,red,thick]C\schemestop
\bigskip
```

```
\schemestart A\arrow{<=>}B\merge^(c1)(c2)--()C\schemestop\quad
\schemestart A\arrow{<=>}B\merge^(c1)(c2)--()[1]C\schemestop\quad
\schemestart A\arrow{<=>}B\merge^(c1)(c2)--()[,1]C\schemestop\quad
\schemestart A\arrow{<=>}B\merge^(c1)(c2)--()[,0.2]C\schemestop\quad
\schemestart A\arrow{<=>}B\merge^(c1)(c2)--()[,0.9,red,thick]C\schemestop
```



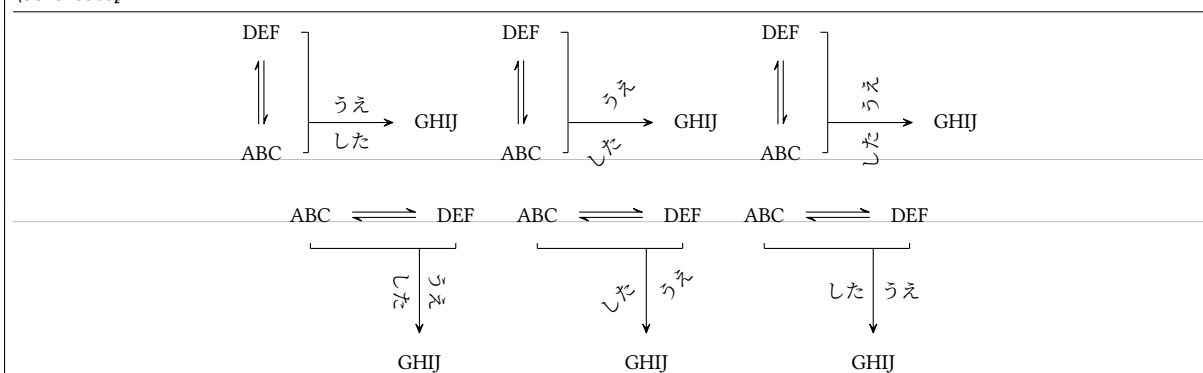
最後に、合流した矢印の上下にラベルを書くことが可能です。方向を示す文字は角括弧で書かれるオプション引数を 2 つ取り、それぞれ矢印の上および下に書かれるラベルです。したがって `\merge` コマンドの完全な構文は次のようになります。

`\merge{方向}[上ラベル][下ラベル](n1.a1)(n2.a2)...(ni.ai)--(n.a[s])[c1,c2,c,スタイル]`

すでに説明した矢印のラベルに関する全ての機能がここでも使用可能であり、ラベルの直前に構文 `*{角度.アンカー}` を書くことで、ラベルの回転角度とアンカーを指定できます。

\merge コマンドのラベル

```
\schemestart
ABC\arrow{<=>}[90]DEF\merge>[うえ][した](c1)(c2)--()[0.25,1,0.75]GHIJ
\schemestop\qqquad
\schemestart
ABC\arrow{<=>}[90]DEF\merge>[*{45.south west}うえ][*{45.north east}した](c1)(c2)--()[0.25,1,0.75]GHIJ
\schemestop\qqquad
\schemestart
ABC\arrow{<=>}[90]DEF\merge>[*{90}うえ][*{90}した](c1)(c2)--()[0.25,1,0.75]GHIJ
\schemestop
\bigskip
\schemestart
ABC\arrow{<=>}DEF\merge v[うえ][した](c1)(c2)--()[0.25,1,0.75]GHIJ
\schemestop\qqquad
\schemestart
ABC\arrow{<=>}DEF\merge v[*{45.north west}うえ][*{45.south east}した](c1)(c2)--()[0.25,1,0.75]GHIJ
\schemestop\qqquad
\schemestart
ABC\arrow{<=>}DEF\merge v[*{0}うえ][*{0}した](c1)(c2)--()[0.25,1,0.75]GHIJ
\schemestop
```



12 プラス記号

`\schemestart` コマンドと `\schemestop` コマンドの間では、“`\+`” マクロを使用して `+` 記号を書くことができます

す。このマクロは、{〈長さ1〉,〈長さ2〉,〈長さ3〉} の構文でオプション引数を取ります。ここで、

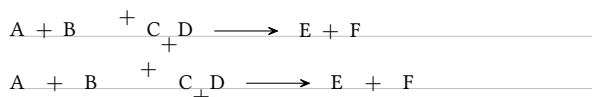
- ・ 〈長さ1〉 と 〈長さ2〉 は、+ 記号の前と後に挿入される長さです。
- ・ 〈長さ3〉 は、+ 記号の垂直オフセットです。

また、〈キーワード〉 + `sep left = 〈長さ〉`、+ `sep right = 〈長さ〉` および + `vshift = 〈長さ〉` を使用することで、それ以降の全てのプラス記号に対してこれらの長さを指定することができます。これらのデフォルト値は、最初の 2 つが 0.5em、3 つ目が 0pt です。

\+ コマンド

```
\schemestart
A\+B\+{2em,5pt}C\+{0pt,0pt,-5pt}D\arrow E\+F
\schemestop

\setchemfig{+ sep left=1em,+ sep right=1em,+ vshift=0pt}
\schemestart
A\+B\+{2em,5pt}C\+{0pt,0pt,-5pt}D\arrow E\+F
\schemestop
```



次の例に示すように \+ コマンドによって挿入されたプラス記号は、化合物の一部であることに注意してください。

化合物と \+

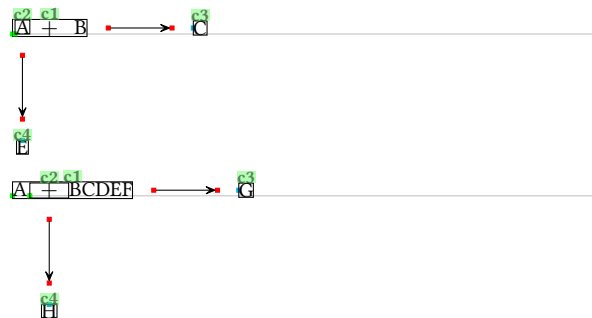
```
\setchemfig{scheme debug=true}
\schemestart A\+ B\+{,5pt}C\arrow D\+ E\schemestop
```



これにより“A”の真下に垂直の矢印を正確に描くことが難しくなります。なぜなら `chemfig` は、この文字を単一の化合物と認識しないからです。この問題は `\subscheme` コマンドを使用して文字“A”を単一の化合物として定義し、後で独自の名前で参照できるようにすることで解決できます。（この方法はプラス記号自体にも適用できます。）

部分化合物と \+

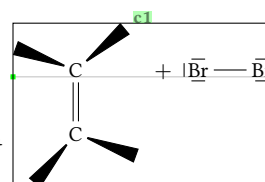
```
\setchemfig{scheme debug=true}
\schemestart
\subscheme{A}\+ B\arrow C
\arrow{@c2--}[ -90]E
\schemestop
\medskip
\schemestart
A\subscheme{\+}BCDEF \arrow G
\arrow{@c2--}[ -90]H
\schemestop
```



よくある問題は、次の例のように“+”記号がその前あるいは後の分子と整列しないことです。

+ 記号の配置

```
\setchemfig{scheme debug=true}
\schemestart
\chemfig{C(<[:40])(<[:160])=[6]C(<[: -130])<[: -20]}
\+
\chemfig{\charge{90=\|,180=\|,270=\|}{Br}-\charge{0=\|,90=\|,-90=\|}{Br}}
\schemestop
```

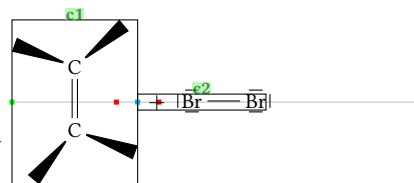


ここでは、“+”記号は直前の化合物のベースライン、すなわち上部の炭素原子と同じベースライン上にあります。“+”記号を移動させることはできますが、それは“|Br — Br|”の垂直位置を変えることにはなりません。上の例で全てが化合物“c1”に含まれていることから分かるように、`chemfig` は“+”記号で化合物の読み取り

を停止しません。したがって最初の分子の直後で化合物を終了させる必要があり、そのために `\arrow{0}[,0]` を使用して長さ 0 の見えない矢印を配置します。反応式全体を垂直方向の中央配置にするには、`\schemestart` コマンドの 2 番目のオプション引数を使って、最初の化合物のアンカーを“west”（またはこれと等価な“180”）に設定する必要があります。

+ 記号の配置

```
\setchemfig{scheme debug=true}
\schemestart[][west]
\chemfig{C(<[:40])(<[:160])=[6]C(<[:130])<[:20]}
\arrow{0}[,0]\+
\chemfig{\charge{90=\|,180=\|,270=\|}{Br}-\charge{0=\|,90=\|,-90=\|}{Br}}
\schemestop
```

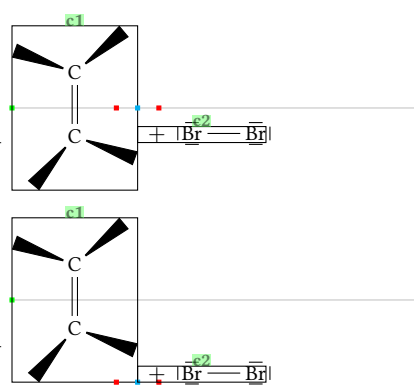


このようにすると、最初の化合物“c1”は最初の分子でのみ構成され、2 番目の化合物はそれ以外の全て、すなわち“+”記号と 2 番目の分子で構成されることになります。あるいは、`\arrow` コマンドでアンカーやスタイルを設定して、2 番目の化合物の位置を移動することもできます。次の例の上のケースでは、2 番目の化合物を 10pt 下に移動しています。下のケースでは、最初の化合物の“south east”アンカーを 2 つ目の化合物の“south west”アンカーと一致させています。

+ 記号の配置

```
\setchemfig{scheme debug=true}
\schemestart[][west]
\chemfig{C(<[:40])(<[:160])=[6]C(<[:130])<[:20]}
\arrow(--[yshift=-10pt]){0}[,0]\+
\chemfig{\charge{90=\|,180=\|,270=\|}{Br}-\charge{0=\|,90=\|,-90=\|}{Br}}
\schemestop
\medskip

\schemestart[][west]
\chemfig{C(<[:40])(<[:160])=[6]C(<[:130])<[:20]}
\arrow(.south east--.south west){0}[,0]\+
\chemfig{\charge{90=\|,180=\|,270=\|}{Br}-\charge{0=\|,90=\|,-90=\|}{Br}}
\schemestop
```



水平方向の反応

水平方向の反応をより簡潔な構文で描くために、“hreact”環境を使うことができます。

```
\hreact[⟨キーワード⟩=⟨値⟩のリスト]
    ⟨反応⟩
\endhreact
```

前述のとおり、以降の文書全体にオプションを適用したい場合は `\setchemfig{⟨キーワード⟩=⟨値⟩のリスト}` で設定できます。

単一の反応にだけ設定を適用するには、その環境のオプション引数を使用します。

以下に、`\hreact` マクロの全てのパラメーター、そのデフォルト値、および簡単な説明を示します。

キーワード	デフォルト値	説明
<code>harrow minwidth</code>	3em	矢印の最小の長さ
<code>label xsep</code>	3pt	矢印がラベルに合わせて延長される場合にラベルの前後に挿入される空白の長さ。したがって矢印の長さは最も長いラベルの長さとの寸法の2倍を足したものに等しくなります。
<code>label align</code>	c	矢印のラベルの文字揃えの位置 (<code>⟨c⟩</code> 、 <code>⟨r⟩</code> または <code>⟨l⟩</code>)
<code>name sep</code>	1.5ex	化合物のボックスとその名前との間の最小の垂直距離
<code>hreact anchor</code>	text	最初の化合物のアンカー
<code>hreact sep</code>	0.5em	化合物のボックス間の水平距離
<code>hreact debug</code>	false	この値が <code>⟨true⟩</code> のとき、化合物のボックス、アンカー、ノード名および水平線を表示
<code>arrow label sep</code>	3pt	これ以降のキーワードについては、 <code>\schemestart</code> のオプションを参照 (58 ページ)
<code>arrow style</code>	⟨空⟩	
<code>arrow double sep</code>	2pt	
<code>arrow double coeff</code>	0.6	
<code>arrow double harpoon</code>	true	
<code>arrow label sep</code>	3pt	
<code>arrow head</code>	-CF	

1 文法

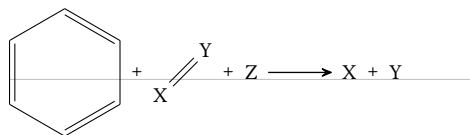
〈反応〉の構文において、トークン \oplus , \boxplus , \boxtimes , \boxdot は特別な意味を持ちます。

〈反応〉では、化合物、+ 記号、および矢印を次の構文と規則に従って記述できます。

- 次に出現する “+”、“>” あるいは環境の終わりまでが 〈化合物〉として扱われます。
- 〈反応〉では、波括弧で囲まれた 〈化合物〉に属するコードは解析されず、そのまま波括弧内に残されます。
- 空白は無視されます。(波括弧内にある場合を除きます。前項参照)
- 〈化合物〉は、上記の予約トークンを除く任意のコードから構成されます。ただし波括弧内にある予約トークンは除外されません。
- “+” は + を描画し、それ自身が 〈化合物〉です。
- “>” は矢印を描画し、それ自身が 〈化合物〉です。

例 1

```
\hreact
\chemfig{*6(---=---)}
+
\chemfig{X=[1]Y}
+
Z > X + Y
\endhreact
```

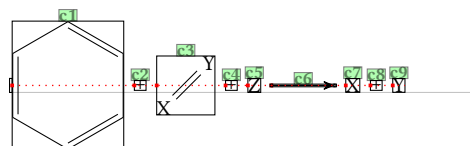


2 化合物の配置

すべての化合物には $c\langle n \rangle$ ($\langle n \rangle$ は整数) と番号が付けられ、ボックス内に配置されます。すべてのボックスの中心は同一水平線上に配置され、`hreact debug` オプションが有効なときはその線が赤い点線で示されます。

化合物の垂直方向の配置

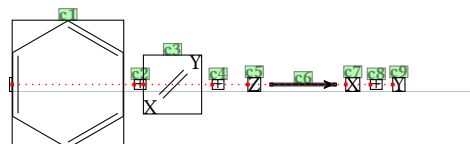
```
\hreact[hreact debug]
\chemfig{*6(---=---)}
+
\chemfig{X=[1]Y}
+
Z > X + Y
\endhreact
```



$\backslash>\langle \text{寸法} \rangle$ が 〈化合物〉 の先頭に書かれた場合、その 〈化合物〉 は $\langle \text{寸法} \rangle$ だけ水平に移動します。

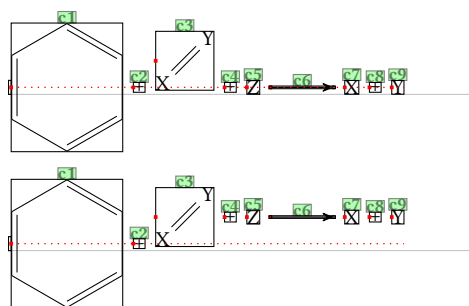
化合物の水平方向の配置の調整

