Assigning Unique Keys to Chemical Compounds for Data Integration: Some Interesting Counter Examples

Greeshma Neglur[1], Robert L. Grossman[2], and Bing Liu[3]

[1] Laboratory for Advanced Computing,
University of Illinois at Chicago, Chicago, IL 60607
neglur@lac.uic.edu

[2] Laboratory for Advanced Computing,
University of Illinois at Chicago, Chicago, IL 60607
grossman@uic.edu

[3] Department of Computer Science,
University of Illinois at Chicago, Chicago, IL 60607
liub@cs.uic.edu

Abstract. Integrating data involving chemical structures is simplified when unique identifiers (UIDs) can be associated with chemical structures. For example, these identifiers can be used as database keys. One common approach is to use the Unique SMILES notation introduced in [2]. The Unique SMILES views a chemical structure as a graph with atoms as nodes and bonds as edges and uses a depth first traversal of the graph to generate the SMILES strings. The algorithm establishes a node ordering by using certain symmetry properties of the graphs. In this paper, we present certain molecular graphs for which the algorithm fails to generate UIDs. Indeed, we show that different graphs in the same symmetry class employed by the Unique SMILES algorithm have different Unique SMILES IDs. We tested the algorithm on the National Cancer Institute (NCI) database [7] and found several molecular structures for which the algorithm also failed. We have also written a python script that generates molecular graphs for which the algorithm fails.

1 Introduction

The volume of biological data, especially chemical structures, is increasing at an unprecedented rate. There are numerous databases today that store chemical substances and thousands of chemical structures are being added to these databases each year. For example, the Chemical Abstracts Service (CAS) alone has more than 71,285,000 records, while the NCI database has close to 250,251 chemical structures. In general, each database uses a different method of assigning keys to the chemical compounds. For example, many databases assign keys based upon the order the compound was added to the database. For this reason, determining whether a compound has been entered into a database more than once or comparing chemical structures across databases is difficult.

This paper is concerned with data integration techniques that use the structural features of chemical compounds to assign unique IDs (UIDs). Using UIDs it is relatively simple to compare chemical structures across different databases, something which facilitates the discovery of new drugs and therapeutic treatments.

In this paper, we consider two schemes for assigning UIDs: Unique SMILES [2] and Universal Chemical Keys (UCKs) [20]. Although Unique SMILES are widely deployed and very useful in practice, we show that the algorithm described in [2] does not lead to unique IDs. We emphasize that the Unique SMILES as deployed by the Daylight Chemical Information System is an enhanced version of the algorithm described in [2], but, as far as we know, there is not a published version of this algorithm.

We believe that our paper makes the following research contributions:

1. We show that the Unique SMILES IDs although extremely useful are not unique.
2. We describe some common circumstances leading to the non-uniqueness of Unique SMILES IDs.

This paper is organized as follows: Section 2 describes related work. Sections 3-4 described one popular technique of assigning IDs to chemical compounds called Unique SMILES [2]. Sections 5-6 explain why Unique SMILES doesn't always generated UIDs. Sections 7 provides some counter examples. The final section summarizes the reason behind the failure of the unique SMILES algorithm and suggests alternate techniques for data integration of chemical compound databases using UIDs.

2 Related Work

The International Union of Pure and Applied Chemistry (IUPAC) rules [13] have been use for several decades. However, these names are growing more complicated and causing inconsistencies and mistakes as compounds become more and more complex [14]. To deal with this problem, the IUPAC has initiated a project [15] to assign unique keys known as IUPAC Chemical identifiers (INChI) to chemical compounds. This approach is based in part on graph theory. The chemical identifiers are alphanumeric text strings obtained from the molecular graph of the compound and are designed so that the chemical structure can be recovered from the UID. However, the details are not yet published.

