Morgan's Algorithm

Eberhard Karls
Universität
Tübingen

Idea

- **Numbering** based on connectivity, i.e., node degree
- Node degree alone is not sufficient to construct a unique numbering
- **Extended connectivity** (EC) includes the node degree of adjacent nodes

Two steps:

1. **Classification by connectivity** (relaxation)

   Iteratively assign extended connectivity to each atom

2. **Canonical enumeration**

   Number all atoms based on the EC,
   resolve ambiguities

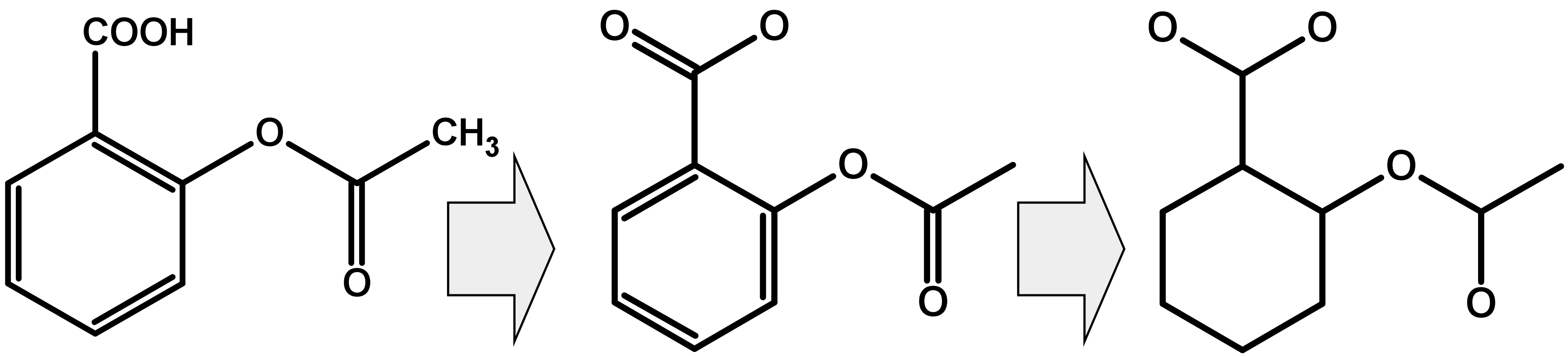
H. L. Morgan, J. Chem. Doc., 1965, 5, 107

