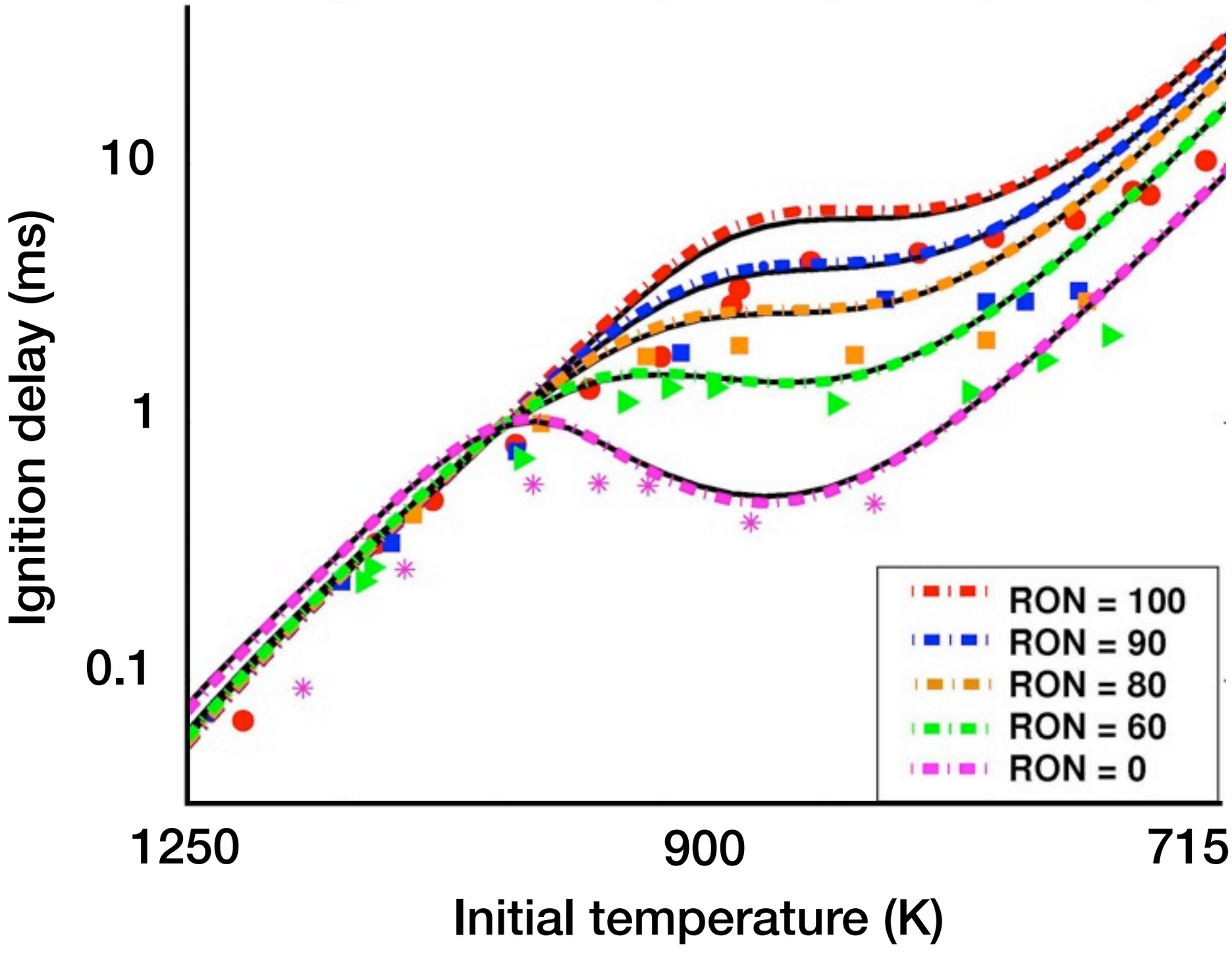
Introduction to RMG-Py

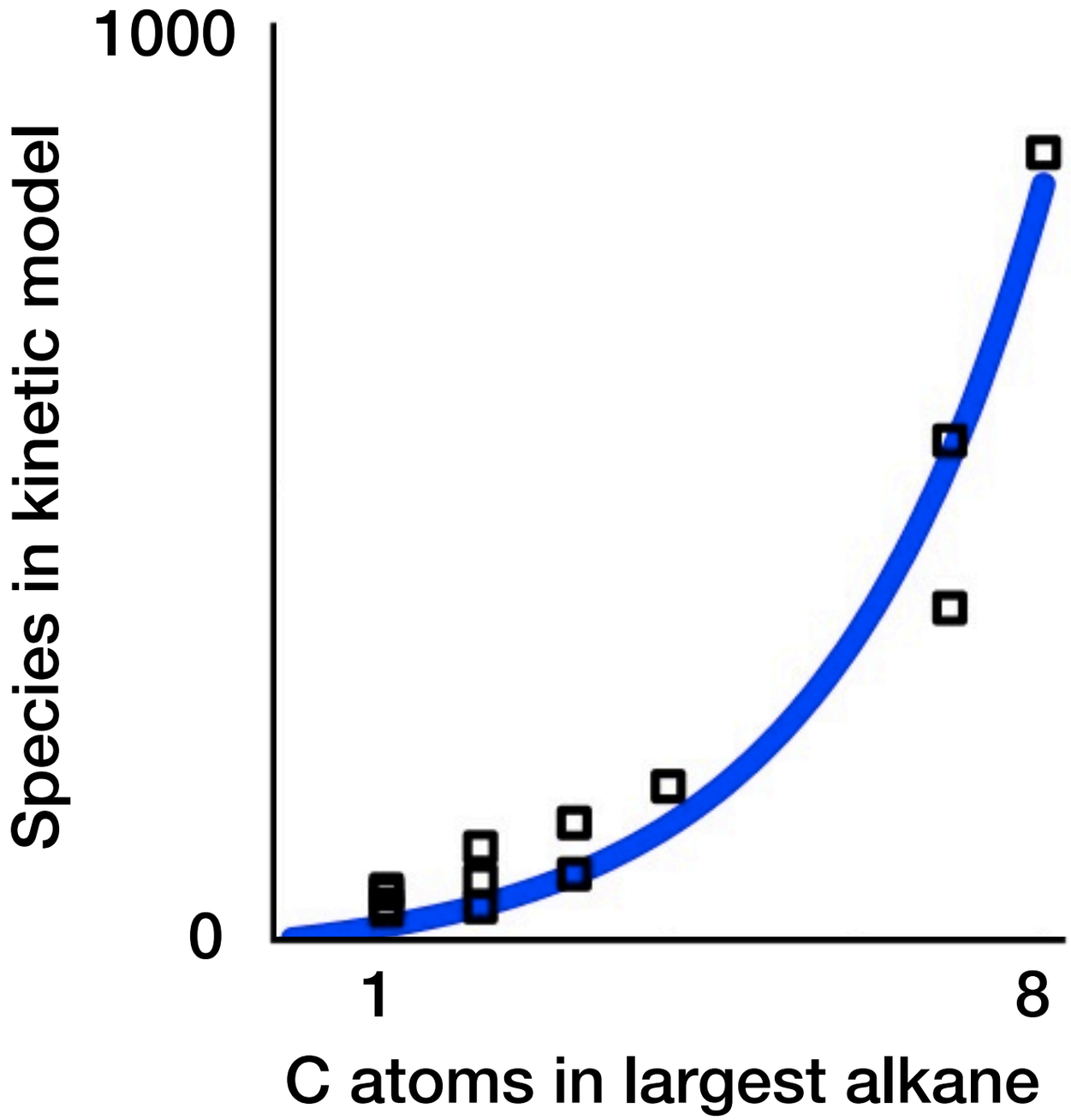
RMG Study Group

February 2, 2015

Combustion chemistry is complex



Modern kinetic models are large



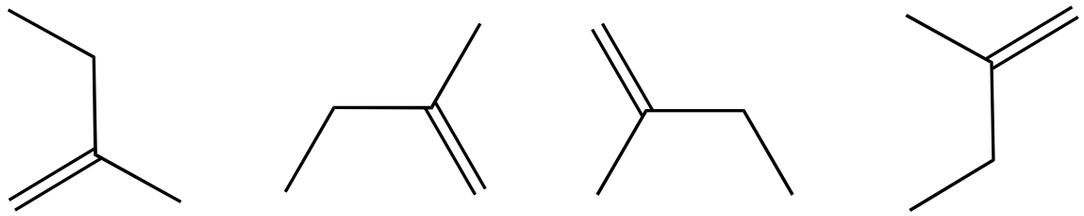
We want to build ***predictive*** kinetic models for any fuel in any conditions.

