





```

\def\CF_warning#1{\immediate\write-1{Package \CFname\space Warn
ing: #1^^J}}
\def\CF_checkprimitive#1#2{% VÃrifie que #1 est une primitive
et sinon, Ãmet le message #2 et exÃcute \endinput
\beginingroup
\edef\__tempa{\meaning#1}\edef\__tempb{\string#1}
\expandafter
\endgroup
\ifx\__tempa\__tempb\else
\CF_error{#2}%
\def\CF_temp{\CFrestorecatcode\endinput}%
\expandafter\CF_temp
\fi
}
\CF_checkprimitive\eTeXversion{You are not using an eTeX engine
, \CFname\space cannot work.}
\CF_checkprimitive\expanded{the \string\expanded\space primitiv
e is not provided by your TeX engine, \CFname\space cannot w
ork.}

%----- Chargement simplekv -----
-----
\input simplekv.tex

%----- Chargement tikz -----
-----
\unless\ifdefined\tikzpicture
\beginingroup\def\CFtemp{\endgroup\input tikz.tex\relax}%
\expandafter\CFtemp
\fi
\usetikzlibrary{arrows.meta}

%----- Allocations -----
-----
\newcount\CF_cntatomgroup
\newcount\CF_cntgroup
\newcount\CF_cntatom
\newcount\CF_cntcycle
\newcount\CF_cntcyclebonds
\newcount\CF_cntcompound

\newif\ifCF_incycle
\newif\ifCF_cyclearc
\newif\ifCF_definesubmol
\newif\ifCF_adjustnamedp
\newif\ifCF_macrofixedbondlength

```

```

\newif\ifCF_compound_is_chemfig

\newdimen\CF_dim
\newdimen\CF_arrowsize
\newdimen\CF_zero \CF_zero=0pt

\newbox\CF_box
\newbox\CF_boxstuff
\newbox\CF_testbox
\newbox\CF_chargebox

\newtoks\CF_substoks

%----- Petites macros -----
-----
\let\CF_begintikzpicture\tikzpicture
\let\CF_endtikzpicture \endtikzpicture
\def\CF_quark{\CF_quark}
\def\CF_execfirst#1#2{#1}
\def\CF_execsecond#1#2{#2}
\def\CF_id#1{#1}
\def\CF_gobarg#1{}
\def\CF_gobtwoargs#1#2{}
\def\CF_firsttonil#1#2\_nil{#1}
\def\CF_sanitizelastitem#1,\empty#2\_nil{#1}
\def\CF_gobtikzinstruction#1;{}
\def\CF_makeother#1{\catcode'#1=12\relax}
\def\CF_lettoken#1#2{\let#1= #2}\CF_lettoken\CF_sptoken{ }
\def\CF_ifx#1#2{\ifx#1#2\expandafter\CF_execfirst\else
\expandafter\CF_execsecond\fi}
\def\CF_ifempty#1{\ifx\empty#1\empty\expandafter\CF_execfirst
\else\expandafter\CF_execsecond\fi}
\def\CF_ifnum#1{\ifnum#1\expandafter\CF_execfirst\else
\expandafter\CF_execsecond\fi}
\def\CF_ifinsidetikz{\ifdefined\pgfpictureid\expandafter\CF_exe
cfirst\else\expandafter\CF_execsecond\fi}
\def\CF_ifzerodim#1{%
\setbox\CF_testbox\hbox{\pgfinterruptpicture\printatom{#1}
\endpgfinterruptpicture}% bugfix 1.53
\CF_ifnum{1\ifdim\wd\CF_testbox=\CF_zero0\fi\ifdim\ht\CF_te
stbox=\CF_zero0\fi\ifdim\dp\CF_testbox=\CF_zero0\fi=1000 }
}
\def\CF_doifempty#1{\ifx\empty#1\empty\expandafter\CF_id\else
\expandafter\CF_gobarg\fi}
\def\CF_doifnotempty#1{\ifx\empty#1\empty\expandafter\CF_gobarg
\else\expandafter\CF_id\fi}

```

```

\def\CF_gobtonil#1\_nil{}
\def\CF_striplastsp#1{%
  \long\def\CF_stripsp##1##2{\expanded{\CF_stripsp_i\_marksp
##1\_nil\_marksp#1\_marksp\_nil{##2}}}%
  \long\def\CF_stripsp_i##1\_marksp#1##2\_marksp##3\_nil{\CF_
stripsp_ii##3##1##2\_nil#1\_nil\_nil}%
  \long\def\CF_stripsp_ii##1#1\_nil##2\_nil{\CF_stripsp_iii
##1##2\_nil}%
  \long\def\CF_stripsp_iii##1##2\_nil##3\_nil##4{\unexpanded
{##4{##2}}}%
  \long\def\CF_striplastsp##1##2{\expanded{\CF_striplastsp_i\
\_marksp##1\_nil#1\_nil\_nil{##2}}}%
  \long\def\CF_striplastsp_i##1#1\_nil##2\_nil{\CF_stripsp_i
ii##1\_nil\_nil}%
}\CF_striplastsp{ }
\edef\CFhash{\string#}
\begingroup
  \catcode'\_8
  \expandafter\gdef\csname CF\string_underscore\endcsname{ }
\endgroup
\def\CF_threera{ \expandafter\expandafter\expandafter}
\def\CF_exptwomacroargs#1#2#3{\CF_expsecond{\CF_expsecond#1{#2}
}{#3}}
\def\CF_expthreemacroargs#1#2#3#4{\CF_expsecond{\CF_exptwomacro
args#1{#2}{#3}}{#4}}
\def\CF_swaparg#1#2{#2{#1}}
\def\CF_expsecond#1#2{\expandafter\CF_swaparg\expandafter{#2}{
#1}}% \CF_expsecond<{arg1}>{<arg2>} donne "<arg1>{*<arg2>}"
\def\CF_eexpsecond#1#2{\expandafter\expandafter\expandafter\CF_
swaparg\expandafter\expandafter\expandafter{#2}{#1}}% \CF_ee
xpsecond{<arg1>}{<arg2>} donne "<arg1>{**<arg2>}"
\def\CF_swapunbrace#1#2{#2#1}
\def\CF_expafter#1#2{\expandafter\CF_swapunbrace\expandafter{#2
}{#1}}% \CF_expafter{<arg1>}{<arg2>} donne "<arg1>{*<arg2>}"
\def\CF_eexpafter#1#2{\expandafter\expandafter\expandafter\CF_s
wapunbrace\expandafter\expandafter\expandafter{#2}{#1}}% \CF
_eexpafter{<arg1>}{<arg2>} donne "<arg1>{**<arg2>}"
\def\CF_addtomacro#1#2{\CF_expsecond{\def#1}{#1#2}}
\def\CF_eaddtomacro#1#2{\CF_expsecond{\CF_addtomacro#1}{#2}}
\def\CF_preaddtomacro#1#2{\CF_expsecond{\CF_preaddtomacro#1{#2
}}{#1}}
\def\CF_preaddtomacroa#1#2#3{\def#1{#2#3}}
\def\CF_addtotoks#1#2{#1\expandafter{\the#1#2}}
\def\CF_eaddtotoks#1#2{\expandafter\CF_addtotoks\expandafter#1
\expandafter{#2}}
\def\CF_assigntonil#1#2\_nil{\def#1{#2}}

```

```

\def\CF_edefadddtomacro#1#2{\CF_expsecond{\CF_addtomacro#1}{
  \expanded{#2}}}
\def\CF_ifnextchar#1#2#3{%
  \let\CF_ifnextchartok=#1% <- espace indÃ©sirable, bugfix v1
  .31
  \def\CF_ifnextcharcodetrue{#2}%
  \def\CF_ifnextcharcodefalse{#3}%
  \futurelet\CF_temptok\CF_ifnextchara
}
\def\CF_ifnextchara{%
  \CF_ifx\CF_temptok\CF_sptoken
  {\CF_ifnextcharb
  }
  {\CF_ifx\CF_temptok\CF_ifnextchartok
  \CF_ifnextcharcodetrue
  \CF_ifnextcharcodefalse
  }%
}
\expandafter\def\expandafter\CF_ifnextcharb\space{\futurelet\CF
_temptok\CF_ifnextchara}
\def\CF_ifstar#1{\CF_ifnextchar*\CF_execfirst{#1}}
\def\CF_testopt#1#2{\CF_ifnextchar[#{1}{#1[#{2}]]}
\def\CF_ifinteger#1{%
  \begingroup
  \afterassignment\CF_afterinteger
  \CF_cntcyclebonds0#1\relax
}
\def\CF_afterinteger#1\relax{%
  \endgroup
  \CF_ifempty{#1}%
}
\def\CF_iffirsttokmatch#1#2{% est ce que #1 et #2 commencent pa
r les mÃªmes tokens ?
  \futurelet\CF_toksa\CF_gobtonil#1\relax\_nil
  \futurelet\CF_toksb\CF_gobtonil#2\relax\_nil
  \CF_ifx\CF_toksa\CF_toksb
}
\def\CF_iffirsttokin#1{% teste si le token qui commence #1 appa
rtient aux tokens mis dans #2
  \futurelet\CF_toksa\CF_gobtonil#1\relax\_nil
  \CF_iffirsttokina
}
\def\CF_iffirsttokina#1{%
  \CF_ifempty{#1}
  {\CF_execsecond
  }
}

```

```

        {\futurelet\CF_toksb\CF_gobtonil#1\relax\_nil
\CF_ifx\CF_toksa\CF_toksb
        {\CF_execfirst}
        {\CF_expsecond\CF_iffirsttokina{\CF_gobarg#1}}}%
    }%
}
\def\CF_ifinstr#1#2{%
    \def\CF_ifinstra##1#2##2\_nil{%
        \ifx\empty##2\empty
            \expandafter\CF_execsecond
        \else
            \expandafter\CF_execfirst
        \fi}%
    \CF_ifinstra#1\_nil#2\_nil
}
\def\CF_afterspace#1 #2\_nil{#2}

%----- Substitution -----
-----
\def\CF_ifstartswith#1#2{% #1=<texte> #2=<motif>
    \CF_ifempty{#1}%
        {\CF_execsecond
        }
        {\def\CF_startwithcode{#1}%
        \def\CF_startwithpattern{#2}%
        \CF_ifstartwitha
        }%
    }
\def\CF_ifstartwitha{%
    \CF_grabfirstarg\CF_startwithcode\CF_firstargcode
    \CF_grabfirstarg\CF_startwithpattern\CF_firstargpattern
    \CF_ifx\CF_firstargcode\CF_firstargpattern
        {\CF_expsecond\CF_ifempty\CF_startwithpattern
        {\CF_execfirst
        }
        {\CF_expsecond\CF_ifempty\CF_startwithcode
        \CF_execsecond
        \CF_ifstartwitha
        }%
        }
    {\CF_execsecond
    }%
}
\def\CF_grabfirstarg#1#2{%
    \CF_ifx#1\empty
        {\let#2\empty

```

```

    }
    {\def\CF_grabmacro{#2}%
    \CF_expsecond\CF_ifbracefirst#1%
    {\expandafter\CF_grabbracearg#1\_nil#1}
    {\CF_expafter{\futurelet\CF_nexttok\CF_grabfirstarg
a}#1\_nil#1}%
    }%
}
\def\CF_grabfirstarga{%
    \CF_ifx\CF_nexttok\CF_sptoken
    \CF_grabspacearg
    \CF_grabnormalarg
}
\def\CF_grabbracearg#1{%
    \expandafter\def\CF_grabmacro{{#1}}%
    \CF_grabargassigntonil\relax
}
\expandafter\def\expandafter\CF_grabspacearg\space{%
    \expandafter\def\CF_grabmacro{ }%
    \CF_grabargassigntonil\relax
}
\def\CF_grabnormalarg#1{%
    \expandafter\def\CF_grabmacro{#1}%
    \CF_grabargassigntonil\relax
}
\def\CF_grabargassigntonil#1\_nil#2{\CF_expsecond{\def#2}{\CF_g
obarg#1}}
\def\CF_ifbracefirst#1{\CF_ifnum{\catcode\CF_threearg'
    \expandafter\CF_firsttonil\detokenize{#1}\_nil=1 }}
\def\CF_substonly#1#2{% #1=entier maxi>0 #2=macro : dans la sc
#1, remplace tous les <motif> par <pattern> sauf lorsque le
motif est suivi d'un caract re >#1
    \def\CF_atendsubstitute{\edef#2{\the\CF_substtoks}}% macro
ex c te    la fin
    \let\CF_substnoglobs\CF_substnoglobonly
    \CF_ifnum{#1>0 }
    {\let\CF_testifx\empty
    \foreach\CF_x in {1,...,#1}
    {\xdef\CF_testifx{\unexpanded\expandafter{\CF_testi
fx}\unexpanded\expandafter{\expandafter\ifx\CF_x\CF_nexttok1
\fi}}}%
    \let\CF_testif\empty
    \foreach\CF_x in {1,...,#1}
    {\xdef\CF_testif{\unexpanded\expandafter{\CF_testif
}\unexpanded\expandafter{\expandafter\if\CF_x\CF_nexttok1\fi
}}}%
}

```

```

\CF_expsecond\CF_substi#2%
}
{\CF_substall#2%
}%
}
\def\CF_substnougrouponly{%
\CF_exptwomacroargs\CF_ifstartwith\CF_substcode\CF_substsub
st
{\CF_grabfirstarg\CF_substcode\CF_temp
\CF_expafter{\futurelet\CF_nexttok\CF_gobtonil}\CF_subst
code\relax\_nil
\CF_ifnum{0\CF_testifx=1 }% si le prochain token est \l
et-Ãlgal Ã 1...#1
{\edef\CF_nexttok{\CF_threeda\CF_firsttonil
\expandafter\string\CF_substcode\_nil}% le dÃltokÃlniser
\CF_ifnum{0\CF_testif=1 }
{\CF_eaddtotoks\CF_substtoks\CF_temp
\CF_grabfirstarg\CF_substcode\CF_temp
\CF_eaddtotoks\CF_substtoks\CF_temp
}
{\CF_eaddtotoks\CF_substtoks\CF_substpattern
}%
}
{\CF_eaddtotoks\CF_substtoks\CF_substpattern
}%
\CF_substgroups
}
{\CF_expsecond\CF_ifempty\CF_substcode
{\CF_atendsubstitute
}
{\CF_grabfirstarg\CF_substcode\CF_substauxarg
\CF_eaddtotoks\CF_substtoks\CF_substauxarg
\CF_substgroups
}%
}%
}
\def\CF_substall#1{% #1=macro
\def\CF_atendsubstitute{\edef#1{\the\CF_substtoks}}% macro
exÃlcutÃle Ã la fin
\let\CF_substnougrouponly\CF_substnougrouppall
\CF_expsecond\CF_substi#1%
}
\def\CF_substnougrouppall{%
\CF_exptwomacroargs\CF_ifstartwith\CF_substcode\CF_substsub
st
{\CF_eaddtotoks\CF_substtoks\CF_substpattern

```

```

\CF_grabfirstarg\CF_substcode\CF_temp
\CF_substgroups
}
{\CF_expsecond\CF_ifempty\CF_substcode
{\CF_atendsubstitute
}
{\CF_grabfirstarg\CF_substcode\CF_substauxarg
\CF_eaddtotoks\CF_substtoks\CF_substauxarg
\CF_substgroups
}%
}%
}
\def\CF_substi#1#2#3{% #1=<texte> #2=<motif> #3=<motif de subst
i>
\def\CF_substcode{#1}\def\CF_substsubst{#2}\def\CF_substpat
tern{#3}%
\CF_substtoks={}
\CF_substgroups
}

\def\CF_substgroups{%
\CF_expsecond\CF_ifbracefirst\CF_substcode
{\CF_grabfirstarg\CF_substcode\CF_substauxarg
\beginingroup
\def\CF_atendsubstitute{%
\expandafter\endgroup\expandafter\CF_addtotoks
\expandafter\CF_substtoks\expandafter{\expandafter{\the\CF_s
ubsttoks}}%
\CF_substgroups
}%
\CF_substtoks{}}% initialiser Ã vide
\expandafter\def\expandafter\CF_substcode\CF_substa
uxarg
\CF_substgroups
}%
{\CF_substnogroups
}%
}

%----- ParamÃtres -----
-----
\def\CF_defifempty#1#2#3{\CF_ifempty{#2}}{\def#1{#3}}{\def#1{#2}
}}
\defKV[chemfig]{%
atom style = \def\CF_atomstyle
{#1},

```

```

chemfig style      = \def\CF_chemfigstyle
{#1},
cram width         = \CF_defifempty\CF_crambasewidth
{#1}{1.5ex},
cram dash width    = \CF_defifempty\CF_cramdashlength
{#1}{1pt},
cram dash sep      = \CF_defifempty\CF_cramdashsep
{#1}{2pt},
atom sep           = \CF_defifempty\CF_atomsep
{#1}{3em},
bond offset        = \CF_defifempty\CF_bondoffset
{#1}{2pt},
double bond sep    = \CF_defifempty\CF_doublesep
{#1}{2pt},
angle increment    = \CF_defifempty\CF_angleincrement
{#1}{45},
node style         = \def\CF_nodestyle
{#1},
bond style         = \def\CF_bondstyle
{#1},
cycle radius coeff = \CF_defifempty\CF_cyclerradiuscoeff
{#1}{0.75},
stack sep          = \CF_defifempty\CF_stacksep
{#1}{1.5pt},
compound style     = \def\CF_defaultcompoundstyle
{#1},
compound sep       = \CF_defifempty\CF_compoundsep
{#1}{5em},
arrow offset       = \CF_defifempty\CF_arrowoffset
{#1}{1em},
arrow angle        = \CF_defifempty\CF_arrowangle
{#1}{0},
arrow coeff        = \CF_defifempty\CF_arrowlength
{#1}{1},
arrow style        = \def\CF_defaultarrowstyle
{#1},
arrow double sep   = \CF_defifempty\CF_arrowdoublesep
{#1}{2pt},
arrow double coeff = \CF_defifempty\CF_arrowdoubleposstart
{#1}{0.6},
arrow label sep    = \CF_defifempty\CF_arrowlabelsep
{#1}{3pt},
arrow head         = \CF_defifempty\CF_arrowhead
{#1}{-CF},
+ sep left         = \CF_defifempty\CF_signspaceante
{#1}{0.5em},