```
\hreact[hreact debug]
\chemfig{*6(---=---)}
+
\>{-5pt}\chemfig{X=[1]Y}
+
\>{5pt}Z > X + Y
\endhreact
```



$\backslash\wedge\langle \text{寸法} \rangle$ が 〈化合物〉 の先頭に書かれた場合、その化合物 (のみ) が垂直方向に $\langle \text{寸法} \rangle$ だけ移動します。* が付いた $\backslash\wedge*\langle \text{寸法} \rangle$ の場合は、それ以降のすべての化合物に適用されます。

```
\hreact[hreact debug]
\chemfig{*6(====)}
+
\^{\10pt}\chemfig{X=[1]Y}
+
Z > X + Y
\endhreact\medbreak
\hreact[hreact debug]
\chemfig{*6(====)}
+
\^*{\10pt}\chemfig{X=[1]Y}
+
Z > X + Y
\endhreact
```



3 化合物名

〈化合物〉の〈名前〉を表示するには、以下のいずれかを使用します。

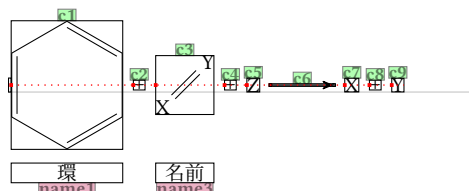
- `\chemname[〈寸法〉]{〈化合物〉}{〈名前〉}`
- `〈化合物〉\name[〈寸法〉]{〈名前〉}` あるいは `\name[〈寸法〉]{〈名前〉}〈化合物〉` (順序に依存しません)

オプションの〈寸法〉は〈化合物〉ボックスと〈名前〉ボックスの最小距離を指定します。省略された場合のデフォルト値はキーワード `name sep` の値です。

名前はすべて `name<n>` (〈n〉はその〈化合物〉に対応する整数) によって番号付けられます。これらは〈化合物〉と同じ幅を持つボックスに配置されます。これらのボックスの上辺は同一水平線上に配置され、その垂直方向の位置は最大の深さをもつ化合物よりも下になるように調節されます。

化合物の名前

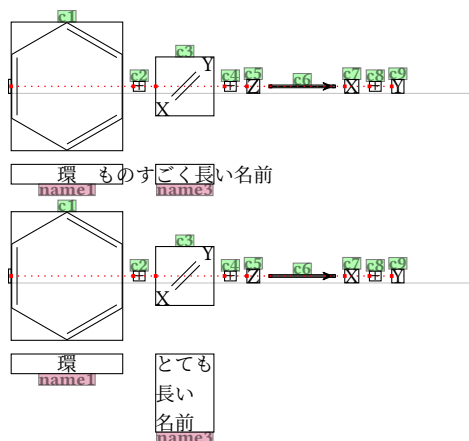
```
\hreact[hreact debug]
\chemfig{*6(====)}\name{環}
+
\chemfig{X=[1]Y}\name{名前}
+
Z > X + Y
\endhreact
```



〈名前〉の幅が〈化合物〉の幅を超過した場合、〈化合物〉の幅は調節されずに〈名前〉がそのボックスの境界をはみだして配置されます。〈名前〉を複数行にわたって表示させたい場合は、ユーザー自身で `\\` マクロを使用する必要があります。

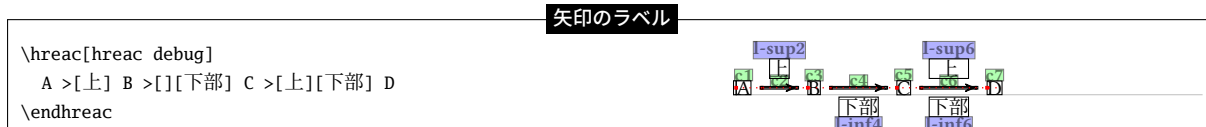
化合物の名前

```
\hreact[hreact debug]
\chemfig{*6(====)}\name{環}
+
\chemfig{X=[1]Y}\name{ものすごく長い名前}
+
Z > X + Y
\endhreact\medbreak
\hreact[hreact debug]
\chemfig{*6(====)}\name{環} +
\chemfig{X=[1]Y}\name{とても\\長い\\名前} +
Z > X + Y
\endhreact
```

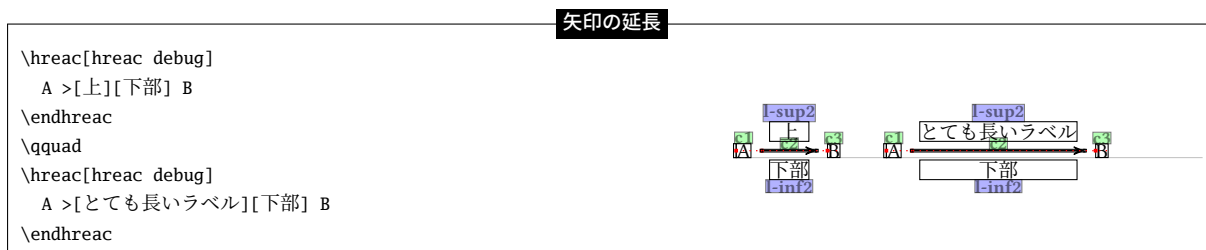


4 矢印のラベル

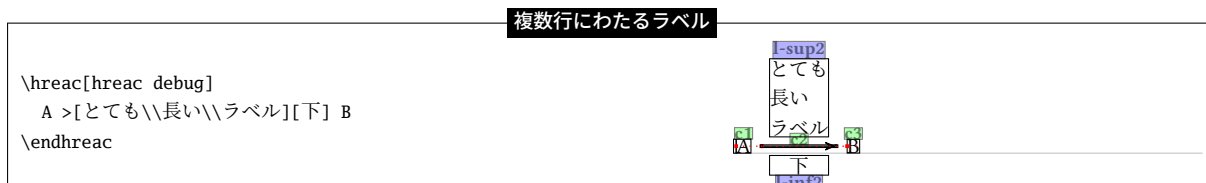
トークン “>” の直後にオプション引数 [\langle ラベル \rangle] が与えられると、この \langle ラベル \rangle は矢印の上に配置されます。さらにもうひとつのオプション引数 [\langle ラベル \rangle] が与えられた場合、2 つめの \langle ラベル \rangle は矢印の下に配置されます。



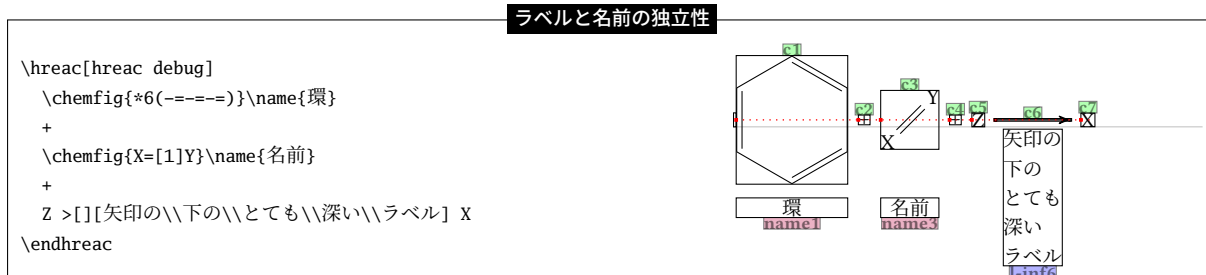
矢印のデフォルトの幅は 3em ですが、ラベルがこの寸法を越えた場合は、矢印が延長されます。



化合物名の場合と同様に、\\ を使用して \langle ラベル \rangle を複数行にすることができます。



矢印の下に配置されるラベルを含むボックスは、化合物名の垂直方向の位置には影響を与えません。



5 矢印の種類

トークン > の直後に波括弧で囲まれたテキストが与えられると、この非必須のテキストは矢印の種類を指定します。このデフォルト値はキーワード **arrow head** の値であり、これはデフォルト値として \langle -CF \rangle を持ちます。したがって、トークン > の完全な構文は次のとおりです。

\langle {(矢印の種類)} \rangle [\langle 上ラベル \rangle][\langle 下ラベル \rangle]

このオプションのパラメーターを指定することで、すでに「反応式」(58 ページ) の節で説明した他の種類の矢印、つまり “ \rightarrow ”、“ \leftarrow ”、“ \leftrightarrow ”、“ \rightleftharpoons ”、“ \longleftrightarrow ” および “ \longleftrightarrow ” を使用することができます。

6 種類の矢印

`\hreact A > B \endhreact\par%` `>{->}` と同一

`\hreact A >{<-} B \endhreact\par`

`\hreact A >{<->} B \endhreact\par`

`\hreact A >{<=>} B \endhreact\par`

`\hreact A >{<<->} B \endhreact\par`

`\hreact A >{<->>} B \endhreact`

$A \longrightarrow B$

$A \longleftarrow B$

$A \longleftrightarrow B$

$A \rightleftharpoons B$

$A \Leftrightarrow B$

$A \Longleftrightarrow B$

コマンドリスト

chemfig が提供するコマンドは以下の通りです。

コマンド	説明
<code>\chemfig[⟨設定⟩]{⟨コード⟩}</code>	⟨コード⟩ で記述された分子を描画します
<code>\setchemfig{⟨設定⟩}</code>	⟨キーワード⟩ = ⟨値⟩ の構文でパラメーターを設定します。\\chemfig の ⟨キーワード⟩ および ⟨値⟩ のリストは 7 ページに、\\schemestart マクロのものは 58 ページに、\\hreact マクロのものは 77 ページにあります。
<code>\resetchemfig</code>	パラメーターをデフォルト値にリセットします
<code>\printatom</code>	分子内に原子を描くためのマクロ。表示をカスタマイズするために再定義できます。 30 ページ参照
<code>\hflipnext</code>	次の分子を水平方向に反転して描画します
<code>\vflipnext</code>	次の分子を垂直方向に反転して描画します
<code>\definesubmol{⟨名前⟩}⟨数字⟩[⟨コード1⟩]{⟨コード2⟩}</code>	描画する分子のコード内で使用可能な別名!⟨名前⟩ を作成します。最後の結合の角度に応じて ⟨コード1⟩ または ⟨コード2⟩ に置き換えられます。 33 ページ参照
<code>\chemskipalign</code>	垂直方向の整列において現在の原子グループを無視させます。 38 ページ参照
<code>\redefinesubmol{⟨名前⟩}⟨数字⟩[⟨コード1⟩]{⟨コード2⟩}</code>	既存の別名 !⟨名前⟩ を新しい ⟨コード⟩ で再定義します。 33 ページ参照
<code>\charge{[⟨パラメーター⟩][⟨場所⟩][⟨tikz⟩]}{⟨原子⟩}</code>	⟨原子⟩ を描画し⟨場所⟩ に従って電荷を配置します。電荷は ⟨原子⟩ のバウンディングボックスの外に置かれます。 38 ページ参照
<code>\Charge{[⟨パラメーター⟩][⟨場所⟩][⟨tikz⟩]}{⟨原子⟩}</code>	\\charge と同様ですが、電荷はバウンディングボックスに含まれます
<code>\chemmove[⟨tikzオプション⟩][⟨tikzコード⟩]</code>	tikzpicture 環境を作成し、⟨tikzオプション⟩ を渡します。⟨tikzコード⟩ に従って、文字 “@” によって定義された分子内のノード間を繋ぎます。 25 ページ参照
<code>\chemabove[⟨長さ⟩]{⟨テキスト1⟩}{⟨テキスト2⟩}</code>	⟨テキスト1⟩ を描画し、その上方に ⟨テキスト2⟩ を ⟨長さ⟩ の垂直方向の距離をおいて配置します。このコマンドは⟨テキスト1⟩ のバウンディングボックスを変更しません。 41 ページ参照
<code>\chembelow[⟨長さ⟩]{⟨テキスト1⟩}{⟨テキスト2⟩}</code>	⟨テキスト1⟩ を描画し、その下方に ⟨テキスト2⟩ を ⟨長さ⟩ の垂直方向の距離をおいて配置します。このコマンドは⟨テキスト1⟩ のバウンディングボックスを変更しません。 41 ページ参照
<code>\Chemabove[⟨長さ⟩]{⟨テキスト1⟩}{⟨テキスト2⟩}</code>	⟨テキスト1⟩ を描画し、その上方に ⟨テキスト2⟩ を ⟨長さ⟩ の垂直方向の距離をおいて配置します。 41 ページ参照
<code>\Chembelow[⟨長さ⟩]{⟨テキスト1⟩}{⟨テキスト2⟩}</code>	⟨テキスト1⟩ を描画し、その下方に ⟨テキスト2⟩ を ⟨長さ⟩ の垂直方向の距離をおいて配置します。 41 ページ参照
<code>\chemname[⟨長さ⟩]{⟨分子⟩}{⟨名前⟩}</code>	⟨名前⟩ を⟨分子⟩ の下に書きます。 28 ページ参照
<code>\chemnameinit</code>	名前の整列を正しく行うために使用する、分子の最大深さを初期化します。 29 ページ参照
<code>\schemestart...\\schemestop</code>	反応式の開始および終了。 57 ページ参照
<code>\arrow</code>	反応式内に矢印を描きます。（このコマンドは反応式内でのみ定義されています。） 58 ページ参照
<code>\+</code>	反応式内に + 記号を表示します。（このコマンドは反応式内でのみ定義されています。） 74 ページ参照

<code>\subscheme{<コード>}</code>	部分反応式を描きます。(このコマンドは反応式内でのみ定義されています。 64 ページ参照)
<code>\definearrow</code>	矢印を定義します。 69 ページ参照
<code>\chemleft<car1><要素>\chemright<car2></code>	伸縮可能な区切り記号<car1> および <car2> をそれぞれ<要素>の左および右に配置します。 65 ページ参照
<code>\chemup<car1><要素>\chemdown<car2></code>	伸縮可能な区切り記号<car1> および <car2> をそれぞれ<要素>の上および下に配置します。 65 ページ参照
<code>\polymerdelim[<設定>]{<ノード1>}{<ノード2>}</code>	指定されたノードに区切り記号を描きます。 52 ページ参照
<code>\hreact... \endhreact</code>	水平方向の反応を描くための開始および終了タグ。 77 ページ参照
<code>>{<type>}[<top label>][<bottom label>]</code>	水平な矢印を描きます。 78 ページ参照
<code>+</code>	水平方向の反応における + 記号を描きます
<code>\>{<dimension>}</code>	現在の化合物を水平方向に移動します。 80 ページ参照
<code>\^>{<寸法>} および \^*>{<寸法>}</code>	現在の化合物を垂直方向に移動します。 78 ページ参照
<code>\chemname{<化合物>}{<名前>}</code>	<名前> を<化合物>の下に書きます。 79 ページ参照
<code>\name{<名前>}</code>	<名前> を現在の化合物の下に書きます。 79 ページ参照

ギャラリー

このマニュアルを締めくくるのは、さまざまな複雑さの分子の図です。

好奇心旺盛なユーザーは各分子の〈コード〉に興味を持つかもしれませんが、分子が複雑になるほど魅力が薄れていきます。実際、ある程度の複雑さを超えると、〈コード〉を書くのは比較的簡単ですが、後になって〈コード〉読んで分析するのはかなり難しくなります。私たちは、複雑な図のコードの即時可読性の限界にすぐ達しました。

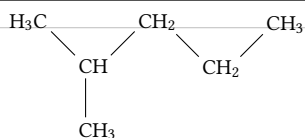
ともかく、このパッケージが分子を描きたい全ての \LaTeX ユーザーに役立つことを願っています。`chemfig` は徹底的にテストされており、バージョン番号は現在 1.0 を超えていますが、遭遇するバグに対して寛容でいていただけると幸いです。何か不具合や改善の提案があれば、[メール](#)を送って知らせてください。

Christian TELLECHEA

★
★ ★

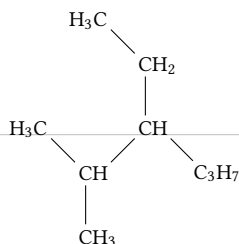
2-メチルペンタン