Use a Reaction Mechanism Generator

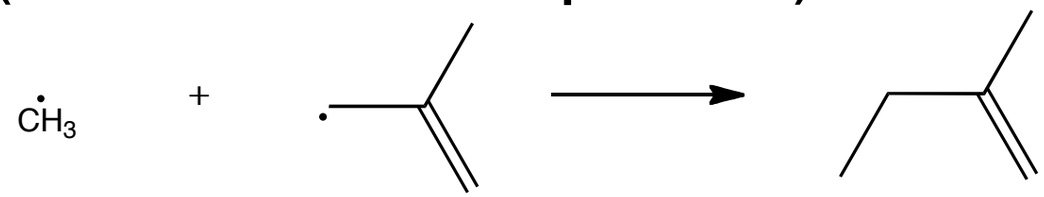
rmg.mit.edu
rmg.coe.neu.edu
facebook.com/rmg.mit

Reaction mechanism generators need to...

1. Represent molecules (and identify duplicates)



2. Create reactions (and then new species)

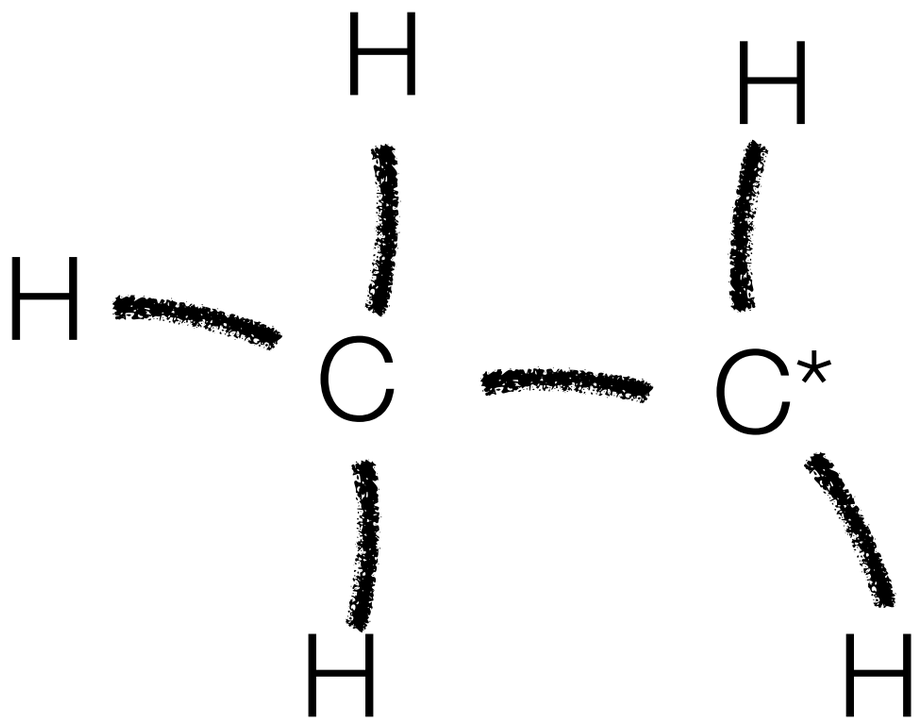


3. Estimate thermo and kinetic parameters (quickly!)

4. Choose what to include and what to leave out

1. Represent Molecules:
2-dimensional graphs

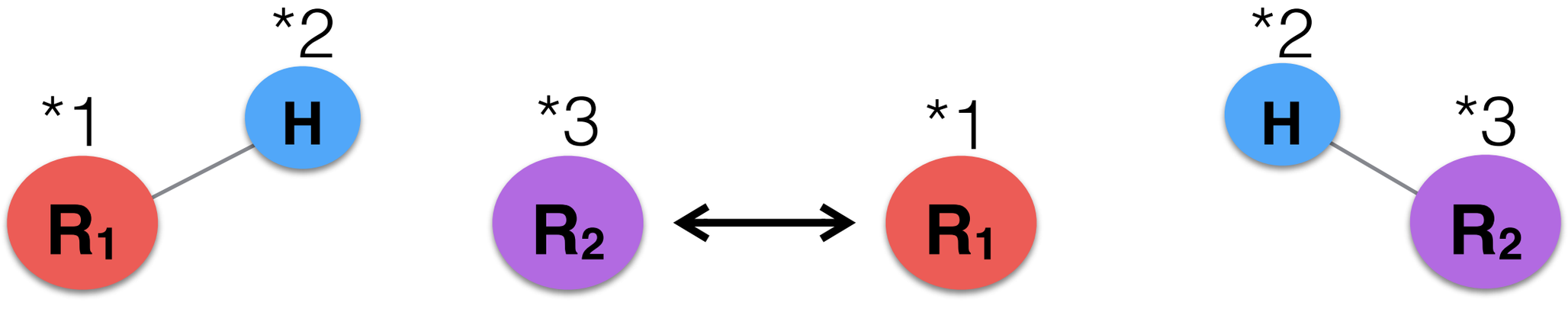
$CH_3CH_2\cdot$ =



multiplicity 2
1 C u0 p0 c0 {2,S} {3,S} {4,S} {5,S}
2 C u1 p0 c0 {1,S} {6,S} {7,S}
3 H u0 p0 c0 {1,S}
4 H u0 p0 c0 {1,S}
5 H u0 p0 c0 {1,S}
6 H u0 p0 c0 {2,S}
7 H u0 p0 c0 {2,S}