Morgan's Algorithm

Eberhard Karls
Universität
Tübingen

1. Computation of extended connectivity

- Consider heavy atom graph $G = (V, E)$

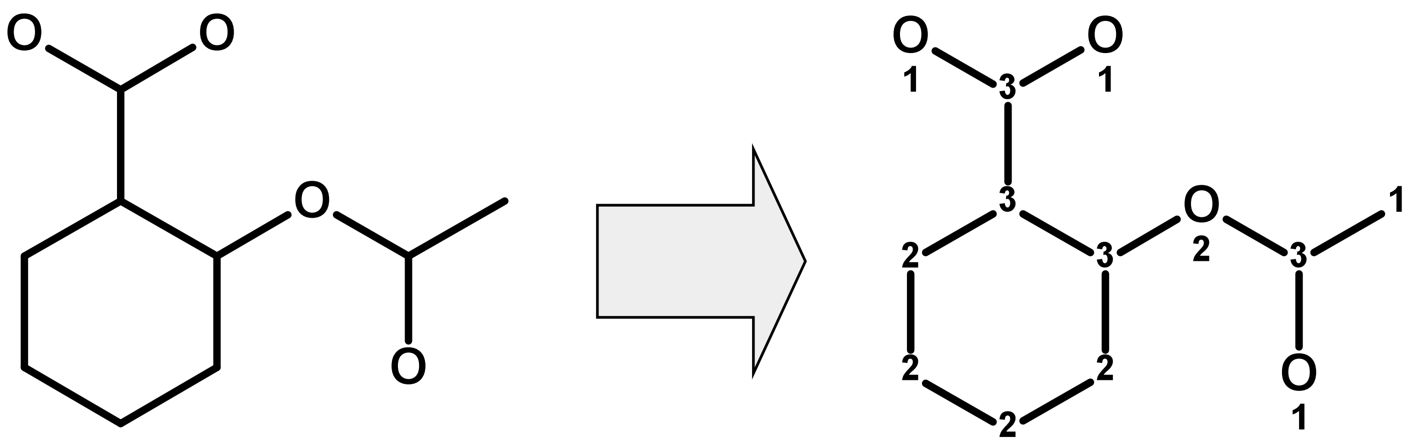


H. L. Morgan, J. Chem. Doc., 1965, 5, 107
LG, S.7

Morgan’s Algorithm

1. Computation of extended connectivity

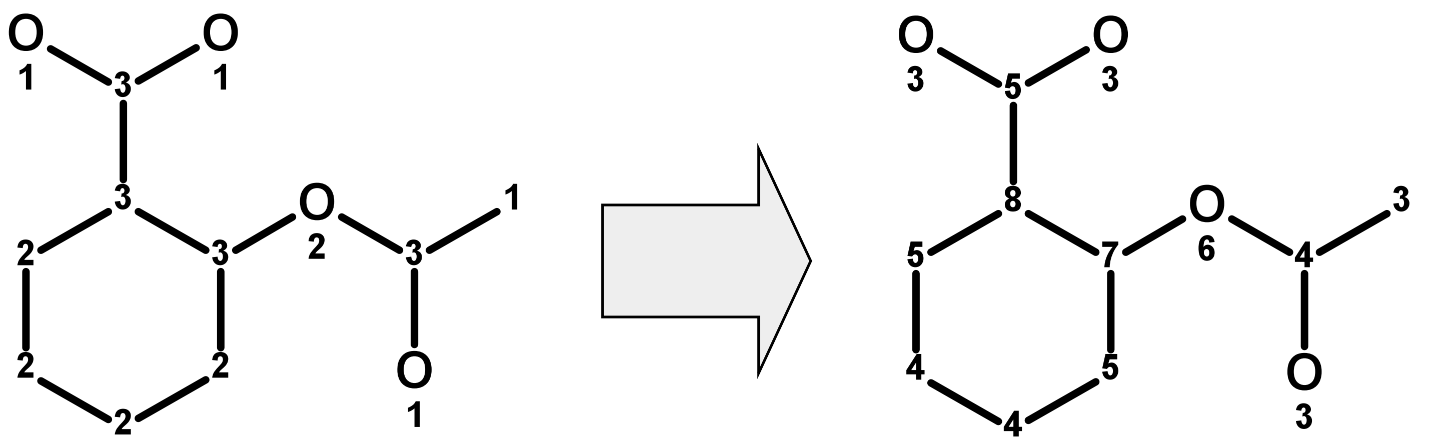
- Consider heavy atom graph $G = (V, E)$
- $EC_0(v) = \deg(v) \quad \forall\, v \in V$



Morgan's Algorithm

1. Computation of extended connectivity

- Consider heavy atom graph $G = (V, E)$
- $EC_0(v) = \deg(v) \;\; \forall\, v \in V$
- For $i > 0$:
  - $EC_i(v) = \sum_{u:(u,v)\in E} EC_{i-1}(u)$
  - Compute number of EC classes $c_i = |\{EC_i(v) \mid v \in V\}|$
  - Abort, if $c_{i+1} = c_i$. Store $EC = EC_i$



$c_0 = 3\ (1, 2, 3)$ $\quad$ $c_1 = 6\ (3, 4, 5, 6, 7, 8)$

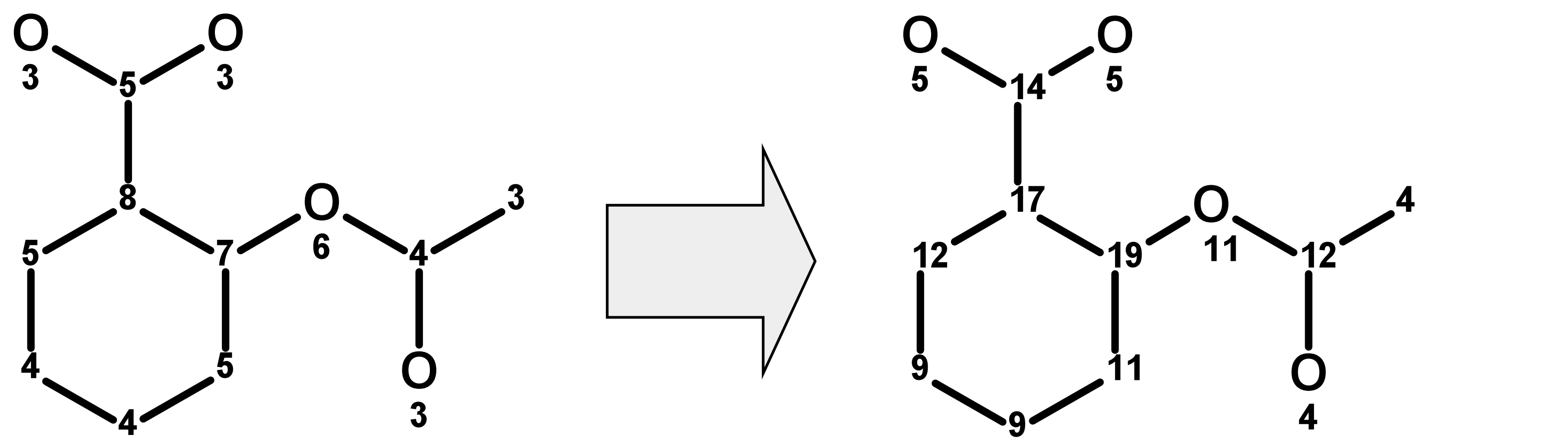
H. L. Morgan, J. Chem. Doc., 1965, 5, 107