```
\chemfig{[7]H_3C-CH(-[6]CH_3)-[1]CH_2-CH_2-[1]CH_3}
```



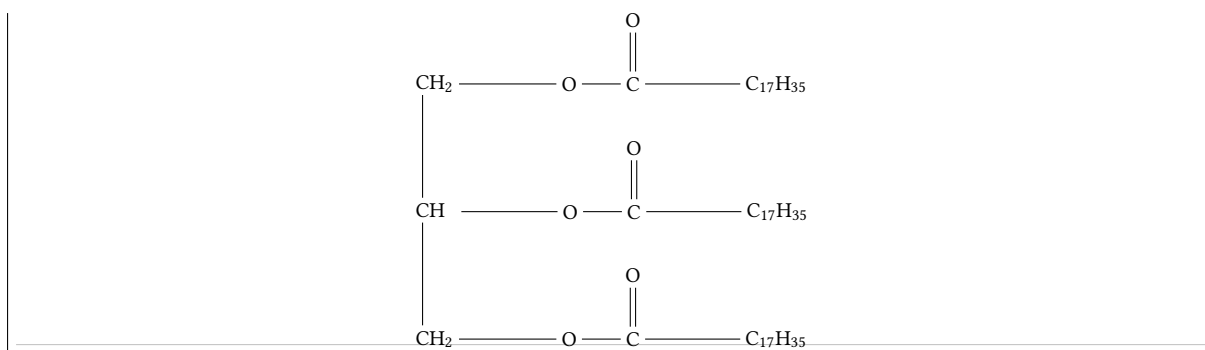
3-エチル-2-メチルヘキサン

```
\chemfig{H_3C-[7]CH(-[6]CH_3)-[1]CH(-[7]C_3H_7)-[2]CH_2-[3]H_3C}
```



ステアリン (示性式)

```
\definesubmol{@}{([0,2]-O-[0,1]C(=[2,1]O)-C_{17}H_{35})}  
\chemfig{[2,2]CH_2!@-CH_{\phantom 2}!@-CH_2!@}
```

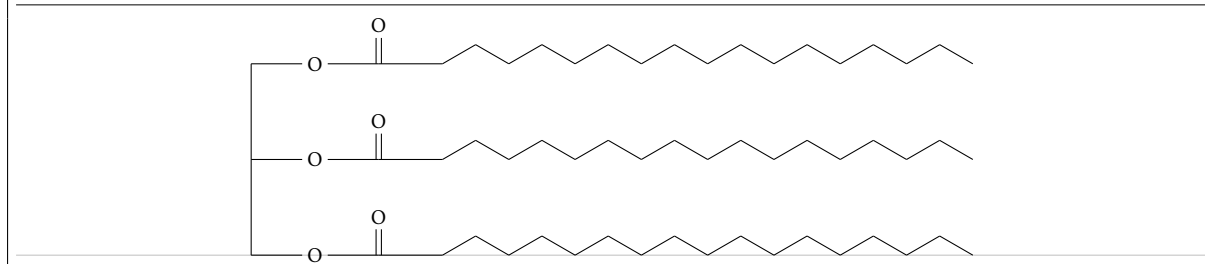


ステアリン (構造式)

```

\definesubmol{x}{-[:+30,.6]-[: -30,.6]}
\definesubmol{y}{-O-(=[2,.6]O)-!x!x!x!x!x!x!x}
\chemfig{[2]([0]!y)-[,1.5]([0]!y)-[,1.5]([0]!y)}

```

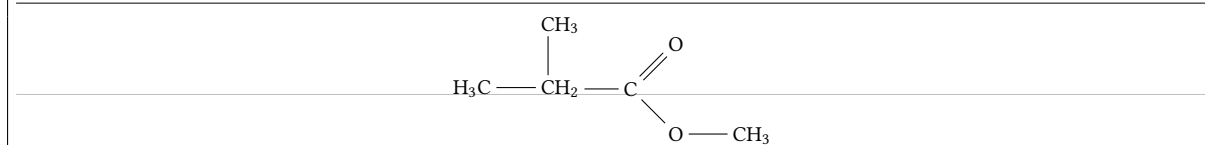


2-メチルプロパン酸メチル

```

\chemfig{H_3C-CH_2(-[2]CH_3)-C(=[1]O)-[7]O-CH_3}

```

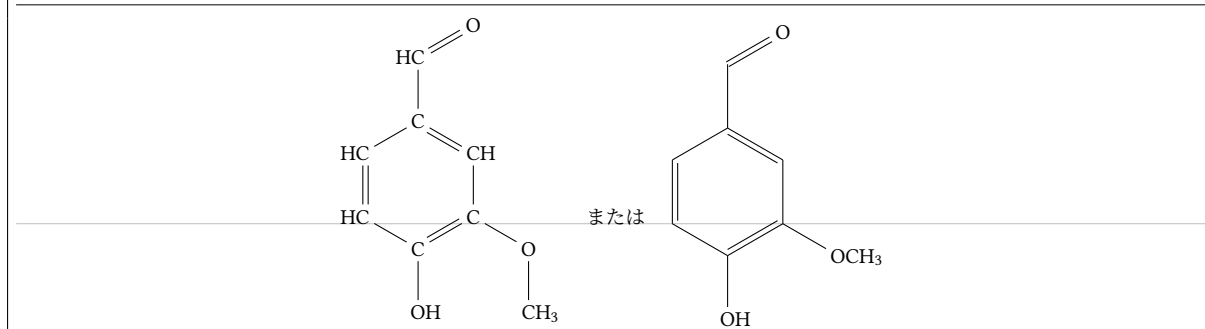


バニリン

```

\chemfig{HC*6(-C(-OH)=C(-O-[: -60]CH_3)-CH=C(-[, ,2]HC=[: -60]O)-HC=[: ,2])} \quad \text{または} \quad \quad
\chemfig{*6(-(-OH)=(-OCH_3)-(-[: -60]O)-=)}

```

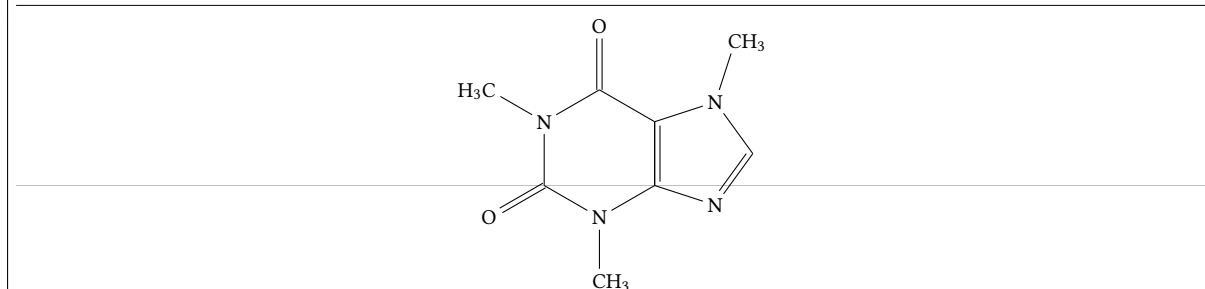


カフェイン

```

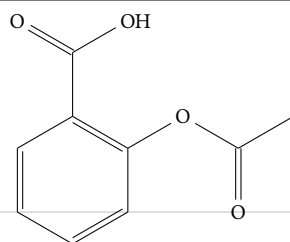
\chemfig{*6((=O)-N(-CH_3)-*5(-N=-N(-CH_3)-=)--(=O)-N(-H_3C)-)}

```



アスピリン (アセチルサリチル酸)

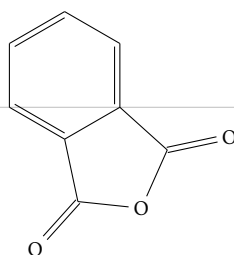
```
\chemfig{*6(=-(=O-[:60])(=[:60]O)-[:+60])=-(=[:+60]O)-[:60]OH)-=O)}
```



アスピリンは多くの国でバイエルの登録商標です。

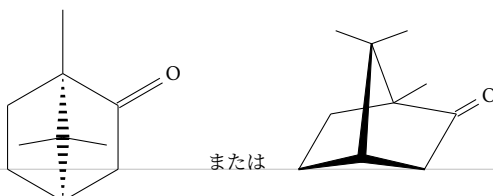
無水フタル酸

```
\chemfig{*6(=*5(-(=O)-O-(=O)-)-===-)}
```



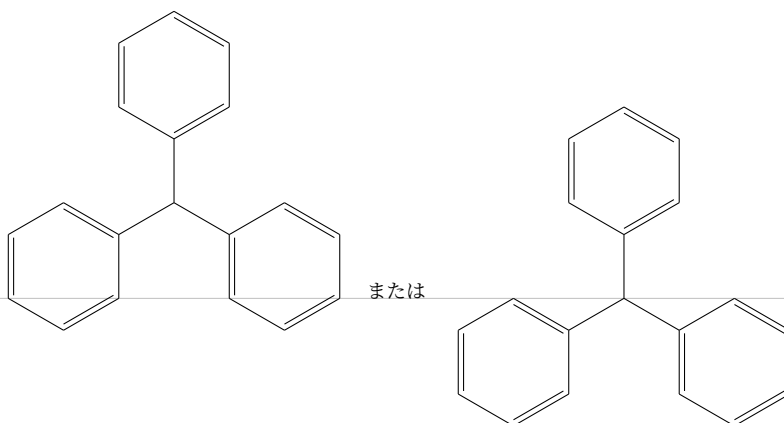
樟脳

```
\chemfig{*6(<[:120](-[:100,0.7])(-[:100,0.7]))--(=O)-(-[:120])--)}  
\quad または \quad  
\setchemfig{cram width=3pt}  
\chemfig{<[:10](>[:85,1.8]?(-[:160,0.6])-[20,0.6])  
>[:10]-[:60](=[:30,0.6]O)-[:170]?(-[:30,0.6])-[190]-[:240]}
```



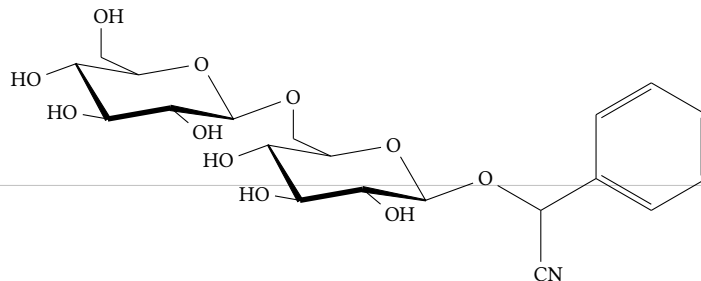
トリフェニルメタン

```
\chemfig{*6(==*6(-(*6(==--)))-*6(==--))==)}  
\quad または \quad  
\definesubmol{@}{*6(==--)}  
\chemfig{(-[:30]!@)(-[:90]!@)(-[:210]!@)}
```



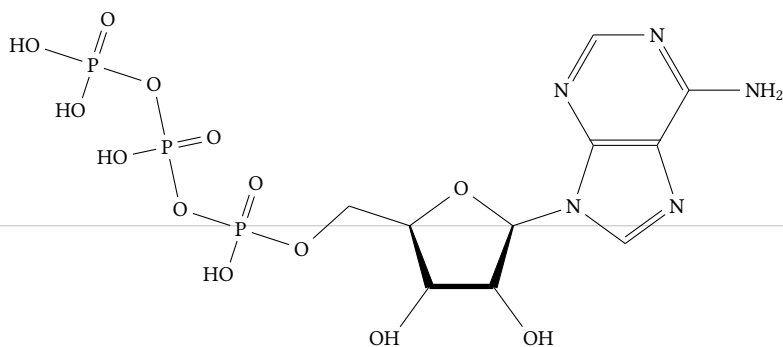
アミグダリン

```
\setchemfig{cram width=2pt}
\definesubmol{c1}{-[:200]-[:120]O-[:190]}
\definesubmol{c2}{-[:170](-[:200,0.7]HO)<[:300](-[:170,0.6]HO)
-[:10,,,line width=2pt](-[:40,0.6]OH)>[:10]}
\definesubmol{csub}{-[:155,0.65]-[:90,0.65]}
\chemfig{O(!{c1}(!{csub}O(!{c1}(!{csub}OH)!{c2}))!{c2})-[:30](-[:90]CN)-[:30]*6(==--==)}
```



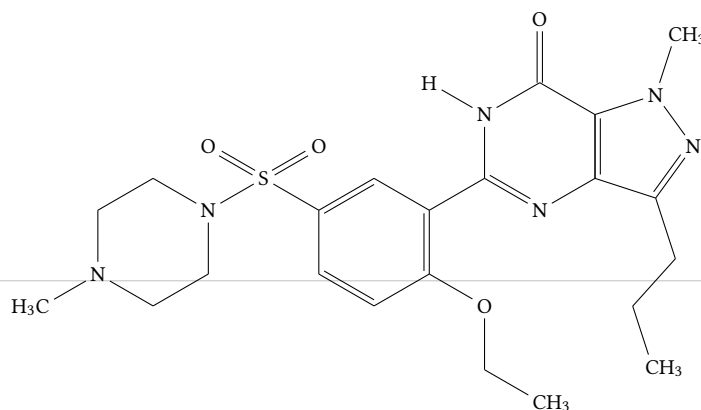
アデノシン三リン酸