2. Create reactions: Using reaction families

- **Template** for recognizing reactive sites

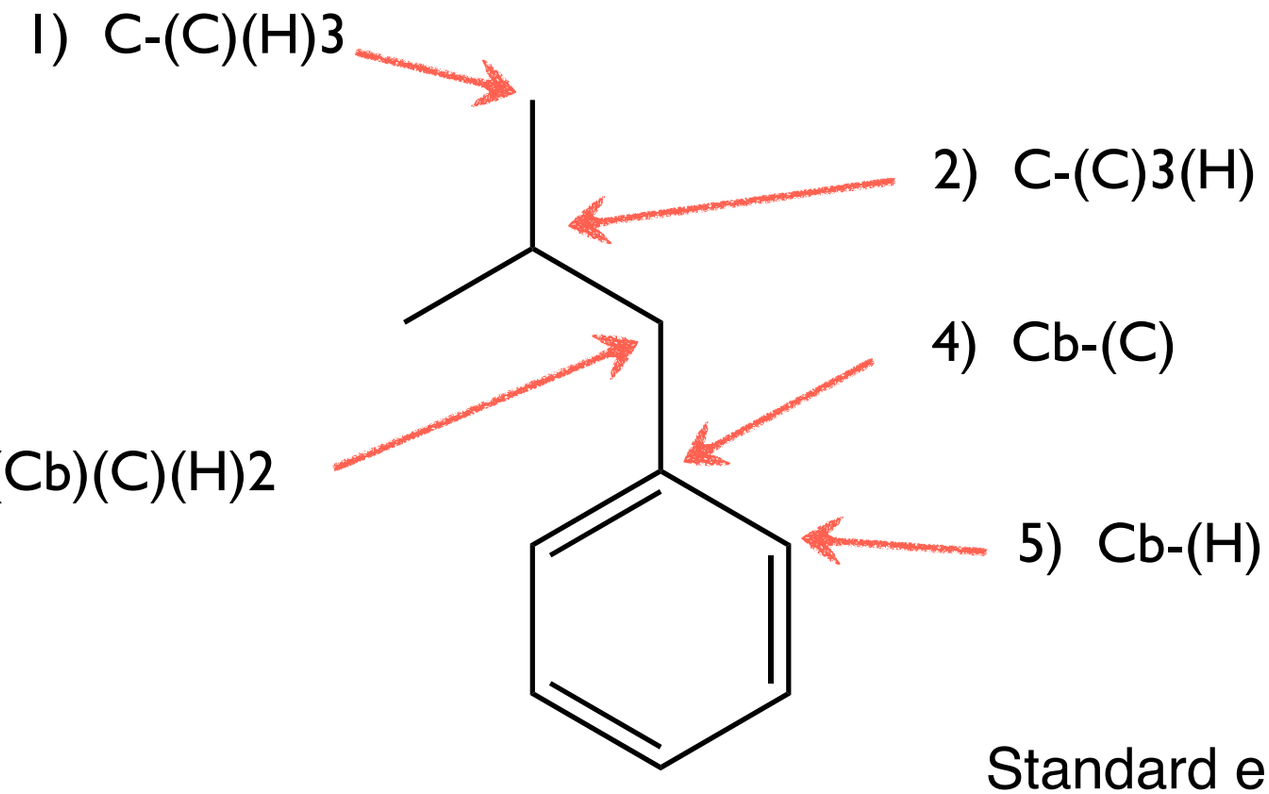


- **Recipe** for changing the bonding at the site

Recipe:
- Break bond {*1, S, *2}
- Form bond {*2, S, *3}
- Gain radical {*1, 1}
- Lose radical {*3, 1}

- **Rules** for estimating the rate

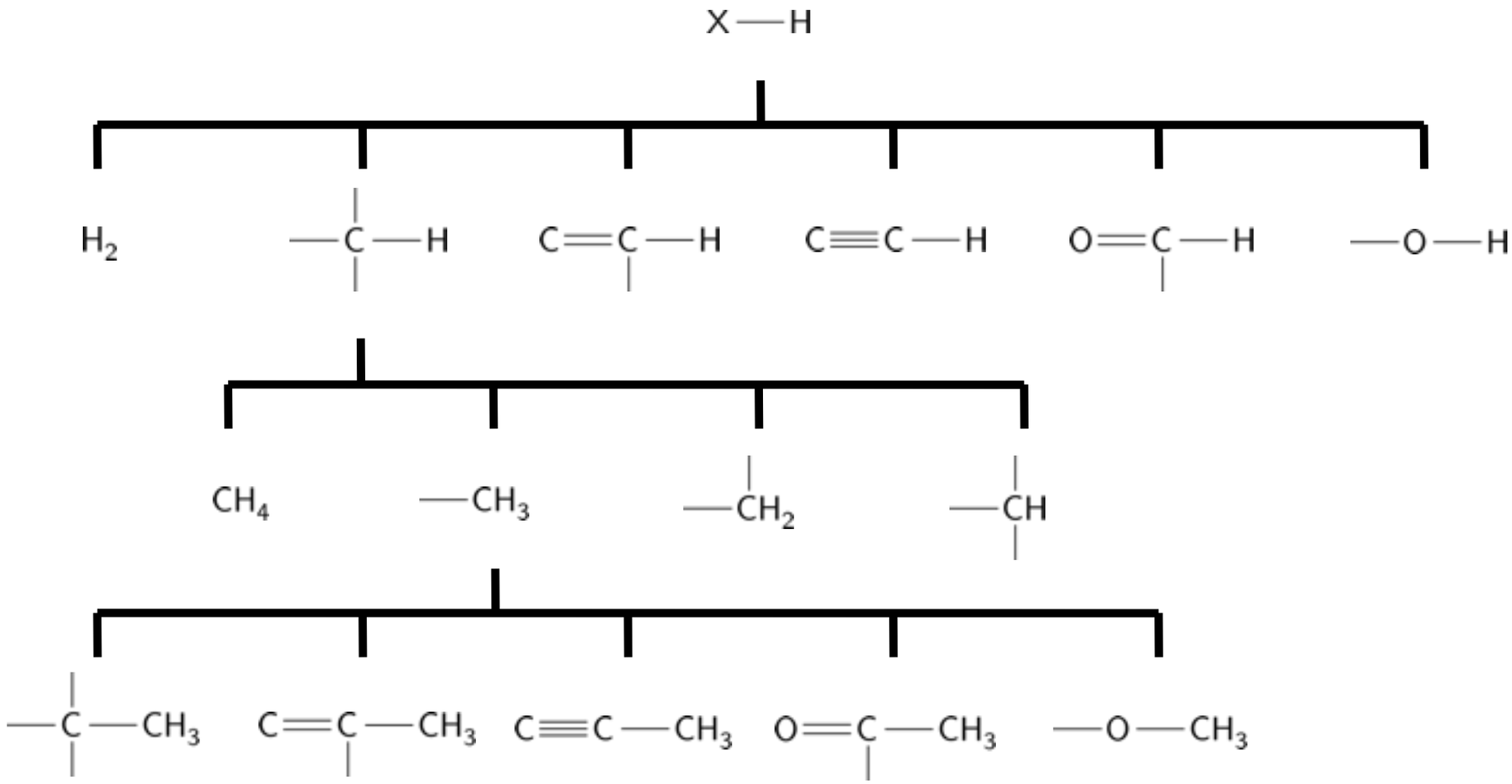
3a. Estimate thermo:
Using Benson's group contributions



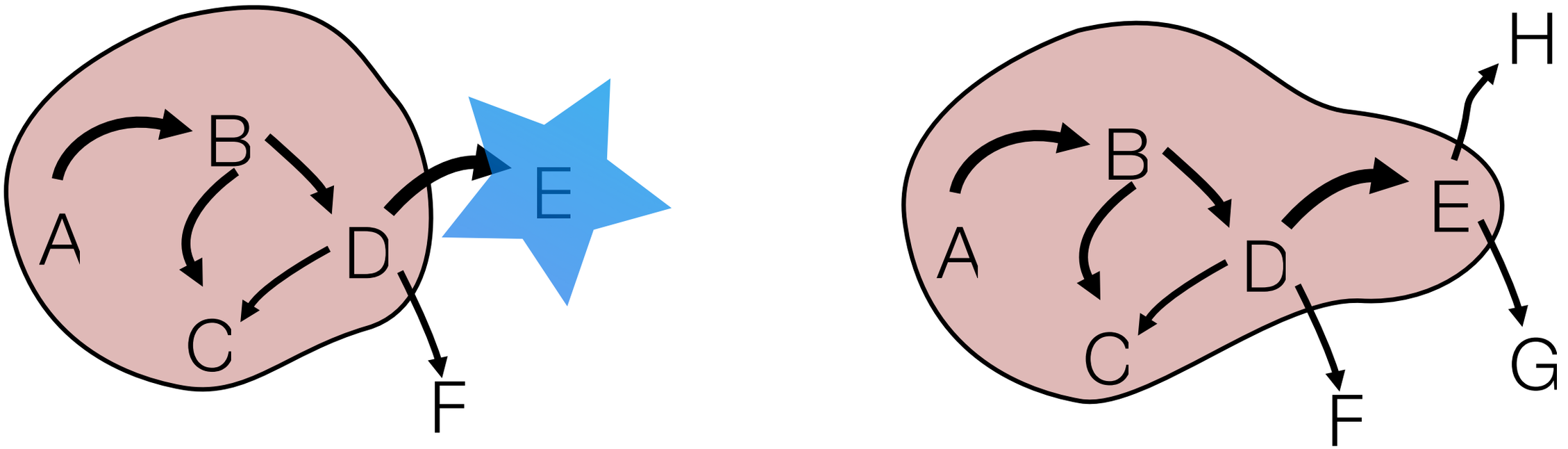
1) 2(-10.20)
2) -1.90
3) -4.86
4) 5.51
5) 5(3.30)