Morgan's Algorithm

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

1. Computation of extended connectivity

- Consider heavy atom graph $G = (V, E)$
- $EC_0(v) = \deg(v) \;\; \forall\, v \in V$
- For $i > 0$:
  - $EC_i(v) = \sum_{u:(u,v)\in E} EC_{i-1}(u)$
  - Compute number of EC classes $c_i = |\{EC_i(v) \mid v \in V\}|$
  - Abort, if $c_{i+1} = c_i$. Store $EC = EC_i$



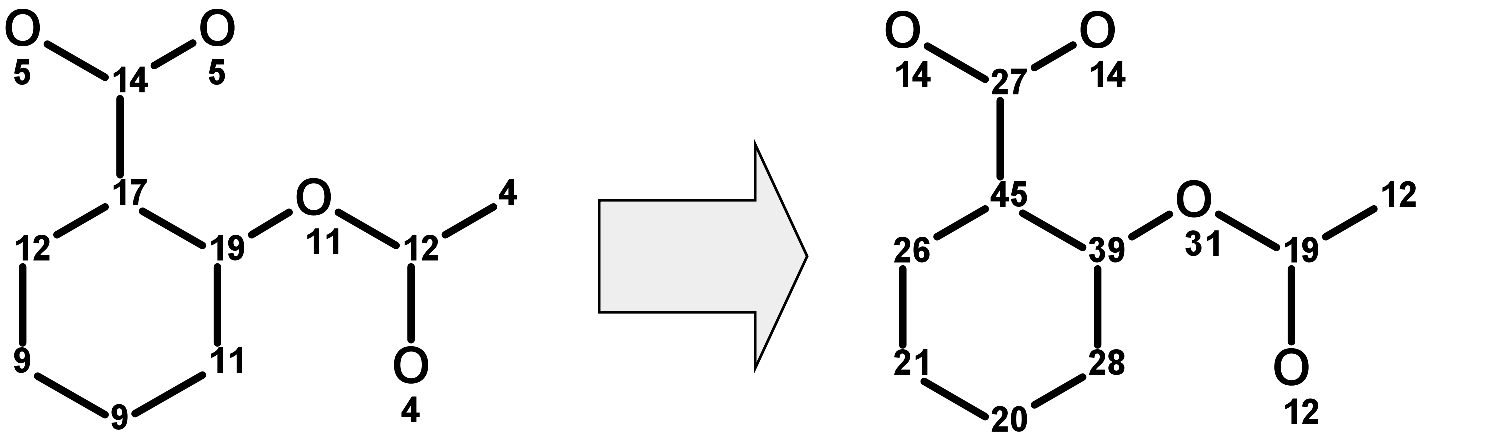
c_1 = 6 (3, 4, 5, 6, 7, 8) c_2 = 8 (4, 5, 9, 11, 12, 14, 17, 19)

H. L. Morgan, J. Chem. Doc., 1965, 5, 107

Morgan's Algorithm

1. Computation of extended connectivity

- Consider heavy atom graph $G = (V, E)$
- $EC_0(v) = \deg(v) \quad \forall\, v \in V$
- For $i > 0$:
  - $EC_i(v) = \sum_{u:(u,v) \in E} EC_{i-1}(u)$
  - Compute number of EC classes $c_i = |\{EC_i(v) \mid v \in V\}|$
  - Abort, if $c_{i+1} = c_i$. Store $EC = EC_i$