```

```

+ sep right          = \CF_defifempty\CF_signspacepost
{#1}{0.5em},
+ vshift             = \CF_defifempty\CF_signvshift
{#1}{0pt}
}
\def\setchemfig{\setKV[chemfig]}
\def\resetchemfig{\restoreKV[chemfig]}
\setKVdefault[chemfig]{%
  atom style          = {},% code tikz mis Å la fin de every
  node/.style
  chemfig style       = {},% code tikz mis Å la fin de l'aru
  gment optionnel de \tikzpicture
  bond join           = false,
  fixed length        = false,
  cram rectangle      = false,
  cram width          = 1.5ex,
  cram dash width     = 1pt,
  cram dash sep       = 2pt,
  atom sep            = 3em,
  bond offset         = 2pt,
  double bond sep     = 2pt,
  angle increment     = 45,
  node style          = {},
  bond style          = {},
  cycle radius coeff  = 0.75,
  stack sep           = 1.5pt,
  autoreset cntcycle  = true,
  show cntcycle       = false,
  debug               =false,
  scheme debug        = false,
  compound style       = {},
  compound sep         = 5em,
  arrow offset        = 1em,
  arrow angle         = 0,
  arrow coeff         = 1,
  arrow style         = {},
  arrow double sep    = 2pt,
  arrow double coeff  = 0.6,
  arrow double harpoon= true,
  arrow label sep     = 3pt,
  arrow head          = -CF,
+ sep left           = 0.5em,
+ sep right          = 0.5em,
+ vshift             = 0pt,
}%

```



```

\CF_cntatomgroup=0% cherche le nombre d'arguments Ã l'en
tuels
}%
}

\def\CF_submoltestnxttoka{\futurelet\CF_toksa\CF_submoltestnxtt
okb}

\def\CF_submoltestnxttokb{%
  \if[\noexpand\CF_toksa\expandafter\CF_execfirst\else
  \expandafter\CF_execsecond\fi
  {\begingroup\CF_sanitizecatcode\CF_expsecond\CF_submolg
  rabopt{\CF_temp}}
  {\CF_expsecond{\def_submola}\CF_temp{}}}%
}

\def\CF_submolgrabopt#1[#2]#{\endgroup\def_submola{#1}{#2}}

\def\def_submola#1{% #1 nom
  \CF_ifnum{0\CF_ifnum{\CF_cntatomgroup<0 }1{\CF_ifnum{\CF_cn
  tatomgroup>9 }10}>0 }
  {\CF_error{Invalid number of arguments in submol
  \detokenize\expandafter{\string#1}. Defining it with 0 argum
  ent}}%
  \CF_cntatomgroup=0
  }
  {}%

  \ifcat\relax\expandafter\noexpand\CF_firsttonil#1\_nil
  \expandafter\CF_execfirst\else\expandafter\CF_execsecond\fi%
  si #1 est une sÃquence de contrÃle
  {\expandafter\ifdefined\CF_firsttonil#1\_nil
  \ifCF_definesubmol
    \CF_warning{the submol \expandafter\string\CF_f
    irsttonil#1\_nil\space is already defined, the previous defi
    nition is lost}%
  \fi
  \fi
  \begingroup
  \CF_sanitizecatcode
  \def_submolb{#1}%
  }
  {\ifcsname CF__#1\endcsname
  \ifCF_definesubmol
    \CF_warning{the submol "#1" is already defined,
    the previous definition is lost}%
  \fi

```

```

\fi
\begingroup
\CF_sanitizecatcode
\expandafter\def\submolb\csname CF__#1\endcsname
}%
}

\def\def_submolb#1#2#3{% #1 nom sous forme de macro, #2 = code
si liaison arrive de droite , #3 = code si liaison arrive de
gauche, \CF_cntatomgroup = nombre d'arguments
\def\CF_tempa{#2}\CF_doifnotempty{#2}{\CF_expsecond{\CF_exp
second\CF_substonly{\number\CF_cntatomgroup}\CF_tempa}{
\CFhash}{\CFhash}}%
\def\CF_tempb{#3}\CF_expsecond{\CF_expsecond\CF_substonly{
\number\CF_cntatomgroup}\CF_tempb}{\CFhash}{\CFhash}%
\CF_expsecond{\CF_expsecond{\def_submolc{#1}}\CF_tempa}\CF_
tempb
}

\def\def_submolc#1#2#3{% #1 nom sous forme de macro, #2 = code
si liaison arrive de droite , #3 = code si liaison arrive de
gauche, \CF_cntatomgroup = nombre d'arguments
\endgroup
\begingroup
\global\toks0{\gdef\CFthesubmol}%
\CF_ifnum{\CF_cntatomgroup>0 }
{\foreach\CF_x in {1,...,\CF_cntatomgroup}{%
\global\toks0\expandafter{\expanded{\the\toks0
\CFhash\CF_x}}%
}%
}
}%
\CF_sanitizecatcode \catcode'\#6 \endlinechar-1
\everyeof{\noexpand}%
\CF_ifempty{#2}%
{\scantokens\expandafter{\the\toks0{\empty#3}}}%
}%
{\scantokens\expandafter{%
\the\toks0{%
\expanded{% bugfix 1.52
\csname CF_exec%
\ifdim\csname CF_currentangle
\endcsname pt>90pt
\ifdim\csname CF_currentangle
\endcsname pt<270pt
first%

```

```

\else
    second%
\fi
\else
    second%
\fi
\endcsname
{\unexpanded{#2}}{\unexpanded{#3}}}%
}%
}%
}%
\endgroup
\let#1\CFthesubmol
}

\def\CF_seeknode#1#2#3{% cherche un noeud au d  but de #1 l'ass
    igne dans la sc #2 et met le reste dans #3
    \let#2\empty
    \def#3{#1}%
    \CF_iffirsttokmatch{#1}{ }%
        {\CF_expsecond\CF_seeknodea{\romannumeral-'\.\noexpand
            #1}#2#3}% ignore les espaces au d  but du groupe d'atome
            {\CF_seeknodea{#1}#2#3}%
    }

\def\CF_seeknodea#1#2#3{%
    \CF_ifempty{#1}%
        {\let#3\empty
        }
        {\futurelet\CF_toks\CF_gobtonil#1\relax\_\nil
        \CF_ifx\CF_toks\CF_sptoken
            {\CF_addtomacro#2{ }%
            \CF_expsecond\CF_seeknodea{\CF_afterspace#1\_\nil}#2
            #3%
            }%
            {\CF_ifx\CF_toks\bgroup
            {\CF_eaddtomacro#2{\expandafter{\CF_firsttonil
            #1\_\nil}}}%
            \CF_expsecond\CF_seeknodea{\CF_gobarg#1}#2#3%
            }%
            {\CF_ifx!\CF_toks% Bugfix v1.5
            {\def\CF_seeksubmloltemp{#1}%
            \CF_seeksubmola
            \CF_expsecond\CF_seeknodea\CF_seeksubmlolte
            mp#2#3%

```

```

    }
    {\CF_iffirsttokina{-(*<>~}%
      {\def#3{#1}%
      }%
      {\CF_eaddtomacro#2{\CF_firsttonil#1\_ni
1}%
      \CF_expsecond\CF_seeknodea{\CF_gobarg#1
}%
      }%
    }
  }%
}%
}%
}

% on sait que #1 commence par -,=,~,<,>. On analyse cette liais
on
% #2 reÃ§oit le type de liaisons (1 pour -, 2 pour =, 3 pour ~)
\def\CF_assignbondcode#1#2{%
  \futurelet\CF_toksa\CF_gobtonil#1\_nil
  \edef#2{%
    \ifx-\CF_toksa1\else
    \ifx=\CF_toksa2\else
    \ifx~\CF_toksa3\else
    \ifx>\CF_toksa4\else
    \ifx<\CF_toksa5\else0% si 0 --> il y a une erreur non d
ue Ã l'utilisateur
    \fi\fi\fi\fi\fi}%
  \ifnum#2>3 % si c'est une liaison de Cram
    \CF_expafter{\futurelet\CF_toksa\CF_gobtonil}{\CF_gobar
g#1\_nil}% chope le caractÃre suivant
    \CF_ifx:\CF_toksa
    {\edef#2{\number\numexpr#2+2}% si c 'est un ":", si
gne du pointillÃ, ajoute 2
    }%
    {\CF_ifx|\CF_toksa% si c 'est un "|", signe du tria
ngle Ã l'vidÃ, ajouter 4
    {\edef#2{\number\numexpr#2+4}}
    {}%
    }%
  \fi
}

\def\CF_grabbondoffseta#1,#2\_nil{%
  \def\CF_startoffset{#1}\def\CF_endoffset{#2}%
}

```

```

\def\CF_grabbondoffset#1(#2)#3\_nil{%
  \CF_doiifnotempty{#2}%
    {\CF_ifinstr{#2},%
      {\CF_grabbondoffseta#2\_nil}%
      {\def\CF_startoffset{#2}}%
    }%
  \def\CF_remainafterbond{#3}%
}

\def\CF_analysebond#1#2{%
  \CF_assignbondcode{#1}#2%
  \CF_expsecond{\def\CF_remainafterbond}{\CF_gobarg#1}%mange
le premier signe de la liaison
  \let\CF_doublebondtype\CF_zero
  \ifnum#2=2 % si c'est une double liaison, regarde s'il y a
un + ou - derri re
    \CF_expafter{\futurelet\CF_toksa\CF_gobtonil}{\CF_gobar
g#1\_nil}%
    \CF_ifx^\CF_toksa
      {\def\CF_doublebondtype{1}%
        \CF_expsecond{\def\CF_remainafterbond}{\CF_gobtwoar
gs#1}% mange le "^
        }
        {\expandafter\ifx\CF_underscore\CF_toksa
          \def\CF_doublebondtype{2}%
          \CF_expsecond{\def\CF_remainafterbond}{\CF_gobt
woargs#1}% mange le "_
          \fi
        }%
    \else
      \ifnum#2>5 % si c'est une liaison de Cram pointill e o
u triangle  vid 
        \CF_expsecond{\def\CF_remainafterbond}{\CF_gobtwoar
gs#1}% mange un caract re de plus
        \fi
      \fi
      \CF_expsecond\CF_iffirsttokmatch\CF_remainafterbond\CFhash
        {\CF_eexpsecond\CF_iffirsttokmatch{\expandafter\CF_goba
rg\CF_remainafterbond.}%si parenth se juste apr s
          {\expandafter\CF_grabbondoffset\CF_remainafterbond\
_nil}
        {}%
      }%
      {}%
    \CF_expsecond\CF_iffirsttokmatch\CF_remainafterbond @%

```

```

        {\expandafter\CF_grabmovearg\CF_remainafterbond\_nil}%
    }%
\CF_expsecond\CF_iffirsttokmatch{\CF_remainafterbond}[%
    {\expandafter\CF_analyseoptarg\CF_remainafterbond\_nil
\CF_remainafterbond
    }%
    {\let\CF_currentstringangle\CF_defaultstringangle
    \let\CF_currentlength\CF_defaultlength
    \let\CF_currentfromatom\CF_defaultfromatom
    \let\CF_currenttoatom\CF_defaulttoatom
    \let\CF_currenttikz\CF_defaulttikz
    \let\CF_movebondname\empty
    }%
\ifCF_incycle
    \pgfmathsetmacro\CF_cycleincrementangle{360/\CF_cyclenu
m+\CF_initcycleangle}%
    \edef\CF_currentstringangle{::+\CF_cycleincrementangle}
%
    \def\CF_initcycleangle{0}%
    \let\CF_currentlength\CF_defaultlength% et on ignore la
longueur de liaison spÅcifiÅe
\fi
\CF_expsecond\CF_setbondangle{\CF_currentstringangle}\CF_cu
rrentangle
}

\def\CF_setbondangle#1#2{% le code de la direction est contenu
dans #1, en sortie, #2 contient l'angle
\CF_ifempty{#1}%
    {\let#2\CF_defaultangle
    }
    {\if:\expandafter\noexpand\CF_firsttonil#1\_nil
        \if:\CF_threeea\noexpand\expandafter\CF_firsttonil
\CF_gobarg#1\_nil
            \pgfmathsetmacro#2{\CF_previousangle+
\expandafter\CF_gobarg\CF_gobarg#1}%
        \else
            \pgfmathsetmacro#2{\CF_gobarg#1}%
        \fi
    \else
        \pgfmathsetmacro#2{#1*\CF_angleincrement}%
        \fi% puis normalise l'angle entre 0 et 360
        \ifdim\ifdim#2pt<0pt -\fi#2pt>360pt
            \pgfmathsetmacro#2{#2-360*floor(#2/360)}%
        \fi% si |#2|>360
        \ifdim#2pt<0pt

```

```

\pgfmathsetmacro#2{#2+360}%
\fi
}%

\def\CF_analysemovearg#1,#2\_nil#3{%
  \def#3{#1}\def\CF_movebondcoeff{#2}%
}

% Argument limitÃs lÃgites ici car #2 (qui est ce qui suit
% "@{<nom>}" dans l'argument optionnel) ne DOIT PAS
% commencer par une accolade.
\def\CF_grabmovearg @#1#2\_nil{%
  \CF_ifinstr{#1},%
    {\CF_analysemovearg#1\_nil\CF_movebondname}%
    {\def\CF_movebondname{#1}\def\CF_movebondcoeff{0.5}}%
  \def\CF_remainoptarg{#2}%
}

\def\CF_testemptyandassign#1#2#3{%
  \CF_ifempty{#2}
    {\let#1#3}
    {\def#1{#2}}%
}

\def\CF_parseoptlist#1,#2,#3,#4,#5\_nil{%
  \CF_testemptyandassign\CF_currentstringangle{#1}\CF_default
stringangle
  \CF_testemptyandassign\CF_currentlength      {#2}\CF_default
length
  \CF_testemptyandassign\CF_currentfromatom    {#3}\CF_default
fromatom
  \CF_testemptyandassign\CF_currenttoatom      {#4}\CF_default
toatom
  \CF_expsecond{\CF_testemptyandassign\CF_currenttikz}{\CF_sa
nitizelastitem#5,\empty\_nil}\CF_defaulttikz
}%

\def\CF_analyseoptarg[#1]{%
  \CF_doifnotempty{#1}%
    {\CF_iffirsttokmatch{#1}{@}%
      {\CF_grabmovearg#1\_nil
      }
      {\let\CF_movebondname\empty
      \def\CF_remainoptarg{#1}%
      }%
    }%
}

```

```

        \expandafter\CF_parseoptlist\CF_remainoptarg,\empty,
        \empty,\empty,\empty\_nil
    }%
    \CF_analyseoptarga\relax
}

\def\CF_analyseoptarga#1\_nil#2{%
    \CF_expsecond{\def#2}{\CF_gobarg#1}%
}

\def\CF_seeksubmol#1#2{% cherche et remplace ! au début de #1.
    #1=code #2=macro recevant le résultat
    \def\CF_seeksubmoltemp{#1}%
    \CF_seeksubmola
    \let#2\CF_seeksubmoltemp
}

\def\CF_seeksubmola{%
    \CF_expsecond{\def\CF_seeksubmoltemp}{\romannumeral-'\.
    \expandafter\noexpand\CF_seeksubmoltemp}%
    \CF_expsecond\CF_iffirsttokmatch{\CF_seeksubmoltemp}!%
        {\CF_eexpsecond{\def\CF_seeksubmoltemp}{\expandafter
        \CF_gobarg\CF_seeksubmoltemp}% enlève le "!"
        \CF_ifx\empty\CF_seeksubmoltemp
        {\CF_error{no submol name found after "!"}}
        }%
    \ifcat\relax\CF_threeta\noexpand\expandafter\CF_firstto
    nil\CF_seeksubmoltemp*\_nil
        \expandafter\CF_seeksubmolb\CF_seeksubmoltemp\_nil
    \else
        \expandafter\CF_seeksubmolc\CF_seeksubmoltemp\_nil
    \fi
    \CF_seeksubmola
    }%
    {}%
}

\def\CF_seeksubmolb#1{\CF_seeksubmold#1\relax}

\def\CF_seeksubmolc#1{\expandafter\CF_seeksubmold\csname CF__#1
\endcsname\relax}% nom de la sous molécule

\def\CF_seeksubmold#1#2\_nil{%#1=macro de la sous molécule #2=
    reste du code commençant par \relax
    \CF_expsecond{\CF_eexpsecond{\def\CF_seeksubmoltemp}}{
    \expandafter#1\CF_gobarg#2}% supprime le \relax puis ajoute

```

```

    la macro au d  but et la 2-d  veloppe
}

\def\CF_insertemptygroup#1{% ins  ire {} au d  but de la sc #1
    \CF_expsecond{\def#1}{\expandafter{\expandafter}#1}%
}

\def\chemfig{\CF_testopt\CF_chemfiga{}}

\def\CF_chemfiga[#1]{%
    \begingroup
        \CF_sanitizecatcode
        \CF_exptwomacroargs\CF_chemfigb\CF_begintikzpicture\CF_
        endtikzpicture[#1]%
    }

\def\CF_chemfigb#1#2[#3]#4{%
    \endgroup
    \begingroup
        \setchemfig{#3}%
        \CF_ifinsidetikz
            {\pgfinterruptpicture
                \let\CF_atendofchemfig\endpgfinterruptpicture
            }
            {\let\CF_atendofchemfig\relax
            }%
        \expanded{% d  but du tikzpicture
            \unexpanded{#1}[%
                remember picture,%
                every node/.style={%
                    anchor=base,%
                    inner sep=0pt,%
                    outer sep=0pt,%
                    minimum size=0pt,%
                    \unexpanded\expandafter{\CF_atomstyle}%
                },%
                baseline,%
                \unexpanded\expandafter{\CF_chemfigstyle}%
            ]%
        }%
        \begingroup% \endgroup rajout   en sortie de trac  
    par \CF_chemfigd
        \let\CF_hooklist\empty
        \ifboolKV[chemfig]{fixed length}
            {\CF_macrofixedbondlengthtrue}
            {\CF_macrofixedbondlengthfalse}%

```

```

\ifboolKV[chemfig]{bond join}
  {\let\CF_drawaxisbond\CF_drawaxisbondjoin}
  {\let\CF_drawaxisbond\CF_drawaxisbondnojoin}
}%

\ifboolKV[chemfig]{cram rectangle}
  {\let\CF_clipcramornot\CF_gobtikzinstructio
n}

  {\let\CF_clipcramornot\clip}%
\CF_incyclefalse
\CF_cntgroup0
\ifboolKV[chemfig]{autoreset cntcycle}
  {\global\CF_cntcycle0 }
  {}%

\let\CF_lastaction\CF_zero% 0=dÃtbut du dessin
1=tracÃt d'un noeud 2=tracÃt d'une liaison
\let\CF_startoffset\empty
\let\CF_endoffset\empty
\let\CF_bondoutcontentsaved\empty
\def\CF_cycleanglecorrection{180/\CF_cyclenum}%
\def\CF_defaultangle{0}%
\def\CF_defaultstringangle{:0}% angle pris par
dÃtfaut si le champ est vide
\def\CF_defaultlength{1}%
\let\CF_defaultfromatom\empty% numero de l'atom
e d'oÃt partent les liaisons par dÃtfaut
\let\CF_defaulttoatom\empty% numÃtro de l'atome
oÃt arrivent les liaisons par dÃtfaut
\let\CF_defaulttikz\empty
\let\CF_previousbondangle\empty
\let\CF_joinbond\CF_zero
\let\CF_previoustikz\empty
\everyeof{\_nil}\endlinechar-1
\CF_sanitizeatcode
\expandafter\CF_assignntonil\expandafter\CF_mole
cule\scantokens{#4}%
\CF_expsecond{\CF_substall\CF_molecule}\CFhash
\CFhash
\CF_expsecond\CF_chemfigc{\CF_molecule}%
%\endgroup <-- rajoutÃt par \CF_chemfigd
#2% fin du tikzpicture
\CF_atendofchemfig
\endgroup
\let\CF_flipstate\CF_zero
}

\def\CF_chemfigc#1{% #1 est le code de la molÃtcle