```
\setchemfig{cram width=3pt}
\definesubmol{a}{-P(=[:90,0.75]O)(-[:90,0.75]HO)-}
\chemfig{[:-54]*5((-[:60]O([::-60]!aO([::-60]!aO([::-60]!aHO))))<(-OH)
-[,,,,line width=2pt](-OH)>(-N*5(-N*6(-(-NH_2)=N=N-)=_)-O-)}
```



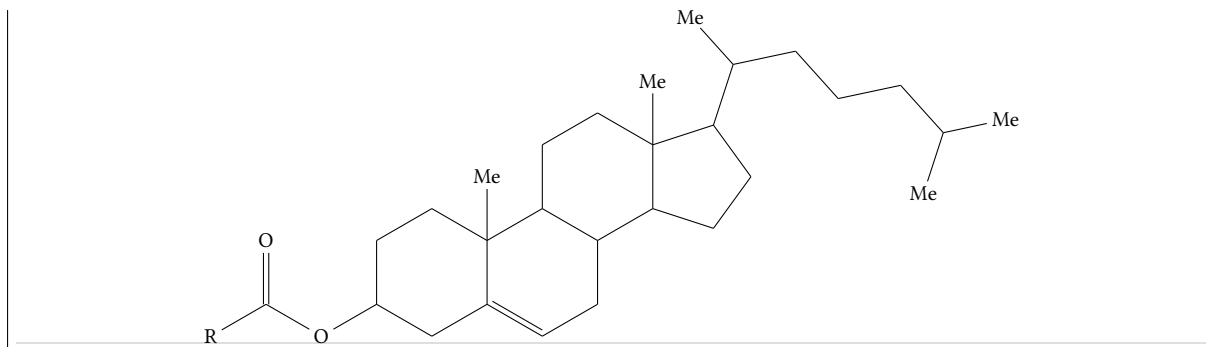
バイアグラ (シルденаフィル)

```
\chemfig{N*6((-H_3C)---N(-S(=[:+120]O)(=[:+0]O)-[:60]*6(==(-O-[:60]-[:60]CH_3)
=(*6(N=*5(-(-[:60]-[:60]CH_3)=N-N(-CH_3)-)=--(=O)-N(-H)-)=))---))}
```



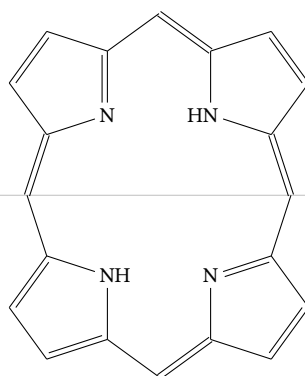
コレステロールエステル

```
\chemfig{[:30]R-(=[:+60]O)-[:60]O-*6(==*6(==*5(---(-(-[:+60]Me)
-[:60]-[:60]-[:60]-[:60](-[:60]Me)-[:60]Me)-(-[:+0]Me)---))--(-[:+0]Me)---))}
```

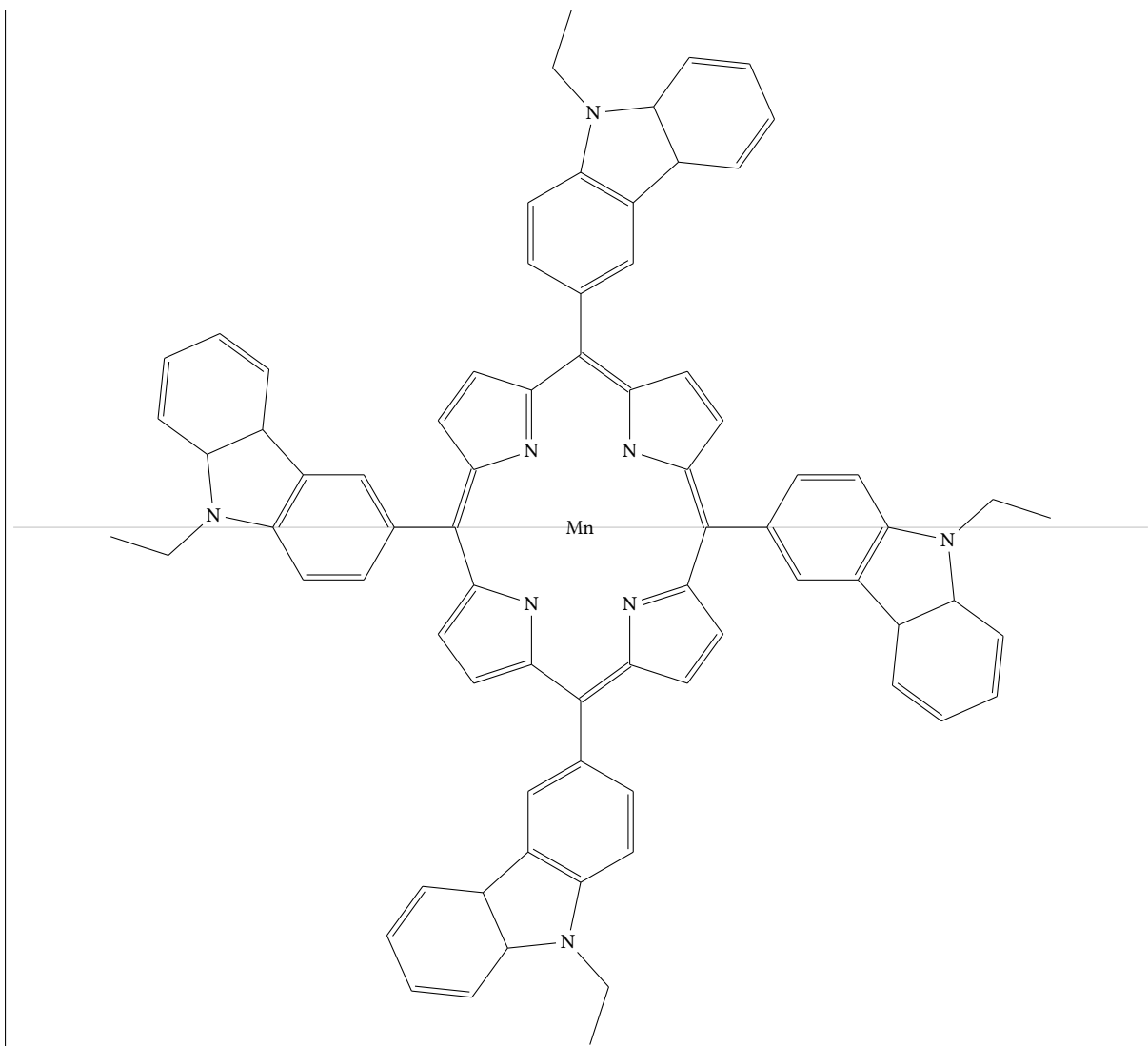
ポルフィリン

```
\chemfig{?[::+72]*5(-N(=[::-72]*5(-[,,,2]HN-[,,2](=[::-36]*5(N(=[::-72]*5(-NH-[,,1]?=-=))
--)))-=-)))-=-)}
```



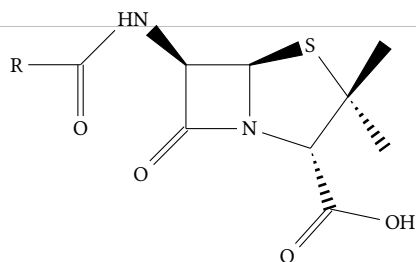
マンガン 5,10,15,20-テトラキス (N-エチル-3-カルバゾリル) ポルフィリン

```
\definesubmol{A}{*6(=-*5(-*6(=-=-)N(-[::-60])-)-=-)}
\chemfig{([::+180]-!A)=[::+72]*5(-N(-(-[::+54]!A)=[::-72]*5(-N(-[::-33,1.5,,,draw=none]Mn)
-(-(-[::+72]!A)-[::-36]*5(N(-(-[::+54]!A)-[::-72]*5(-N(-(-)=)))-=-)))-=-)))-=-)}
```



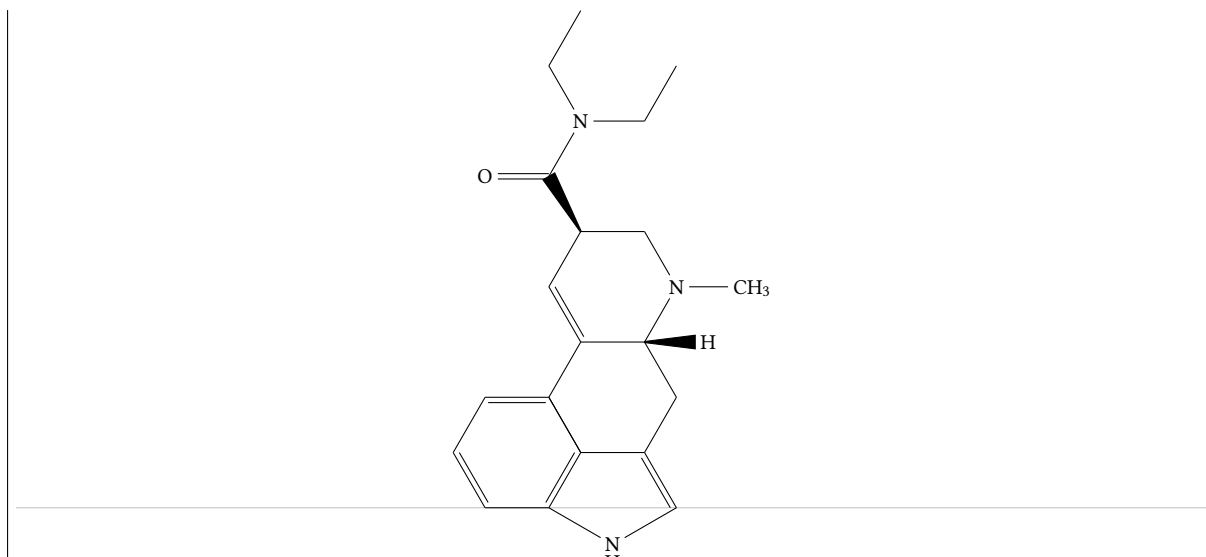
ペニシリン

\chemfig{[: -90]HN(-[: -45](-[: -45]R)=[: +45]O)>[: +45]*4(- (=O)-N*5(-(<[: -60]O)-[: +60]OH)-(<[: +0])(<[: -108])-S>--))}



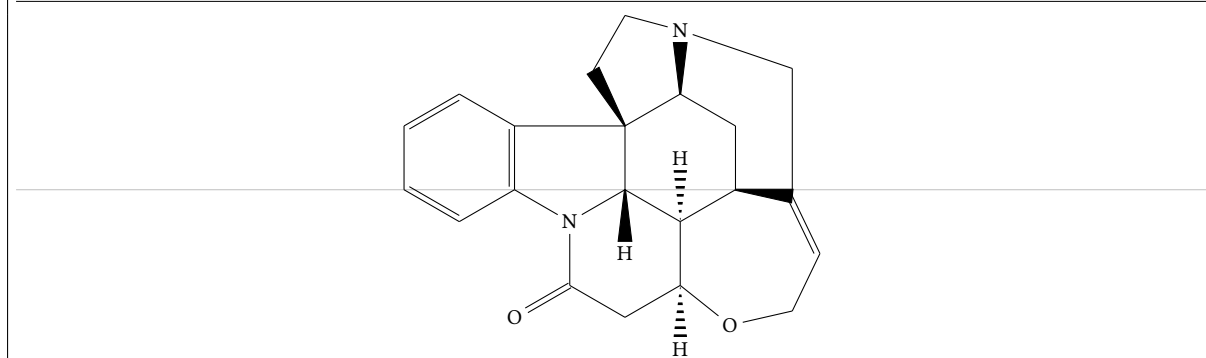
LSD

\chemfig{[:150]?*6(=*6(--*6(-N(-CH_3)--(<[: +60]O)-[: -60]N(-[: +60]-[: -60])--[: -60]-[: +60])=>)([: -120]<H)---)*6(-==(-[: -30,1.155]\chembelow{N}{H}?=))}



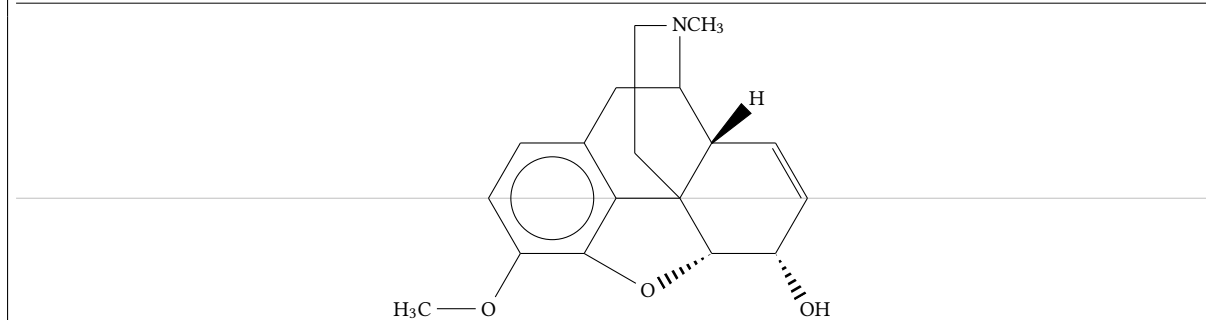
ストリキニーネ

```
\chemfig{*6(=*6(-N*6(- (=O)--([::-120]<:H)*7(-O---?[0]([::-25.714]-[ ,2]?[1]))
-*6(-?[0,{>}--(<N?[1]?[2])-(<[: -90]-[: -60]?[2]))(<[: +0]H)-([::+120]<H)--?)=?-=-)}
```



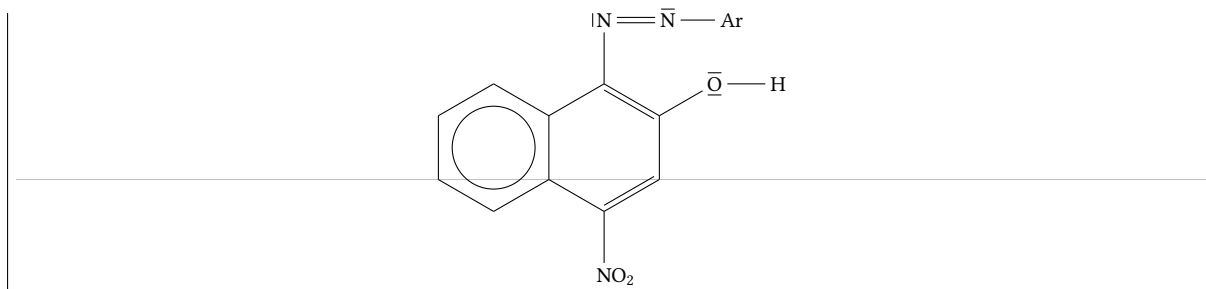
コデイン