Standard enthalpy : -5.15 kcal/mole of isobutylbenzene

3b: Estimate kinetics:
Using rules based on hierarchical trees

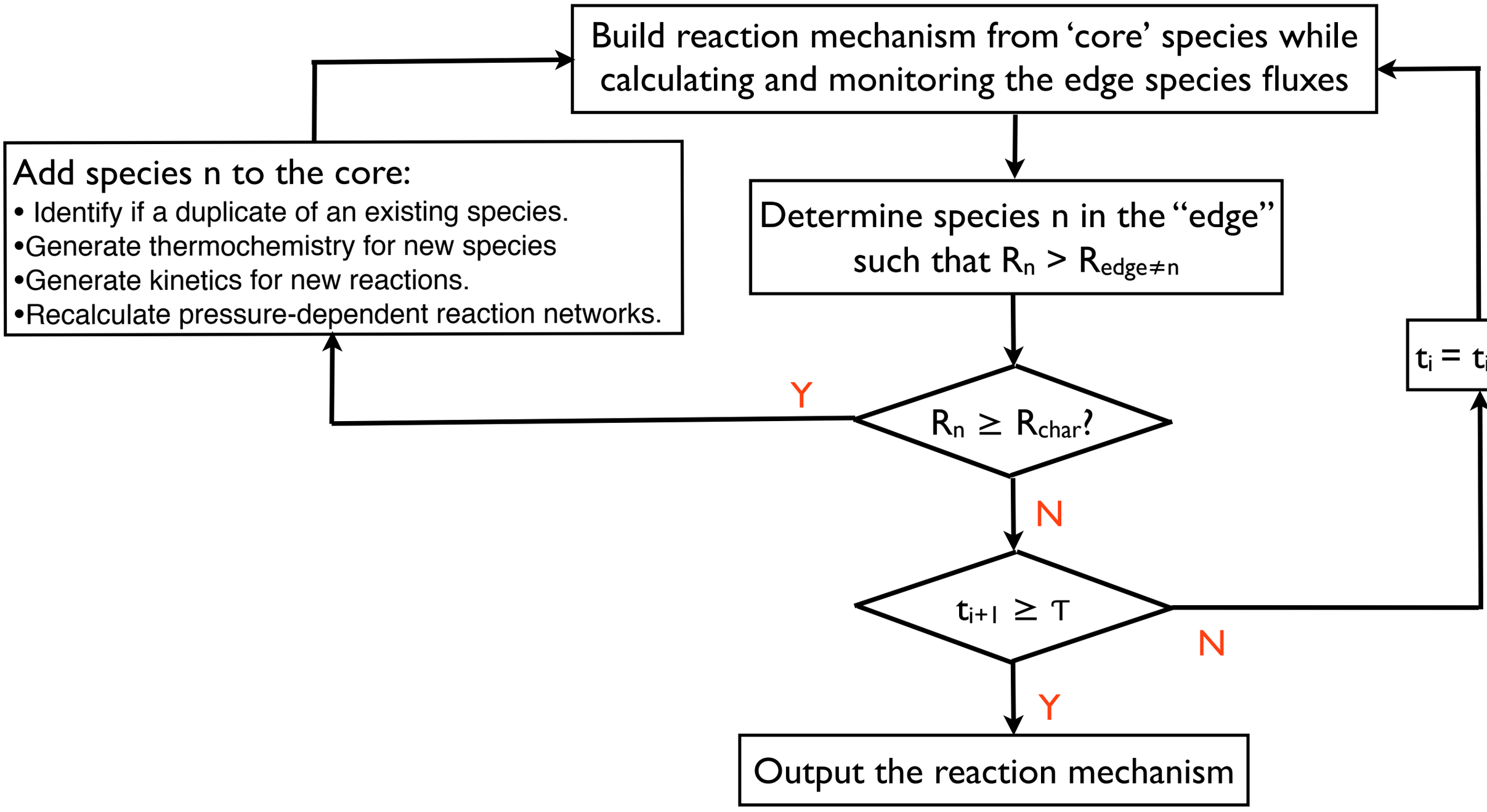


4. Choose what to include:
Rate-based expansion of the model core



Whether species “E” is brought in depends on rate of formation of species in the core and a user-defined tolerance

RMG's rate-based algorithm



Now more details...

- How to define species
- Controlling the flux-based expansion algorithm
  - Reactor conditions, Tolerance, Pruning
- Controlling how to get parameters
  - Seeds, Libraries, Estimates
- Extra features
  - Constraints, Pressure dependence, QMTP, Liquid Phase

Defining Species:
Input file syntax

```
species(
    label='n-butane',
    reactive=True,
    structure=smiles("CCCC"),
)
species(
    label='N2',
    reactive=False,
    structure=InChI("InChI=1/N2/c1-2"),
)
```

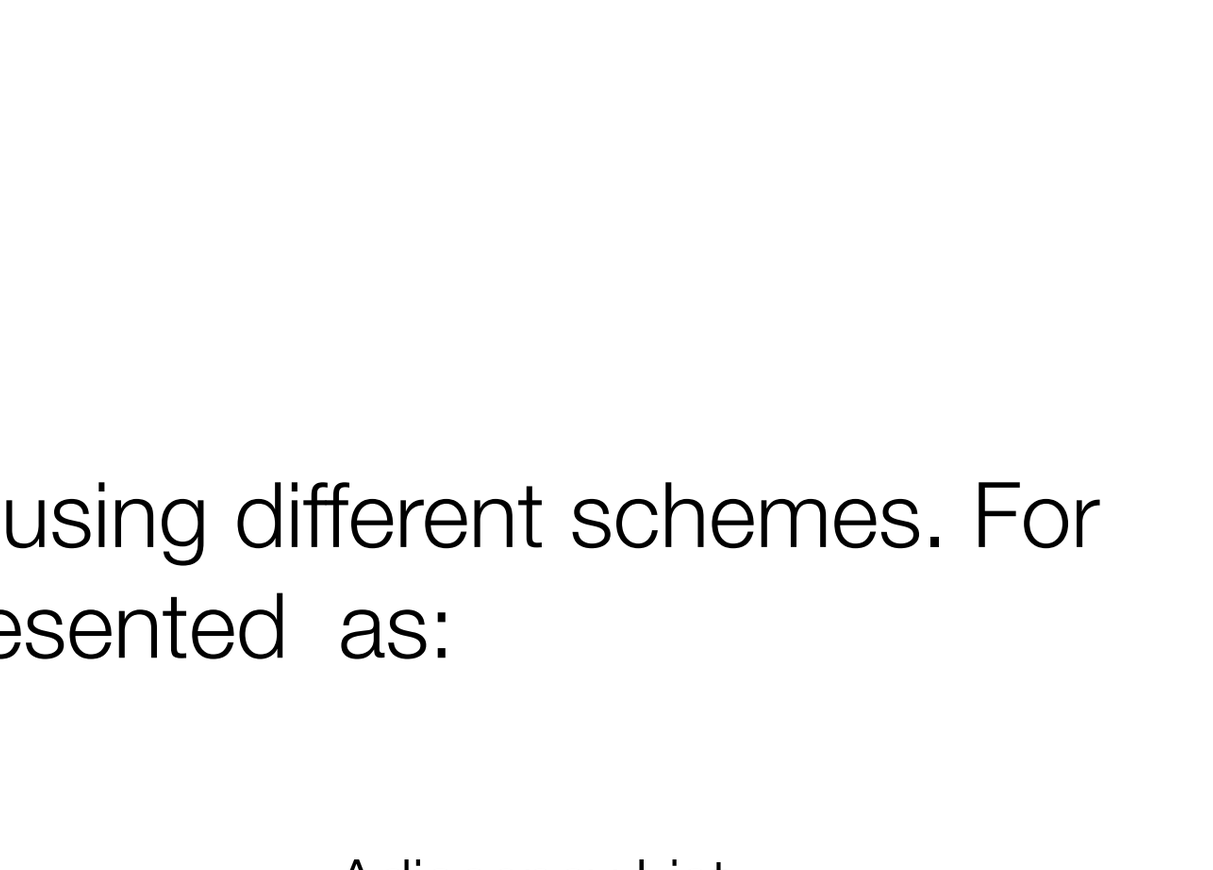
Defining Species:
Chemical Identifiers

Molecules can be described using different schemes. For example Ethene can be represented as:

SMILES
C=C

InChI
1S/C2H4/c1-2/h1-2H2

Adjacency List



Controlling the flux-based expansion: Reactor conditions

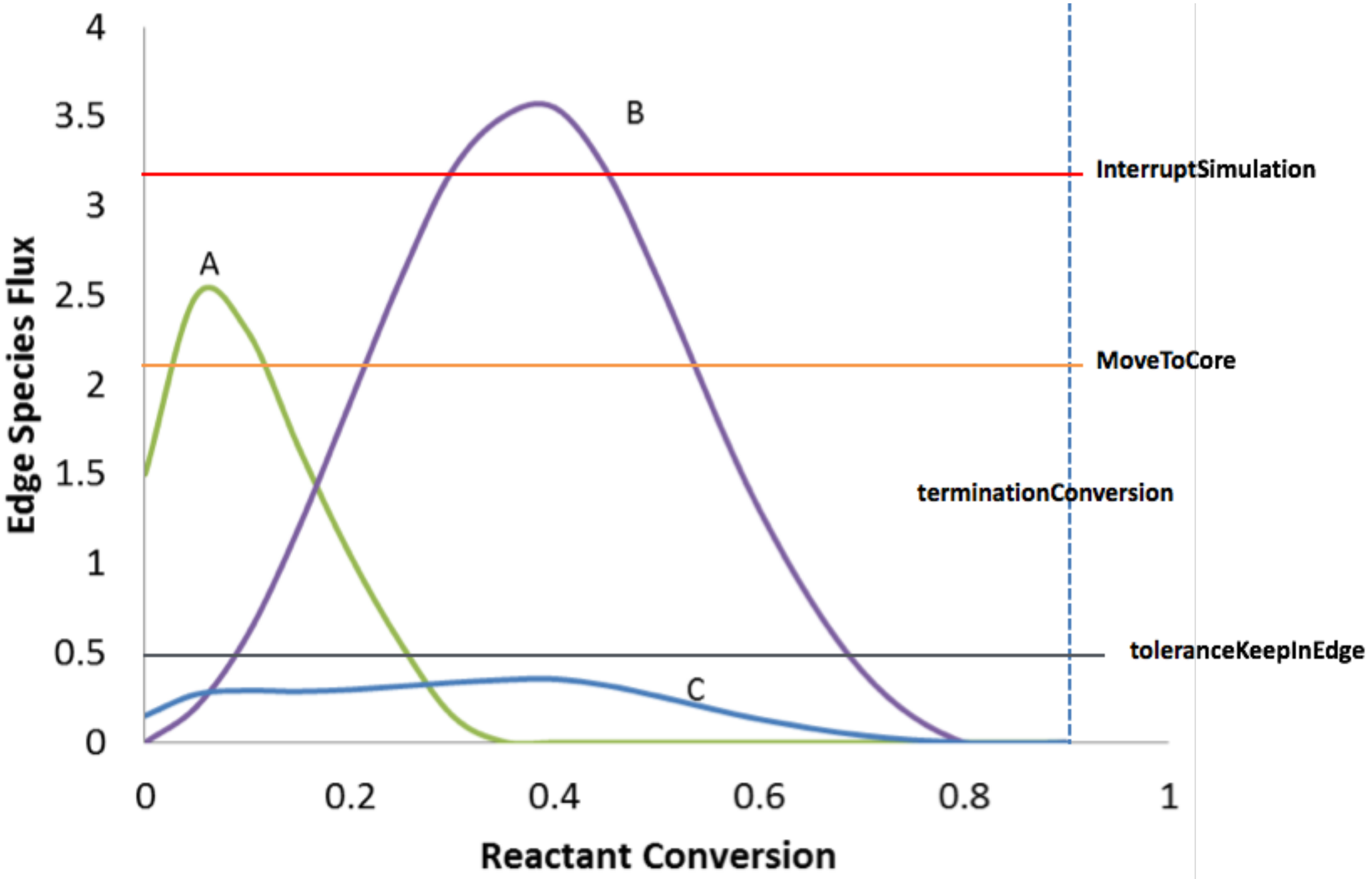
```
simpleReactor(
    temperature=(1200,‘K’),
    pressure=(1.0,‘bar’),
    initialMoleFractions={
        ‘n-butane’: 0.1,
        ‘N2’: 0.5,
    },
    terminationConversion={
        ‘n-butane’: 0.5,
    }
    terminationTime=(1e-1,‘s’),
)
```

Controlling the flux-based expansion: Tolerance and Pruning

```
simulator(
    atol=1e-16,
    rtol=1e-8,
)


model(
    toleranceKeepInEdge=0.1,
    toleranceMoveToCore=0.5,
    toleranceInterruptSimulation=0.9,
    maximumEdgeSpecies=100000
)
```

Controlling the flux-based expansion: Tolerance and Pruning



Kehang Han, RMG study group, 7/10/14. http://cheme.scripts.mit.edu/green-group/rmg/

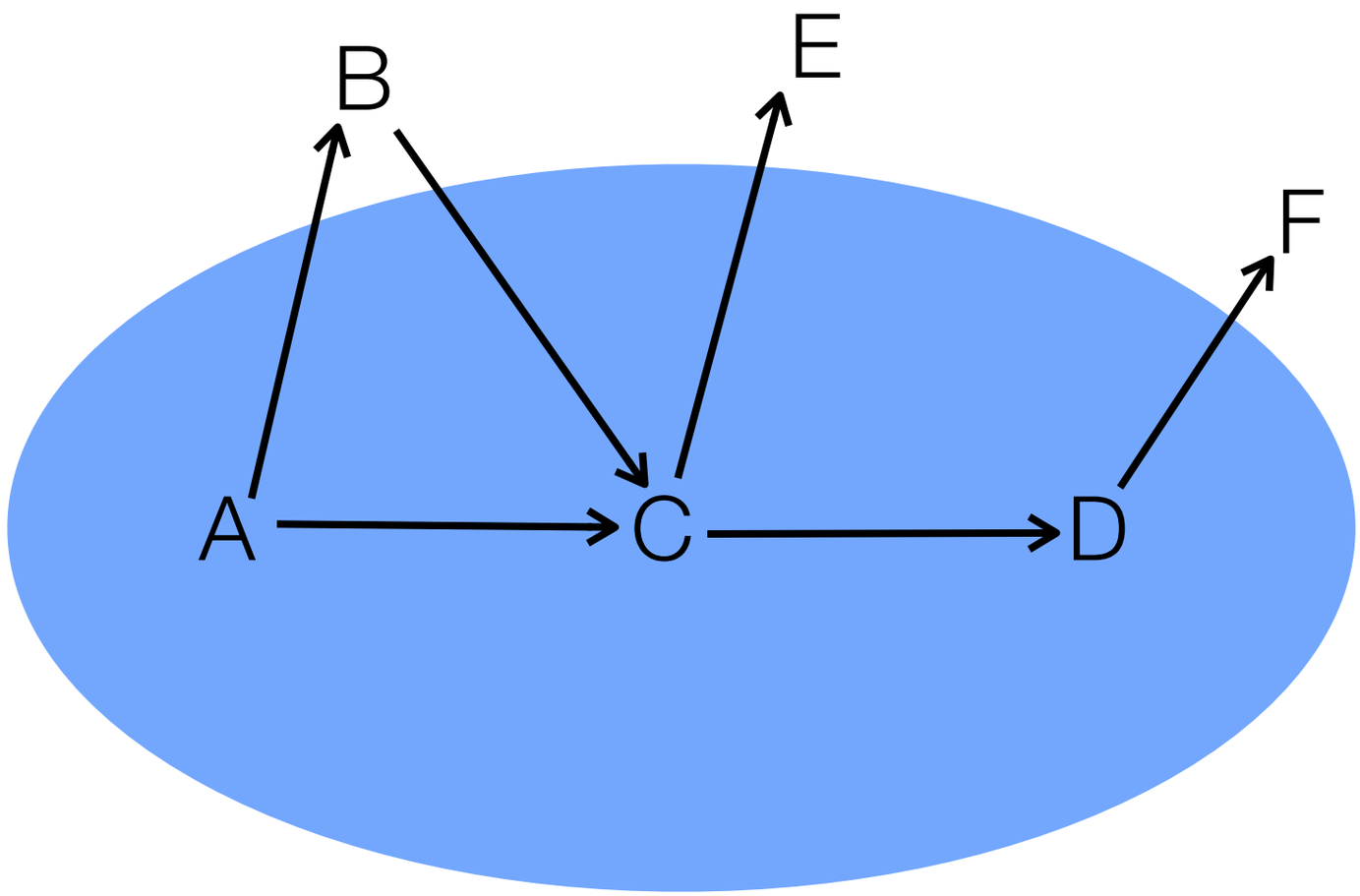
Controlling how to get parameters: Libraries and Seeds

```
database(
    thermoLibraries = ['GRI-Mech3.0', 'DFT_QCI_thermo'],
    reactionLibraries = [],
    seedMechanisms = [],
    kineticsDepositories = ['training'],
    kineticsFamilies = ['!Intro_Disproportionation', '!Substitution_O'],
    kineticsEstimator = 'rate rules',
)
```