```

```

\ifnum\CF_lastaction=3
  \ifCF_incycle
    \def\CF_defaultangle{0}%
  \else
    \ifnum\CF_cntcyclebonds=0 % si c'est le d  but d'un
cycle
      \pgfmathsetmacro\CF_defaultangle{\CF_previousangle+180+\CF_cycleanglecorrection}% on met la liaison    +180   + correction
    \else
      \pgfmathsetmacro\CF_defaultangle{\CF_previousangle-90+180/\CF_cyclenum}% sinon    la bissectrice du sommet du cycle
    \fi
  \fi
  \let\CF_defaultstringangle\empty
\fi
\let\CF_currentangle\CF_defaultangle
\def\CF_molecule{#1}%
\CF_expsecond\CF_seeksubmol\CF_molecule\CF_molecule% alias en premier ?
\if[\CF_threeda\noexpand\expandafter\CF_firsttonil\CF_molecule\_nil
  \expandafter\CF_analyseoptarg\CF_molecule\_nil\CF_molecule
  \CF_expsecond\CF_setbondangle{\CF_currentstringangle}
\CF_currentangle
  \let\CF_defaultangle\CF_currentangle
  \let\CF_previousangle\CF_currentangle
  \CF_doifnotempty\CF_currentstringangle{\let\CF_defaultangle\CF_currentangle}%
  \CF_doifnotempty\CF_currentlength      {\let\CF_defaultlength\CF_currentlength}%
  \CF_doifnotempty\CF_currentfromatom    {\let\CF_defaultfromatom\CF_currentfromatom}%
  \CF_doifnotempty\CF_currenttoatom      {\let\CF_defaulttoatom\CF_currenttoatom}%
  \CF_doifnotempty\CF_currenttikz        {\let\CF_defaulttikz\CF_currenttikz}%
  \CF_expsecond\CF_seeksubmol\CF_molecule\CF_molecule
\fi
\edef\CF_defaultstringangle{:\CF_defaultangle}%
\let\CF_currentlength\CF_defaultlength
\let\CF_currentfromatom\CF_defaultfromatom
\let\CF_currenttoatom\CF_defaulttoatom
\let\CF_currenttikz\CF_defaulttikz

```

```

\ifCF_incycle% si on commence un cycle
  \let\CF_currentangle\CF_previousangle
  \pgfmathsetmacro\CF_cyclearcinitangle{\CF_currentangle+
\CF_initcycleangle+180/\CF_cyclenum+90}%
  \pgfmathsetmacro\CF_centeroffset{\CF_currentlength*\CF_
atomsep/(2*sin(180/\CF_cyclenum))}%
  \node[at=(\CF_bondoutnode),shift=(\CF_cyclearcinitangle
:\CF_centeroffset pt),anchor=center](cyclecenter\number\CF_
cntcycle){};% le centre du cycle
  \ifboolKV[chemfig]{show cntcycle}
    {\node[at=(cyclecenter\number\CF_cntcycle),anchor=c
enter,overlay]{\tiny\number\CF_cntcycle};}
    {}%
  \ifCF_cyclearc% on doit tracer l'arc de cercle dans le
cycle ?
    \pgfmathsetmacro\CF_cyclearcradius{\CF_cyclerradiusc
oeff*\CF_currentlength*\CF_atomsep/(2*tan(180/\CF_cyclenum))
}%
    \node[at=(cyclecenter\number\CF_cntcycle),shift=(
\CF_cyclearcstartangle:\CF_cyclearcradius pt)](initarc){};%
    le début de l'arc
    \CF_expafter{\draw[\CF_cyclearcdirecttikz](initarc
) arc (\CF_cyclearcstartangle:\CF_cyclearcendangle:\CF_cycle
arcradius pt);%
    \fi
  \else
    \let\CF_currentangle\CF_defaultangle
  \fi
  \ifnum\CF_lastaction=0
    \let\CF_previousangle\CF_defaultangle
    \node(CF_node){};
    \CF_expsecond\CF_iffirsttokin{\CF_molecule}{-(*~?<>)%
    {\CF_insertemptygroup\CF_molecule}%
    {}%
  \fi
  \CF_chemfigd
}

\def\CF_chemfigd{%
  \let\CF_nextaction\CF_chemfigd% Ã priori, on reboucle
  \CF_ifx\CF_molecule\empty
    {\let\CF_nextaction\endgroup
    }
    {\CF_expsecond\CF_seeknode{\CF_molecule}\CF_currentatom
group\CF_molecule
  \CF_ifx\empty\CF_currentatomgroup% pas de noeud pour co

```

```

mmencer ?
    {\def\CF_bondoutnode{%
      n\CF_lastgroupnumber-%
      \ifx\CF_currentfromatom\empty
        \ifdim\CF_currentangle pt<90pt
          \number\CF_cntatomgroup
        \else
          \ifdim\CF_currentangle pt>270pt
            \number\CF_cntatomgroup
          \else
            1%
          \fi
        \fi
      \else
        \CF_currentfromatom
      \fi}%
    \CF_expafter{\futurelet\CF_toks\CF_gobtonil}{\CF_m
olecule\relax\_nil}%
    \CF_iffirsttokina{<~}% la suite est une liaison
    {\ifnum\CF_lastaction=2 % c'est la deuxi me li
aion cons cutive ?
      \CF_insertemptygroup\CF_molecule% ins re u
n groupe vide
      \edef\CF_bondoutnode{\CF_bondoutnode}%
    \else
      \ifCF_incycle
        \advance\CF_cntcyclebonds1
      \fi
      \CF_expsecond\CF_analysebond{\CF_molecule}
\CF_bondtype
      \edef\CF_bondoutnode{\CF_bondoutnode}%
      \let\CF_molecule\CF_remainafterbond
      \ifCF_incycle
        \ifnum\CF_cntcyclebonds=\CF_cyclenum
\relax
          \CF_threeda\CF_execfirst
        \else
          \ifnum\CF_cntcyclebonds=1
            \let\CF_cyclefirsttikz\CF_curre
nttikz
            \CF_doifnotempty\CF_startoffset
{\let\CF_cyclejoinlast\CF_zero}%
          \fi
          \CF_threeda\CF_execsecond
        \fi
      \else

```

```

\expandafter\CF_execsecond
\fi
{\let\CF_nextaction\endgroup
\CF_drawbond\CF_bondtype{\CF_bondoutnod
e}{\CF_hookcycle}\CF_previousatomgroup\CF_hookatomgroup
}%
{\node[at=(\CF_bondoutnode\ifCF_incycle
\else\ifCF_macrofixedbondlength.\CF_currentangle\fi\fi),shif
t=(\ifcase\CF_flipstate\or180-\or-\fi\CF_currentangle:\CF_cu
rrentlength*\CF_atomsep)](CF_node){};
\let\CF_previousangle\CF_currentangle
\def\CF_lastaction{2}%
}%
\fi
\ifcat\relax\detokenize\expandafter{
\romannumeral-'\.\expandafter\noexpand\CF_molecule}\relax
% s'il ne reste plus rien après la liaison (sa
ns tenir compte de l'espace devant)-> insérer un groupe vide
\CF_insertemptygroup\CF_molecule
\fi
}%
{\edef\CF_bondoutnode{\CF_bondoutnode}% À la value
le l'atome de départ de liaison
\CF_ifx(\CF_toksa% une parenthèse pour commenc
er ?
{\ifnum\CF_lastaction=2 % il y avait une li
aison juste avant ?
\CF_insertemptygroup\CF_molecule
\else
\CF_expsecond\CF_grabsubmol{\CF_molecul
e}%
\begingroup
\ifCF_incycle\def\CF_lastaction{3}
\fi% on était dans un cycle
\CF_incyclefalse
\aftergroup\CF_chemfigd
\def\CF_nextaction{\CF_expsecond\CF
_chemfigc{\CF_molinparen}}}%
\fi
}%
{\CF_ifx\CF_molecule\empty
{\let\CF_nextaction\endgroup
}
}% ce qui reste après le noeud courant
n'est pas vide, ne commence pas par "--~", ni par une paren
thèse

```

```

\CF_ifx*\CF_toksa% un cycle ?
{\ifnum\CF_lastaction=2
\CF_insertemptygroup\CF_molecul
e% ins  tre un groupe vide
\else
\ifCF_incycle
\def\CF_lastaction{3}%
\fi% on   tait dans un cycle
\ifnum\CF_lastaction=3
\let\CF_lastcyclex\CF_cyc
lex
\fi
\CF_eexpsecond\CF_iffirsttokmat
ch{\expandafter\CF_gobarg\CF_molecule}%
{\CF_eexpsecond{\def\CF_mol
ecule}{\expandafter\CF_gobarg\CF_molecule}% enl  ve la 1er   
toile
\CF_eexpsecond\CF_iffirstto
kmatch{\expandafter\CF_gobarg\CF_molecule}{% un crochet ensu
ite ?
{\expandafter\CF_cyclep
arsepreamblewithoptarg\CF_molecule\_nil% \begin{group} inclus
}%
{\def\CF_cyclearcstarta
ngle{0}\def\CF_cyclearcendangle{360}%
\let\CF_cyclearcdirectt
tikz\empty
\expandafter\CF_cyclepa
rsepreable\CF_molecule\_nil% \begin{group} inclus
}%
\CF_cyclearctrue
}%
{\expandafter\CF_cyclepa
rsepreable\CF_molecule\_nil% \begin{group} inclus
\CF_cyclearcfalse
}%
\CF_cntcyclebonds0
\edef\CF_hookcycle{\CF_bondoutn
ode}%
\let\CF_hookatomgroup\CF_previo
usatomgroup
\CF_ifzerodim\CF_hookatomgroup
{\def\CF_cyclejoinlast{1}}%
joindre le dernier
{\def\CF_cyclejoinlast{0}}%
\CF_incycletrue

```

```

\global\advance\CF_cntcycle1
\ifnum\CF_lastaction=3
\pgfmacthsetmacro\CF_initcyc
leangle{360/\CF_lastcyclenum-180}% c'est un cycle dans un cy
cle
\else
\pgfmacthsetmacro\CF_initcyc
leangle{-180/\CF_cyclenum-90+\CF_cycleanglecorrection}%
\fi
\aftergroup\CF_chemfigd
\def\CF_nextaction{\CF_expsecon
d\CF_chemfigc{\CF_molinparen}}}%
\fi
}%
{\CF_error{something went wrong her
e: \detokenize\expandafter{\CF_molecule}^^}If you think it's
a bug, please, send a Minimal Example to the author}%
}%
}%
}%
}%
{\CF_expthreemacroargs\CF_drawatomgroup\CF_currenta
ngle\CF_currenttoatom\CF_currentatomgroup
}%
}%
\CF_nextaction
}

\def\CF_cycleparsepreamble*#1#2\_nil{%
\ifnum#1<3
\CF_error{a cycle must be at least a triangle.^^}The nu
mber following "*" must be 3 or more}%
\fi
\def\CF_molecule{#2}%
\CF_expsecond\CF_grabsubmol{\CF_molecule}%
\beginingroup
\def\CF_cyclenum{#1}%
}

\def\CF_cycleparsepreamblewithoptarg*[#1]#2#3\_nil{%
\CF_cycleparseoptarg#1,\empty,\empty,\empty\_nil
\CF_cycleparsepreamble*#2#3\_nil
}

\def\CF_cycleparseoptarg#1,#2,#3\_nil{%

```

```

\CF_ifempty{#1}
  {\def\CF_cyclearcstartangle{0}}
  {\def\CF_cyclearcstartangle{#1}}%
\CF_ifempty{#2}
  {\def\CF_cyclearcendangle{360}}
  {\def\CF_cyclearcendangle{#2}}%
\CF_expsecond{\def\CF_cyclearcdirecttikz}{\CF_sanitizelastitem#3,\empty\_nil}%
}

\def\CF_grabsubmol#1{%
  \beginingroup
    \catcode'(1 \catcode')2
    \CF_threearc
  \endgroup
  \expandafter\CF_grabsubmola\scantokens{\relax#1}%
}

\def\CF_grabsubmola#1\_nil{%
  \expandafter\CF_assigntonil\expandafter\CF_molecule
  \scantokens\CF_threearc{\expandafter\CF_gobarg \CF_gobarg
#1}%
  \expandafter\CF_assigntonil\expandafter\CF_molinparen
  \scantokens\CF_threearc{\expandafter\CF_firsttonil\CF_gobarg
#1\_nil}%
}

\def\CF_ifcarisupperletter#1{%
  \ifcat\relax\noexpand#1%
    \let\CF_next\CF_execsecond% faux si c'est une sc
  \else
    \ifnum'#1<'A
      \let\CF_next\CF_execsecond
    \else
      \ifnum'#1>'Z
        \let\CF_next\CF_execsecond
      \else
        \let\CF_next\CF_execfirst
      \fi
    \fi
  \fi
  \CF_next
}

% CrÃÃl 4 noeuds au dessus et au dessous des noeuds #1 et #2
% Ã une distance de #3 du noeud #1 et #4 du noeud #2

```

```

\def\CF_createnormnodes#1#2#3#4{%
  \CF_doi fnotempty{#3}
    {\node[shape=coordinate,at=(#1),xshift=#3*\CF_normx,ysh
ift=#3*\CF_normy](#11){};
    \node[shape=coordinate,at=(#1),xshift=-#3*\CF_normx,ysh
ift=-#3*\CF_normy](#12){};
    }%
  \CF_doi fnotempty{#4}
    {\node[shape=coordinate,at=(#2),xshift=#4*\CF_normx,ysh
ift=#4*\CF_normy](#21){};
    \node[shape=coordinate,at=(#2),xshift=-#4*\CF_normx,ysh
ift=-#4*\CF_normy](#22){};
    }%
}

\def\CF_distancebetweenpoints#1#2#3#4#5{%
  \pgfextractx\CF_dim{\pgfpointanchor{#1}{#2}}\edef\CF_dimax{
\the\CF_dim}%
  \pgfextracty\CF_dim{\pgfpointanchor{#1}{#2}}\edef\CF_dimay{
\the\CF_dim}%
  \pgfextractx\CF_dim{\pgfpointanchor{#3}{#4}}\edef\CF_dimbx{
\the\CF_dim}%
  \pgfextracty\CF_dim{\pgfpointanchor{#3}{#4}}\edef\CF_dimby{
\the\CF_dim}%
  \pgfmathsetmacro#5{veclen(\CF_dimbx-\CF_dimax,\CF_dimby-\CF
_dimay)}%
}

\def\CF_computenodevect#1#2{%
  \CF_distancebetweenpoints{#1}{center}{#2}{center}\CF_vector
len
  \pgfmathsetmacro\CF_normx{(\CF_dimay-\CF_dimby)/\CF_vectorl
en}%
  \pgfmathsetmacro\CF_normy{(\CF_dimbx-\CF_dimax)/\CF_vectorl
en}%
}

\def\CF_setoffset#1#2{%
  \CF_doi fempty#1{%
    \CF_ifzerodim{#2}
      {\def#1{0pt}}
      {\edef#1{\CF_bondoffset}}%
    }%
}

\def\CF_drawbond#1#2#3#4#5{% #1=type de liaison #2 et #3:nom de

```

```

    noeuds de d'Albut et fin #4 et #5: contenu des atomes de d'Al
but et fin
\CF_setoffset\CF_startoffset#4%
\CF_setoffset\CF_endoffset#5%
\let\CF_currentbondstyle\CF_bondstyle
\CF_doifnotempty\CF_currenttikz{\CF_eaddtomacro\CF_currentb
ondstyle{\expandafter,\CF_currenttikz}}%
\path(#2)--(#3)coordinate[pos=0](#2@)coordinate[pos=1](#3@)
;%
\CF_computenodevect{#2@}{#3@}%
\pgfmathsetmacro\CF_startcoeff{\CF_startoffset/\CF_vectorle
n}%
\pgfmathsetmacro\CF_endcoeff{1-\CF_endoffset/\CF_vectorlen}
%
\path(#2@)--(#3@)coordinate[pos=\CF_startcoeff](#2@@)coordi
nate[pos=\CF_endcoeff](#3@@);%
\CF_doifnotempty\CF_movebondname% on doit poser un noeud su
r la liaison
    {\path(#2@@)--(#3@@)coordinate[overlay,pos=\CF_movebond
coeff](\CF_movebondname);
    \let\CF_movebondname\empty
    }%
\ifcase#1\relax
    \CF_error{unknown bond type, this error should not occu
r^^}If you think it's a bug, send a Minimal Example to the a
uthor}%
\or% 1 = liaison simple
    \CF_drawaxisbond{#2}{#3}% trace la liaison simple dans
l'axe
\or% 2 = liaison double
    \ifCF_incycle
        \ifnum\CF_doublebondtype=0
            \def\CF_doublebondtype{1}%
        \fi
        \ifnum\CF_flipstate>0
            \def\CF_doublebondtype{2}%
        \fi
        \pgfmathsetmacro\CF_doublebondlengthcorrection{\CF_
doublesep*tan(180/\CF_cyclenum)}%
        \fi
        \ifcase\CF_doublebondtype
            \CF_createnormnodes{#2@@}{#3@@}{\CF_doublesep/2}{
\CF_doublesep/2}%
            \CF_drawbonda(#2@@1)--(#3@@1);
            \CF_drawbonda(#2@@2)--(#3@@2);
            \let\CF_joinbond\CF_zero

