New Adjacency List and Multiplicity

Beat Buesser

20.05.2014

Reasons for Adjacency List Changes

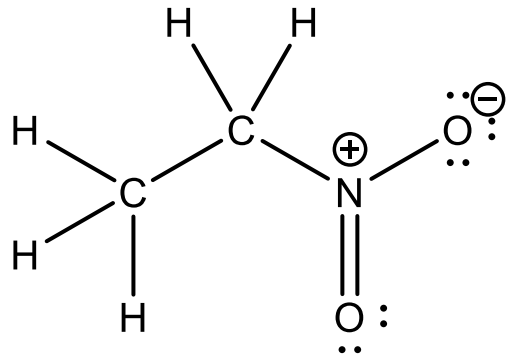
- New elements with variable valency, beyond hydrocarbons
  - Oxygen could change valency but hardly does because of electronegativity
  - Sulphur was introduced only as divalent, although it is often hexavalent
  - Nitrogen changes has many tetra- and pentavalent species
  - Phosphorous, Metals also have varying valency
- Varying valency “old” adjacency list
  - No clean solution
  - A lot of code and difficult to maintain
  - Electron balance by tracking lone electron pairs much simpler

New Adjacency Lists - Python

- Generic format

<ID> [<label>] <element> **U**<unpaired electrons> **L**<lone pairs> **E**<formal charge> <bond list>

```
nitro ethane
1  C U0 L0 E0   {2,S} {4,S} {5,S} {6,S}
2  C U0 L0 E0   {1,S} {3,S} {7,S} {8,S}
3  N U0 L0 E+1  {2,S} {9,D} {10,S}
4  H U0 L0 E0   {1,S}
5  H U0 L0 E0   {1,S}
6  H U0 L0 E0   {1,S}
7  H U0 L0 E0   {2,S}
8  H U0 L0 E0   {2,S}
9  O U0 L2 E0   {3,D}
10 O U0 L3 E-1  {3,S}
```

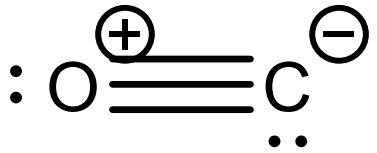


U – unpaired electrons: always required
L – lone electron pairs: optional, if not defined assuming L0
E – formal charges: is being read but not used, currently replaced by value calculated from U and bonds, calculated values is printed

Adjacency Lists – Python
Other examples

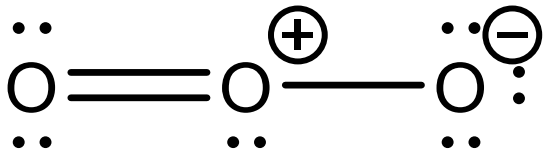
- e.g. carbon monoxide (NEW: atom type Ot)

```
CO
1 C U0 L1 E-1 {2,T}
2 O U0 L1 E+1 {1,T}
```



- e.g. ozone

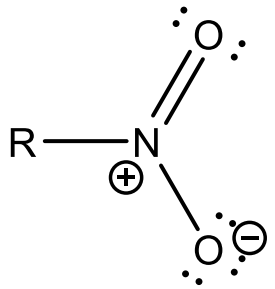
```
O3
1 O U0 L2 E0  {2,D}
2 O U0 L1 E+1 {1,D} {3,S}
3 O U0 L3 E-1 {2,S}
```



Group Adjacency Lists - Python

- e.g. nitro group

```
1 * N    U0 L0 {2,D} {3,S} {4,S}
2   O    U0    {1,D}
3   O    U0    {1,S}
4   R!H  U0    {1,S}
```



So far, groups only compare lone electron pairs
for labeled atoms, if defined
Lx, Ux, and Ex are treated as wild cards

Multiplicity/Electronic States

- Forward reaction

NO2 + CH2(S) -> HONO + CH

- RMG broke by finding only reverse to triplet

HONO + CH -> NO2 + CH2(T)

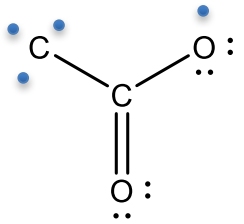
- Extended to find all possible reverse reactions and electronic states

HONO + CH -> NO2 + CH2(T)
HONO + CH -> NO2 + CH2(S)