See http://rmg.mit.edu/database/
for other libraries and depositories

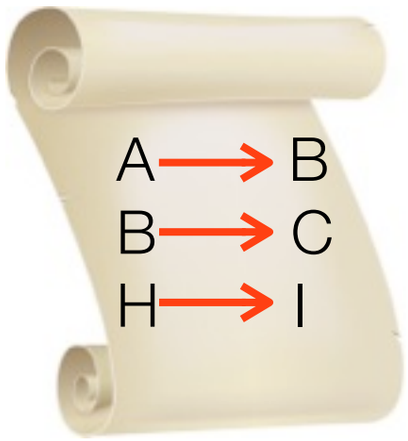
Controlling how to get parameters: Libraries and Seeds

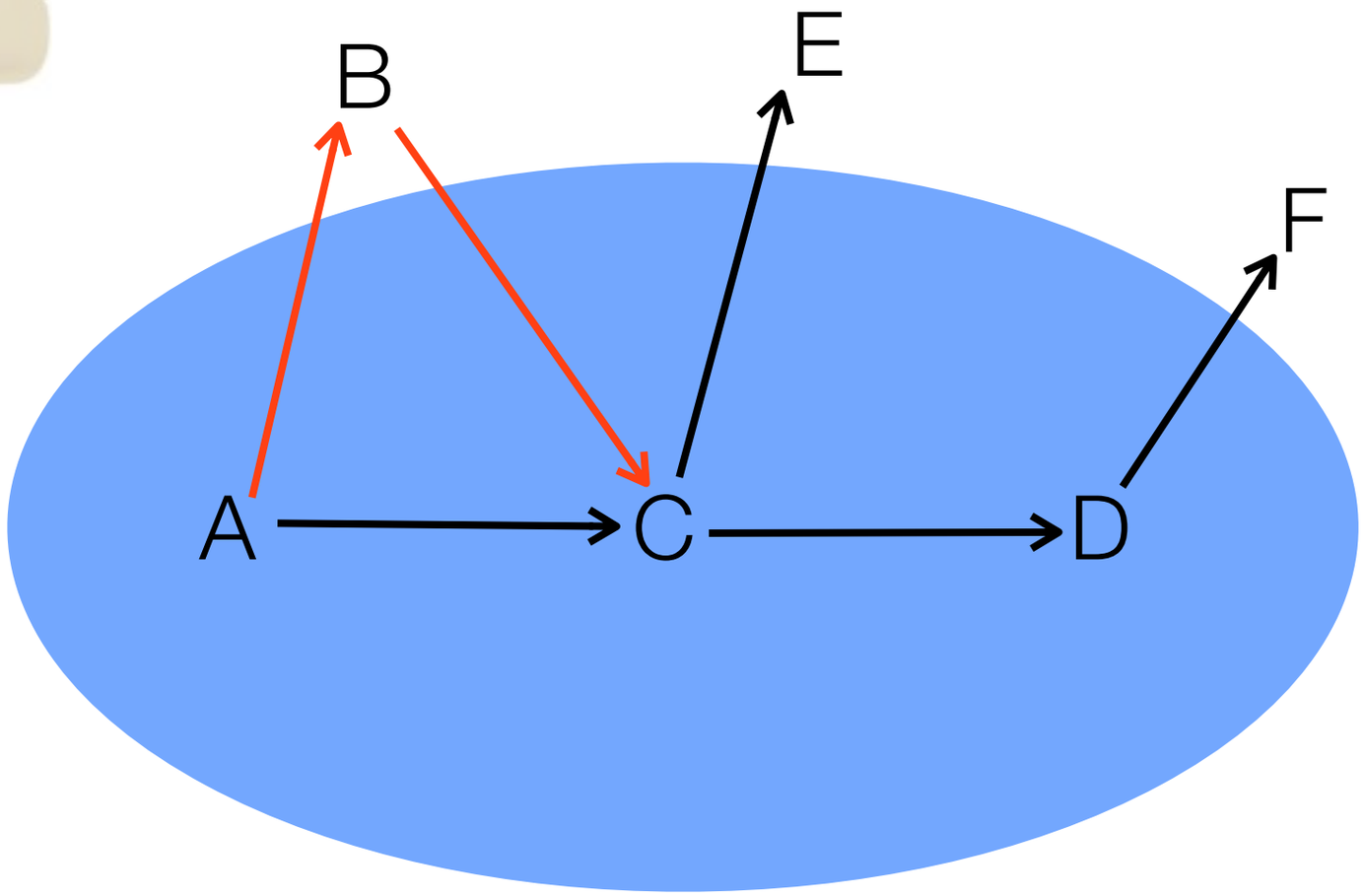
General RMG Algorithm



Controlling how to get parameters: Libraries and Seeds

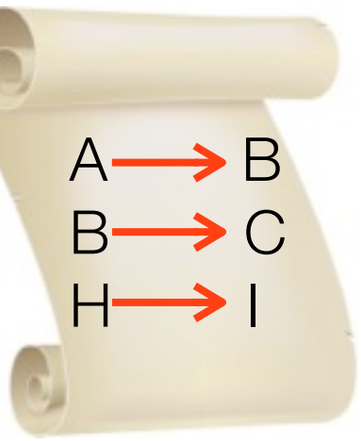
Reaction library