```

```

\or
\CF_createnormnodes{#2@@}{#3@@}\CF_doublesep\CF_doublesep
\CF_drawaxisbond{#2}{#3}% trace la liaison simple dans l'axe\CF_drawbonda(#2@@)--(#3@@);
\beginngroup% ajuste Ã l'eventuellement les longueurs des liaisons doubles
\ifCF_incycle
\ifdim\CF_startoffset=0pt
\CF_edefadddtomacro\CF_currentbondstyle{
,shorten <=\CF_doublebondlengthcorrection pt}%
\fi
\ifdim\CF_endoffset=0pt
\CF_edefadddtomacro\CF_currentbondstyle{
,shorten >=\CF_doublebondlengthcorrection pt}%
\fi
\fi
\CF_drawbonda(#2@@1)--(#3@@1);
\endgroup
\or
\CF_createnormnodes{#2@@}{#3@@}\CF_doublesep\CF_doublesep
\CF_drawaxisbond{#2}{#3}% trace la liaison simple dans l'axe\CF_drawbonda(#2@@)--(#3@@);
\beginngroup% ajuste Ã l'eventuellement les longueurs des liaisons doubles
\ifCF_incycle
\ifdim\CF_startoffset=0pt
\CF_edefadddtomacro\CF_currentbondstyle{
,shorten \ifnum\CF_flipstate=0 <=\else>=\fi\CF_doublebondlengthcorrection pt}%
\fi
\ifdim\CF_endoffset=0pt
\CF_edefadddtomacro\CF_currentbondstyle{
,shorten \ifnum\CF_flipstate=0 >=\else<=\fi\CF_doublebondlengthcorrection pt}%
\fi
\fi
\CF_drawbonda(#2@@2)--(#3@@2);
\endgroup
\fi
\or% 3 = liaison triple
\CF_createnormnodes{#2@@}{#3@@}\CF_doublesep\CF_doublesep
\CF_drawaxisbond{#2}{#3}% trace la liaison simple dans l'axe\CF_drawbonda(#2@@)--(#3@@);

```

```

\CF_drawbonda(#2@@1)--(#3@@1);
\CF_drawbonda(#2@@2)--(#3@@2);
\or% 4 = liaison Cram pleine de #2 vers #3
\CF_createnormnodes{#2@@}{#3@@}{\CF_crambasewidth/2}{}%
\CF_expafter{\filldraw[]\CF_currentbondstyle,line join=
bevel](#2@@1)--(#2@@2)--(#3@@)--cycle;
\let\CF_joinbond\CF_zero
\or% 5 = liaison Cram creuse de #3 vers #2
\CF_createnormnodes{#3@@}{#2@@}{\CF_crambasewidth/2}{}%
\CF_expafter{\filldraw[]\CF_currentbondstyle,line join=
bevel](#3@@1)--(#3@@2)--(#2@@)--cycle;
\let\CF_joinbond\CF_zero
\or% 6 = liaison Cram pointill  te de #2 vers #3
\scope
\CF_createnormnodes{#2@@}{#3@@}{\CF_crambasewidth/2
}{}%
\CF_clipcramornot(#2@@1)--(#2@@2)--(#3@@)--(#2@@1);
\CF_expafter{\draw[]\CF_currentbondstyle,dash patte
rn=on \CF_cramdashlength off \CF_cramdashsep,line width=\CF_
crambasewidth](#2@@)--(#3@@);
\endscope
\let\CF_joinbond\CF_zero
\or% 7 = liaison Cram pointill  te de #3 vers #2
\scope
\CF_createnormnodes{#3@@}{#2@@}{\CF_crambasewidth/2
}{}%
\CF_clipcramornot(#3@@1)--(#3@@2)--(#2@@)--(#3@@1);
\CF_expafter{\draw[]\CF_currentbondstyle,dash patte
rn=on \CF_cramdashlength off \CF_cramdashsep,line width=\CF_
crambasewidth](#3@@)--(#2@@);
\endscope
\let\CF_joinbond\CF_zero
\or% 8 = liaison cram rectangle          de #2 vers #3
\CF_createnormnodes{#2@@}{#3@@}{\CF_crambasewidth/2}{}%
\CF_expafter{\draw[]\CF_currentbondstyle,line join=beve
l](#2@@1)--(#2@@2)--(#3@@)--cycle;
\let\CF_joinbond\CF_zero
\or% 9 = liaison cram rectangle          de #3 vers #1
\CF_createnormnodes{#3@@}{#2@@}{\CF_crambasewidth/2}{}%
\CF_expafter{\draw[]\CF_currentbondstyle,line join=beve
l](#3@@1)--(#3@@2)--(#2@@)--cycle;
\let\CF_joinbond\CF_zero
\else
\CF_error{unknown bond type, this error should not occu
r^^}If you think it's a bug, send a Minimal Example to the a
uthor}%

```

```

\fi
\let\CF_startoffset\empty
\let\CF_endoffset\empty
\let\CF_previoustikz\CF_currenttikz
\let\CF_previousbondangle\CF_previousangle
\def\CF_previousbond{#1}%
}

\def\CF_drawaxisbondnojoin#1#2{\CF_drawbonda(#1@@)--(#2@@);}

\def\CF_drawaxisbondjoin#1#2{% dessine une liaison simple dans
l'axe avec raccord r\l'trograde
\ifCF_incycle\ifnum\CF_cntcyclebonds=\CF_cyclenum\relax
\let\CF_nexttikz\CF_cyclefirsttikz
\fi\fi
\ifnum\CF_joinbond=0
\ifCF_incycle
\ifnum\CF_cntcyclebonds=\CF_cyclenum\relax
\ifnum\CF_cyclejoinlast=1
\CF_drawbonda(#1@@)--(#2@@)--%
([shift=(\CF_previousbondangle+2*\CF_cy
cleincrementangle:.5\pgflinewidth)]#2@@);
\else
\CF_drawbonda(#1@@)--(#2@@);
\fi
\else
\CF_drawbonda(#1@@)--(#2@@);
\fi
\else
\CF_drawbonda(#1@@)--(#2@@);
\fi
\def\CF_joinbond{1}%
\else
\CF_ifx\CF_previoustikz\CF_currenttikz
{\def\CF_joinbond{1}%
\CF_ifzerodim\CF_previousatomgroup
{\CF_ifx\CF_previousbondangle\empty% si d\l'tbut
mol\l'cule
{\CF_drawbonda(#1@@)--(#2@@);
}
{\ifdim\CF_startoffset=0pt
\ifCF_incycle
\ifnum\CF_cntcyclebonds=\CF_cyclenu
m\relax
\ifnum\CF_cyclejoinlast=1
\CF_ifx\CF_cyclefirsttikz

```

```
\CF_currenttikz
\CF_previousbondangle:-.5\pgflinewidth)]#1@@)--(#1@@)--(#2@@)
)--%
\CF_previousbondangle+2*\CF_cycleincrementangle:.5\pgflinewidth)]#2@@);
}
\CF_drawbonda([shift=(\CF_previousbondangle:-.5\pgflinewidth)]#1@@)--(#1@@)--(#2@@)
);
}%
\else
\CF_drawbonda([shift=(\CF_previousbondangle:-.5\pgflinewidth)]#1@@)--(#1@@)--(#2@@);
\fi
\else
\CF_drawbonda([shift=(\CF_previousbondangle:-.5\pgflinewidth)]#1@@)--(#1@@)--(#2@@);
\fi
\else
\CF_drawbonda([shift=(\CF_previousbondangle:-.5\pgflinewidth)]#1@@)--(#1@@)--(#2@@);
\fi
\else
\CF_drawbonda(#1@@)--(#2@@);
\fi
}%
}
{\CF_drawbonda(#1@@)--(#2@@);} %
}
{\ifCF_incycle
\ifnum\CF_cntcyclebonds=\CF_cyclenum\relax
\ifnum\CF_cyclejoinlast=1
\CF_ifx\CF_nexttikz\CF_currenttikz
{\CF_drawbonda(#1@@)--(#2@@)--([shift=(\CF_previousbondangle+2*\CF_cycleincrementangle:.5\pgflinewidth)]#2@@)};%
\CF_drawbonda(#1@@)--(#2@@);} %
\else
\CF_drawbonda(#1@@)--(#2@@);
\fi
\else
\CF_drawbonda(#1@@)--(#2@@);
\fi
\else
\CF_drawbonda(#1@@)--(#2@@);
```

```

        \fi
    }%
\fi
\ifdim\CF_endoffset=0pt \else
    \let\CF_joinbond\CF_zero
\fi
}

\def\CF_drawbonda{\CF_expafter{\draw[]\CF_currentbondstyle}}

\def\CF_hookdrawall{% dessine tous les crochets contenus dans l
a sc \CF_hookdraw
\CF_doifnotempty\CF_hookdrawlist
{ \expandafter\CF_hookdrawfirst\CF_hookdrawlist\_nil% tr
ace un lien de crochet Ã crochets
\CF_hookdrawall
}%
}

\def\CF_hookdrawfirst[#1,#2,#3]#4#5#6#7#8\_nil{%
\def\CF_hookdrawlist{#8}%
\begin{group}
\let\CF_joinbond\CF_zero
\def\CF_currenttikz{#3}%
\def\CF_hookstartcontent{#6}\def\CF_hookendcontent{#7}%
\CF_ifinteger{#2}%
{ \CF_drawbond{#2}{#4}{#5}\CF_hookstartcontent\CF_ho
okendcontent
}%
{ \CF_assignbondcode{#2}\CF_bondcurrentnum
\CF_drawbond\CF_bondcurrentnum{#4}{#5}\CF_hookstart
content\CF_hookendcontent
}%
\end{group}
}

\def\CF_extractatom#1-#2\_nil{#2}% transforme le bound@outnode
en nÂ de l'atome

\def\CF_gobblemovearg @#1#2\_nil#3{%
\expandafter\def\csname atom\_number\CF_cntatom\endcsname{
#2}%
\CF_ifinstr{#1},%
{ \CF_analysemovearg#1\_nil#3\let\CF_movebondcoeff\empty
}%
{ \def#3{#1}}%

```

```

\CF_doiempty{#2}{\let\CF_nodestru\empty}%
}%

\def\hflipnext{\def\CF_flipstate{1}}

\def\vflipnext{\def\CF_flipstate{2}}
\let\CF_flipstate\CF_zero

\def\CF_drawatomgroup#1#2#3{% #1=angle d'arrivee de la liaison
#2=numero atome sur lequel arrive la liaison #3=groupe d'atomes
\expandafter\let\expandafter\CF_bondoutcontent% assigne le
contenu de l'atome d'où part la liaison
\csname
\ifdefined\CF_bondoutnode
atom_\expandafter\CF_extractatom\CF_bondoutnode
\_nil
\else
empty%
\fi
\endcsname
\global\advance\CF_cntgroup1
\let\CF_currentatom\empty
\global\let\CF_hookdrawlist\empty
\CF_cntatomgroup0 % est le nombre d'atome dans le groupe que
va calculer \CF_drawatomgroupa
\CF_iffirsttokmatch{#3}?
{\CF_drawatomgroupa{#3}}
{\CF_drawatomgroupa{#3}}%
\def\CF_currentatomgroup{#3}%
\CF_remove\movearg\CF_currentatomgroup% enlève les "@{<nom>}"
\CF_ifinstr{#3}?%
{\CF_removehook\CF_currentatomgroup
\ifcat\relax\detokenize\expandafter{\romannumeral-'\.
\expandafter\noexpand\CF_currentatomgroup}\relax
\let\CF_currentatomgroup\empty
\fi
}%
}%
\CF_doiifnotempty{#2}
{\ifnum#2<1
\CF_warning{no atom found at position #2, perhaps
you misspelled the optional argument of the bond.}%
\else
\ifnum#2>\CF_cntatomgroup

```

```

\CF_error{no atom found at position #2, pershap
s you misspelled the optional argument of the bond.}%
\fi
\fi
}%
\edef\CF_hookatomnumber{%
\CF_ifempty{#2}
{\ifdim#1pt>90pt
\ifdim#1pt<270pt
\number\CF_cntatomgroup
\else
1%
\fi
\else
1%
\fi
}
{#2}%
}%
}%
\CF_cntatom\CF_hookatomnumber
\CF_ifzerodim\CF_currentatomgroup
{\let\CF_nodestrut\empty
}
{\CF_ifx\empty\CF_bondoutcontentsaved
{\def\CF_nodestrut{\vphantom\CF_bondoutcontent}}}%
{\def\CF_nodestrut{\vphantom\CF_bondoutcontentsaved
}}}%
}%
\edef\CF_optstring{anchor=\ifnum\CF_lastaction=0 base\else
\ifCF_incycle center\else\ifCF_macrofixedbondlength 180+#1
\else center\fi\fi\fi,at=(\CF_node),\CF_nodestyle}% premier a
tome de la mol\AA'cule affich\AA'
\loop
\unless\ifnum\CF_cntatom>\CF_cntatomgroup
\CF_eexpafter
{\futurelet\CF_toks}
{\expandafter\expandafter\expandafter\CF_gobtonil
\csname atom_\number\CF_cntatom\endcsname\_nil}%
\CF_ifx @\CF_toks% l'atome courant commence par un "@"
{\CF_threera\CF_gobblemovearg\csname atom_\number
\CF_cntatom\endcsname\_nil\CF_moveatomname
\CF_expafter{\node[]\CF_optstring,overlay](\CF_move
atomname){\phantom{\CF_nodecontent}};%
\let\CF_moveatomname\empty
}
}

```

```

    {}%
    \ifboolKV[chemfig]{debug}
    {
      \CF_expafter{\node[]\CF_optstring,draw=gray}(n
\number\CF_cntgroup-\number\CF_cntatom){\CF_nodecontent};%
      \CF_show_debug_atom
    }
    \CF_expafter{\node[]\CF_optstring}(n\number\CF_cnt
group-\number\CF_cntatom){\CF_nodecontent};%
  }%
  \let\CF_nodestrut\empty
  \advance\CF_cntatom1
  \edef\CF_optstring{anchor=base \ifnum\CF_flipstate=1 ea
st\else west\fi,at=(n\number\CF_cntgroup-\number\numexpr\CF_
cntatom-1.base \ifnum\CF_flipstate=1 west\else east\fi),\CF_
nodestyle}%
  \repeat
  \CF_cntatom\CF_hookatomnumber
  \ifnum\CF_lastaction=2 % s'il faut tracer une liaison
    \gdef\CF_cycleanglecorrection{0}% alors c'est qu'un cyc
le ne peut pas commencer la mol cule : annulation de la cor
rection d'angle
    \CF_drawbond\CF_bondtype{\CF_bondoutnode}{n\number\CF_c
ntgroup-\number\CF_cntatom}\CF_previousatomgroup\CF_currenta
tomgroup
  \fi
  \def\CF_lastaction{1}% met la derni re action   1 : affic
hage d'un noeud
  \loop
    \ifnum\CF_cntatom>1
      \advance\CF_cntatom-1
      \edef\CF_optstring{anchor=base \ifnum\CF_flipstate=1 we
st\else east\fi,at=(n\number\CF_cntgroup-\number\numexpr\CF_
cntatom+1.base \ifnum\CF_flipstate=1 east\else west\fi),\CF_
nodestyle}%
      \CF_eexpafter
      {
        \futurelet\CF_toksa{
        \expandafter\expandafter\expandafter\CF_gobtonil
\csname atom_\number\CF_cntatom\endcsname\_nil}%
        \CF_ifx @\CF_toksa% l'atome courant commence par un "@"
        {\CF_threeda\CF_gobblemovearg\csname atom_\number
\CF_cntatom\endcsname\_nil\CF_moveatomname
        \CF_expafter{\node[]\CF_optstring,overlay}(\CF_move
atomname){\phantom{\CF_nodecontent}};%
        \let\CF_moveatomname\empty
      }
    }%
  {}%

```

```

\ifboolKV[chemfig]{debug}
{
\CF_expafter{\node[]\CF_optstring,draw=gray](n
\number\CF_cntgroup-\number\CF_cntatom){\CF_nodecontent}};%
\CF_show_debug_atom
}
\CF_expafter{\node[]\CF_optstring](n\number\CF_cnt
group-\number\CF_cntatom){\CF_nodecontent}};%
}%
\repeat
\ifboolKV[chemfig]{debug}
\CF_show_debug_atomgroup
{}%
\CF_hookdrawall
\edef\CF_lastgroupnumber{\number\CF_cntgroup}%
\let\CF_previousatomgroup\CF_currentatomgroup
}

\def\CF_show_debug_atom{%
\node[at=(n\number\CF_cntgroup-\number\CF_cntatom.south),an
chor=north,outer sep=1pt,overlay]{$\scriptscriptstyle\color{
gray}\number\CF_cntatom$};%
}

\def\CF_show_debug_atomgroup{%
\draw[red,overlay] ([xshift=-.5pt,yshift=.5pt]n\number\CF_c
ntgroup-1.north west) rectangle ([xshift=.5pt,yshift=-.5pt]n
\number\CF_cntgroup-\number\CF_cntatomgroup.south east);%
\path (n\number\CF_cntgroup-1.north west) -- (n\number\CF_c
ntgroup-\number\CF_cntatomgroup.north east)
node [midway,yshift=1pt,overlay] {$\scriptscriptstyle
\color{red}\number\CF_cntgroup$};
}

\def\CF_keeppmovearg @#1#2\_nil{\def\CF_currentatom{@{#1}}}%

\def\CF_drawatomgroupa#1{% transforme #1 en un groupe d'atomes
\CF_ifempty{#1}
{
\expandafter\let\csname atom\_number\CF_cntatomgroup
\endcsname\CF_currentatom
}
{
\advance\CF_cntatomgroup1
\futurelet\CF_toks\CF_gobtonil#1\_nil
\CF_ifx @\CF_toks
{
\CF_keeppmovearg#1\_nil
\CF_removemovearga#1\_nil\CF_aftermovearg
\CF_expsecond\CF_drawatomgroupb{\CF_aftermovearg}%

```

```

    }%
    {\let\CF_currentatom\empty
    \CF_drawatomgroupb{#1}%
    }%
  }%
}

\def\CF_drawatomgroupb#1{%
  \CF_ifempty{#1}
  {\expandafter\let\csname atom_\number\CF_cntatomgroup
  \endcsname\CF_currentatom
  }
  {\futurelet\CF_toks\CF_gobtonil#1\_nil
  \CF_ifx\bgroup\CF_toks
  {\CF_eaddtomacro\CF_currentatom{\expandafter{\CF_fi
  rsttonil#1\_nil}}%
  \CF_expsecond\CF_drawatomgroupba{\CF_gobarg#1}%
  }%
  {\CF_ifx\CF_sptoken\CF_toks
  {\CF_addtomacro\CF_currentatom{ }%
  \CF_expsecond\CF_drawatomgroupba{\CF_afterspace
  #1\_nil}%
  }%
  {\CF_eaddtomacro\CF_currentatom{\CF_firsttonil
  #1\_nil}%
  \CF_expsecond\CF_drawatomgroupba{\CF_gobarg#1}%
  }%
  }%
  }%
}

% enlève tous les "@{nom}" de la sc #1
\def\CF_removevearg#1{%
  \CF_expsecond\CF_ifinstr{#1}@%
  {\expandafter\CF_removevearga#1\_nil#1%
  \CF_removevearg#1%
  }%
  {}%
}

% enlève le premier "@{<nom>}" de l'argument et l'assigne à #
2
\def\CF_removevearga#1\_nil#2{%
  \def\CF_removeveargb##1@{%
    \CF_expsecond{\def#2}{\CF_gobarg##1}% mange le \relax
    \CF_removeveargc\relax
  }
}

```

```

}%
\def\CF_removemoveargc##1\_nil{\CF_eaddtomacro#2{\CF_gobtwo
args##1}}% mange le \relax et le <nom>
\CF_removemoveargb\relax#1\_nil
}

\def\CF_drawatomgroupba#1{% transforme #1 en un groupe d'atomes
\CF_ifempty{#1}
{
\expandafter\let\csname atom\_number\CF_cntatomgroup
\endcsname\CF_currentatom
\let\CF_currentatom\empty
}
{\futurelet\CF_toks\CF_gobtonil#1\_nil
\CF_ifx @\CF_toks
{
\expandafter\let\csname atom\_number\CF_cntatomgro
up\endcsname\CF_currentatom
\let\CF_currentatom\empty
\CF_drawatomgroupa{#1}%
}%
{\CF_ifx|\CF_toks
{
\expandafter\let\csname atom\_number\CF_cntato
mgroup\endcsname\CF_currentatom
\let\CF_currentatom\empty
\CF_expsecond\CF_drawatomgroupa{\CF_gobarg#1}%
}%
{\CF_ifx\CF_sptoken\CF_toks
{\CF_addtomacro\CF_currentatom{ }%
\CF_expsecond\CF_drawatomgroupba{\CF_afters
pace#1\_nil}%
}%
{\CF_ifx\bgroup\CF_toks
{\CF_eaddtomacro\CF_currentatom{
\expandafter{\CF_firsttonil#1\_nil}}%
\CF_expsecond\CF_drawatomgroupba{\CF_go
barg#1}%
}%
{\CF_expsecond\CF_ifcarisupperletter{
\CF_firsttonil#1\_nil}%
{\expandafter\let\csname atom_
\_number\CF_cntatomgroup\endcsname\CF_currentatom
\let\CF_currentatom\empty
\CF_drawatomgroupa{#1}%
}%
{\CF_ifx?\CF_toks
{\CF_expsecond\CF_iffirsttokmat
ch{\CF_gobarg#1}[% un crochet apr s le "?"