Multiplicity/Electronic States

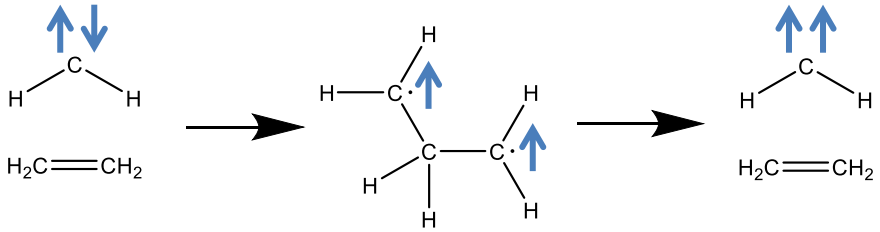
- Problematic if 2-centered radicals

```
radical
1 C 3Q 0 {3,S}
2 O 1 2 {3,S}
3 C 0 0 {1,S} {2,S} {4,D}
4 O 0 2 {3,D}
```

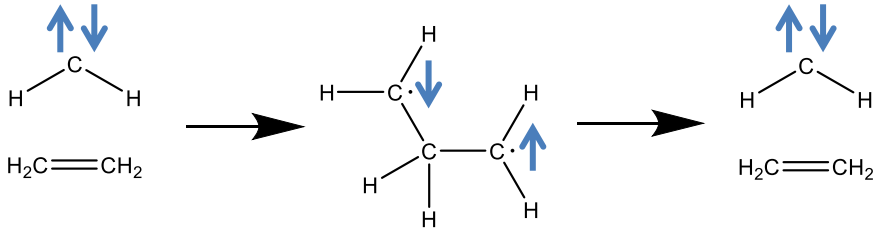


- Problem: Loss of information in radical reactions

RMG so far



RMG from now on



Extend database - Multiplicity

- Store multiplicity as species property
- Consequences
  - Remove 2S and 2T labels from database
  - Define label for species in thermo and kinetic libraries

```
entry(
    index = 318,
    label = "HONO",
    multiplicity = 1
    molecule =        """
        1 O 0 2 {2,D}
        2 N 0 1 {1,D} {3,S}
        3 O 0 2 {2,S} {4,S}
        4 H 0 0 {3,S}
        """,
)
```

Extend database – Multiplicity Kinetics Libraries

- Read multiplicity as part of adjacency list
- Future development: move multiplicity from adjacency list to species attribute and store species in separate library and link thermo and kinetics information

```
entry(
    index       = 1,
    reactant1 =
    """
    HCO
    multiplicity 2
    1 C U1 L0 E0  {2,D} {3,S}
    2 O U0 L2 E0  {1,D}
    3 H U0 L0 E0  {1,S}
    """,
…
```

Extend database – Multiplicity Groups

- Multiplicity attribute is a list

```
entry(
    index = 7,
    label = "Ct_H",
    multiplicity = [1, 2, 3, 4, 5],
    group =
"""
1 *2 Ct U0 {2,S}
2 *3 H  U0 {1,S}
""",
    kinetics = None,
    shortDesc = u"""""",
```

Multiplicity from Molecule Input Functions

- Functions like fromInChI(), fromSMILES(), fromSMARTS() based on fromRDKitMol() assume maximum multiplicity

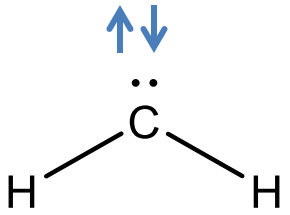
- multiplicity = 2*total spin + 1

  = number of unpaired electrons + 1

New Reaction Families - Outlook

Combination of lone pairs and multiplicity might allow introducing/distinguishing new chemistry

1-centered di-radicals, e.g. CH2 singlet



Currently new style

```
multiplicity 1
1 C U2 L0 E0 {2,S} {3,S}
2 H U0 L0 E0 {1,S}
3 H U0 L0 E0 {1,S}
```