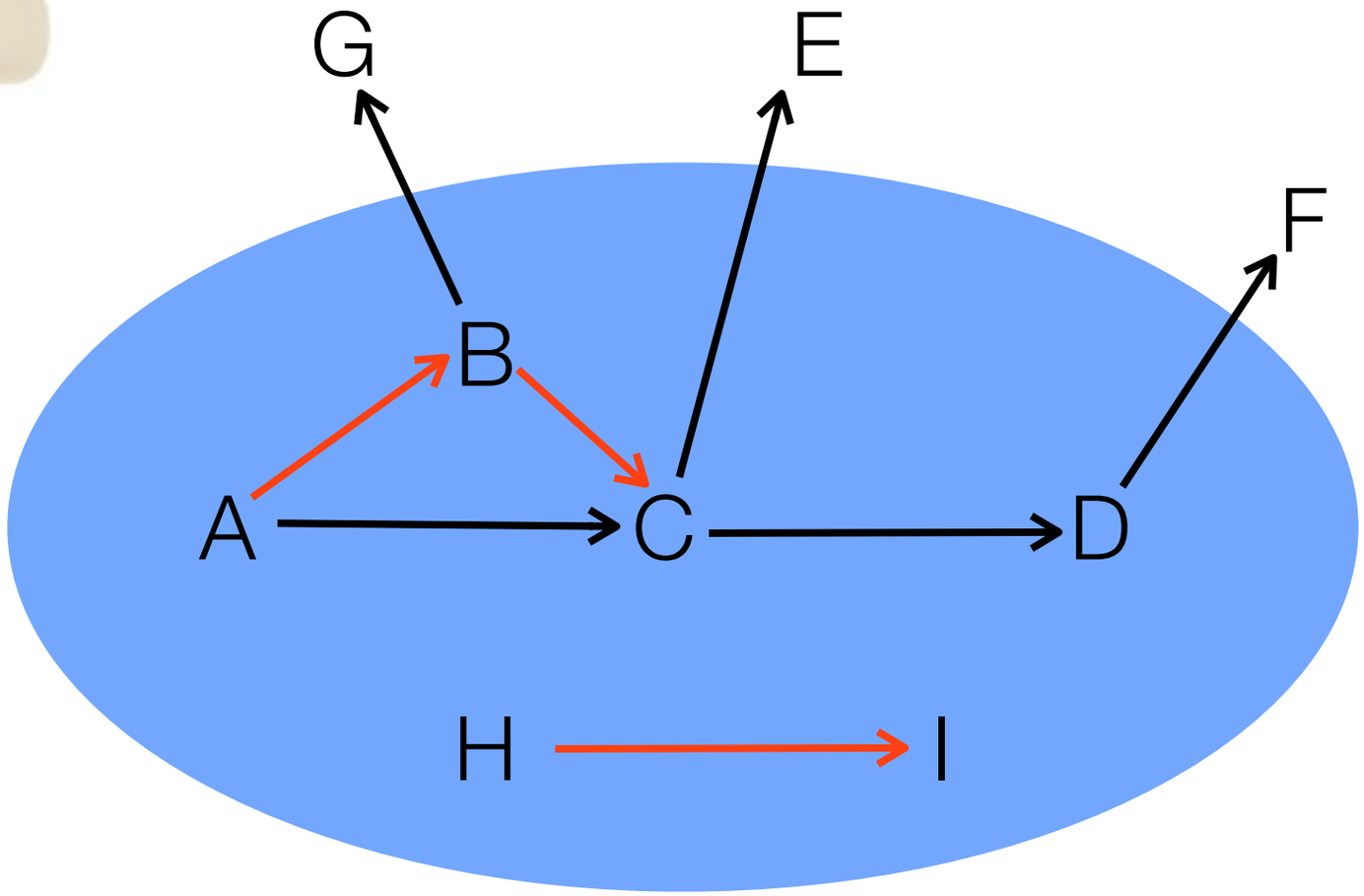




Controlling how to get parameters: Libraries and Seeds

Seed mechanism



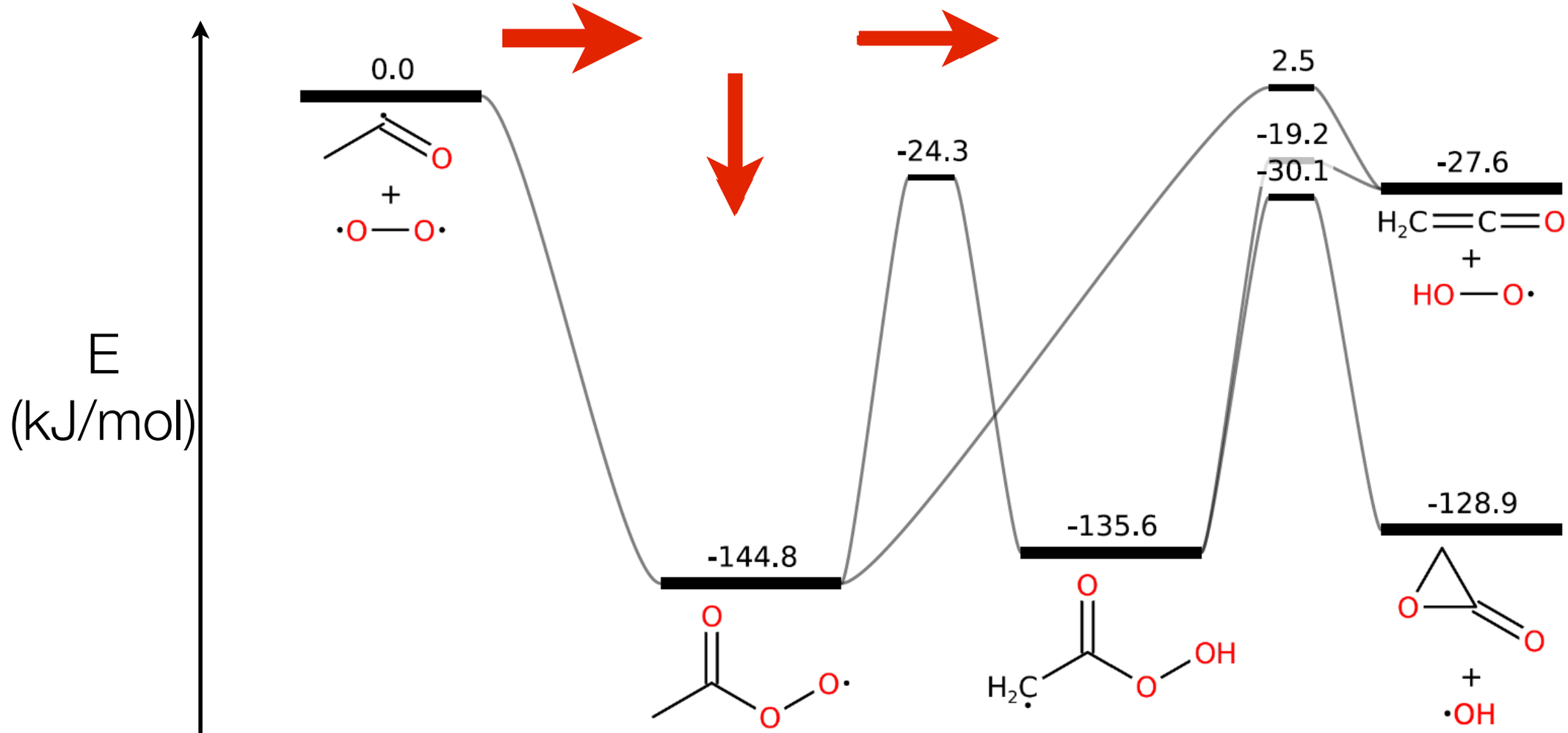


Extra features:
Species Constraints

```
generatedSpeciesConstraints(
    maximumRadicalElectrons = 2,
)
```

maximum<Element>Atoms (e.g. Carbon, Sulfur)
maximumHeavyAtoms
maximumRadicalElectrons