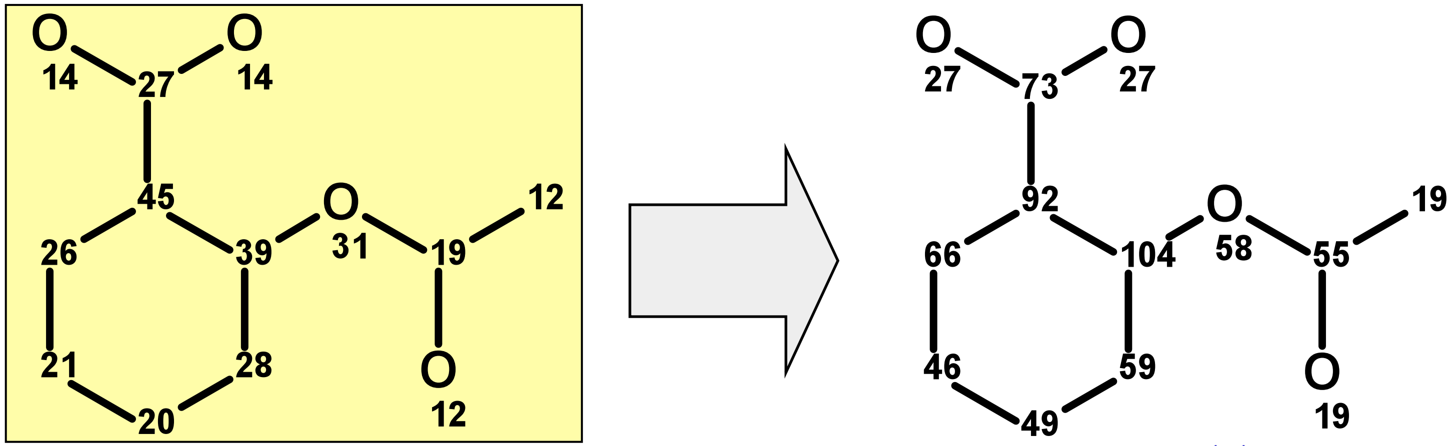


$c_2 = 8$ $\qquad$ $c_3 = 11$

Morgan's Algorithm

1. Computation of extended connectivity

- Consider heavy atom graph $G = (V, E)$
- $EC_0(v) = \deg(v) \quad \forall\, v \in V$
- For $i > 0$:
  - $EC_i(v) = \sum_{u:(u,v)\in E} EC_{i-1}(u)$
  - Compute number of EC classes $c_i = |\{EC_i(v) \mid v \in V\}|$
  - **Abort, if $c_{i+1} = c_i$. Store $EC = EC_i$**



$c_3 = 11$ $\qquad$ $c_4 = 11$

H. L. Morgan, J. Chem. Doc., 1965, 5, 107

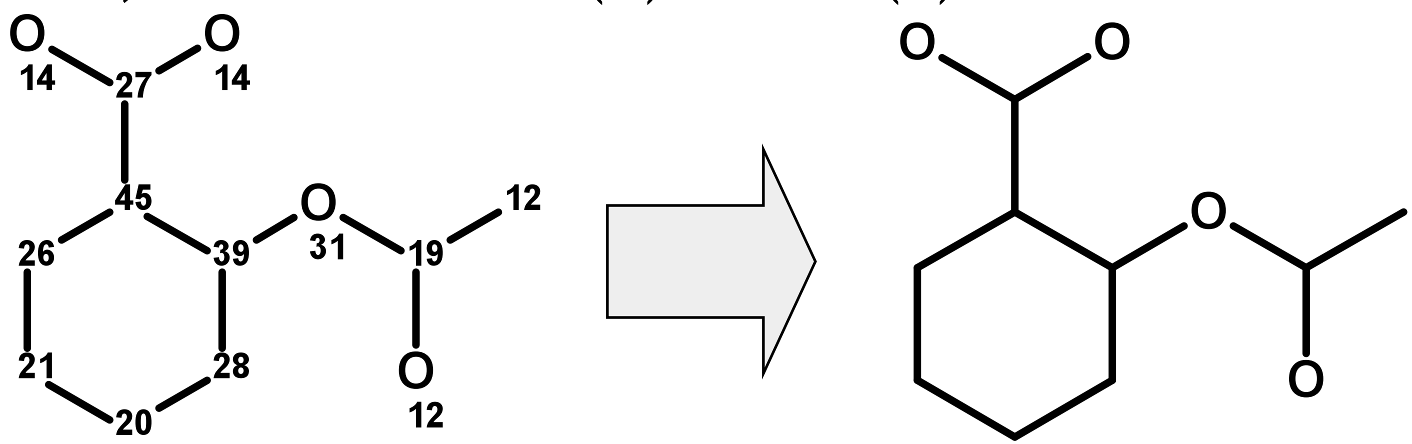
Morgan's Algorithm

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

2. Canonical enumeration

- *label*(*v*) = 0 ∀ *v*∈*V*, *i* = 1, *v* = argmax *EC*(*v*) for *v* ∈ *V*
- Label *v*: *label(v)* ← *i*, *i* ← *i* + 1
- For current node *v*:
  - For all nodes *u adjacent to v* with *label*(*u*) = 0:
    - Label *u* as above with decreasing *EC(u)*
    - Ambiguities: atomic number, bond order
- v ← *u*, *u*∈*V* with label(u) = label(v) + 1

EC(*v*)