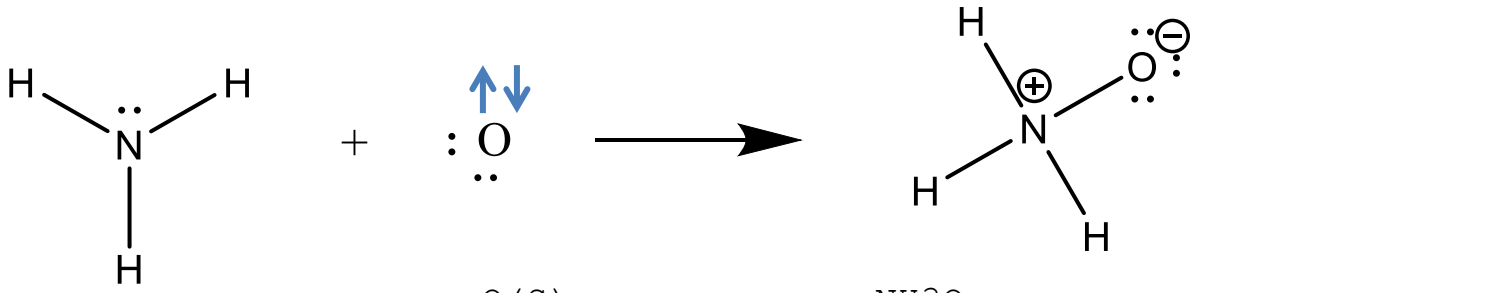
Actual true singlet spin state

```
multiplicity 1
1 C U0 L1 E0 {2,S} {3,S}
2 H U0 L0 E0 {1,S}
3 H U0 L0 E0 {1,S}
```

not to concentrate only on unpaired electrons,
filled and empty orbitals might define reaction mechanisms too

Reaction Family lone_electron_pair_bond

```
recipe(actions=[
    ['FORM_BOND', '*1', 'S', '*2'],
    ['LOSE_RADICAL', '*2', '2'],
    ['LOSE_PAIR', '*1', '1'],
    ['GAIN_PAIR', '*2', '1'],
])
```

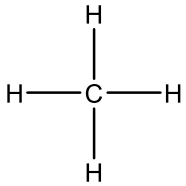


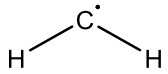
```
NH3
1 *1 N 0 1 {2,S} {3,S} {4,S}
2    H 0 0 {1,S}
3    H 0 0 {1,S}
4    H 0 0 {1,S}
```

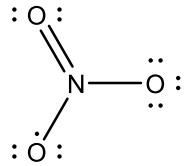
```
O(S)
1 *2 O 2 2
```

```
NH3O
1 *1 N 0 1 {2,S} {3,S} {4,S}
2    H 0 0 {1,S}
3    H 0 0 {1,S}
4    H 0 0 {1,S}
5 *2 O 0 3 {1,S}
```

The function isMoleculeForbidden(self, molecule) allows so far only neutral species

CH_4 

$CH_2(T)$ 

NO_3 

Java Style

```
1 C 0
```

```
1 C 2T
```

Py Style / Lone Pair Style

```
1 C 0 0 {2,S} {3,S} {4,S} {5,S}
2 H 0 0 {1,S}
3 H 0 0 {1,S}
4 H 0 0 {1,S}
5 H 0 0 {1,S}
```

```
1 C 2T 0 {2,S} {3,S}
2 H 0  0 {1,S}
3 H 0  0 {1,S}
```

```
1 N 0 0 {2,D} {3,S} {4,S}
2 O 0 2 {1,D}
3 O 0 3 {1,S}
4 O 1 2 {1,S}
```

New Py Style

```
1 C U0 L0 E0 {2,S} {3,S} {4,S} {5,S}
2 H U0 L0 E0 {1,S}
3 H U0 L0 E0 {1,S}
4 H U0 L0 E0 {1,S}
5 H U0 L0 E0 {1,S}
```

```
1 C U2 L0 E0 {2,S} {3,S}
2 H U0 L0 E0 {1,S}
3 H U0 L0 E0 {1,S}
```

```
1 N U0 L0 E-1 {2,D} {3,S} {4,S}
2 O U0 L2 E0  {1,D}
3 O U0 L3 E+1 {1,S}
4 O U1 L2 E0  {1,S}
```

Adjacency Lists
Java and former Python

- Generic format

  <number> [<label>] <element> <radicals> <bond list>

- Example

```
propene radical
 1 *1 C 0 {2,D}
 2 *2 C 0 {1,D} {3,S}
 3    C 1 {2,S}
```