The most common approach for integrating information about chemical compounds across databases is to use a unique key assigned by one of the databases, such as an acquisition-based or Chemical Abstracts Service (CAS) based registry numbers, as the foreign key for the other databases. For example, the NCI database stores the corresponding CAS registry number for its chemical compounds. Integrating databases in this way is labor intensive and does not easily scale.

Another approach is to view the molecular structures as a graph and to compare them directly using a graph isomorphism algorithm. There are several algorithms [11,12,17] which test for graph isomorphism. The problem with this approach is the amount of computing required to compare two structures. A more important problem is that just identifying two graphs as isomorphic does not directly provide a UID.

Several graph based techniques to solve the problem of assigning unique keys to chemical structures are known. For example, Randic and coworkers [16] developed a technique that canonically orders the adjacency matrix to produce an ID. Another popular method to discriminate molecular graphs is by means of graph invariants and vertex-in-graph invariants. One such method is the Morgan algorithm [18] which uses extended sum connectivities to distinguish atoms in a molecule. Another molecular graph canonizer is MOLGEN-CID [19].

In contrast, the Universal Chemical Key (UCK) algorithm [20] enumerates all paths up to a specified depth d in the molecular graph, lexicographically orders them, and concatenates them to produce an ID. These strings are long and cannot be used to recover the graphs. On the other hand, it is easy to use them to integrate distributed bioinformatics databases [20]. For databases of chemical compounds examined to date, a depth of $d = 3$ or 4 produces UIDs.

3 The Unique SMILES Algorithm

SMILES [1] (Simplified Molecular Input Line Entry System) is a popular chemical notation system used for computerized processing of chemical information. SMILES is a string obtained by enumerating the atomic symbols and bond types via a depth-first tree-traversal of a molecular graph, where, as usual, the nodes represent atoms and the edges represent bonds.

The problem with all such approaches is that there is no natural order to nodes in a molecular graph, and different depth-first traversals will result from different starting points. This means that there may be more than one correct SMILES string obtained from the same molecular graph. For this reason, SMILES strings, which in general are not unique, cannot be used as database keys.

To overcome this disadvantage, the creators of SMILES came up with a 2-stage algorithm called CANGEN [2] to generate a unique SMILES string for a given molecular structure. The first stage, CANON, involves CANonicalization of the structure represented as a molecular graph. The second stage, GENES, GENerates the unique SMILES notation as a depth-first traversal of the canonicalized molecular graph.

For most chemical structures the CANGEN algorithm as described in [2] generates UIDs. However, as we show below by counter examples, there are exceptions. These exceptions need not be complicated. See Section 7. The reason is simple: if the graph is symmetric enough, it is possible for the CANON stage of the Unique SMILES algorithm to generate different canonical labels for the nodes of the molecular graph. This results in several different Unique SMILES strings.

The Unique SMILES algorithm consists of the following two stages [2]:

1. The CANON stage labels a molecular graph with canonical labels. Each atom/node is given a numerical label on the basis of its topology.

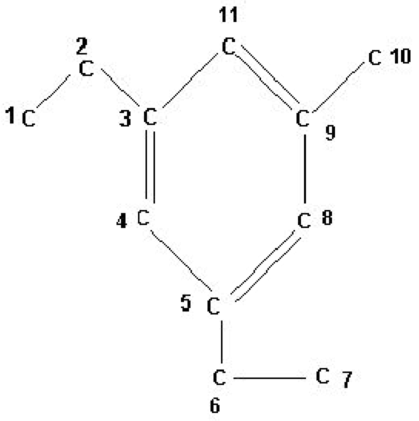

Fig. 1. Molecular graph of 3,5 di-ethyl toluene. NSC number 62141

2. The GENES stage generates unique SMILES notation as a tree representation of the graph. GENES selects the starting atom and makes branching decisions by referring to the canonical labels as needed.

The algorithm and its non-uniqueness will be explained with the example of chemical compound 3,5 di-ethyl toluene, It is stored in the NCI database with NSC number 62141. The molecular graph of this compound is described in Figure 1, where the number beside each atom is just assigned for brevity to refer to the atom in describing the following steps of the algorithm.