```
\chemfig{[: -30]**6(-(-O[:180]H_3C)-?-*6(-(-[3]-[2,2]-[0, .5])*6(-(<[: -150,1.155]O?)
-(<[:0]H)-=-)-(<[1]H)-(-[2]NCH_3)--)---)}
```



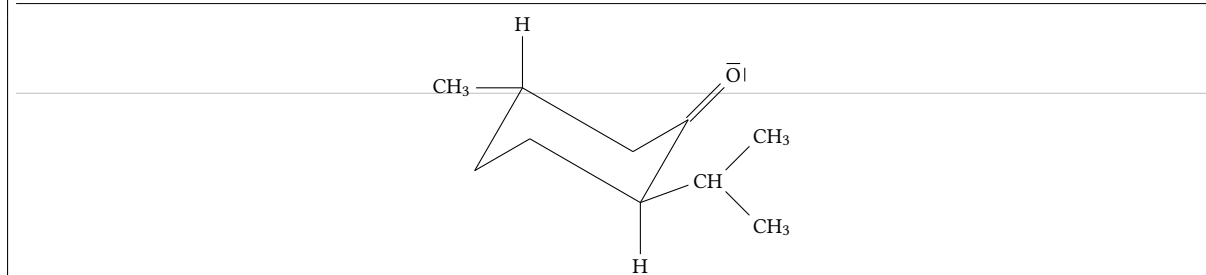
染料 (赤)

```
\chemfig{*6(---*6(-(-NO_2)=-(\charge{90=\|,-90=\|}{0})-([0]H)=(-\charge{180=\|}{N}=[0]\charge{90=\|}{N}-[0]Ar)-)---
)})
```



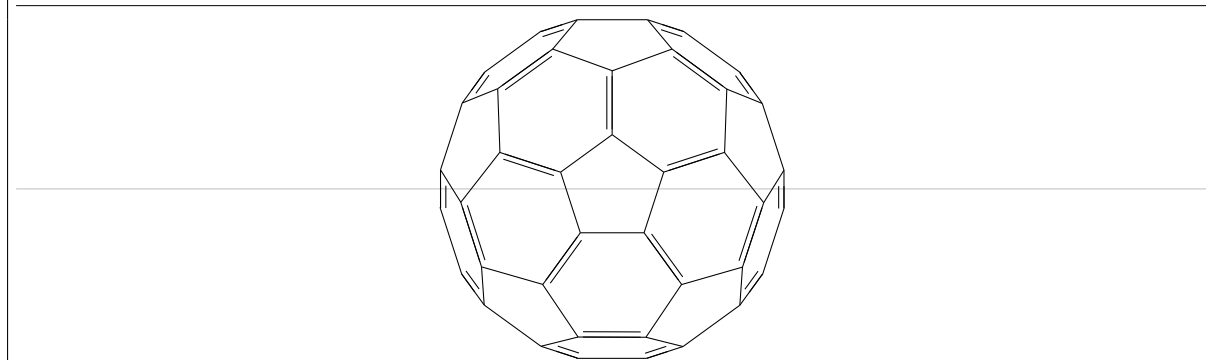
メントン

```
\chemfig{CH_3-?(-[2]H)(-[::-30,2]-[::+60](=[1]\charge{0=\,90=\}|){0})
-[::-150,1.5](-[:20]CH(-[1]CH_3)(-[7]CH_3))(-[6]H)-[::+90,2]-[::+60]?}}
```



フラレン

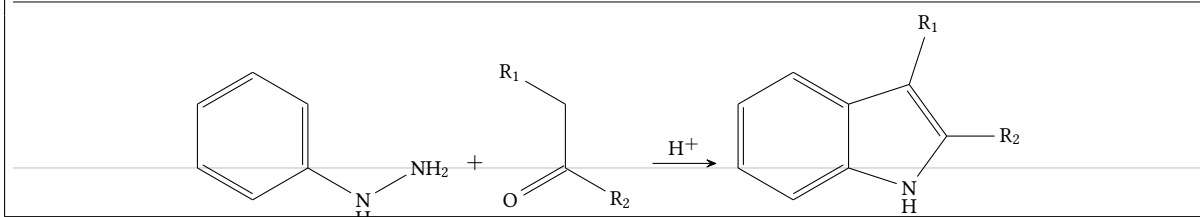
```
\definesubmol\fragment1{
  (-[:#1,0.85,,,draw=none]
  -[:126]-[:54](=_(2pt,2pt)[::180])
  -[:70](-[:56.2,1.07]=^(2pt,2pt)[::180,1.07])
  -[:110,0.6](-[:148,0.60](=^[::180,0.35])-[::-18,1.1])
  -[:50,1.1](-[:18,0.60]=_[::180,0.35])
  -[:50,0.6]
  -[:110])
}
\chemfig{
  !\fragment{18}
  !\fragment{90}
  !\fragment{162}
  !\fragment{234}
  !\fragment{306}
}
```



フィッシャー インドール合成

```
\schemestart
\chemfig{*6(=*6(-\chembelow{N}{H}-NH_2)=-=)}
\+
\chemfig{([:-150]0)(-[:30]R_2)-[2]-[:150]R_1}
\arrow(.mid east--.mid west){->[\chemfig{H^+}]}
\chemfig{*6(=*5(-\chembelow{N}{H})-(-R_2)=(-R_1)-=-=)}
```

\schemestop



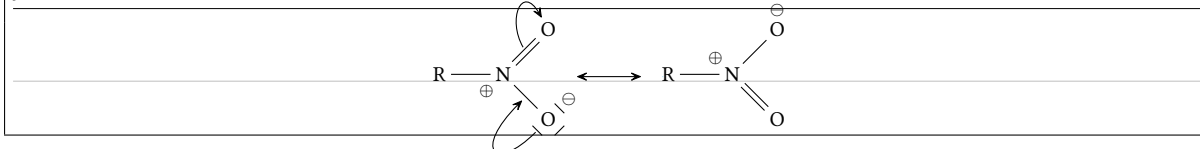
反応機構：カルボニル基

```
\schemestart
\chemfig{C([3-])([5-])=[@{db,.5}]@{atoo}\charge{0=\,|-90=\,}{0}}
\arrow(.mid east--.mid west){<->}
\chemfig{\charge{90:3pt=\scriptstyle\oplus}{C}([3-])([5-])-\%
\charge{0=\,|,90=\,|-90=\,|,45:3pt=\scriptstyle\ominus}{0}}
\schemestop
\chemmove{\draw[shorten <=2pt, shorten >=2pt](db) ..controls +(up:5mm) and +(up:5mm)..(atoo);}
```



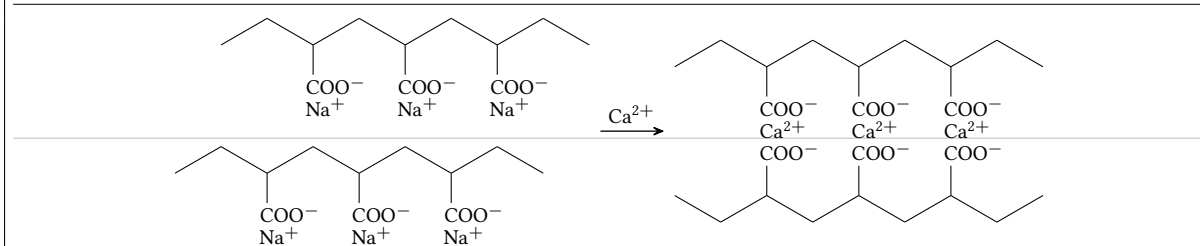
反応機構：ニトロ基

```
\schemestart
\chemfig{R-\charge{225:3pt=\scriptstyle\oplus}{N}([1]=[@{db}]@{atoo1}O)([7]-[@{sb}]@{atoo2}
\charge{45=\,|-45=\,|-135=\,|,45:5pt=\scriptstyle\ominus}{0})}
\arrow(.mid east--.mid west){<->}
\chemfig{R-\charge{135:3pt=\scriptstyle\oplus}{N}([1]-\charge{90:3pt=\scriptstyle\ominus}{O})([7]=O)}
\schemestop
\chemmove{
\draw[shorten <=2pt, shorten >=2pt](db) ..controls +(120:5mm) and +(120:7mm)..(atoo1);
\draw[shorten <=3pt, shorten >=2pt](atoo2) ..controls +(225:10mm) and +(225:10mm)..(sb);
}
```



アルギン酸カルシウムゲル

```
\definesubmol{0}{-[-3,.15,,,draw=none]}% invisible bond to stack
\definesubmol{COO/Na+}1{-[7]((-[-3,.66]COO^{\scriptstyle\ominus}N|a^{\scriptstyle\oplus}1))}
\definesubmol{chain}{-[-5]!{COO/Na+}{-[-5]!{COO/Na+}{-[-5]-[7]}}
\definesubmol{COO/Ca2+}1{-[7]((-[-3,.66]COO^{\scriptstyle\ominus}!0C|a^{\scriptstyle\oplus}2+!0COO^{\scriptstyle\ominus}-[-3,.66](-[7]-[5])#1))}
\hreact[fixed length,atom sep=2.5em,angle increment=30]
\chemfig{-[-5]!{COO/Na+}{!0\phantom{C}!0!{COO/Na+}{!{chain}}(-[-1])!{chain}}
>[\printatom{Ca^{2+}}]}
\chemfig{-[-5]!{COO/Ca2+}{(-[-1]-[1])}-[-5]!{COO/Ca2+}{-[-5]!{COO/Ca2+}{-[-5]-[7]}}
\endhreact
```



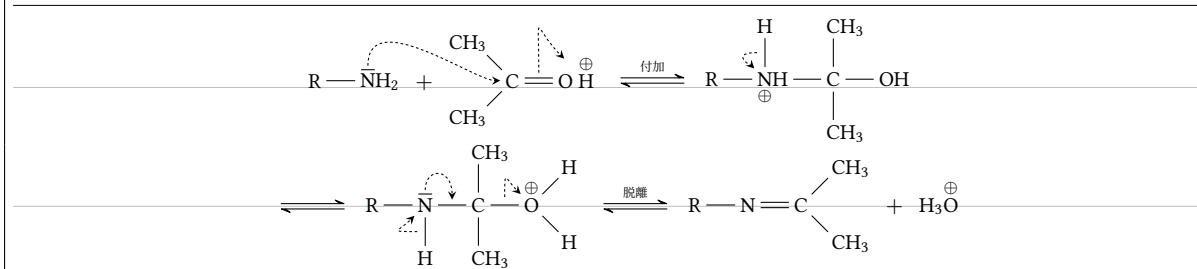
求核付加反応：第一級アミン

```
\setchemfig{atom sep=2.5em,compound sep=5em}
\schemestart
```

```

\chemfig{R-@{aton}\charge{90=\}\{N\}H_2}
\+
\chemfig{@{atoc}C([3]-CH_3)([5]-CH_3)=[@{atoo1}]O}
\chemfig{@{atoo2}\chemabove{H}{\scriptstyle\oplus}}
\chemmove[-stealth,shorten <=3pt,dash pattern= on 1pt off 1pt,thin]{
  \draw[shorten >=2pt](aton) ..controls +(up:10mm) and +(left:5mm)..(atoc);
  \draw[shorten >=8pt](atoo1) ..controls +(up:10mm) and +(north west:10mm)..(atoo2);}
\arrow{<=>[\tiny 付加]}
\chemfig{R-@{aton}\chembelow{N}{\scriptstyle\oplus}H([2]-[ @{sb} ]H)-C(-[2]CH_3)-([6]CH_3)-OH}
\schemestop
\chemmove{
  \draw[-stealth,dash pattern= on 1pt off 1pt,shorten <=3pt, shorten >=2pt]
  (sb)..controls +(left:5mm) and +(135:2mm)..(aton);}
\par
\schemestart
\arrow{<=>}
\chemfig{R-@{aton}\charge{90=\}\{N\}([6]-[ @{sbh} ]H)-[ @{sb} ]C(-[2]CH_3)-([6]CH_3)-[ @{sbo} ]@{atoo}}
\chemabove{O}{\scriptstyle\oplus}(-[1]H)(-[7]H)}
\chemmove[-stealth,shorten <=3pt,shorten >=2pt,dash pattern= on 1pt off 1pt,thin]{
  \draw(aton) ..controls +(up:5mm) and +(up:5mm)..(sb);
  \draw(sbh) ..controls +(left:5mm) and +(south west:5mm)..(aton);
  \draw(sbo) ..controls +(up:5mm) and +(north west:5mm)..(atoo);}
\arrow{<=>[\tiny 脱離]}\chemfig{R-N=C(-[1]CH_3)-([7]CH_3)}
\+
\chemfig{H_3\chemabove{O}{\scriptstyle\oplus}}
\schemestop

```

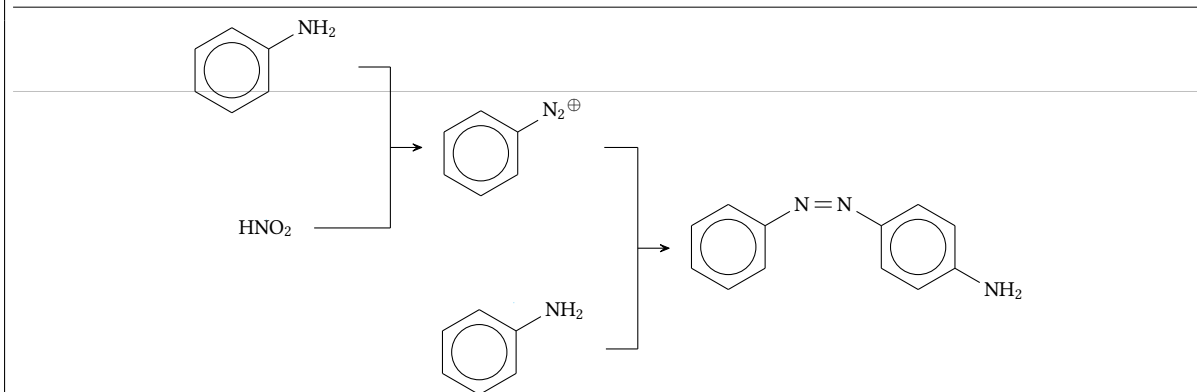


反応式

```

\setchemfig{atom sep=2em}
\schemestart[-90]
\chemfig{**6(---(-NH _2)---)}\arrow{0}\chemfig{HNO_2}
\merge(c1)(c2)--()
\chemfig{**6(---(-N_2|{\scriptstyle\oplus})---)}\arrow{0}\chemfig{**6(---(-NH _2)---)}
\merge(c3)(c4)--()
\chemfig{**6(---(-N=[:::30]N-[::-30]**6(---(-NH _2)---))---)}
\schemestop

```



付加反応