```

```

\CF_gobarg#1\_nil\CF_afterhook}%
\CF_expafter{\CF_graboptarg
g[]}{\CF_gobarg#1\_nil\CF_afterhook}%
\CF_exptwomacroargs\CF_ifinstr{
\CF_hooklist}{\expandafter(\CF_hookcurrentname)}% crochet d'Ã
l'Ã d'Ãfini ?
\CF_expsecond\CF_hookparse
list{\CF_hookcurrentname}% chercher les caractÃristiques du
crochet sauvegardÃ
\CF_edefaddtomacro\CF_hookd
rawlist{%
[\CF_hookcurrentname,
\CF_hookcurrentlink,\CF_hookcurrenttikz]{\CF_hooksavedcoord}
{n\CF_cntgroup-\CF_cntatomgroup}}%
\CF_eaddtomacro\CF_hookdraw
list{\expandafter{\CF_hooksavedcontent}}%
\CF_eaddtomacro\CF_hookdraw
list{\expandafter{\CF_currentatom}}% ajoute les 4 arguments
Ã la liste des crochets Ã tracer
\global\let\CF_hookdrawlist
\CF_hookdrawlist
}%
\CF_edefaddtomacro\CF_hook
list{(\CF_hookcurrentname)|n\CF_cntgroup-\CF_c
ntatomgroup|}%
\CF_eaddtomacro\CF_hooklist
{\CF_currentatom|}%
\global\let\CF_hooklist\CF_
hooklist
}%
\CF_expsecond\CF_drawatomgroupb
a{\CF_afterhook}%
}%
{\CF_eaddtomacro\CF_currentatom
\CF_expsecond\CF_drawatomgroupb
a{\CF_gobarg#1}%
}%
}%
}%
}%
}%
}%
}%
}%
}%
}

```

```

\def\CF_kookdefaultname{a}
\def\CF_hookdefaultlink{-}
\def\CF_hookdefaulttikz{}

\def\CF_hookparseoptarg#1,#2,#3\_nil{%
  \CF_testemptyandassign\CF_hookcurrentname{#1}\CF_kookdefaultname
  \CF_testemptyandassign\CF_hookcurrentlink{#2}\CF_hookdefaultlink
  \CF_testemptyandassign\CF_hookcurrenttikz{#3}\CF_hookdefaulttikz
}

\def\CF_graboptarg[#1]#2\_nil#3{%
  \CF_hookparseoptarg#1,,\_nil
  \def#3{#2}%
}

\def\CF_hookparselist#1{% #1 est le nom du noeud Ã  retrouver
  \def\CF_hookparselist##1(#1)|##2|##3|##4\_nil{\def\CF_hook
    savedcoord{##2}\def\CF_hooksavedcontent{##3}}%
  \expandafter\CF_hookparselist\CF_hooklist\_nil
}

\def\CF_removehook#1{%
  \CF_expsecond\CF_ifinstr{#1}?%
  {\CF_expafter{\CF_removehooka\relax}#1\_nil#1%
  \CF_removehook#1%
  }
  {}%
}

\def\CF_removehooka#1?#2\_nil#3{%
  \CF_iffirsttokmatch{#2}[%
  {\CF_removehookb#1?#2\_nil#3}
  {\CF_expsecond{\def#3}{\CF_gobarg#1#2}}%
  }

\def\CF_removehookb#1?[#2]#3\_nil#4{\CF_expsecond{\def#4}{\CF_g
  obarg#1#3}}

\defKV[charge]{%
  .radius      = \CF_defifempty\CF_dotradius {#1}{0.15ex},
  .sep         = \CF_defifempty\CF_dotsep    {#1}{0.3em},
  .style       = \CF_defifempty\CF_dotstyle  {#1}{fill=black}

```

```

    ,
    "length      = \CF_defifempty\CF_rectlength{#1}{1.5ex},
    "width       = \CF_defifempty\CF_rectwidth {#1}{.3ex}
}
\def\setcharge#{\setKV[charge]}
\def\resetcharge{\restoreKV[charge]}
\setKVdefault[charge]{%
    debug        = false,% trace les contours des noeuds
    macro atom    = \printatom,%macro qui prendra comme argument
        l'atome recevant la charge
    circle       = false,% false => noeud atome = rectangle
    macro charge  = ,% macro attendue (\printatom ou \ensuremat
        h, par exemple) qui prendra comme argument la charge
    extra sep     = 1.5pt,% s'Alparation additionnelle entre le n
        oeud (cercle ou rectangle) et la position des charges
    overlay      = true,% charges en "surimpression"
    shortcuts     = true,% raccourcis \. \: \| et \" actifs pour
        Lewis
    lewisautorot  = true,% rotation auto charge Lewis
    .radius       = 0.15ex,% rayon du point
    :sep          = 0.3em,% s'Alparation des deux points
    .style        = {fill=black},% style des points
    "length       = 1.5ex,% longueur rectangle
    "width        = .3ex,% largeur rectangle
    "style        = {black,line width=0.4pt},% style rectangle
    |style        = {black,line width=0.4pt},% style ligne
}%
\def\chargedot{\CF_testopt\chargedot_a{}}
\def\chargedot_a[#1]{%
    \begingroup
        \setKV[charge]{#1}%
        \CF_expafter{\tikz\draw[]{\CF_dotstyle}}(0,0)circle(\CF
            _dotradius);%
    \endgroup
}
\def\chargeddot{\CF_testopt\chargeddot_a{}}
\def\chargeddot_a[#1]{%
    \begingroup
        \setKV[charge]{#1}%
        \ifboolKV[charge]{lewisautorot}
            {\pgfmathsetmacro\CF_lewisrot{90+\chargeangle}}
            {\def\CF_lewisrot{0}}%
        \pgfmathsetmacro\CF_halfsep{\CF_dotsep/2}%
        \tikzpicture[anchor=center,rotate=\CF_lewisrot]%
            \CF_expafter{\draw[]{\CF_dotstyle}}%
            (-\CF_halfsep pt,0)circle(\CF_dotradius)%
    \endgroup
}

```

```

(\CF_halfsep pt,0)circle(\CF_dotradius);%
\endtikzpicture
\endgroup
}
\def\chargerect{\CF_testopt\chargerect_a{}}
\def\chargerect_a[#1]{%
\begingroup
\setKV[charge]{#1}%
\ifboolKV[charge]{lewisautorot}
{\pgfmathsetmacro\CF_lewisrot{90+\chargeangle}}
{\def\CF_lewisrot{0}}%
\pgfmathsetmacro\CF_halfwidth{\CF_rectwidth/2}%
\pgfmathsetmacro\CF_halflength{\CF_rectlength/2}%
\tikzpicture[anchor=center,rotate=\CF_lewisrot]%
\CF_eexpafter{\draw[]{\useKV[charge]{|style}}}{(-\CF_halflength pt,-\CF_halfwidth pt)rectangle(\CF_halflength pt,\CF_halfwidth pt)};% bugfix 1.51
\endtikzpicture
\endgroup
}
\def\chargeline{\CF_testopt\chargeline_a{}}
\def\chargeline_a[#1]{%
\begingroup
\setKV[charge]{#1}%
\ifboolKV[charge]{lewisautorot}
{\pgfmathsetmacro\CF_lewisrot{90+\chargeangle}}
{\def\CF_lewisrot{0}}%
\pgfmathsetmacro\CF_halflength{\CF_rectlength/2}%
\tikzpicture[anchor=center,rotate=\CF_lewisrot]%
\CF_eexpafter{\draw[]{\useKV[charge]{|style}}}{(-\CF_halflength pt,0)--(\CF_halflength pt,0)};% bugfix 1.51
\endtikzpicture
\endgroup
}
\def\CF_enablesHORTcuts{%
\let\CF_saveddot \.\let\.\chargedot
\let\CF_savedddot\:\let\:\chargeddot
\let\CF_savedrect\" \let\"chargerect
\let\CF_savedline\" \let\"chargeline
\let\enablesHORTcuts\relax
\let\disablesHORTcuts\CF_disablesHORTcuts
}
\def\CF_disablesHORTcuts{%
\let\.\CF_saveddot
\let\:\CF_savedddot
\let\" \CF_savedrect

```

```

\let\|\CF_savedline
\let\enablesshortcuts\CF_enablesshortcuts
\let\disablesshortcuts\relax
}
\def\charge{%
  \beginingroup
    \catcode'\: 12
    \charge_a{true}%
  }
\def\Charge{%
  \beginingroup
    \catcode'\: 12
    \charge_a{false}%
  }
\def\charge_a#1#2{% #1=TF #2=liste emplacements
  \CF_testopt{\charge_b{#1}}{#2}\_nil
}
\def\charge_b#1[#2]#3\_nil{%
  \charge_c{#1}[#2]{#3}%
}
\def\charge_c#1[#2]#3#4{% #1=TF pour overlay, #2= rÅlglages, #3
  =liste d'emplacements, #4=atome
  \setcharge{overlay=#1,#2}%
  \setbox\CF_chargebox\hbox{\useKV[charge]{macro atom}{#4
  }}%
  \CF_ifinsidetikz
  {\pgfinterruptpicture
    \let\CF_atendofcharge\endpgfinterruptpicture
  }
  {\let\CF_atendofcharge\relax
  }%
  \expanded{\noexpand
  \tikzpicture[every node/.style={%
    \ifboolKV[charge]{debug}{draw=red,}{}%
    anchor=base,%
    inner sep=0pt,%
    outer sep=0pt,%
    minimum size=0pt},%
    baseline]}%
  \expanded{\noexpand
  \node[%
    \ifboolKV[charge]{circle}{circle,}{}%
    \ifboolKV[charge]{debug}{draw=green,}{}%
    anchor=base%
  ]}%
  (atombox)at(0,0)%

```

```

        {\copy\CF_chargebox};% noeud contenant l'atome
\expanded{\noexpand
\node[%
\ifboolKV[charge]{circle}{circle,}{}%
\ifboolKV[charge]{debug}{draw=blue,}{}%
anchor=base,%
inner sep=\useKV[charge]{extra sep},%
overlay%
]}%
(atom)at(0,0){%
\vrule width0pt height\ht\CF_chargebox
depth\dp\CF_chargebox
\vrule width\wd\CF_chargebox height\CF_
zero depth\CF_zero};% noeud pour placer les charges
\let\enablesshortcuts\relax
\let\disablesshortcuts\relax
\ifboolKV[charge]{shortcuts}\CF_enablesshortcuts{}%
l'atome n'est PAS concerné par les raccourcis
\charge_d#3,\CF_quark=%
\endtikzpicture
\CF_atendofcharge
\endgroup
}
\def\charge_d#1={%
\CF_ifx\CF_quark{#1}%
{}
{\CF_striplastsp{#1}\charge_e=}% bugfix 1.54
}
\def\charge_e#1={%
\CF_ifinstr{#1}[
{\charge_f#1=}
{\charge_f#1[]}=}%
}
\def\charge_f#1[#2]={%
\CF_ifinstr{#1}:
{\charge_g#1[#2]=}
{\charge_g#1:0pt[#2]=}%
}
\def\charge_g#1:#2[#3]=#4,{% #1=angle, #2=offset, #3=code tikz
charge, #4=charge
\CF_stribsp{#1}\CF_ifinteger
{\pgfmacthsetmacro\chargeangle{mod(#1,360)}}%
}
{\pgfmacthanglebetweenpoints{\pgfpointanchor{atom}{cente
r}}{\CF_stribsp{#1}\pgfpointanchor{atom}}}%
\let\chargeangle\pgfmacthresult% incorrect si (atom.cent

```

```

er==atom.#1) && (extra sep==0) TODO: mettre un warning ?
}%
\edef\CF_offset{\the\dimexpr#2+0pt}%
\CF_striple{#1}{\CF_distancebetweenpoints{atom}{center}{atom}}\CF_chargedistance
\CF_eexpafter{\node[anchor=center,]{\ifboolKV[charge]{overlay}{overlay}{overlay,}{}}#3}%
at([shift=(\chargeangle:\CF_chargedistance pt+\CF_offset)]atom.center){\useKV[charge]{macro charge}{#4}};%
\charge_d
}

\def\Chembelow{\begingroup\let\CF_temp\CF_gobarg\CF_chembelowa}

\def\chembelow{\begingroup\let\CF_temp\CF_id\CF_chembelowa}

\def\CF_chembelowa{\CF_testopt\CF_chembelowb\CF_stacksep}

\def\CF_chembelowb[#1]#2#3{%
\setbox\CF_box\hbox{\printatom{#2}}%
\expandafter\top\CF_temp{to\ht\CF_box}{%
\offinterlineskip
\hbox{\printatom{#2}}%
\kern#1\relax
\hbox to\wd\CF_box{\hss\printatom{#3}\hss}%
\CF_temp\vss
}%
\endgroup
}

\def\Chemabove{\begingroup\let\CF_temp\CF_gobarg\CF_chemabovea}

\def\chemabove{\begingroup\let\CF_temp\CF_id\CF_chemabovea}

\def\CF_chemabovea{\CF_testopt\CF_chemaboveb\CF_stacksep}

\def\CF_chemaboveb[#1]#2#3{%
\setbox\CF_box\hbox{\printatom{#2}}%
\expandafter\box\CF_temp{to\ht\CF_box}{%
\offinterlineskip
\CF_temp\vss
\hbox to\wd\CF_box{\hss\printatom{#3}\hss}%
\kern#1\relax
\hbox{\printatom{#2}}%
}%
\endgroup

```

```

}

\def\chemmove{\CF_testopt\CF_chemmove{}}

\def\CF_chemmove[#1]#2{%
  \CF_doifnotempty{#2}%
    {\expandafter\tikzpicture\expanded{[overlay,remember picture,-CF\CF_ifempty{#1}]{},\unexpanded{#1}}}%
    #2%
  \endtikzpicture
}%

}

\def\chemnameinit#1{%
  \setbox\CF_boxstuff\hbox{#1}%
  \xdef\CF_dpmax{\the\dp\CF_boxstuff}%
}
\let\CF_dpmax\CF_zero

\def\CF_parsemolname#1\#2\_nil{%
  \hbox to\CF_wdstuffbox{\hss#1\hss}%
  \CF_doifnotempty{#2}{\CF_parsemolname#2\_nil}%
}

\def\chemname{%
  \CF_ifstar
    {\CF_adjustnamedpfalse\CF_chemnamea}
    {\CF_adjustnamedptrue \CF_chemnamea}%
}

\def\CF_chemnamea{\CF_testopt\CF_chemnameb{1.5ex}}

\def\CF_chemnameb[#1]#2#3{%
  \setbox\CF_boxstuff\hbox{#2}%
  \edef\CF_wdstuffbox{\the\wd\CF_boxstuff}\edef\CF_dpstuffbox
  {\the\dp\CF_boxstuff}%
  \leavevmode
  \ifdim\CF_dpmax<\CF_dpstuffbox\global\let\CF_dpmax\CF_dpstuffbox\fi
  \vtop{%
    \box\CF_boxstuff
    \nointerlineskip
    \kern\dimexpr#1\ifCF_adjustnamedp+\CF_dpmax-\CF_dpstuffbox\fi\relax
    \CF_parsemolname#3\\\_nil
  }%
}

```



```

        {$\left.\vrule height\CF_delimdim depth\CF_delimdim width0pt
        \right#3$};%
        \endtikzpicture
    \endgroup
}

\def\chemdown#1{%
    \CF_warning{"\string\chemdown\string#1" ignored! No \string
    \chemup\space previously found.}%
}

\def\CF_setstyle#1,#2,#3\_nil#4#5#6{%
    \def#4{#1}\let#5\empty\let#6\empty
    \CF_iffirsttokmatch\CF_quark{#2\relax}
    {}%
    {\def#5{#2}%
    \CF_iffirsttokmatch\CF_quark{#3\relax}
    {}%
    {\CF_setstylea#3\_nil#6}%
    }%
}

\def\CF_setstylea#1,\CF_quark#2\_nil#3{\def#3{#1}}

\def\CF_and{\futurelet\CF_toks\CF_anda}

\def\CF_anda{%
    \CF_ifx\CF_toks\bgroup
    {\CF_andb}
    {\CF_andb{}}%
}

\def\CF_andb#1{%
    \CF_setstyle#1,\CF_quark,\CF_quark\_nil\CF_signspaceante_
    \CF_signspacepost\_CF_signvshift_
    \CF_doifnotempty\CF_signspaceante_{\let\CF_signspaceante\CF
    _signspaceante_}%
    \CF_doifnotempty\CF_signspacepost_{\let\CF_signspacepost\CF
    _signspacepost_}%
    \CF_doifnotempty\CF_signvshift_{\let\CF_signvshift\CF_signv
    shift_}%
    \raise\CF_signvshift\hbox{\kern\CF_signspaceante$+\kern\CF
    _signspacepost}%
}

\def\schemestart{%
    \begingroup