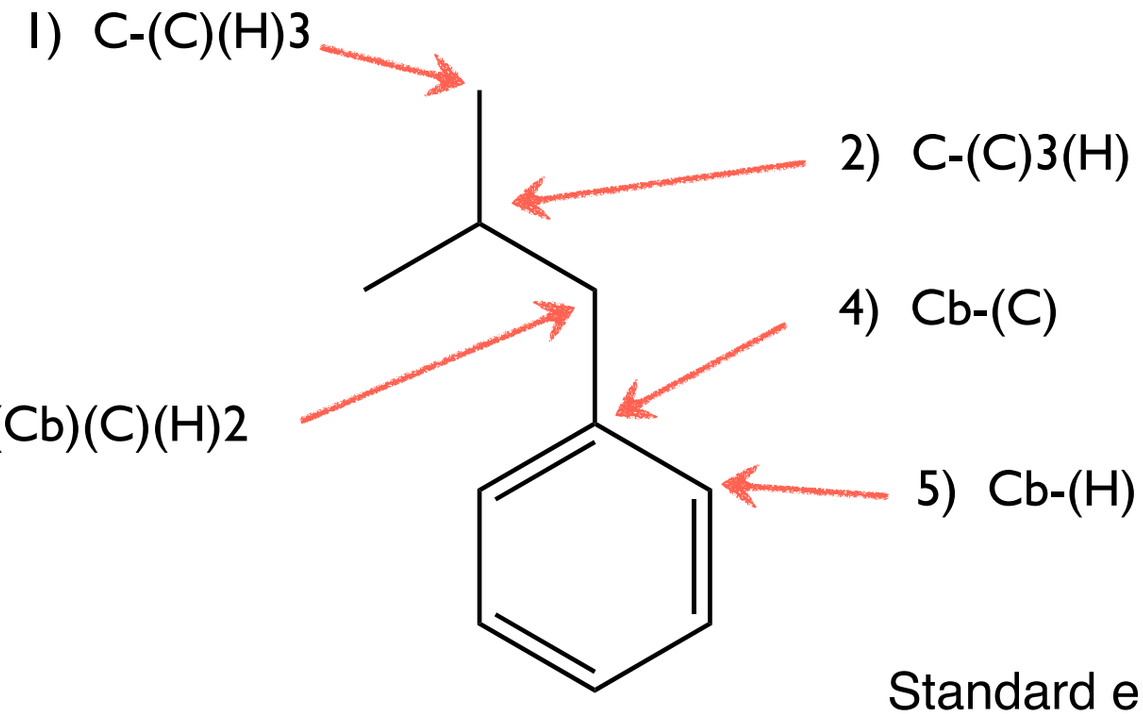
Extra features:
Pressure Dependence



Extra features:
Pressure Dependence

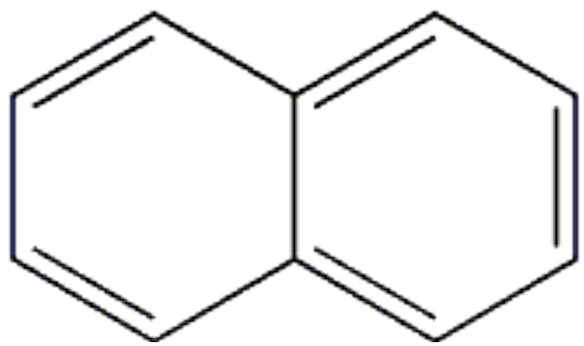
```
pressureDependence(
    method='modified strong collision',
    maximumGrainSize=(0.5,'kcal/mol'),
    minimumNumberOfGrains=250,
    temperatures=(300,2000,'K',8),
    pressures(0.01,100,'bar',5),
    interpolation=('Chebyshev',6,4),
)
```

Extra features:
QMTP

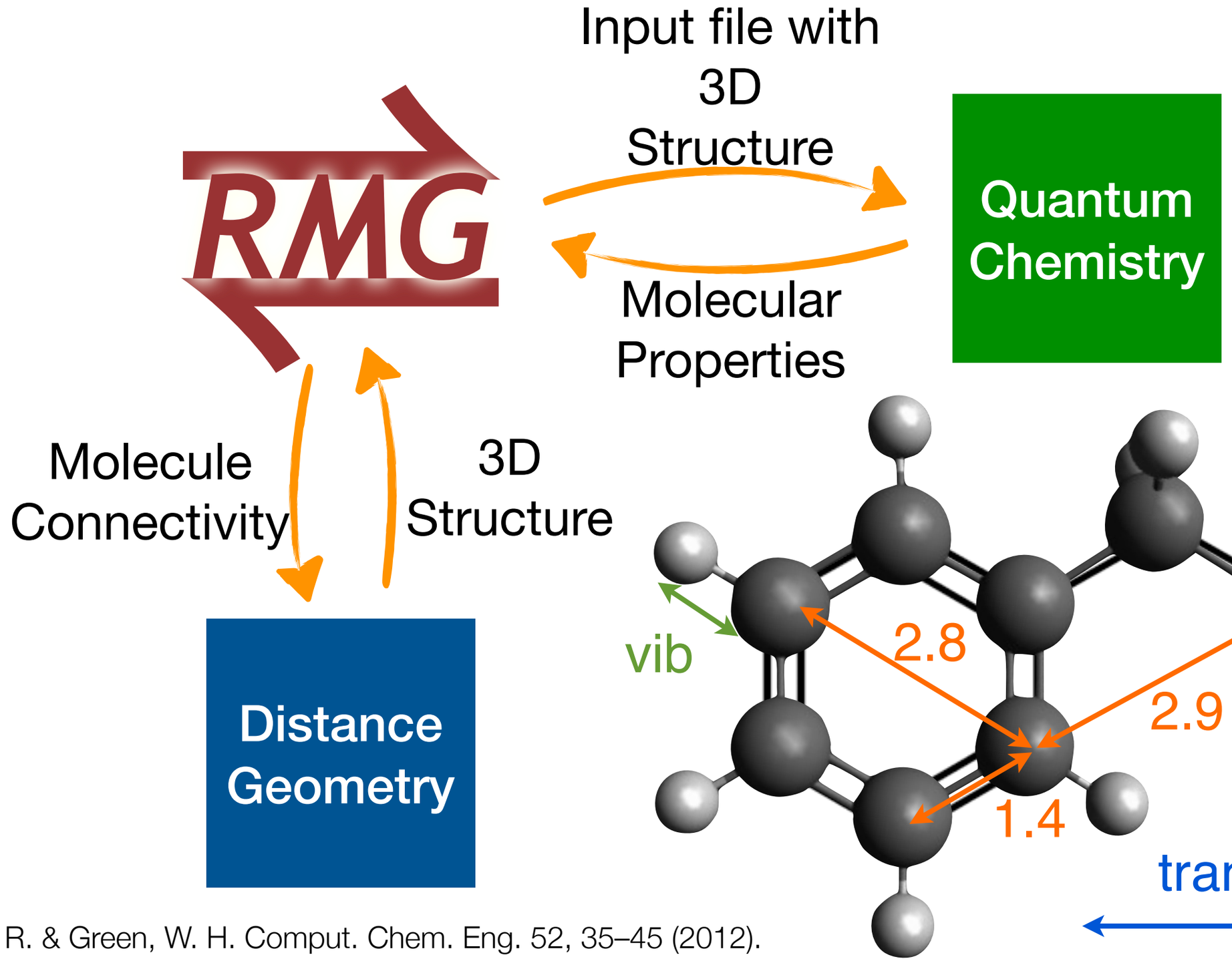


1) 2(-10.20)
2) -1.90
3) -4.86
4) 5.51
5) 5(3.30)

Standard enthalpy : -5.15 kcal/mole
of isobutylbenzene



Extra features:
QMTP



Magoon, G. R. & Green, W. H. Comput. Chem. Eng. 52, 35–45 (2012).

Extra features:
QMTP

```
quantumMechanics(
    software='mopac',
    method='pm3',
    fileStore='QMfiles',
    scratchDirectory='QMscratch',
    onlyCyclics=True,
    maxRadicalNumber=0,
)
```

Extra features:
Liquid Phase

```
liquidReactor(
    temperature=(500,‘K’),
    initialConcentrations={
        ‘octane’: (6.0e-3,‘mol/cm^3’),
        ‘oxygen’: (5.0e-6,‘mol/cm^3’),
    },
    terminationTime=(5,‘s’),
)

solvation(
    solvent=‘octane’
)
```

Final Options

```
options(
    units='si',
    saveRestartPeriod=(1,'day'),
    drawMolecules=False,
    generatePlots=False,
)
```

saveConcentrationProfiles=True/False # liquid phase

Example Input File

https://github.com/GreenGroup/RMG-Py/blob/master/examples/rmg/1%2C3-hexadiene/input.py