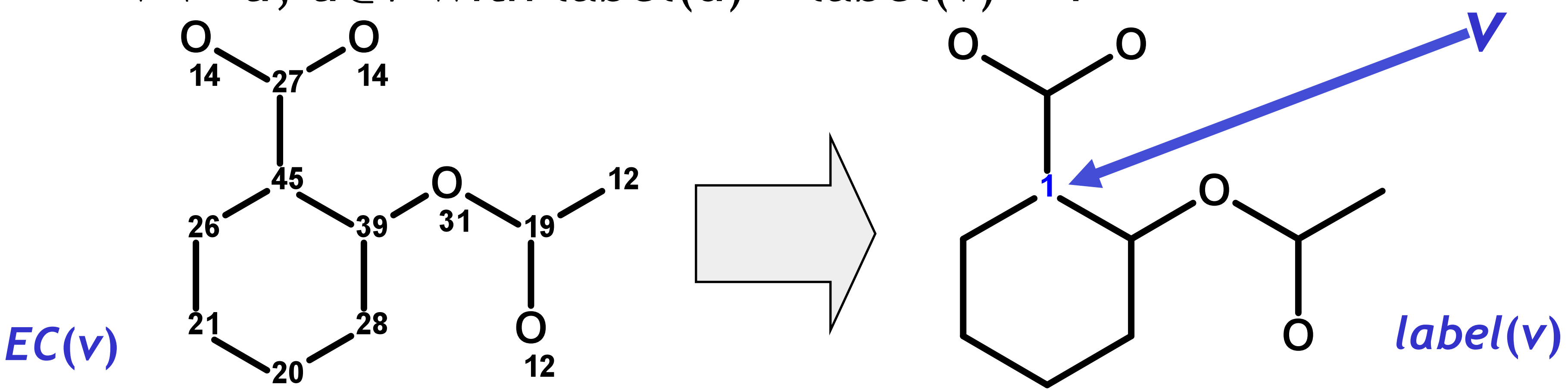


label(*v*)

Morgan's Algorithm

2. Canonical enumeration

- *label*(*v*) = 0 ∀ *v*∈*V*, *i* = 1, *v* = argmax *EC*(*v*) for *v* ∈ *V*
- Label *v*: *label(v)* ← *i*, *i* ← *i* + 1
- For current node *v*:
  - For all nodes *u adjacent to v* with *label*(*u*) = 0:
    - Label *u* as above with decreasing *EC(u)*
    - Ambiguities: atomic number, bond order
- v ← *u*, *u*∈*V* with label(u) = label(v) + 1

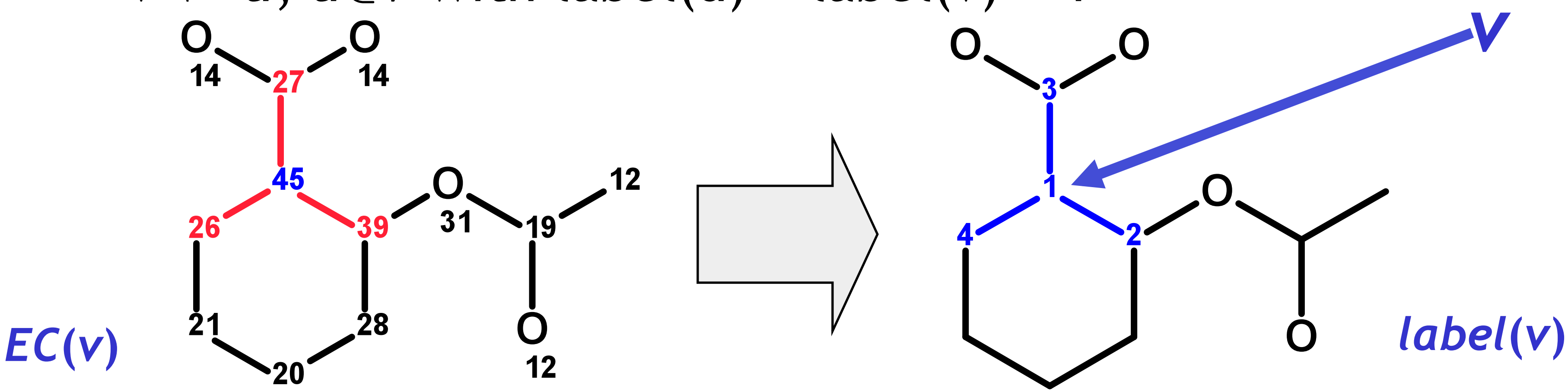


H. L. Morgan, J. Chem. Doc., 1965, 5, 107

Morgan's Algorithm

2. Canonical enumeration

- *label*(*v*) = 0 ∀ *v*∈*V*, *i* = 1, *v* = argmax *EC*(*v*) for *v* ∈ *V*
- Label *v*: *label(v)* ← *i*, *i* ← *i* + *1*
- For current node *v*:
  - For all nodes *u adjacent to v* with *label*(*u*) = 0:
    - Label *u* as above with decreasing *EC(u)*
    - Ambiguities: atomic number, bond order
- v ← *u*, *u*∈*V* with label(u) = label(v) + 1