```

```

\edef\CF_schemenest{\number\numexpr\CF_schemenest+1}%
\CF_testopt\CF_schemestarta{}}%
}

\def\CF_schemestarta[#1]{%
  \CF_setstyle#1,\CF_quark,\CF_quark\_nil\CF_arrowangle\_CF_a
rrowlength\_CF_arrowstyle_
  \CF_doifnotempty\CF_arrowangle_{\let\CF_arrowangle\CF_arrow
angle_}%
  \CF_doifnotempty\CF_arrowlength_{\let\CF_arrowlength\CF_arr
owlength_}%
  \CF_eexpsecond{\def\CF_arrowtip}{\expandafter\CF_gobarg\CF_
arrowhead}%
  \CF_expsecond{\CF_preaddtomacro\CF_defaultarrowstyle}{\CF_a
rrowhead,}%
  \let\CF_arrowstyle\CF_defaultarrowstyle
  \CF_doifnotempty\CF_arrowstyle_{\CF_eaddtomacro\CF_arrowsty
le{\expandafter,\CF_arrowstyle_}}%
  \pgfmathsetmacro\CF_arrowdoublesep{\CF_arrowdoublesep/2}%
  \pgfmathsetmacro\CF_arrowdoubleposstart{(1-\CF_arrowdoublep
osstart)/2}%
  \pgfmathsetmacro\CF_arrowdoubleposend{1-\CF_arrowdoublepos
start}%
  \ifboolKV[chemfig]{scheme debug}
    {\tikzpicture[every node/.style={draw,anchor=base,inner
sep=0pt,outer sep=0pt,minimum size=1.5pt},baseline,remember
picture]}
    {\tikzpicture[every node/.style={anchor=base,inner sep=
0pt,outer sep=0pt,minimum size=0pt},baseline,remember pictur
e]}%
  \let\merge\CF_merge
  \expandafter\let\csname+\endcsname\CF_and
  \let\arrow\CF_arrow
  \let\schemestop\CF_schemestop
  \let\subscheme\CF_subscheme
  \CF_testopt{\CF_schemestartb}{text}%
}

\def\CF_schemestartb[#1]{%
  \ifnum\CF_schemenest=1 % la commande n'est pas imbriquée ?
    \CF_cntcompound0
  \fi
  \edef\CF_currentnodename{c\number\CF_cntcompound}%
  \let\CF_nextnodename\empty
  \let\CF_nextnodestyle\empty
  \let\CF_directarrowlist\empty

```

```

\ifboolKV[chemfig]{scheme debug}
  {\node[fill,green](\CF_currentnodename){};}
  {\node(\CF_currentnodename){};}%
\def\CF_nextnodeanchor{#1}%
\CF_doiempty\CF_nextnodeanchor{\def\CF_nextnodeanchor{text
}}%
\let\CF_compound\empty
\CF_schemestartc
}

\def\CF_schemestartc{%
  \futurelet\CF_toks\CF_schemestarte
}

\expandafter\def\expandafter\CF_schemestartd\space{\futurelet
\CF_toks\CF_schemestarte}

\def\CF_schemestarte{% ... et l'examine :
  \CF_iffirsttokina{\arrow\schemestop\merge}%
  {}
  {\CF_ifx\CF_toks\bgroup
    {\ifCF_compound_is_chemfig% bugfix 1.6
      \edef\CF_restore_hashcatcode{\catcode\number'\#
=\number\catcode'\# \relax}%
      \catcode'\#12 % TODO ou carrÃ¢lement mettre # Ã¢
12 dans tout l'environnement ?
      \fi
      \CF_addnextarg
    }
    {\CF_ifx\CF_toks\CF_sptoken
      {\CF_addtomacro\CF_compound{ }%
      \CF_schemestartd
      }
      {\CF_ifx\CF_toks\chemfig
        \CF_compound_is_chemfigtrue% mettre le flag
        Ã¢ vrai
        {}%
        \afterassignment\CF_schemestartc
        \CF_addtomacro\CF_compound
        }%
      }%
    }%
  }

\def\CF_addnextarg#1{%
  \CF_addtomacro\CF_compound{{#1}}%

```

```

\ifCF_compound_is_chemfig% bugfix 1.6
  \CF_restore_hashcatcode
  \CF_compound_is_chemfigfalse% mettre le flag Ã  faux
\fi
\CF_schemestartc
}

\def\CF_displaycompound#1#2{% #1 = nom et #2 = style
  \CF_doifnotempty\CF_compound
  {\global\advance\CF_cntcompound1
  \CF_ifx\CF_defaultcompoundstyle\empty
  {\let\CF_currentnodestyle\empty}
  {\CF_expsecond{\def\CF_currentnodestyle}{\CF_defaul
tcompoundstyle,}}%
  \CF_addtomacro\CF_currentnodestyle{anchor=\CF_nextnodea
nchor,at=(\CF_currentnodename)}%
  \CF_ifempty{#2}%
  {\CF_doifnotempty\CF_nextnodestyle
  {\CF_eaddtomacro\CF_currentnodestyle{
\expandafter,\CF_nextnodestyle}}%
  }
  {\CF_doifnotempty\CF_nextnodestyle
  {\CF_warning{two styles for the same node, firs
t style "\CF_nextnodestyle" ignored}%
  }%
  \CF_addtomacro\CF_currentnodestyle{,#2}%
  }%
  \CF_ifempty{#1}
  {\edef\CF_temp{%
    \CF_ifempty\CF_nextnodename
    {c\number\CF_cntcompound}
    {\CF_nextnodename}%
  }%
  }
  {\CF_doifnotempty\CF_nextnodename
  {\CF_warning{two names for the same node, first
name "\CF_nextnodename" ignored}%
  }%
  \edef\CF_temp{#1}%
  }%
  \CF_expafter{\node[]\CF_currentnodestyle}(\CF_temp){\CF
_compound};%
  \ifboolKV[chemfig]{scheme debug}%
  {\node[draw=none,anchor=270,at=(\CF_temp.90),fill=g
reen!60,overlay,opacity=0.5]{\scriptsize\bfseries\CF_temp};%
  }
}

```

```

        {}%
        \let\CF_currentnodename\CF_temp
    }%
}

\def\CF_schemestop{%
    \CF_displaycompound{}{}%
    \CF_directarrowlist
    \endtikzpicture
    \xdef\CF_schemenest{\number\numexpr\CF_schemenest-1}%
    \endgroup
}

\def\CF_analysearrowarg#1{\CF_analysearrowarg#1[]\_nil}

\def\CF_analysearrowarg#1[#2]#3\_nil{%
    \CF_ifinstr{#1}.
        {\CF_addtomacro\CF_temp{#1[#2]}}
        {\CF_addtomacro\CF_temp{#1.[#2]}}%
}

\def\CF_arrow{%
    \CF_ifnextchar(
        {\CF_arrowa
        }
        {\CF_ifnextchar\bgroup
            {\CF_arrowb(.[]--.[])}
            {\CF_arrowb(.[]--.[]){}}%
        }%
    }

\def\CF_arrowa(#1--#2){%
    \def\CF_temp{ }%
    \CF_analysearrowarg{#1}%
    \CF_addtomacro\CF_temp{--}%
    \CF_analysearrowarg{#2}%
    \CF_addtomacro\CF_temp)%
    \CF_ifnextchar\bgroup
        {\expandafter\CF_arrowb\CF_temp}
        {\expandafter\CF_arrowb\CF_temp{}}%
    }

\def\CF_arrowb(#1.#2[#3]--#4.#5[#6])#7{%
    \def\CF_currentarrowtype{#7}% nom de la fl che
    \CF_doifempty\CF_currentarrowtype{\def\CF_currentarrowtype{
    ->}}%

```

```

\CF_testopt{\CF_arrowc(#1.#2[#3]--#4.#5[#6])}{}%
}

\def\CF_arrowc(#1.#2[#3]--#4.#5[#6])[#7]{%
  \def\CF_temp{\CF_arrowe(#1.#2[#3]--#4.#5[#6])}%
  \CF_arrowd#7,\empty,\empty\_nil
}

\def\CF_arrowd#1,#2,#3\_nil{%
  \CF_addtomacro\CF_temp{{#1}}%
  \CF_eaddtomacro\CF_temp{\expandafter{#2}}%
  \expandafter\CF_eaddtomacro\expandafter\CF_temp\expandafter
  {\expandafter\expandafter\expandafter{\expandafter\CF_saniti
  zelastitem#3,\empty\_nil}}%
  \CF_temp
}

% #1, #4 : nom des nodes      #2, #5 : ancrs des nodes      #3, #6
% : styles des nodes
% #7 : angle fl  che      #8 : longueur fl  che      #9 : style tik
% z de la fl  che
\def\CF_arrowe(#1.#2[#3]--#4.#5[#6])#7#8#9{%
  \let\CF_arrowcurrentstyle\CF_arrowstyle
  \if @\expandafter\CF_firsttonil\detokenize{#1.}\_nil% si #1
  commence par @
    \if @\expandafter\CF_firsttonil\detokenize{#4.}\_nil
      \CF_eaddtomacro\CF_directarrowlist{\expandafter\CF_
      directarrow\expandafter{\CF_currentarrowtype}{#1}{#2}{#4}{#5
      }{#9}}%
      \let\CF_nextaction\CF_schemestartc
    \else
      \CF_doifnotempty\CF_arrowcurrentstyle{\CF_addtomacr
      o\CF_arrowcurrentstyle,%
      \CF_doifnotempty{#9}{\CF_addtomacro\CF_arrowcurrent
      style{#9,}}%
      \CF_displaycompound{{#3}}%
      \def\CF_nextnodename{#4}%
      \CF_expsecond{\def\CF_currentnodename}{\CF_gobarg#1
      }%
      \let\CF_arrowstartname\CF_currentnodename
      \let\CF_arrowendname\CF_nextnodename
      \CF_arrowf{#7}{#8}{#2}{#5}%
      \def\CF_nextnodestyle{#6}%
    \fi
  \else
    \CF_doifnotempty\CF_arrowcurrentstyle{\CF_addtomacro\CF

```

```

_arrowcurrentstyle,}%
\CF_doiifnotempty{#9}{\CF_addtomacro\CF_arrowcurrentstyl
e{#9,}}}%
\if @\expandafter\CF_firsttonil\detokenize{#2.}\_nil
\CF_error{syntax "(<name>--@<name>)" is not allowed
}%
\else
\CF_displaycompound{#1}{#3}%
\edef\CF_arrowstartname{%
\CF_ifempty{#1}
\CF_currentnodename
{#1}%
\CF_doiifnotempty{#2}{.#2}%
}%
\CF_arrowf{#7}{#8}{#2}{#5}%
\def\CF_nextnodename{#4}%
\def\CF_nextnodestyle{#6}%
\fi
\fi
\CF_arrowgobspaces% mange les espaces puis ex  cute \CF_nex
taction
}

\def\CF_arrowgobspaces{\futurelet\CF_toks\CF_arrowgobspacesa}

\def\CF_arrowgobspacesa{%
\CF_ifx\CF_sptoken\CF_toks
\CF_arrowgobspacesb
\CF_nextaction
}

\expandafter\def\expandafter\CF_arrowgobspacesb\space{
\futurelet\CF_toks\CF_arrowgobspacesa}

\def\CF_arrowf#1#2#3#4{% #1=angle #2=longueur #3=ancre d  t
part #4=ancre arriv  e
\def\CF_nextaction{\let\CF_compound\empty\CF_schemestartc}%
\def\CF_arrowcurrentangle{#1}\CF_doiifempty\CF_arrowcurrenta
ngle{\let\CF_arrowcurrentangle\CF_arrowangle}%
\def\CF_currentarrowlength{#2}\CF_doiifempty\CF_currentarrow
length{\let\CF_currentarrowlength\CF_arrowlength}%
\node[at=(\CF_currentnodename.\CF_ifempty{#3}\CF_arrowcurre
ntangle{#3}),shift=(\CF_arrowcurrentangle:\CF_currentarrowle
ngth*\CF_compoundsep),cyan,fill](end@arrow@i@number\CF_sche
menest){};%
\edef\CF_arrowendname{end@arrow@i@number\CF_schemenest\CF_

```

```

doifnotempty{#4}{.#4}}%
\ifboolKV[chemfig]{scheme debug}
{
  \node[at=(\CF_currentnodename.\CF_ifempty{#3}\CF_arrow
currentangle{#3}),shift=(\CF_arrowcurrentangle:\CF_arrowoffs
et),red,fill](start@arrow){};%
  \node[at=(\CF_currentnodename.\CF_ifempty{#3}\CF_arrowc
urrentangle{#3}),shift=(\CF_arrowcurrentangle:\CF_currentarr
owlength*\CF_compoundsep-\CF_arrowoffset),red,fill](end@arro
w){};%
}
{
  \node[at=(\CF_currentnodename.\CF_ifempty{#3}\CF_arrow
currentangle{#3}),shift=(\CF_arrowcurrentangle:\CF_arrowoffs
et)](start@arrow){};%
  \node[at=(\CF_currentnodename.\CF_ifempty{#3}\CF_arrowc
urrentangle{#3}),shift=(\CF_arrowcurrentangle:\CF_currentarr
owlength*\CF_compoundsep-\CF_arrowoffset)](end@arrow){};%
}%
\def\CF_arrowstartnode{start@arrow}\def\CF_arrowendnode{end
@arrow}%
\csname\expandafter\CF_grabarrownam\CF_currentarrowtype[\_
nil\CF_threeea\endcsname
\expandafter\CF_grabarrowargs\CF_currentarrowtype[\_ni
l[] [] [] [] [] [] []]\_nil
\def\CF_currentnodename{end@arrow@i@\number\CF_schemenest}%
\edef\CF_nextnodeanchor{\CF_ifempty{#4}{180+\CF_arrowcurren
tangle}{#4}}%
}

% trace un fl che initi le par (@nom--@nom)
% #1=type de fl che #2=nom depart #3=ancre d part #4=n
om arriv le #5=ancre arriv le #6=style fl che
\def\CF_directarrow#1#2#3#4#5#6{%
  \CF_expsecond{\def\CF_arrowstartname}{\CF_gobarg#2}%
  \CF_expsecond{\def\CF_arrowendname}{\CF_gobarg#4}%
  \path[sloped,allow upside down](\CF_gobarg#2\ifx\empty#3
\empty\else.#3\fi)--(\CF_gobarg#4\ifx\empty#5\empty\else.#5
\fi)%
  coordinate[pos=0,xshift=\CF_arrowoffset](start@direct@a
rrow)%
  coordinate[pos=1,xshift=-\CF_arrowoffset](end@direct@ar
row);%
  \def\CF_arrowstartnode{start@direct@arrow}%
  \def\CF_arrowendnode{end@direct@arrow}%
  \pgfmathanglebetweenpoints
  {\pgfpointanchor{\CF_gobarg#2}{\ifx\empty#3\empty cente
r\else#3\fi}}% Ne pas utiliser \CF_ifempty ici !!!

```

```

        {\pgfpointanchor{\CF_gobarg#4}{\ifx\empty#5\empty center\else#5\fi}}%
        \let\CF_arrowcurrentangle\pgfmathresult
        \CF_doiifnotempty{#6}{\CF_addtomacro\CF_arrowcurrentstyle{#6,}}}%
        \csname\CF_grabararrowname#1[]\_nil\expandafter\endcsname\CF_grabararrowargs#1[]\_nil[] [] [] [] [] [] []\_nil
    }

\def\CF_mergegrabchardir#1[#2][#3]#4\_nil{%
    \CF_expafter{\futurelet\CF_toksa\CF_gobtonil}{\CF_firsttonil#1>\_nil}\_nil
    \ifx>\CF_toksa
        \def\CF_mergeangle{0}\def\CF_mergeextreme{xmax}\def\CF_mergesign{+}%
    \else
    \ifx<\CF_toksa
        \def\CF_mergeangle{180}\def\CF_mergeextreme{xmin}\def\CF_mergesign{-}%
    \else
    \ifx^\CF_toksa
        \def\CF_mergeangle{90}\def\CF_mergeextreme{ymax}\def\CF_mergesign{+}%
    \else
    \ifx v\CF_toksa
        \def\CF_mergeangle{-90}\def\CF_mergeextreme{ymin}\def\CF_mergesign{-}%
    \fi\fi\fi\fi
    \def\CF_mergelabelup{#2}\def\CF_mergelabeldo{#3}%
}

\def\CF_merge#1({%
    \CF_mergegrabchardir#1[]\_nil
    \CF_mergea(
}

\def\CF_mergea#1--(#2){\CF_testopt{\CF_mergeb#1--(#2)}{}}

\def\CF_mergeb#1--(#2)[#3]{%
    \CF_displaycompound{}}}%
    \CF_parsemergeopt#3,\CF_quark,\CF_quark,\CF_quark\_nil
    \def\CF_mergexmax{-16383.99999pt}\let\CF_mergeymax\CF_mergexmax
    \def\CF_mergexmin{16383.99999pt}\let\CF_mergeymin\CF_mergexmin
    \CF_mergeparsenodelist#1(\relax)% calcule les maxi des posi

```

```

tions
\pgfmathsetmacro\CF_mergeextremeresult{\csname CF_merge\CF_
mergeextreme\endcsname\CF_mergesign\CF_mergefromcoeff*\CF_co
mpoundsep}%
\CF_mergec#1(\relax)% trace les lignes entre les noeuds prÃ
l'Ãdents et la ligne de jonction
\CF_expsecond{\def\CF_temp}{\expandafter[\CF_mergestyle,sho
rten <=0,shorten >=0,-]}%
\if x\expandafter\CF_firsttonil\CF_mergeextreme\_nil
\CF_addtomacro\CF_temp{(\CF_mergeextremeresult pt,\CF_m
ergeymax)--(\CF_mergeextremeresult pt,\CF_mergeymin)}%
\else
\CF_addtomacro\CF_temp{(\CF_mergexmin,\CF_mergeextreme
result pt)--(\CF_mergexmax,\CF_mergeextremeresult pt)}%
\fi
\expandafter\draw\CF_temp node[pos=\CF_mergesplitcoeff](mer
ge@point){}% trace la ligne de jonction
node[at=(merge@point),shift=(\CF_mergeangle:\CF_compoun
dsep*\CF_mergetocoeff-\CF_arrowoffset)](end@merge){}%
node[at=(merge@point),shift=(\CF_mergeangle:\CF_compoun
dsep*\CF_mergetocoeff)](end@merge@i){};%
\let\CF_arrowcurrentangle\CF_mergeangle
\CF_expafter{\draw[\CF_mergestyle,shorten <=0](merge@point
)--(end@merge)%
\expandafter\CF_arrowdisplaylabela\expandafter{\CF_merg
elabelup}{.5}+\expandafter\CF_arrowdisplaylabela\expandafter
{\CF_mergelabeldo}{.5}-;%
\def\CF_currentnodename{end@merge@i}%
\let\CF_temp\empty
\CF_analysearrowarg{#2}%
\expandafter\CF_merged\CF_temp\_nil
}

\def\CF_mergec(#1){%
\if\relax\expandafter\noexpand\CF_firsttonil#1\_nil
\else
\CF_ifdot{#1}%
{\edef\merge_currentnodename{\CF_beforedot#1\_nil}%
\edef\merge_currentanchor{\CF_afterdot#1\_nil}%
}%
{\def\merge_currentnodename{#1}%
\let\merge_currentanchor\CF_mergeangle
}%
\if x\expandafter\CF_firsttonil\CF_mergeextreme\_nil
\pgfextracty\CF_dim{\pgfpointanchor\merge_currentno
denname\merge_currentanchor}%