4 The CANON Stage of Unique SMILES

Node ordering for the generation of unique SMILES is obtained by developing topological symmetry classes, using the product of corresponding primes as illustrated below.

Graph Invariants. The algorithm claims that a set of six atomic invariants is sufficient for the purpose of obtaining a unique notation for simple SMILES. (More invariants are added for cases like Absolute SMILES to differentiate between structural and stereo-isomers).

The set is described below in descending order of priority :

1. number of connections
2. number of non-hydrogen bonds
3. atomic number
4. sign of charge
5. absolute charge
6. number of attached hydrogen

For the molecular graph in Figure 1, the initial atomic invariants for the atoms is described in Table 1(a) row labeled 'A'.

Rank Equivalence. The algorithm replaces the initial node invariant values by smaller numbers based on their sorted order to avoid numerical overflow since there is nothing intrinsically meaningful in their specific values. The row labeled 'B' in Table 1(a) describes the initial ranks.

Products of Primes. To obtain a canonical ordering of the nodes, and to obtain and identify all the symmetry classes of the nodes, an extended connectivity method using the product of the corresponding primes is used. This method is essentially used only to break ties between the initial node ordering to obtain a canonical order of the nodes.

The corresponding primes for the atoms of the molecular graph are described in the row labeled ' B* ' in Table 1(b). The product of the corresponding primes, which is the product of the primes associated with the atoms adjacent to a given atom, is displayed in the row labeled 'C' of Table 1(b).

Notice that node '10' was initially ranked '1' and appeared to belong to the same symmetry class as the other two nodes (1,7) with rank '1' when actually it did not, but by using the product of the corresponding primes we have been able to break the tie. (row 'D' of Table 1(b)).

By further following the steps of the algorithm as described in the unique SMILES algorithm [9], we obtain the final node partitioning as in Table 2: (the details of the steps are described in Table 1).

Table 1. Perception of Topological Symmetry classes for 3,5 di-ethyl toluene

Node id	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***
(a) Initial atomic invariants											
A	1,01, 06,0, 0,3	2,02, 06,0, 0,2	4,04, 06,0, 0,0	3,03, 06,0, 0,1	4,04, 06,0, 0,0	2,02, 06,0, 0,2	1,01, 06,0, 0,3	3,03, 06,0, 0,1	4,04, 06,0, 0,0	1,01, 06,0, 0,3	3,03, 06,0, 0,1
B	1	2	4	3	4	2	1	3	4	1	3
(b) Classification by product of primes											
B*	2	3	7	5	7	3	2	5	7	2	5
C	3	14	75	49	75	14	3	49	50	7	49
D	1	3	6	4	6	3	1	4	5	2	4
D*	2	5	13	7	13	5	2	7	11	3	7
E	5	26	245	169	245	26	5	143	147	11	143
F	1	3	7	5	7	3	1	4	6	2	4
F*	2	5	17	11	17	5	2	7	13	3	7
G	5	34	385	289	385	34	5	221	147	13	221
H	1	3	7	5	7	3	1	4	6	2	4

Table 2. Invariant partitioning and symmetry classes of nodes

Canon-ical label	***1***	***2***	***3***	***4***	***5***	***6***	***7***
Node ids	1,7	10	2,6	8,11	4	9	3,5

5 Explanation of Non-uniqueness

Breaking Ties. We have observed that the extended connectivity method using the product of corresponding primes was able to generate 7 different symmetry classes. Since the highest rank/label (7) is smaller than the number of nodes (11), there is more than one atom in certain symmetry classes. To avoid an arbitrary decision among these atoms in a given symmetry class, the algorithm proceeds to define a next step called 'breaking ties'. In this step, all the ranks of the atoms are doubled and the value of the first (lowest valued) atom that is tied is reduced by one. This set is then treated as a new invariant set and the previous algorithm for generating an invariant partitioning is repeated until the highest rank is equal to the number of nodes.