```

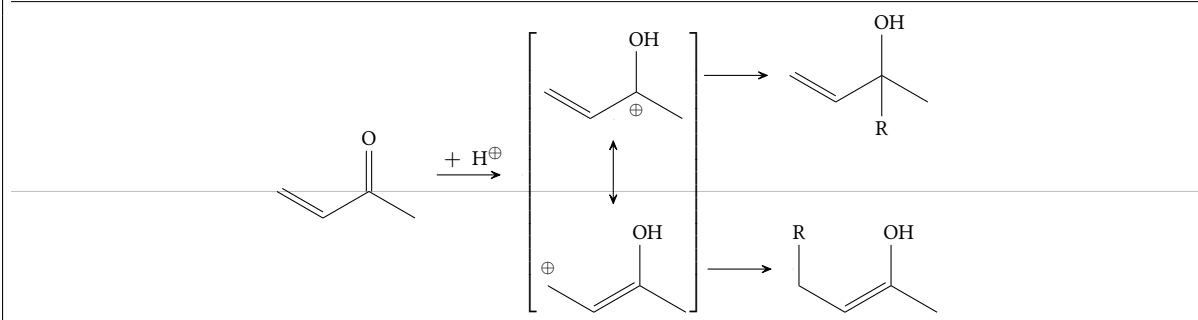
\setchemfig{atom sep=2.5em}

```

```

\schemestart
\chemfig{*6(=-(=([2]O))}
\arrow{>[\+\chemfig{H^\oplus}]}
\chemleft[\\subscheme[90]{%
\chemfig{*6((-[2,0.33,,draw=none]\scriptstyle\oplus)-(-)-OH)}
\arrow{<->}
\chemfig{*6(=-(=([6,0.33,,draw=none]\scriptstyle\oplus)-OH))}\chemright]
\arrow{@c3--}\chemfig{*6((-[2]R)-(-)-OH)}
\arrow{@c4--}\chemfig{*6(=-(=([6]R)-OH)}
\schemestop

```

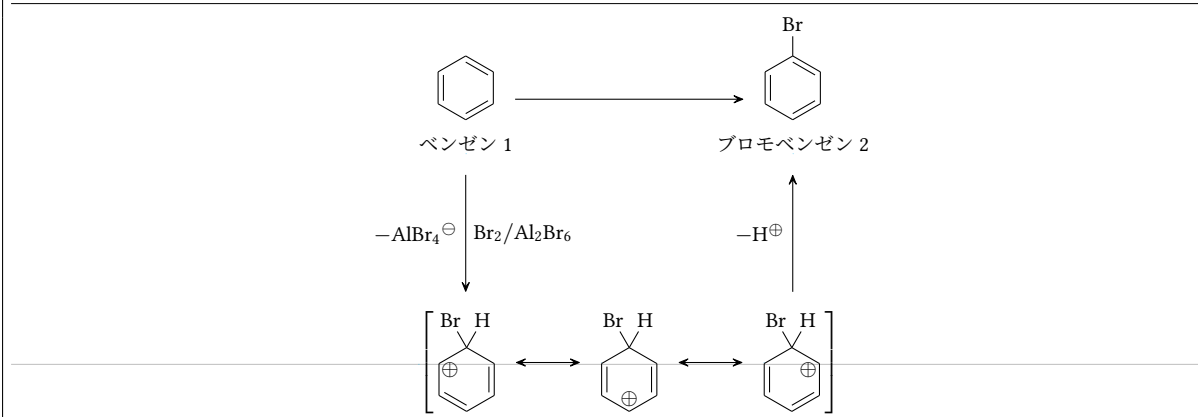


芳香族求電子置換反応

```

\setchemfig{atom sep=1.5em}%
\definesubmol{+}{-[-0.4,,draw=none]\oplus}%
\schemestart
\arrow{0}[,0]
\chemleft[\\subscheme{\chemfig{*6(=--(-[:120]Br)-[:60]H)-(!+)-)}}
\arrow{<->}
\chemfig{*6(-(!+)---(-[:120]Br)-[:60]H)-=)}
\arrow{<->}
\chemfig{*6(=--(!+)-(-[:120]Br)-[:60]H)-=)}\chemright]
\arrow{@c2--}{<-[*0\chemfig{-}AlBr_4|\^ominus][*0\chemfig{Br_2/Al_2Br_6}][90,1.5]}
\chemname{\chemfig{*6(=--=)}}{ベンゼン 1}
\arrow{@c4--}{>[*0\chemfig{-}H^\oplus][90,1.5]}
\chemname{\chemfig{*6(=--(-Br)-=)}}{プロモベンゼン 2}
\arrow{@c5.mid east--@c6.mid west}
\schemestop
\chemnameinit{}

```



塩素化の反応機構

```

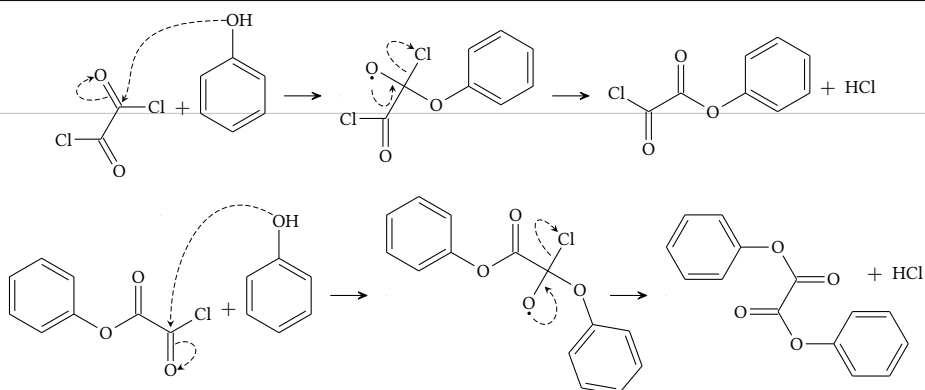
\scriptsize\setchemfig{bond offset=1pt,atom sep=2em,compound sep=4em}
\schemestart
\chemfig{Cl-[@{a0}]=[@{a1}:120]@{a2}O)-[:120]([:-60]O)-[4]Cl}\+\chemfig{*6(=--(-@{oh1}OH)-=)}\arrow
\chemfig{*6((-O-[:150](-[:@{o0}:150]@{o1}\charge{-90=\.}{O}))(-[:@{cl0}:60]@{cl1}Cl)-[:240](-[4]Cl)=[:6]O)=--=)}
\arrow\chemfig{*6((-O-[:150]([2]O)-[:150]([6]O)-[:150]Cl)=--=)}\+\chemfig{HCl}
\arrow{@c1--}{0}[-90,0.5]
\chemfig{*6(=--6(-O-6(-@{o2}=[@{o3}]@{o4}O)-Cl)([:60]O)=--=)}\+\chemfig{*6(=--(-@{oh2}OH)-=)}\arrow

```

```

\chemfig{*6(==6(-0(-(-[c12]:60)[c13]Cl)(-[o5]:-120)[o6]\charge{-90=\.}{0})-0[-:-40]*6(==--)))(=[::60]O)-
==)}
\kern-3em \arrow\chemfig{[:30]*6(==(-0-[:60](=0)-[:120](=[4]O)-[:60]O-*6(==--))=--)}
\kern-3em \+\chemfig{HCl}
\schemestop
\chemmove[line width=0.2pt,-stealth,dash pattern = on 2pt off 1pt]{
\draw[shorten <=2pt](a1)..controls+(200:5mm)and+(200:5mm)..(a2);
\draw[shorten >=2pt](oh1.west)..controls+(180:15mm)and+(60:5mm)..(a0);
\draw[shorten <=6pt,shorten >=2pt](o1)..controls+(270:5mm)and+(270:5mm)..(o0);
\draw[shorten <=2pt](c10)..controls+(150:5mm)and+(150:5mm)..(c11.150);
\draw[shorten <=2pt](o3)..controls +(30:3mm) and +(30:5mm)..(o4.east);
\draw[shorten >=2pt](oh2.135).. controls +(150:10mm) and +(90:10mm).. (o2);
\draw[shorten >=2pt,shorten <=5pt]([xshift=-1.5mm]o6.315)..controls +(315:5mm) and +(315:5mm)..(o5);
\draw[shorten <=2pt](c12)..controls +(135:5mm) and +(135:5mm)..(c13.north west);}

```

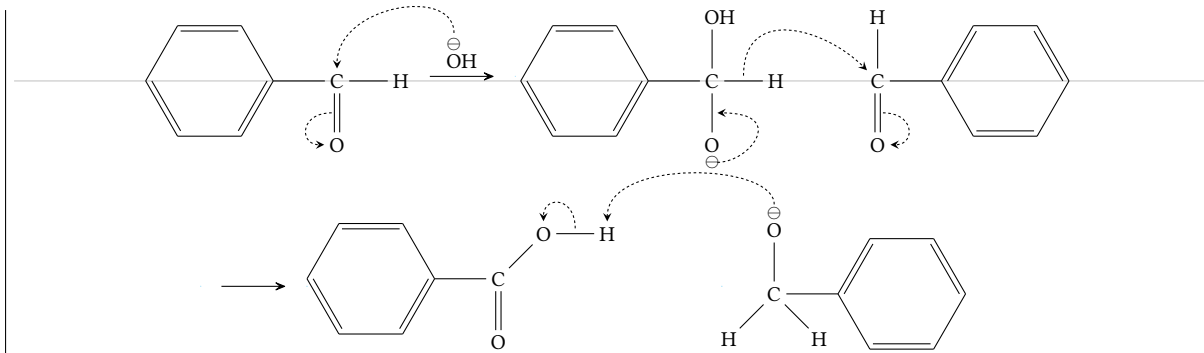


カニッツァー口反応

```

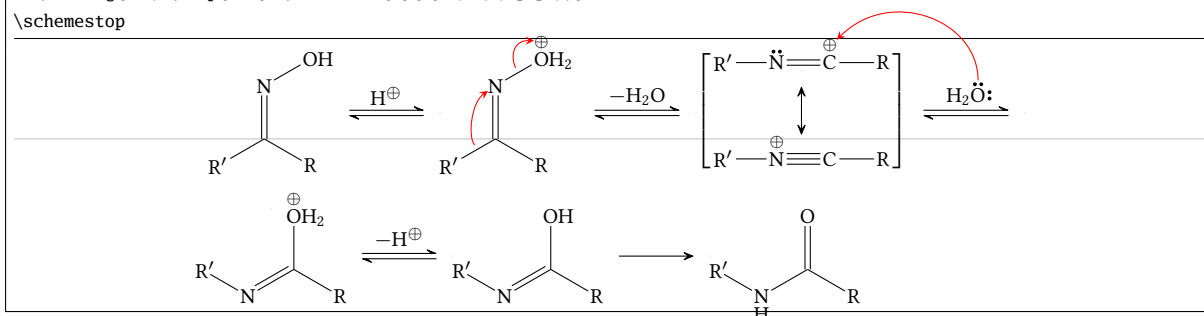
\schemestart
\chemfig{[:30]*6(==(-@{atoc}C([6]=[db])@{atoo1}O)-H)-==)}
\arrow[start.mid east--.mid west]{->\chemfig{@{atoo2}\chemabove{0}{\scriptstyle\ominus}}H}
\chemmove[-stealth,shorten >=2pt,dash pattern=on 1pt off 1pt,thin]{
\draw[shorten <=8pt](atoo2) ..controls +(up:10mm) and +(up:10mm)..(atoc);
\draw[shorten <=2pt](db) ..controls +(left:5mm) and +(west:5mm)..(atoo1);}
\chemfig{[:30]*6(==(-C([6]-[sb1])@{atoo1}\chembelow{0}{\scriptstyle\ominus})([2]-OH)-[sb2]H)-==)}
\hspace{1cm}
\chemfig{[:30]*6((-@{atoc}C([6]=[db])@{atoo2}O)-[2]H)-==)}
\chemmove[-stealth,shorten <=2pt,shorten >=2pt,dash pattern=on 1pt off 1pt,thin]{
\draw([yshift=-4pt]atoo1.270) ..controls +(0:5mm) and +(right:10mm)..(sb1);
\draw(sb2) ..controls +(up:10mm) and +(north west:10mm)..(atoc);
\draw(db) ..controls +(right:5mm) and +(east:5mm)..(atoo2);}
\arrow(@start.base west--){0}[-75,2]
{}
\arrow
\chemfig{[:30]*6(==(-C([1]-@{atoo2}O-[sb]0)[atoh]H)([6]=O))-==)}
\arrow{0}
\chemfig{[:30]*6((-C(-[5]H)-[7]H)-[2]@{atoo1}\chemabove{0}{\scriptstyle\ominus})=--)}
\chemmove[-stealth,shorten >=2pt,dash pattern=on 1pt off 1pt,thin]{
\draw[shorten <=7pt](atoo1.90) ..controls +(90:8mm) and +(up:10mm)..(atoh);
\draw[shorten <=2pt](sb) ..controls +(up:5mm) and +(up:5mm)..(atoo2);}
\schemestop

```

ベックマン転位

```
\setchemfig{bond offset=1pt,atom sep=2.5em,compound sep=5em,arrow offset=6pt}
\schemestart
\chemfig{(-[:150]R')(-[:30]R)=[2]N-[:30]OH}
\arrow{<=>[\chemfig{H^{\oplus}}]}
\chemfig{(-[:150]R')(-[:30]R)=[2]@{a1}N-@{b0}:30@{b1}\chemabove{O}{\scriptstyle\oplus}H_2}
\chemmove[red,-stealth,red,shorten <=2pt]{
\draw(a0)..controls +(135:2mm) and +(215:4mm).. (a1);
\draw(b0)..controls +(120:2mm) and +(180:3mm).. ([yshift=7pt]b1.180);}
\arrow{<=>[\chemfig{H_2O}]}[1.1]
\chemleft[\subscheme{90}]%%
\chemfig{R'-\chemabove{N}{\scriptstyle\oplus}~C-R}
\arrow{<->}[0.75]
\chemfig{R'-\charge{90=}{N}=@{a1}\chemabove{C}{\scriptstyle\oplus}-R}\chemright]
\arrow{<=>[\chemfig{H_2O@{a0}\charge{0=}{,90=}{O}}]}[1.1]
\chemmove[red,-stealth,red,shorten <=3pt]{
\draw(a0)..controls+(90:10mm)and+(45:10mm)..([yshift=6pt]a1.45);}
\arrow{@c1--}{0}[-90,0.333]
\chemfig{*6(R\rlap{$'$}-N=(-R)-\chemabove{O}{\scriptstyle\oplus}H_2)}
\arrow{<=>[\chemfig{H^{\oplus}}]}
\chemfig{*6(R\rlap{$'$}-N=(-R)-OH)}
\arrow
\chemfig{*6(R\rlap{$'$}-\chembelow{N}{H}-(-R)(=[2]O))}
\schemestop
```

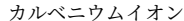


反応式