```

```

\CF_expafter{\draw[]\CF_mergestyle,shorten >=0,-}([
shift=(\CF_mergeangle:\CF_arrowoffset)]\merge_currentnodenam
e.\merge_currentanchor)--(\CF_mergeextremeresult pt,\CF_dim)
;%
\else
\pgfextractx\CF_dim{\pgfpointanchor\merge_currentno
denam\merge_currentanchor}%
\CF_expafter{\draw[]\CF_mergestyle,shorten >=0,-}([
shift=(\CF_mergeangle:\CF_arrowoffset)]\merge_currentnodenam
e.\merge_currentanchor)--(\CF_dim,\CF_mergeextremeresult pt)
;%
\fi
\expandafter\CF_mergec
\fi
}

\def\CF_merged#1.#2[#3]\_nil{%
\def\CF_nextnodename{#1}%
\edef\CF_nextnodeanchor{%
\CF_ifempty{#2}
{180+\CF_mergeangle}
{#2}%
}%
\def\CF_nextnodestyle{#3}%
\let\CF_compound\empty
\CF_schemestartc
}

\def\CF_parsemergeopt#1,#2,#3,#4\_nil{%
\CF_ifempty{#1}
{\def\CF_mergefromcoeff{0.5}}
{\def\CF_mergefromcoeff{#1}}%
\def\CF_mergetocoeff{0.5}%
\def\CF_mergesplitcoeff{0.5}%
\CF_expsecond{\def\CF_mergestyle}{\CF_arrowhead}%
\CF_iffirsttokmatch\CF_quark{#2\relax}
{}
{\CF_ifempty{#2}
{\def\CF_mergetocoeff{0.5}}
{\def\CF_mergetocoeff{#2}}}%
\CF_iffirsttokmatch\CF_quark{#3\relax}
{}
{\CF_ifempty{#3}
{\def\CF_mergesplitcoeff{0.5}}
{\def\CF_mergesplitcoeff{#3}}}%
\CF_iffirsttokmatch\CF_quark{#4\relax}

```

```

        {}
        {\CF_parsemergeopta#4\_nil}%
    }%
}%
}

\def\CF_parsemergeopta#1,\CF_quark#2\_nil{%
    \CF_ifempty{#1}
    {}
    {\CF_addtomacro\CF_mergestyle{,#1}}%
}

\def\CF_mergeparsenodelist(#1){%
    \if\relax\expandafter\noexpand\CF_firsttonil#1\_nil
    \else
        \CF_ifdot{#1}%
        {\edef\merge_currentnodename{\CF_beforedot#1\_nil}
        \edef\merge_currentanchor{\CF_afterdot#1\_nil}}%
        {\def\merge_currentnodename{#1}\let\merge_currentan
        chor\CF_mergeangle}%
        \pgfextractx\CF_dim{\pgfpointanchor\merge_currentnodena
        me\merge_currentanchor}%
        \ifdim\CF_dim>\CF_mergemax
            \edef\CF_mergemax{\the\CF_dim}%
        \fi
        \ifdim\CF_dim<\CF_mergemin
            \edef\CF_mergemin{\the\CF_dim}%
        \fi
        \pgfextracty\CF_dim{\pgfpointanchor\merge_currentnodena
        me\merge_currentanchor}%
        \ifdim\CF_dim>\CF_mergeymax
            \edef\CF_mergeymax{\the\CF_dim}%
        \fi
        \ifdim\CF_dim<\CF_mergeymin
            \edef\CF_mergeymin{\the\CF_dim}%
        \fi
        \expandafter\CF_mergeparsenodelist
    \fi
}

\def\CF_grabarrownames#1[#2\_nil]{\detokenize{CF_arrow(#1)}}
\def\CF_grabarrowargs#1[#2\_nil]{#2}

\def\CF_makeparametertertext#1{%
    \toks0{}%
    \CF_cntgroup#1\relax

```

```

\CF_makeparametertertexta#1%
}

\def\CF_makeparametertertexta#1{%
  \unless\ifnum#1>\CF_cntgroup
    \toks0\expandafter{\the\toks0[###1]}%
    \expandafter\CF_makeparametertertexta\expandafter{\number
\numexpr#1+1\expandafter}%
  \fi
}

% #1 est le nombre d'arguments optionnels, #2 est le nom et #3
% le code
\def\definearrow#1#2#3{%
  \beginingroup
    \CF_makeparametertertext{#1}%
  \expandafter\endgroup
  \expandafter\def\csname\detokenize{CF_arrow(#2)}
  \expandafter\endcsname\the\toks0{#3\CF_gobtonil}%
}

\def\CF_ifdot#1{\CF_ifdota#1.\_nil}
\def\CF_ifdota#1.#2\_nil{\ifx\empty#2\empty\expandafter\CF_exec
second\else\expandafter\CF_execfirst\fi}
\def\CF_beforedot#1.#2\_nil{#1}
\def\CF_afterdot#1.#2\_nil{#2}

\def\CF_rotatenode*#1#2\_nil{%
  \CF_ifdot{#1}
  {\CF_beforedot#1\_nil}
  {#1}%
}

\def\CF_anchornode*#1#2\_nil#3{%
  \CF_ifdot{#1}
  {\CF_afterdot#1\_nil}
  {\CF_arrowcurrentangle-#390-#1}%
}

% #1 = label #2 = position #3 = + ou - (au dessus ou au desso
us) #4 : nom du noeud de dÃpart
% #5 = label #6 = position #7 = + ou - (au dessus ou au desso
us) #8 : nom du noeud de fin
\def\CF_arrowdisplaylabel#1#2#3#4#5#6#7#8{%
  \CF_doifnotempty{#1#5}
  {\path(#4)--(#8)\CF_arrowdisplaylabela{#1}{#2}{#3}\CF_a
rrowdisplaylabela{#5}{#6}{#7};}%
}

```

```

}

\def\CF_arrowdisplaylabela#1#2#3{%
  \CF_doiifnotempty{#1}
  {\if*\expandafter\CF_firsttonil\detokenize{#1}\_nil
    \ifboolKV[chemfig]{scheme debug}
    {node[pos=#2,sloped,yshift=#3\CF_arrowlabelsep,
draw,fill,cyan](shifted@node){}%
    node[draw,rotate=\CF_rotatenode#1\_nil,anchor=
\CF_anchornode#1\_nil#3,at=(shifted@node)]{\expandafter\CF_g
obarg\CF_gobarg#1}%
    }
    {node[pos=#2,sloped,yshift=#3\CF_arrowlabelsep]
(shifted@node){}%
    node[rotate=\CF_rotatenode#1\_nil,anchor=\CF_an
chornode#1\_nil#3,at=(shifted@node)]{\expandafter\CF_gobarg
\CF_gobarg#1}%
    }%
  }%
  \else
    \ifboolKV[chemfig]{scheme debug}
    {node[pos=#2,sloped,yshift=#3\CF_arrowlabelsep,
draw,fill,cyan](shifted@node){}%
    node[draw,pos=#2,anchor=-#390,sloped,yshift=#3
\CF_arrowlabelsep]{#1}%
    }
    {node[pos=#2,anchor=-#390,sloped,yshift=#3\CF_a
rrowlabelsep]{#1}%
    }
  }%
  \fi
}%
}

% pose des noeuds d'ÃcalÃs de la dimension #1 ÃÃ (\CF_arrowsta
rtnode) et (\CF_arrowendnode)
\def\CF_arrowshiftnodes#1{%
  \unless\ifdim\CF_ifempty{#1}\CF_zero{#1}=0pt
  \expanded{%
    \noexpand\path(\CF_arrowstartnode)--(\CF_arrowendno
de)%
    node[pos=0,sloped,yshift=#1](\CF_arrowstartnode1){}
    node[pos=1,sloped,yshift=#1](\CF_arrowendnode1){};}%
  \edef\CF_arrowstartnode{\CF_arrowstartnode1}\edef\CF_ar
rowendnode{\CF_arrowendnode1}%
  \fi
}

```

```

\definearrow3{->}{%
  \CF_arrowshiftnodes{#3}%
  \CF_expafter{\draw[]\CF_arrowcurrentstyle}(\CF_arrowstartno
de)--(\CF_arrowendnode);%
  \CF_arrowdisplaylabel{#1}{0.5}+\CF_arrowstartnode{#2}{0.5}-
  \CF_arrowendnode
}

\definearrow3{<-}{%
  \CF_arrowshiftnodes{#3}%
  \CF_expafter{\draw[]\CF_arrowcurrentstyle}(\CF_arrowendnode
)--(\CF_arrowstartnode);%
  \CF_arrowdisplaylabel{#1}{0.5}+\CF_arrowstartnode{#2}{0.5}-
  \CF_arrowendnode
}

\definearrow5{-/>}{%
  \CF_arrowshiftnodes{#3}%
  \CF_expafter{\draw[]\CF_arrowcurrentstyle}(\CF_arrowstartno
de)--(\CF_arrowendnode)%
  coordinate[midway,shift=(\CF_arrowcurrentangle:-1pt)](mi
dway@i)%
  coordinate[midway,shift=(\CF_arrowcurrentangle:1pt)](mi
dway@ii)%
  coordinate[at=(midway@i),shift=(\CF_ifempty{#4}{225}{#4
+180}+\CF_arrowcurrentangle:\CF_ifempty{#5}{5pt}{#5})](line@
start)%
  coordinate[at=(midway@i),shift=(\CF_ifempty{#4}{45}{#4
+\CF_arrowcurrentangle:\CF_ifempty{#5}{5pt}{#5})](line@end)%
  coordinate[at=(midway@ii),shift=(\CF_ifempty{#4}{225}{
#4+180}+\CF_arrowcurrentangle:\CF_ifempty{#5}{5pt}{#5})](lin
e@start@i)%
  coordinate[at=(midway@ii),shift=(\CF_ifempty{#4}{45}{#4
+\CF_arrowcurrentangle:\CF_ifempty{#5}{5pt}{#5})](line@end@
i);
  \draw(line@start)--(line@end);%
  \draw(line@start@i)--(line@end@i);%
  \CF_arrowdisplaylabel{#1}{0.5}+\CF_arrowstartnode{#2}{0.5}-
  \CF_arrowendnode
}

\definearrow3{<->}{%
  \CF_arrowshiftnodes{#3}%
  \CF_expafter{\draw[]\CF_arrowcurrentstyle,\CF_arrowtip-\CF_
arrowtip}(\CF_arrowstartnode)--(\CF_arrowendnode);%
  \CF_arrowdisplaylabel{#1}{0.5}+\CF_arrowstartnode{#2}{0.5}-

```

```

\CF_arrowendnode
}

\definearrow3{<=>}{%
\CF_arrowshiftnodes{#3}%
\path[allow upside down](\CF_arrowstartnode)--(\CF_arrowend
node)%
node[pos=0,sloped,yshift=\CF_arrowdoublesep](\CF_ar
rowstartnode @u0){}%
node[pos=0,sloped,yshift=-\CF_arrowdoublesep](\CF_a
rrowstartnode @d0){}%
node[pos=1,sloped,yshift=\CF_arrowdoublesep](\CF_ar
rowstartnode @u1){}%
node[pos=1,sloped,yshift=-\CF_arrowdoublesep](\CF_a
rrowstartnode @d1){};%
\beginpgfgroup
\ifboolKV[chemfig]{arrow double harpoon}
{\pgfarrowharpoontrue}
{}%
\CF_expafter{\draw[\CF_arrowcurrentstyle](\CF_arrowsta
rtnode @u0)--(\CF_arrowstartnode @u1);%
\CF_expafter{\draw[\CF_arrowcurrentstyle](\CF_arrowsta
rtnode @d1)--(\CF_arrowstartnode @d0);%
\endpgfgroup
\CF_arrowdisplaylabel{#1}{0.5}+\CF_arrowstartnode{#2}{0.5}-
\CF_arrowendnode%
}

\definearrow3{<->}{%
\CF_arrowshiftnodes{#3}%
\path[allow upside down](\CF_arrowstartnode)--(\CF_arrowend
node)%
node[pos=0,sloped,yshift=1pt](\CF_arrowstartnode @u
0){}%
node[pos=\CF_arrowdoubleposstart,sloped,yshift=-1pt
](\CF_arrowstartnode @d0){}%
node[pos=1,sloped,yshift=1pt](\CF_arrowstartnode @u
1){}%
node[pos=\CF_arrowdoubleposend,sloped,yshift=-1pt](
\CF_arrowstartnode @d1){};%
\beginpgfgroup
\ifboolKV[chemfig]{arrow double harpoon}
{\pgfarrowharpoontrue}
{}%
\CF_expafter{\draw[\CF_arrowcurrentstyle](\CF_arrowsta
rtnode @u0)--(\CF_arrowstartnode @u1);%

```

```

\CF_expafter{\draw[\CF_arrowcurrentstyle](\CF_arrowsta
rtnode @d1)--(\CF_arrowstartnode @d0);%
\endgroup
\CF_arrowdisplaylabel{#1}{0.5}+\CF_arrowstartnode{#2}{0.5}-
\CF_arrowendnode%
}

\definearrow3{<->}{%
\path[allow upside down](\CF_arrowstartnode)--(\CF_arrowend
node)%
node[pos=\CF_arrowdoubleposstart,sloped,yshift=1pt]
(\CF_arrowstartnode @u0){}%
node[pos=0,sloped,yshift=-1pt](\CF_arrowstartnode @
d0){}%
node[pos=\CF_arrowdoubleposend,sloped,yshift=1pt](
\CF_arrowstartnode @u1){}%
node[pos=1,sloped,yshift=-1pt](\CF_arrowstartnode @
d1){};%
\begingroup
\ifboolKV[chemfig]{arrow double harpoon}
{\pgfarrowharpoontrue}
{ }%
\CF_expafter{\draw[\CF_arrowcurrentstyle](\CF_arrowsta
rtnode @u0)--(\CF_arrowstartnode @u1);%
\CF_expafter{\draw[\CF_arrowcurrentstyle](\CF_arrowsta
rtnode @d1)--(\CF_arrowstartnode @d0);%
\endgroup
\CF_arrowdisplaylabel{#1}{0.5}+\CF_arrowstartnode{#2}{0.5}-
\CF_arrowendnode
}

\definearrow30{%
\CF_arrowshiftnodes{#3}%
\CF_arrowdisplaylabel{#1}{0.5}+\CF_arrowstartnode{#2}{0.5}-
\CF_arrowendnode
}

\definearrow5{-U>}{%
\CF_arrowshiftnodes{#3}%
\CF_expafter{\draw[\CF_arrowcurrentstyle](\CF_arrowstartno
de)--(\CF_arrowendnode)node[midway](Uarrow@arctangent){};%
\CF_ifempty{#4}
{\def\CF_Uarrowradius{0.333}}
{\def\CF_Uarrowradius{#4}}%
\CF_ifempty{#5}%
{\def\CF_Uarrowabsangle{60}}

```

```

        {\pgfmathsetmacro\CF_Uarrowabsangle{abs(#5)}}% ne prend
        re en compte que la valeur absolue de l'angle
        \expandafter\draw\expanded{[\CF_ifempty{#1}{draw=none}{
        \unexpanded\expandafter{\CF_arrowcurrentstyle}},-]}(Uarrow@ar
        ctangent)%
        arc[radius=\CF_compoundsep*\CF_currentarrowlength*\CF_U
        arrowradius,start angle=\CF_arrowcurrentangle-90,delta angle
        =-\CF_Uarrowabsangle]node(Uarrow@start){};
        \expandafter\draw\expanded{[\CF_ifempty{#2}{draw=none}{
        \unexpanded\expandafter{\CF_arrowcurrentstyle}}]}(Uarrow@arc
        tangent)%
        arc[radius=\CF_compoundsep*\CF_currentarrowlength*\CF_U
        arrowradius,start angle=\CF_arrowcurrentangle-90,delta angle
        =\CF_Uarrowabsangle]node(Uarrow@end){};
        \pgfmathsetmacro\CF_temp{\CF_Uarrowradius*cos(\CF_arrowcurr
        entangle)<0?"-":"+"}%
        \ifdim\CF_Uarrowradius pt>0pt
            \CF_arrowdisplaylabel{#1}{0}\CF_temp{Uarrow@start}{#2}{
            1}\CF_temp{Uarrow@end}%
        \else
            \CF_arrowdisplaylabel{#2}{0}\CF_temp{Uarrow@start}{#1}{
            1}\CF_temp{Uarrow@end}%
        \fi
    }

\def\CF_grabdelim#1#2#3\CFnil{\def\CF_leftdelim{#1}\def\CF_rig
htdelim{#2}}
\defKV[CFdelimiters]{%
    delimiters    = \CF_grabdelim#1()\CFnil,
    height        = \def\CF_delimheight{#1},
    depth         = \CF_expsecond{\CF_defifempty\CF_delimdepth{
    #1}}{\CF_delimheight},
    open xshift   = \edef\CF_leftdelimxshift{\the\dimexpr#1},
    close xshift  = \edef\CF_rightdelimxshift{\CF_ifempty{#1}{-
    \CF_leftdelimxshift}{-\the\dimexpr#1}}
}
\setKVdefault[CFdelimiters]{%
    delimiters    = (),
    height        = 10pt,
    depth         = ,
    open xshift   = 0pt,
    close xshift  = ,
    h align       = true,
    auto rotate   = false,
    rotate        = 0,
    indice        = n