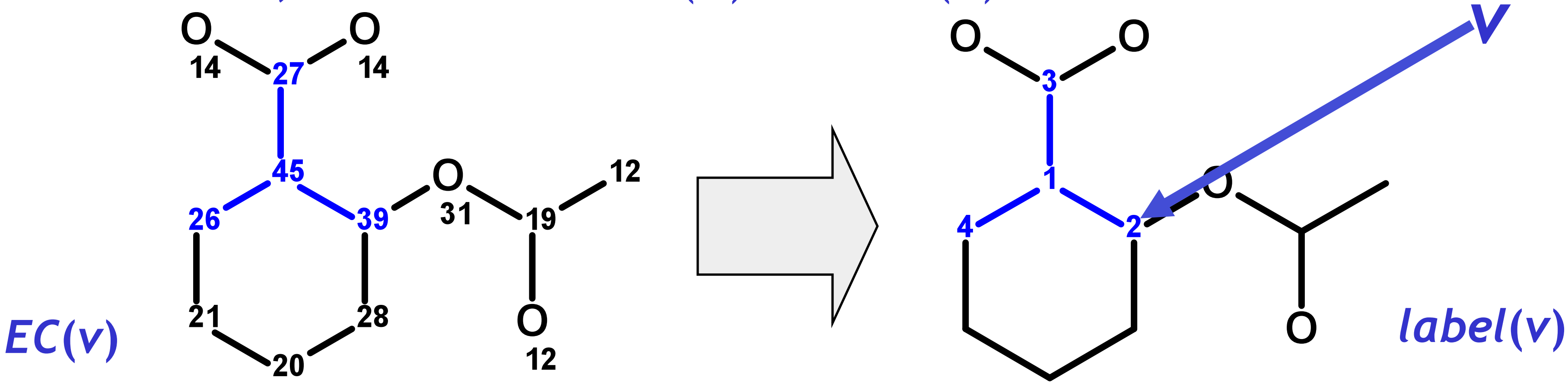
H. L. Morgan, J. Chem. Doc., 1965, 5, 107
LG, S.7

Morgan's Algorithm

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

2. Canonical enumeration

- *label*(*v*) = 0 ∀ *v*∈*V*, *i* = 1, *v* = argmax *EC*(*v*) for *v* ∈ *V*
- Label *v*: *label(v)* ← *i*, *i* ← *i* + *1*
- For current node *v*:
  - For all nodes *u adjacent to v* with *label*(*u*) = 0:
    - Label *u* as above with decreasing *EC(u)*
    - Ambiguities: atomic number, bond order
- v ← *u*, *u*∈*V* with label(u) = label(v) + 1

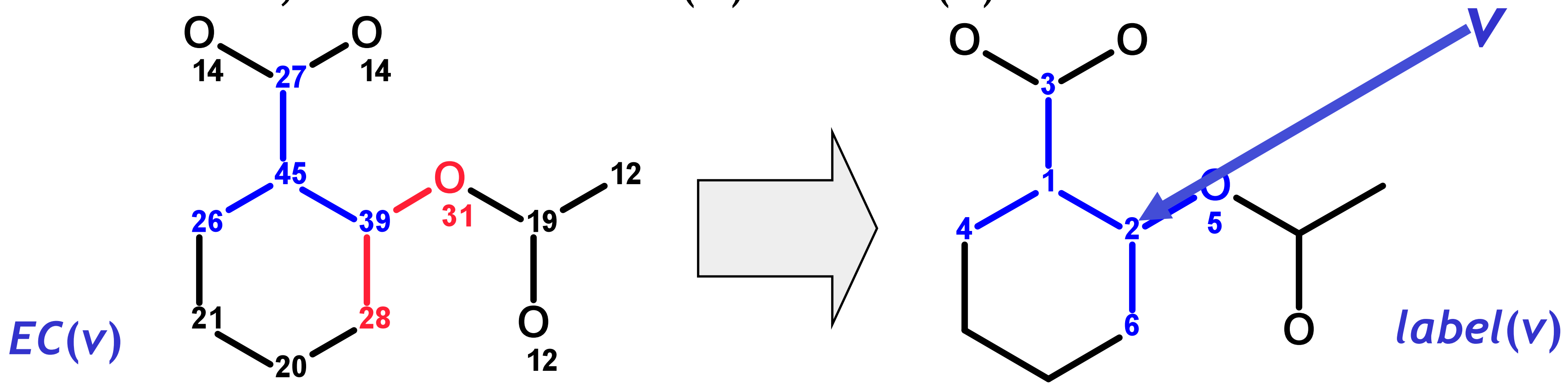


Morgan's Algorithm

Eberhard Karls Universität Tübingen

2. Canonical enumeration

- *label*(*v*) = 0 ∀ *v*∈*V*, *i* = 1, *v* = argmax *EC*(*v*) for *v* ∈ *V*
- Label *v*: *label(v)* ← *i*, *i* ← *i* + *1*
- For current node *v*:
  - For all nodes *u adjacent to v* with *label*(*u*) = 0:
    - Label *u* as above with decreasing *EC(u)*
    - Ambiguities: atomic number, bond order
- v ← *u*, *u*∈*V* with label(u) = label(v) + 1