```
\setchemfig{atom sep=1.5em,compound sep=4em}
\schemestart
\chemfig{-[:30]=[:60](-[:60])-[:60]}
\arrow{->[\chemfig{HCl}]}
\chemfig{-[:30]-[:60](-[:120]Cl)(-[:60])-[:60]}+\chemfig{-[:30](-[:60]Cl)-[:60](-[:60])-[:60]}
\arrow{@c1--.north west}{->[\chemfig{H_2O}]}[-45,1.7]
\chemfig{-[:30]-[:60](-[:120]OH)(-[:60])-[:60]}+\chemfig{-[:30](-[:60]OH)-[:60](-[:60])-[:60]}
\schemestop
```

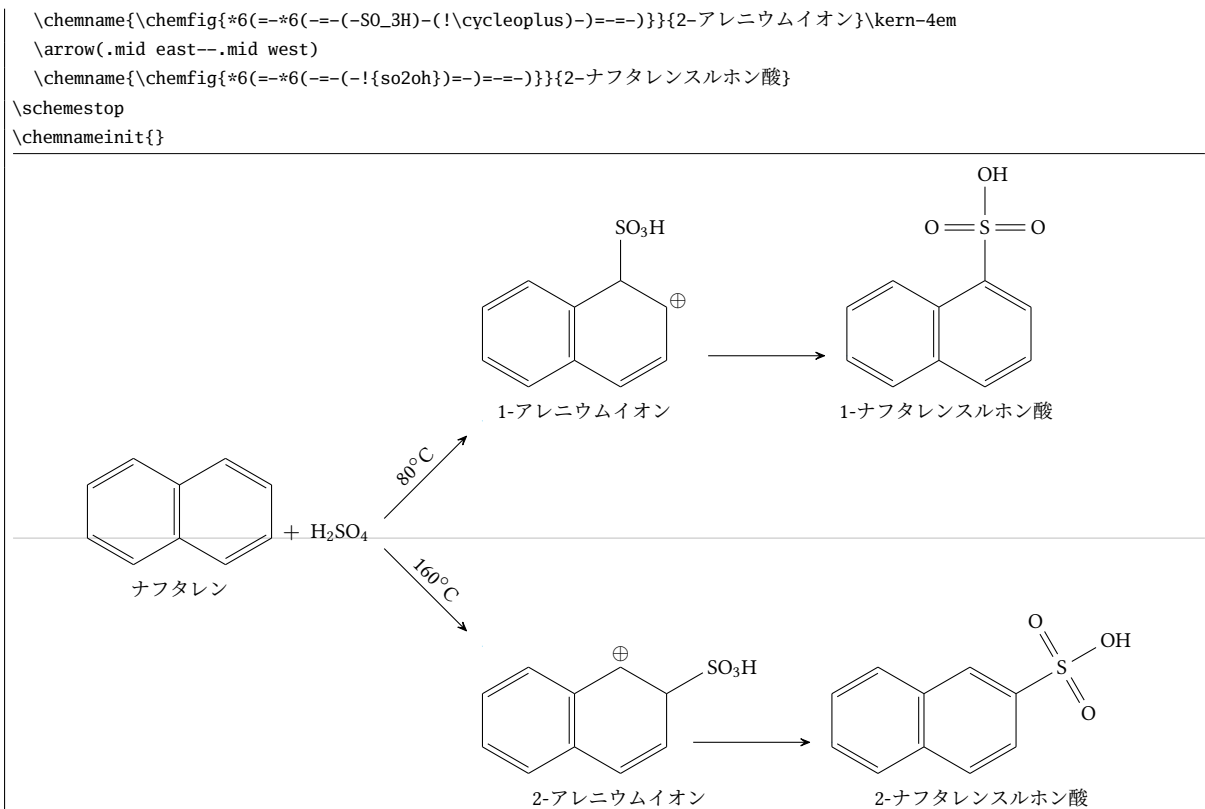


```
\schemestop
\chemnameinit{}
```

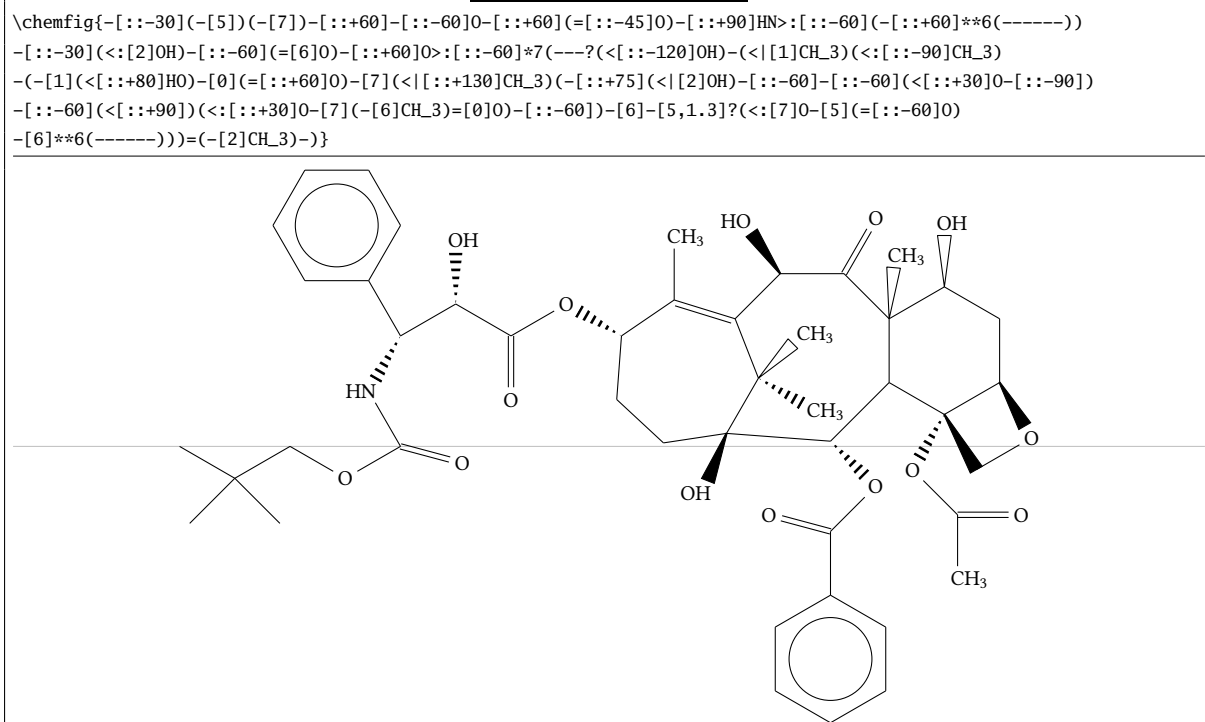


ナフタレンのスルホン化

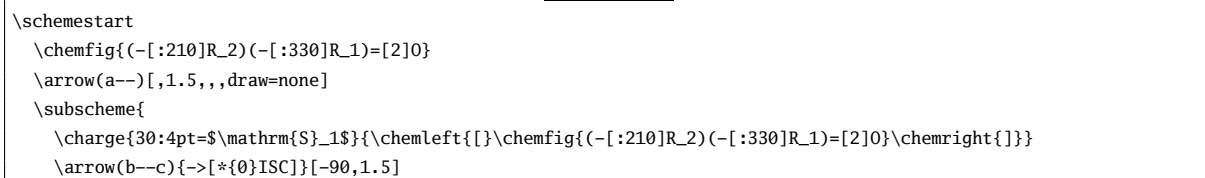
コンパイル日：2025 年 11 月 3 日



タキソテール (ドセタキセル)



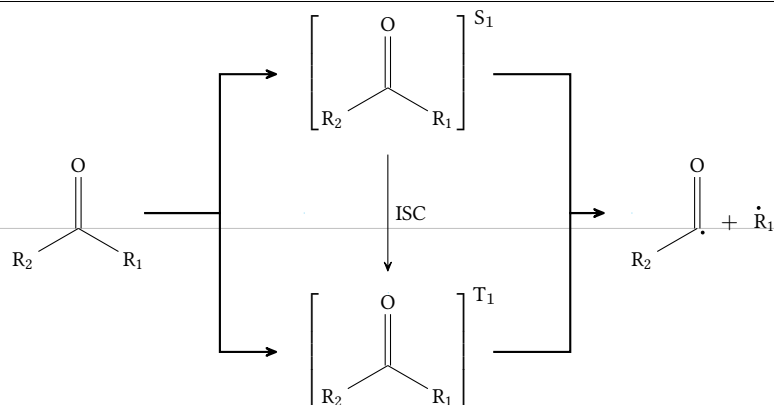
分岐する矢印



```

\charge{30:4pt}=\mathrm{T}_1$\chemleft[\chemfig{(-[:210]R_2)(-[:330]R_1)=[2]O}\chemright{}}
}
\arrow{--d}[,1.5,,draw=none]
\chemfig{(-[:210]R_2)(-[:330,0.1,,draw=none]\charge{330:-1pt=\.,\,})=[2]O}
\+
\chemfig{\charge{90:1pt=\.,\,}{R}_1}
\schemestop
\chemmove{
\draw[thick,shorten >=10pt] (a.east) -- ++(1,0) |- (b.west);
\draw[thick,shorten >=10pt] (a.east) -- ++(1,0) |- (c.west);
\draw[-,thick,shorten >=15pt,shorten <=8pt] (b.east) -- ++(1.3,0) |- (d.west);
\draw[thick,shorten >=10pt,shorten <=8pt] (c.east) -- ++(1.3,0) |- (d.west);
}

```



変更履歴

最初のバージョン以降の `chemfig` における変更点は以下の通りです（最新のものが先頭）。この節は英語に翻訳されていません。

Version 1.71 du 30/10/2025

- bugfix : trop de mauvais raccords de liaisons avec la nouvelle macro `\CF_ifzerodim`. Retour à la précédente définition (la macro `\CF_ifzerodim` n'est utilisée que si `use atom strut = <true>`).

Les codes donnent à nouveau les bons raccords de liaisons :

1. `\chemfig{?-A-[:90]B?}`
2. `\chemfig{A-@{a}-B}`
3. `\chemfig{[:30]-\charge{90=a}{}-[: -30]}`

On obtient bien 3,6 avec le code suivant :