This concept of *double-and-tie-break* works for certain highly symmetric structures like cubane (consisting of eight carbon atoms at the vertices of a cube) irrespective of the initial ordering of the nodes. However, for our example in Figure 1, this ends up generating different canonical orderings of the graph resulting in different unique SMILES strings.

In our example following the *double-and-tie-break* step, we detect the first tie among the nodes with id's 1,7. We need to reduce the first lowest valued atom (out of nodes with id's 1,7) that is tied by one. In our example since we can have two starting nodes, and the notion of 'first' in this case is ambiguous, we can either choose node '1' or '7'. The algorithm fails to establish a mechanism of preference within the nodes belonging to the same symmetry class. It assumes that choosing any of the nodes within a symmetry class will result in the same unique SMILES string. This assumption works for certain regular graphs, however for graphs similar to our example, it does not work as desired.

For our example, by merely changing the input order of the nodes we can choose either node with id '1' or '7' as the first lowest valued atom and reduce its rank by '1', totally changing the start node for the depth-first traversal (DFT). If the graph was entered as shown in Figure 2, we would have ended up choosing the node with id '7' of Figure 1 as the first node and reduced its rank making it the start node for DFT.

By choosing the node with id '1' of Figure 1 as the first lowest valued atom to break the tie and continuing the algorithm, we obtain a canonical ordering as in Table 3. (This is just one of the many canonical orderings we can obtain, and is explained later).

However, if we had chosen the node with id '7' of Figure 1 as the first node, and continuing the algorithm one of the many canonical orderings we would obtain is shown in Table 4. This will be the case if we had input the graph as in Figure 2.

The problem of establishing an order within a given symmetry class can be solved for a limited enough collection of molecular graphs by considering more chemical/topological characteristics to distinguish between these atoms and establish a precedence order.

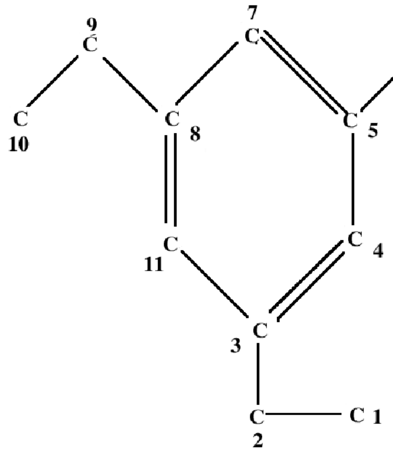

Fig. 2. Alternate input graph of 3,5 di-ethyl toluene

Table 3. One of the final canonical orderings choosing node with id 1 of Figure 1

Node id	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***
Canonical label	1	4	10	8	11	5	2	7	9	3	6

Table 4. One of the final canonical orderings choosing node with id 7 of Figure 1

Node id	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***
Canonical label	2	4	10	8	11	5	1	7	9	3	6

Obtaining the Unique SMILES String via GENES. By following the CANON process we have obtained a canonicalization of the graph. According to the CANON process, the nodes with the same rank are supposed to belong to the same symmetry class. The GENES process treats this structure as a tree and generates a SMILES string by Depth-First Traversal.

1. *Initial node selection*: The lowest canonical numbered atom is chosen as the starting point and it becomes the root of the Depth-First Traversal tree.

2. *Branching decision*: The following two rules apply :
   (a) Branch to double or triple bond in the ring if one exists or
   (b) Branch to the lower canonically numbered atom.

In this particular case, we observe that we can have two initial node selections, resulting in two different depth-first traversal trees from the two different canonical orderings described in Table 3 and Table 4.