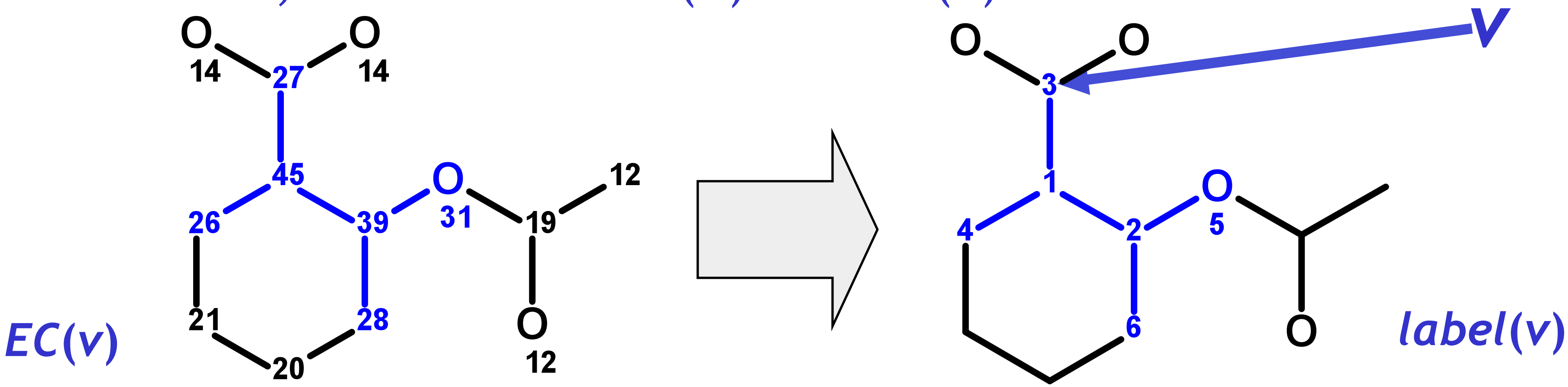


H. L. Morgan, J. Chem. Doc., 1965, 5, 107

Morgan's Algorithm

2. Canonical enumeration

- *label*(*v*) = 0 ∀ *v*∈*V*, *i* = 1, *v* = argmax *EC*(*v*) for *v* ∈ *V*
- Label *v*: *label(v)* ← *i*, *i* ← *i* + 1
- For current node *v*:
  - For all nodes *u adjacent to v* with *label*(*u*) = 0:
    - Label *u* as above with decreasing *EC(u)*
    - Ambiguities: atomic number, bond order
- v ← *u*, *u*∈*V* with label(u) = label(v) + 1



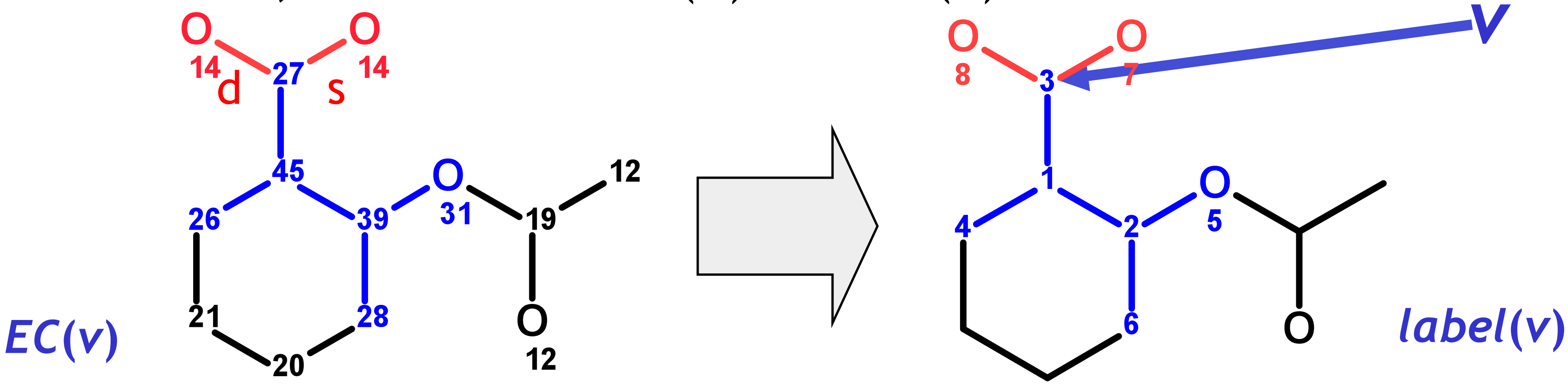
Morgan's Algorithm

Eberhard Karls Universität Tübingen

2. Canonica

- *label*(*v*) = 0
- Label *v*: *lab*
- For current
  - For all nodes *u adjacent to v* with *label*(*u*) = 0:
    - Label *u* as above with decreasing *EC*(*u*)
    - **Ambiguities: atomic number, bond order**
- v ← *u*, *u*∈*V* with label(u) = label(v) + 1

Atomic number:
Increasing priority(C > N > P etc.)
Bond order:
Increasing priority (single, double, ...)