```
\newcount\xx
\begingroup
  \def\printatom#1{\global\advance\xx1 \ensuremath{\mathrm{#1}}}
  \xx=0 \setbox0\hbox{\chemfig{A-B-C}} \the\xx,
  \xx=0 \setbox0\hbox{\chemfig[use atom strut]{A-B-C}}\the\xx
\endgroup
```

- bugfix : prise en compte de l'argument optionnel de `\chemfig` dans l'environnement `hreact`.
Le code `\hreact\chemfig[angle increment=+30]{A-[1]B} > C\endhreact` ne plante plus à cause du + dans l'argument optionnel
- bugfix : prise en compte de l'argument optionnel de `\chemname` dans l'environnement `hreact`.
- bugfix : dans l'environnement `hreact`, l'argument de `\^{<dimension>}` n'est plus évaluée à 0pt si exprimé en ex ou em.
- ajout : par souci de cohérence, dans l'environnement `hreact`, la macro `\name[<dim>]{<nom>}` a un argument optionnel pour spécifier l'espacement du `<nom>` avec le `<composé>`
- mise en place de warnings si utilisation d'un ou deux arguments optionnels de `\schemestart`. Création de la clé `init anchor` pour accéder au réglage possible avec le deuxième argument optionnel.
- mise en cohérence : la clé `name sep` vaut désormais 1.5ex par défaut et s'applique à `\chemname` si son argument optionnel est vide

Version 1.7 du 26/10/2025

- réaction horizontales avec l'environnement

`\hreact... \endhreact`

qui offre une syntaxe plus simple que `\schemestart... \schemestop`

- par défaut désormais, l'argument de `\printatom` est développé et dépourvu de strut¹².
Afin d'assurer la compatibilité, le nouveau booléen `use atom strut` peut être mis à `<true>` afin de retrouver le comportement précédent

¹²dysfonctionnement aimablement signalé sur 2 sites par Joseph Wright avec son amusante habitude d'exprimer publiquement le mal qu'il pense de mes packages.

- lorsque la clé `use atom strut = <false>`, chaque atome n'est composé typographiquement qu'une seule fois alors que c'était parfois 5 fois ou plus auparavant puisque `\vphantom` compose son argument dans une boîte de \TeX . La macro `\CF_ifzerodim` a été rapidement corrigée pour que ce soit le cas. Il reste encore des investigations à faire pour que le code soit plus propre...
Il se peut que ce changement provoque des petits bugs d'affichage dans le dessin de certaines molécules. Merci de me les signaler.
- suppression de l'historique des changements dans `chemfig.tex` pour le mettre ici dans le manuel.

Version 1.66 du 28/12/2023

- les liaisons de Cram pleines sont jointes entre elles ou aux liaisons simples lorsque `bond join = <true>`

Version 1.6e du 30/06/2023

- nouvelle clé `baseline` pour régler finement l'alignement vertical d'une molécule

Version 1.6d du 18/02/2023

- les ancres 'b' et 'd' n'étaient pas prises en compte pour le tracé de flèches directes de type `(@a.b-@c.d)` dans les schémas réactionnels
- correction d'un bug : `\CF_currentfromatom` n'était pas initialisé au début d'un cycle et donc le code suivant plantait
`\chemfig{AB-[,2]*3(---)}`

Version 1.6c du 27/09/2022

- nouvelles clés `gchemname`, `schemestart code` et `schemestop code` (suggestion de Balazs Debrececi)

Version 1.6b du 01/08/2021

- encodage UTF-8
- la macro `\#` n'était pas définie pour remplacer `#(...)` lorsque `\chemfig` se trouve dans l'argument d'une macro.

Version 1.6a du 28/02/2021

- le fichier `lewis.tex` a été renommé `chemfig-lewis.tex`

Version 1.6 du 26/02/2021

- les macros des formules de Lewis sont retirées et placées dans le fichier séparé "`lewis.tex`" que l'utilisateur peut charger s'il le souhaite
- ajout d'une clé `debug` pour le trousseau `[chemfig]`
- à l'intérieur d'un schéma, le token '#' est permis dans l'argument de `\chemfig`

Version 1.56 du 13/07/2020

- le centre des cycles est désormais accessible via un nœud spécifique pour chacun d'eux.

Version 1.55 du 15/06/2020

- `chemfig` est incompatible avec `conTeXt`, vu que ce moteur redéfinit des primitives telles que `\expanded`, `\unexpanded` et peut être d'autres.

Version 1.54 du 21/05/2020

- chemfig ne peut plus fonctionner sans `\expanded`
- bug : un signe "=" laissé par erreur dans le flux

Version 1.53 du 27/04/2020

- mise à jour en fonction des nouvelles fonctionnalités de l'extension `simplekv`
- bug : `\CF_ifzerodim` interrompt maintenant le tracé dans la `\hbox`

Version 1.52 du 14/04/2020

- bug : définition corrigée de `\CFthesubmol` dans `\CF_defsubmolc` pour qu'elle se développe en 1 coup seulement

Version 1.51 du 06/04/2020

- bug corrigé dans `\chargerect_a` et `\chargeline_a`

Version 1.5 du 05/03/2020

- nouvelles macros `\charge` et `\Charge`. Les macros `\lewis` et `\Lewis` sont obsolètes et amenées à disparaître à moyen terme (au moins 9 mois), soit fin 2020
- prise en compte de la dimension d'un groupe d'atome pour tracer des liaisons jointives
- bug corrigé dans `\CF_searchnode`
- ajout d'une section dans le manuel (placement des atomes)

Version 1.41 du 21/05/2019

- utilisation de la nouvelle primitive `\expanded`
- nouvelle clé `h align` (`<true>` par défaut) pour les délimiteurs de `\polymerdelim`. Lorsque à `<false>`, les délimiteurs ne sont plus alignés horizontalement mais positionnés aux noeuds demandés
- nouvelle clé `auto rotate` qui n'a de sens que si `h align = <false>` : les délimiteurs sont automatiquement inclinés
- nouvelle clé `rotate` qui n'a de sens que si `h align = <false>` ET `auto rotate = <false>` : l'inclinaison des délimiteurs peut être à false choisie

Version 1.4 du 18/04/2019

- corrections de nombreux bugs
- caractère privé "_" et non plus "@" d'où des modifications à prévoir notamment dans la doc avec les codes spécifiques aux flèches, ça risque de couiner sur tex.stackexchange.com
- anciennes macros abandonnées et désormais indéfinies :
`\setcrambond`, `\setatomsep`, `\setbondoffset`, `\setdoublesep`, `\setangleincrement`, `\enablefixedbondlength`, `\disablefixedbondlength`, `\setnodestyle`, `\setbondstyle`, `\setlewis`, `\setlewisdist`, `\setstacksep`, `\setarrowdefault`, `\setcompoundstyle`, `\setandsign`, `\setarrowoffset`, `\setcompoundsep`, `\setarrowlabelsep`, `\enablebondjoin`, `\disablebondjoin` et `\schemedebbug`
- l'ancienne syntaxe `\chemfig[[[]]{}` est abandonnée et n'est plus acceptée, désormais c'est
`\chemfig[<clés>=<valeurs>]{<code molécule>}`
- l'ancienne syntaxe `\lewis[<coeff>]` ou `\Lewis[<coeff>]` n'est plus acceptée au profit de `\lewis[<clés>=<valeurs>]`

Version 1.34 du 23/02/2019

- bug dans la flèche " \leftrightarrow " corrigé

Version 1.33 du 31/10/2018

- les macros définies par `\definesubmol` peuvent désormais avoir un ou plusieurs arguments
- macro `\polymerdelim` documentée

Version 1.32 du 23/08/2018

- définition de `\printatom`, `\CF_begintikzpicture` et `\CF_endtikzpicture` dans le fichier `t-chemfig.tex`

Version 1.31 du 05/04/2018

- correction d'un espace indésirable dans `\CF_ifnextchar`

Version 1.3 du 08/03/2018

- tous les paramètres sont désormais passés via `\setchemfig` qui fait appel à "`simplekv`". Par conséquent, *toutes* les macros qui réglaient des paramètres deviennent obsolètes, à savoir :
`\setcrambond`, `\setatomsep`, `\setbondoffset`, `\setdoublesep`, `\setangleincrement`, `\enablefixedbondlength`, `\disablefixedbondlength`, `\setnodestyle`, `\setbondstyle`, `\setlewis`, `\setlewisdist`, `\setstacksep`, `\setarrowdefault`, `\setcompoundstyle`, `\setandsign`, `\setarrowoffset`, `\setcompoundsep`, `\setarrowlabelsep`, `\enablebondjoin`, `\disablebondjoin` et `\schemedebug`
Ces macros seront **supprimées** dans une future version.
- la version étoilée `\chemfig*` et les deux arguments optionnels de la macro `\chemfig[[]]` sont également optionnels et seront **supprimés** dans une future version afin d'accéder à la syntaxe `\chemfig[clés=valeurs]{code}`
- 6 nouveaux paramètres : `lewis radius`, `arrow double sep`, `arrow double coeff`, `arrow double harpoon`, `cycle radius coeff`, `arrow head`.
- correction d'un bug dans `\CF_parsemergeopt` qui dans certains cas, envoyait vers l'affichage des caractères
- petit toilettage du code
- macro `\polymerdelim` (non documentée) expérimentale et encore en phase de tests
- suppression d'un registre d'écriture de fichier

Version 1.2e du 20/05/2017

- la macro contenant la définition d'une flèche est désormais "`\CF_arrow(<nom>)`", ainsi la macro `\0` n'est plus définie par `\definearrow`
- remerciements rajoutés après une suppression induite, pour ne froisser aucune susceptibilité

Version 1.2d du 01/12/2015

- correction d'un bug dans la flèche " \dashv "
- la version étoilée de `\setcrambond` dessine les liaisons de Cram en pointillés sous forme de trait large et non pas sous forme de triangle.

Version 1.2c du 20/11/2015

- Correction d'un bug dans `\CF_setbondangle` : l'angle renvoyé pouvait être négatif

- Correction d'un bug dans `\CF_directarrow` : la macro `\CF_ifempty` n'est pas correctement développée dans l'argument de `\pgfpoinanchor`

Version 1.2b du 15/11/2015

- bug dans `\CF_searchsubmol` qui laissait "*" dans le flux de lecture de TeX. Un message d'erreur est également ajouté en cas de "!" en fin de traitement.
- correction d'un bug dans `\CF_setbondangle` où l'angle $[\langle :a \rangle]$ n'était pas évalué par `\pgfmithsetmacro`.

Version 1.2a du 21/10/2015

- erreur de copier-coller dans le code: une adresse url était malencontreusement présente en plein milieu d'une ligne de code

Version 1.2 du 08/10/2015

- correction d'un bug dans le tracé des liaisons de Cram.
- création de `\setangleincrement`.
- chargement de "arrows.meta" et définition de la flèche "CF" basée sur "Stealth" et définie avec `\pgfdeclarearrow`. Les anciennes flèches "CF_full" et "CF_half" sont abandonnées puis définies avec `\pgfarrowsdeclare`.
- flèche "-U>" corrigée : le placement des labels est maintenant correct dans tous les cas. Ainsi :

$$-U>[\langle a \rangle][\langle b \rangle][\langle d \rangle][\langle r \rangle][\langle a \rangle]$$
place le label $\langle a \rangle$ près du début de la flèche, quels que soient les signes du rayon $\langle r \rangle$ et de l'angle $\langle a \rangle$.
- `\chemrel`, `\setchemrel` et `\chemsign` sont supprimées.
- compatibilité, avec les limitations d'usage, avec la librairie "externalize" : le `\begin{tikzpicture}` voit désormais le `\end{tikzpicture}` correspondant dans la macro `\CF_chemfigb`.

Version 1.1a du 23/02/2015

- correction d'un bug dans `\CF_grabbondoffset`. Si `\chemfig` est dans l'argument d'une macro, les "#" sont doublés par l'action de `\scantokens` de la macro `\CF_chemfigb` et il faut un argument délimité avant "(" pour absorber tous les "#".

Version 1.1 du 13/02/2015

- correction d'un bug dans `\CF_searchsubmol` : la macro `\CF_molecule` est dépouillé de son éventuel espace en première position.
- correction d'un bug dans `\CF_arrowf` : le nom du prochain nœud courant "end@arrow@i" était erroné dans le cas où une flèche contenait un sous schéma. Ce nom doit dépendre de `\CF_schemenest`.
- la jonction entre deux liaisons consécutives dans l'axe peut être activé avec `\enablebondjoin` et désactivé avec `\disablebondjoin` (préférable, état par défaut).
- `\chemfig` suivi d'une "*" demande à ce que les liaisons aient une longueur invariable : la distance inter-atome devient donc variable. Cette fonctionnalité est désactivé dans les cycles afin que les polygones soient réguliers. `\enablefixedbondlength` permet cette fonctionnalité pour toutes les macros `\chemfig` (même non étoilée) tandis que `\disablefixedbondlength` le désactive.

Version 1.0h du 28/11/2013

- `\chemname` admet maintenant une version étoilé qui ne tient pas compte des profondeurs précédentes.
- `\CF_dpmax` est géré globalement.
- correction d'un bug dans "-U>" : le style de la flèche n'était pris en compte pour l'arc.
- correction d'un bug dans `\CF_directarrow` : l'angle de la flèche n'était pas calculé

Version 1.0g du 16/11/2012

- correction d'un bug dans `\CF_directarrow` pour faire prendre en compte le style des flèche par défaut
- correction d'un bug dans `\CF_lewisc` : la boîte *doit* être composée en dehors de l'environnement `tikzpicture` pour éviter `nullfont` si jamais `\printatom` ne passe pas en mode math.
- correction d'un bug dans `\CF_chemfigc` : si une longueur par défaut est modifiée par `[, <l]` au début d'une molécule et si des cycles étaient emboîtés, cette longueur n'était pas appliquée aux sous-cycles.
- ré-écriture des macros `\chemabove` et `\chembelow` pour prendre en compte le bug (désormais corrigé) dans `luatex`.
- nouvelle macro `\setstacksep` qui définit l'espacement par défaut dans les macros `\chemabove` et `\chembelow`.

Version 1.0f du 24/02/2012

- correction d'un bug avec `\definesubmol`, les catcodes n'étaient pas correctement gérés.

Version 1.0e du 13/01/2012

- la gestion des espaces dans les groupes d'atomes est désormais plus rigoureuse. Plusieurs bugs ont été corrigés

Version 1.0d du 19/12/2011

- les cercles des cycles étaient tracés au mauvais moment. La longueur de la liaison qui les précédait influait sur le rayon du cercle :

`\chemfig{--[,0.5]**6(-----)}`

donnait un bug à l'affichage.

Version 1.0c du 30/11/2011

- la macro `\+` n'est plus explicitement écrite
- vérifie que eTeX est le moteur utilisé

Version 1.0b du 29/11/2011

- la commande `\merge` est désormais protégée entre `\schemestart` et `\schemestop` contre des définitions par d'autres packages.
- `\box0` est utilisé au lieu du maladroit `\unhbox0`

Version 1.0a du 18/09/2011

- les macros `\Lewis` et `\lewis` admettent un argument optionnel
- la macro `\setlewisdist` règle la distance entre les 2 électrons

Version 1.0 du 15/06/2011

- les schémas réactionnels sont désormais disponibles.
- `\Chemabove` et `\Chembelow` modifient la boîte englobante.
- `\Lewis` modifie la boîte englobante
- les macros `\chemleft`, `\chemright`, `\chemup` et `\chemdown` affichent des délimiteurs extensibles à gauche, à droite, au dessus et au dessous d'un matériel.

Version 0.4b du 24/04/2011

- l'argument de `\chemfig` est tokénisé avec `\scantokens` ce qui rend caduc tout souci de code de catégorie, à part #.
- la commande `\setbondstyle` permet de définir le style des liaisons.
- correction de l'affichage incorrect des doubles liaisons dans les cycles après les commandes `\hflipnext` et `\vflipnext`
- correction d'un bug lorsqu'un alias commence une molécule

Version 0.4a du 10/04/2011

- Correction d'un bug concernant l'argument optionnel en début de molécule.

Version 0.4 du 07/03/2011

- `chemfig` est désormais écrit en plain-etex et donc utilisable par d'autres formats que LaTeX.
- Un peu plus de rigueur avec les catcodes des caractères spéciaux, notamment lorsque la commande `\chemfig` se trouve dans l'argument de `\chemmove`, `\chemabove`, `\chembelow`, `\chemrel`. TODO : faut-il `\scantoken` l'argument de `\chemfig` pour être définitivement débarrassé de ces histoires de catcode ???
- Correction d'un bug dans le calcul de l'angle des liaisons

Version 0.3a du 08/01/2011

- Correction d'un bug dans l'argument optionnel de `\definesubmol` lorsque celui-ci comporte des crochets.
- Mise à jour du manuel en anglais.
- Ajout de `\vflipnext` et `\hflipnext` pour retourner horizontalement ou verticalement la prochaine molécule.

Version 0.3 du 21/11/2010

- Amélioration de `\definesubmol` qui accepte les séquences de contrôle. On peut aussi choisir un alias dont la substitution est différente selon l'orientation de la liaison qui lui arrive dessus.
- Le caractère `"|"` force la fin d'un atome. Si on écrit `"D|ef"` alors, `chemfig` verra deux atomes `"D"` et `"ef"`.
- Le caractère `"#"` est reconnu lorsqu'il suit un caractère de liaison. Il doit être suivi d'un argument entre parenthèses qui contient l'offset de début et de fin qui s'appliqueront à cette liaison.
- La macro `\chemfig` admet un argument optionnel qui sera passé à l'environnement `tikzpicture` dans lequel elle est dessinée
- Mise en place de la représentation des mécanismes réactionnels avec la syntaxe `"@{<nom>}"` devant un atome où `"@{<nom>,<coeff>}"` au tout début de l'argument d'une liaison. Cette syntaxe permet de placer un nœud (au sens de `tikz`) qui deviendra l'extrémité des flèches des mécanismes. Le tracé des flèches est faite par la macro `\chemmove` dont l'argument optionnel devient celui de l'environnement `tikzpicture` dans lequel sont faites les flèches.
- Pour le mécanisme d'alignement vertical via le `\vphantom`, la commande `\chemskipalign` permet d'ignorer le groupe d'atomes dans lequel elle est écrite.
- La commande `\chemname` permet d'afficher un nom sous une molécule. la commande `\chemnameinit` initialise la plus grande profondeur rencontrée.
- La commande `\lewis` a été modifiée de telle sorte que les dessins des décorations soient proportionnels à la taille de la police.

Version 0.2 du 31/08/2010

- Ajout de la documentation en anglais.
- Correction de bugs.
- `\printatom` est désormais une macro publique.

- Les espaces sont permis dans les molécules. Ils seront ignorés par défaut puisque les atomes sont composés en mode math par `\printatom`.
- Une paire de Lewis peut être représentée ”:”.
- Dans les cycles, une correction de la longueur du trait déporté des liaisons doubles est fait de telle sorte que si l’on écrit `\chemfig{*5(====)}`, on obtient deux polygones réguliers concentriques.
- La séquence de contrôle `\setnodelistyle` permet de spécifier le style des noeuds dessinés par tikz.

Version 0.1 du 23/06/2010

- Première version publique sur le CTAN