```

```

}%
\def\polymerdelim{\CF_ifnextchar[\CF_polymerdelima]{\CF_polymerdelima[]}}
\def\CF_polymerdelima[#1]#2#3{%
  \restoreKV[CFdelimiters]%
  \CF_doiifnotempty{#1}{\setKV[CFdelimiters]{#1}}%
  \edef\CF_delimhalfdim{\the\dimexpr(\CF_delimheight+\CF_delimdepth)/2}%
  \edef\CF_delimvshift {\the\dimexpr(\CF_delimheight-\CF_delimdepth)/2}%
  \chemmove{%
    \nulldelimiterspace0pt
    \pgfextractx\CF_dim{\pgfpointanchor{#2}{center}}\edef
  \CF_leftdelimx{\the\CF_dim}%
    \pgfextracty\CF_dim{\pgfpointanchor{#2}{center}}\edef
  \CF_leftdelimy{\the\CF_dim}%
    \pgfextractx\CF_dim{\pgfpointanchor{#3}{center}}\edef
  \CF_rightdelimx{\the\CF_dim}%
    \pgfextracty\CF_dim{\pgfpointanchor{#3}{center}}\edef
  \CF_rightdelimy{\the\CF_dim}%
  \def\CF_autorotate{0}%
    \ifboolKV[CFdelimiters]{h align}
      {\let\CF_rightdelimy\CF_leftdelimy}
    }
    {%
      \ifboolKV[CFdelimiters]{auto rotate}
        {\pgfmathatantwo{\CF_rightdelimy-\CF_leftdelimy}
        }{\CF_rightdelimx-\CF_leftdelimx}%
        \let\CF_autorotate\pgfmathresult
      }
      {\CF_eexpsecond\CF_ifempty{\useKV[CFdelimiters]
      {rotate}}}
    }
    {\edef\CF_autorotate{\useKV[CFdelimiters]{rotate}}}%
  }%
}%
\def\CF_delim{\node[at={(\CF_leftdelimx+\CF_leftdelimxshift,\CF_leftdelimy+\CF_delimvshift)},rotate=\CF_autorotate]
  {$\left\CF_leftdelim\vrule height\CF_delimhalfdim depth\CF_delimhalfdim width0pt\right.$};%
  \node[at={(\CF_rightdelimx+\CF_rightdelimxshift,\CF_rightdelimy+\CF_delimvshift)},rotate=\CF_autorotate]
  {$\left.\vrule height\CF_delimhalfdim depth\CF_delimhalfdim width0pt\right\CF_rightdelim
  \CF_eexpsecond\CF_doiifnotempty{\useKV[CFdelimit

```

```

ers]{indice}}
        {\CF_underscore{\rlap{$\scriptstyle\useKV[C
Fdelimiters]{indice}$}}}
        $};
    }%
}

\catcode'\@11
\pgfdeclarearrow{%
    name = CF,%
    defaults = {%
        length = 3pt 5 1,%
        width' = 0pt .8,%
        inset' = 0pt .5,%
        line width = 0pt 1 1,%
        round%
    },%
    setup code = {%
        % Cap the line width at 1/4th distance from inset to ti
        p
        \pgf@x\pgfarrowlength
        \advance\pgf@x by-\pgfarrowinset
        \pgf@x.25\pgf@x
        \ifdim\pgf@x<\pgfarrowlinewidth\pgfarrowlinewidth\pgf@x
        \fi
        % Compute front miter length:
        \pgfmathdivide@{\pgf@sys@tonumber\pgfarrowlength}{\pgf@
        sys@tonumber\pgfarrowwidth}%
        \let\pgf@temp@quot\pgfmathresult%
        \pgf@x\pgfmathresult pt%
        \pgf@x\pgfmathresult\pgf@x%
        \pgf@x4\pgf@x%
        \advance\pgf@x by1pt%
        \pgfmathsqrt@{\pgf@sys@tonumber\pgf@x}%
        \pgf@xc\pgfmathresult\pgfarrowlinewidth% xc is front mi
        ter
        \pgf@xc.5\pgf@xc
        \pgf@xa\pgf@temp@quot\pgfarrowlinewidth% xa is extra ha
        rpoon miter
        % Compute back miter length:
        \pgf@ya.5\pgfarrowwidth%
        \csname pgfmathatan2@\endcsname{\pgfmath@tonumber
        \pgfarrowlength}{\pgfmath@tonumber\pgf@ya}%
        \pgf@yb\pgfmathresult pt%
        \csname pgfmathatan2@\endcsname{\pgfmath@tonumber
        \pgfarrowinset}{\pgfmath@tonumber\pgf@ya}%

```

```

\pgf@ya\pgfmathresult pt%
\advance\pgf@yb by-\pgf@ya%
\pgf@yb.5\pgf@yb% half angle in yb
\pgfmathatan@{\pgf@sys@tonumber\pgf@yb}%
\pgfmathreciprocal@{\pgfmathresult}%
\pgf@yc\pgfmathresult\pgfarrowlinewidth%
\pgf@yc.5\pgf@yc%
\advance\pgf@ya by\pgf@yb%
\pgfmathsincos@{\pgf@sys@tonumber\pgf@ya}%
\pgf@ya\pgfmathresulty\pgf@yc% ya is the back miter
\pgf@yb\pgfmathresultx\pgf@yc% yb is the top miter
\ifdim\pgfarrowinset=0pt
    \pgf@ya.5\pgfarrowlinewidth% easy: back miter is ha
lf linewidth
    \fi
    % Compute inset miter length:
    \pgfmathdivide@{\pgf@sys@tonumber\pgfarrowinset}{\pgf@s
ys@tonumber\pgfarrowwidth}%
    \let\pgf@temp@quot\pgfmathresult%
    \pgf@x\pgfmathresult pt%
    \pgf@x\pgfmathresult\pgf@x%
    \pgf@x4\pgf@x%\pgf@ya
    \advance\pgf@x by1pt%
    \pgfmathsqrt@{\pgf@sys@tonumber\pgf@x}%
    \pgf@yc\pgfmathresult\pgfarrowlinewidth% yc is inset mi
ter
    \pgf@yc.5\pgf@yc%
    % Inner length (pgfutil@tempdima) is now arrowlength -
    front miter - back miter
    \pgfutil@tempdima\pgfarrowlength%
    \advance\pgfutil@tempdima by-\pgf@xc%
    \advance\pgfutil@tempdima by-\pgf@ya%
    \pgfutil@tempdimb.5\pgfarrowwidth%
    \advance\pgfutil@tempdimb by-\pgf@yb%
    % harpoon miter correction
    \ifpgfarrowroundjoin
        \pgfarrowssetbackend{\pgf@ya\advance\pgf@x by-.5
\pgfarrowlinewidth}%
    \else
        \pgfarrowssetbackend{0pt}
    \fi
    \ifpgfarrowharpoon
        \pgfarrowssetlineend{\pgfarrowinset\advance\pgf@x b
y\pgf@yc\advance\pgf@x by.5\pgfarrowlinewidth}%
    \else
        \pgfarrowssetlineend{\pgfarrowinset\advance\pgf@x b

```

```

y\pgf@yc\advance\pgf@x by-.25\pgfarrowlinewidth}%
\ifpgfarrowreversed
\ifdim\pgfinnerlinewidth>0pt
\pgfarrowssetlineend{\pgfarrowinset}%
\else
\pgfarrowssetlineend{\pgfutil@tempdima
\advance\pgf@x by\pgf@ya\advance\pgf@x by-.25
\pgfarrowlinewidth}%
\fi
\fi
\fi
\ifpgfarrowroundjoin
\pgfarrowssettipend{\pgfutil@tempdima\advance\pgf@x
by\pgf@ya\advance\pgf@x by.5\pgfarrowlinewidth}%
\else
\pgfarrowssettipend{\pgfarrowlength
\ifpgfarrowharpoon\advance\pgf@x by\pgf@xa\fi}%
\fi
% The hull:
\pgfarrowshullpoint{\pgfarrowlength\ifpgfarrowroundjoin
\else\ifpgfarrowharpoon\advance\pgf@x by\pgf@xa\fi\fi}{
\ifpgfarrowharpoon-.5\pgfarrowlinewidth\else0pt\fi}%
\pgfarrowsupperhullpoint{0pt}{.5\pgfarrowwidth}%
\pgfarrowshullpoint{\pgfarrowinset}{\ifpgfarrowharpoon-
.5\pgfarrowlinewidth\else 0pt\fi}%
% Adjust inset
\pgfarrowssetvisualbackend{\pgfarrowinset}%
\advance\pgfarrowinset by\pgf@yc%
% The following are needed in the code:
\pgfarrowssavethe\pgfutil@tempdima
\pgfarrowssavethe\pgfutil@tempdimb
\pgfarrowssavethe\pgfarrowlinewidth
\pgfarrowssavethe\pgf@ya
\pgfarrowssavethe\pgfarrowinset
},%
drawing code = {%
\pgfsetdash{}{0pt}%
\ifpgfarrowroundjoin\pgfsetroundjoin\else
\pgfsetmiterjoin\fi
\ifdim\pgfarrowlinewidth=\pgflinewidth\else
\pgfsetlinewidth{\pgfarrowlinewidth}\fi
\pgfpathmoveto{\pgfqpoint{\pgfutil@tempdima\advance\pgf
@x by\pgf@ya}{0pt}}%
\pgfpathlineto{\pgfqpoint{\pgf@ya}{\pgfutil@tempdimb}}%
\pgfpathlineto{\pgfqpoint{\pgfarrowinset}{0pt}}%
\ifpgfarrowharpoon \else

```



- La séquence de contrôle `\setnodestyle` permet de spécifier le style des nœuds dessinés par tikz.

v0.3            2010/11/21

- 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'appliquent 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 ou "@{<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.

-----  
-----  
v0.3a            2011/01/08  
- 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.

-----  
v0.4            2011/03/07  
- `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 scanner 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

-----  
-----  
v0.4a            2011/04/10  
- Correction d'un bug concernant l'argument optionnel en début  
  de molécule.

-----  
v0.4b            2011/04/24  
- l'argument de `\chemfig` est désormais 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

-----  
-----  
v1.0            2011/06/15  
- 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.

-----  
-----  
v1.0a           2011/09/18  
- les macros \Lewis et \lewis admettent un argument optionnel  
- la macro \setlewisdist règle la distance entre les 2 électrons

-----  
-----  
v1.0b           2011/11/29  
- la commande \merge est désormais protégée entre  
\schemestart et \schemestop contre des définitions par d'autres  
packages.  
- \box est utilisé au lieu du maladroit \unhbox

-----  
-----  
v1.0c           2011/11/30  
- la macro \+ n'est plus explicitement écrite  
- vérifie que eTeX est le moteur utilisé

-----  
-----  
v1.0d           2011/12/19  
- 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.

-----  
-----  
v1.0e           2012/01/13

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

-----  
-----  
v1.0f        2012/02/24

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

-----  
-----  
v1.0g        2012/11/16

- 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'aurait pas été appliquée aux sous-cycles.
- réécriture 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.

-----  
-----  
v1.0h        2013/11/28

- \chemname admet maintenant une version étendue qui ne tient pas compte des profondeurs primitives.
- \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

v1.1            2015/02/13

- correction d'un bug dans `\CF_seeksubmol` : la macro `\CF_molecule` est d'abord poussée 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 d'abord prendre `\CF_schemenest`.
- la jonction entre deux liaisons consécutives dans l'axe peut être activée avec `\enablebondjoin` et désactivée avec `\disablebondjoin` (préférable, l'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 désactivée dans les cycles afin que les polygones soient réguliers. `\enablefixedbondlength` permet cette fonctionnalité pour toutes les macros `\chemfig` (même non étiquetées) tandis que `\disablefixedbondlength` le désactive.

-----

-----

v1.1a           2015/02/23

- 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 d'limité avant "(" pour absorber tous les #.

-----

-----

v1.2            2015/10/08

- 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 puisque définies avec `\pgfarrowsdeclare`.
- flèche "-U>" corrigée : le placement des labels est maintenant correct dans tous les cas. Ainsi :  

$$-U>[\langle a \rangle][\langle b \rangle][\langle d \rangle][r][a]$$

```

        place le label <a> près du début de la flèche, quels q
ue
        soient les signes du rayon r et de l'angle a.
- \chemrel, \setchemrel et \chemsign sont supprimés.
- compatibilité, avec les limitations d'usage, avec la
  librairie "externalize" : le \begin{tikzpicture} voit
  désormais le \end{tikzpicture} correspondant dans la mac
ro
  \CF_chemfigb.
-----
-----
v1.2a      2015/10/21
- erreur de copier-coller dans le code: une adresse url était
  malencontreusement présente en plein milieu d'une ligne
  de
  code
-----
-----
v1.2b      2015/11/15
- bug dans \CF_seeksubmol qui laissait "*" dans le flux de
  lecture de TeX. Un message d'erreur est également ajouté
  l'
  en cas de "!" en fin de traitement.
- correction d'un bug dans \CF_setbondangle où l'angle [<:
  a>]
  n'était pas évalué par \pgfmathsetmacro.
-----
-----
v1.2c      2015/11/20
- Correction d'un bug dans \CF_setbondangle : l'angle renvoyé
  pouvait être négatif
- Correction d'un bug dans \CF_directarrow : la macro \CF_i
  fempty
  n'est pas correctement développée dans l'argument de
  \pgfpointanchor
-----
-----
v1.2d      2015/12/01
- correction d'un bug dans la flèche "-U"
- la version étoilée de \setcrambond dessine les liaisons
  de
  Cram en pointillés sous forme de trait large et non pas
  sous
  forme de triangle.

```

```

-----
-----
v1.2e      2017/05/20
- la macro contenant la définition d'une flèche est
  désormais "\CF_arrow(<nom>)", ainsi la macro \0 n'est plu
  s
  définie par \definearrow
- remerciements rajoutés après une suppression induite, pou
  r ne
  froisser aucune susceptibilité
-----
-----
v1.3      2018/03/08
- tous les paramètres sont désormais passés via
  \setchemfig qui
  fait appel à "simplekv". Par conséquent, _toutes_ les m
  acros qui
  étaient des paramètres deviennent obsolètes, à savo
  ir :
    \setcrambond, \setatomsep, \setbondoffset,
  \setdoublesep,
    \setangleincrement, \enablefixedbondlength,
    \disablefixedbondlength, \setnodestyle, \setbondstyle
  ,
    \setlewis, \setlewisdist, \setstacksep,
  \setcompoundstyle,
    \setarrowdefault, \setandsign, \setarrowoffset,
    \setcompoundsep, \setarrowlabelsep, \enablebondjoin,
    \disablebondjoin et \schemedebug.
  et ces macros seront *supprimées* dans une future versio
  n.
- la version à l'œuvre "\chemfig*" et les deux arguments op
  tionnels
  de la macro "\chemfig[[]]" sont également optionnels et
  seront
  *supprimés* dans une future version afin d'accroître la
  syntaxe
  \chemfig[clés=valeurs]{code}
- 6 nouveaux paramètres : "lewis radius", "arrow double se
  p",
  "arrow double coeff", "arrow double harpoon", "cycle radi
  us
  coeff", "arrow head".
- correction d'un bug dans \CF_parsemergeopt qui dans certa
  ins
  cas, envoyait vers l'affichage des caractères

```

- petit toilettage du code
- macro `\polymerdelim` (non documenté) expérimentale et en phase de tests
- suppression d'un registre d'écriture de fichier

---

-----

v1.31            2018/04/05

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

---

-----

v1.32            2018/08/23

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

---

-----

v1.33            2018/10/31

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

---

-----

v1.34            2019/02/23

- bug dans la flèche "`<->`" corrigé

---

-----

v1.4             2019/04/18

- corrections de nombreux bugs
- caractère privé "\_" et non plus "@" -> modifications à prévoir notamment dans la doc avec les codes spécifiques aux flèches, à la
- risque de couiner sur [tex.stackexchange.com](https://tex.stackexchange.com)
- anciennes macros abandonnées et désormais définies :  
`\setcrambond`, `\setatomsep`, `\setbondoffset`,  
`\setdoublesep`,  
`\setangleincrement`, `\enablefixedbondlength`,  
`\disablefixedbondlength`, `\setnodestyle`, `\setbondstyle`  
, `\setlewis`, `\setlewisdist`, `\setstacksep`,  
`\setcompoundstyle`,  
`\setarrowdefault`, `\setandsign`, `\setarrowoffset`,  
`\setcompoundsep`, `\setarrowlabelsep`, `\enablebondjoin`,  
`\disablebondjoin` et `\schemedebug`
- l'ancienne syntaxe `\chemfig[][]{}` est abandonnée et n'est

```

t plus
acceptÃle, dÃsormais c'est
\chemfig[<clÃs>=<valeurs>]{<code molÃcule>}
- l'ancienne syntaxe \lewis[<coeff>] ou \Lewis[<coeff>] n'e
st
plus acceptÃle au profit de \lewis[<clÃs>=<valeurs>]
-----
-----
v1.41      2019/05/21
- utilisation de la nouvelle primitive \expanded
- nouvelle clÃ "h align" (true par dÃfaut) pour les dÃli
miteurs
  de \polymerdelim. Lorsque Ã false, les dÃlimateurs ne s
ont
  plus alignÃs horizontalement mais positionnÃs aux noeu
s demandÃs
- nouvelle clÃ "auto rotate" qui n'a de sens que si h alig
n=false :
  les dÃlimateurs sont automatiquement inclinÃs
- nouvelle clÃ "rotate" qui n'a de sens que si halign=fals
e ET
  auto rotate=false : l'inclinaison des dÃlimateurs peut Ã
Ãtre
  choisie
-----
-----
v1.5       2020/03/05
- nouvelles macros \charge et \Charge. Les macros \lewis et
\Lewis
  sont obsolÃtes et amenÃes Ã disparaÃtre Ã 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 corrigÃ dans \CF_seeknode
- ajout d'une section dans le manuel (placement des atomes)
-----
-----
v1.51      2020/04/06
- bug corrigÃ dans \chargerec_a et \chargeline_a
-----
-----
v1.52      2020/04/14
- bug : dÃfinition corrigÃe de \CFthesubmol dans \def_sub
molc pour

```

qu'elle se d  veloppe en 1 coup seulement

---

-----

v1.53            2020/04/27

- mise    jour en fonction des nouvelles fonctionnalit  s d
- e
- l'extension simplekv
- bug : \CF\_ifzerodim interrompt maintenant le trac   dans
- la \hbox

---

-----

v1.54            2020/05/21

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

---

-----

v1.55            2020/06/15

- chemfig est incompatible avec conTeXt, vu que ce moteur r
- ed  finit
- des primitives telles que \expanded, \unexpanded et peut
-   tre
- d'autres.

---

-----

v1.56            2020/07/13

- le centre des cycles est d  sormais accessible via un noe
- ud
- sp  cifique pour chacun d'eux.

---

-----

v1.6             2021/02/26

- les macros des formules de Lewis sont retir  es et plac  
- es dans
- le fichier s  par   "lewis.tex" que l'utilisateur peut ch
- arger
- s'il le souhaite
- ajout d'une cl   <debug> pour le trousseau [chemfig]
-    l'int  rieur d'un sch  ma, le token '#' est permis dan
- s
- l'argument de \chemfig

---

-----

v1.6a            2021/02/28

- le fichier lewis.tex a   t   renomm   chemfig-lewis.tex

---

-----

v1.6b            2021/08/01  
- encodage UTF-8  
- la macro \# n'Ãtait pas dÃfinie pour remplacer Ân#(...)  
Âž lorsque  
  \chemfig se trouve dans l'argument d'une macro.