H. L. Morgan, J. Chem. Doc., 1965, 5, 107
LG, S.7

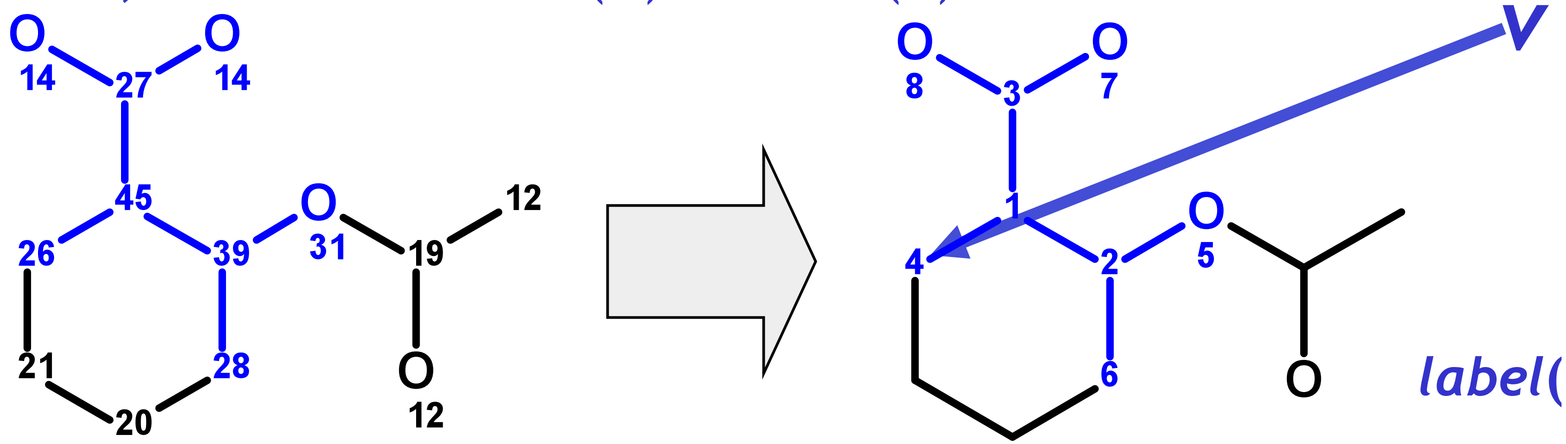
Morgan's Algorithm

Eberhard Karls Universität Tübingen

2. Canonical enumeration

- *label*(*v*) = 0 ∀ *v*∈*V*, *i* = 1, *v* = argmax *EC*(*v*) for *v* ∈ *V*
- Label *v*: *label(v)* ← *i*, *i* ← *i* + 1
- For current node *v*:
  - For all nodes *u adjacent to v* with *label*(*u*) = 0:
    - Label *u* as above with decreasing *EC(u)*
    - Ambiguities: atomic number, bond order
- v ← *u*, *u*∈*V* with label(u) = label(v) + 1

EC(*v*)



label(*v*)

ZBIT

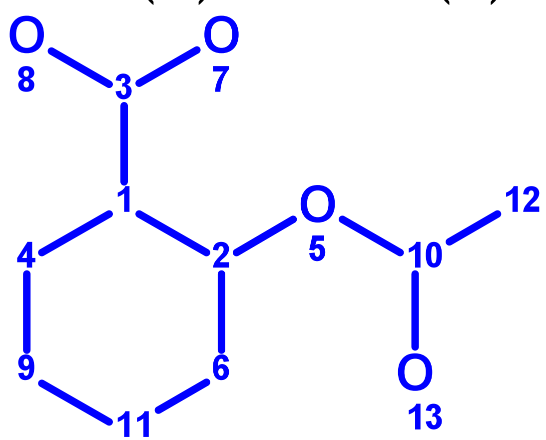
Morgan's Algorithm

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

2. Canonical enumeration

- *label*(*v*) = 0 $\forall$ *v*$\in$*V*, *i* = 1, *v* = argmax *EC*(*v*) for *v* $\in$ *V*
- Label *v*: *label(v)* $\leftarrow$ *i*, *i* $\leftarrow$ *i + 1*
- For current node *v*:
  - For all nodes *u adjacent to v* with *label*(*u*) = 0:
    - Label *u* as above with decreasing *EC(u)*
    - Ambiguities: atomic number, bond order
- v $\leftarrow$ *u*, *u*$\in$*V* with label(u) = label(v) + 1

Result:



Morgan's Algorithm

- Morgan's algorithm allows the canonical enumeration of a molecular graph
- **Problems**
  - No provably unique enumeration!
  - Problems in phase 1: **oscillations**
    - Oscillating values of $c \Rightarrow$ Algorithm does not terminate!
  - Problems in phase 2: **ambiguities**
    - Not all ambiguities can be resolved

$\Rightarrow$ There are quite a few **improved variants** of the algorithm, **problems are not entirely resolved,** but much less likely

USMILES

- David Weininger proposed an algorithm for **unique SMILES** (USMILES) in 1989
- Based on Morgan‘s alg. ⇒ same problems
  - No provably unique solution
  - Problems with certain structures
- Since 1989 the algorithm had to be changed a few times
- What exactly Daylight‘s USMILES algorithm does in detail has not been published so far

D. Weininger, A. Weininger, and J.L. Weininger, J. Chem. Inf. Comput. Sci. 1989, 29, 97