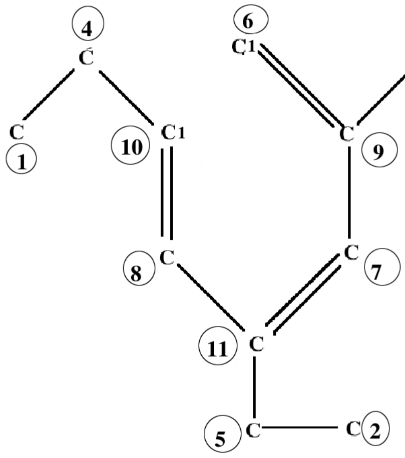

Fig. 3. Depth-first traversal associated with the initial node '1' and the canonical labeling described in Table 3. This gives the Unique SMILES **CCC1=CC(=CC(=C1)C)CC**

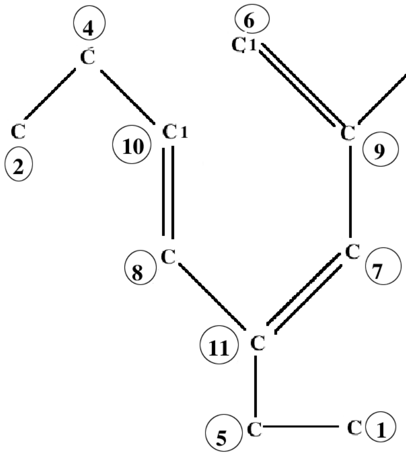

Fig. 4. Depth-first traversal associated with the initial node '7' and the canonical labeling described in Table 4. This gives the Unique SMILES **CCC1=CC(=CC(=C1)CC)C**

- The USMILES with start node id '1' is CCC1=CC(=CC(=C1)C)CC. The node IDs are described in Figure 1. The canonical labeling is described in Table 3. The depth-first traversal is described in Figure 3.
- The USMILES with start node id '7' is CCC1=CC(=CC(=C1)CC)C. The node IDs are described in Figure 1. The canonical labeling is described in Table 4. The depth-first traversal is described in Figure 4.

– The UCK algorithm generated using the web-service at [10], generates one unique key for this molecular graph, which is 85C7DC186897FD83D8ECB6B 167D988BE.

6 Experimental Studies

We have written a Python program implementing the CANGEN algorithm. The program takes as input an adjacency list of the molecular graph (described in [8]) and generates all possible unique SMILES strings for the graph. Once an invariant partitioning is obtained and it is determined that there is more than one node in any symmetry class, the script permutes the individual nodes within a symmetry class and generates all possible node selections.

For the example above, since there are 4 symmetry classes, we will get 16 different final invariant partitionings. Once we obtain these different partitionings we proceed to break ties in each of them and continue the remaining steps of the algorithm. Not all of the 16 final canonical orderings obtained from these different invariant partitionings generate different SMILES strings — only a subset of these generate different unique SMILES strings. In our example only two of the different final canonical orderings (Table 3 and Table 4) generate two different unique SMILES strings.

A web interface to this program can be accessed at [6].

7 Examples from the NCI Database

Here are some counter examples found in the NCI Database [7]. A web interface to these counter examples can be accessed at [6]. For each of the examples in this section:

1. We verified that the two different unique SMILES strings obtained map onto the same molecular graph via the on-line implementation [4] of the depict algorithm [3] provided by Daylight software [4].
2. We also verified this using another on-line implementation [5] of the CANGEN algorithm provided by the cactus service of the NCI chemical structure database. Using this service, one can input a SMILES string and get the unique SMILES for it.

NSC ID 4420. Here are two different Unique SMILES strings for N, N-Diallylmelamine with NSC id 4420:

– NC1=NC(=NC(=N1)N(CC=C)CC=C)N
– NC1=NC(=NC(=N1)N)N(CC=C)CC=C

See Figure 5 for the molecular graph. The unique key generated by UCK for this compound is: 020A134950962096577666701295295E.

NSC ID 10392. Here are two different Unique SMILES strings for 2, 4-Mesitylenediamine with NSC ID 10392.

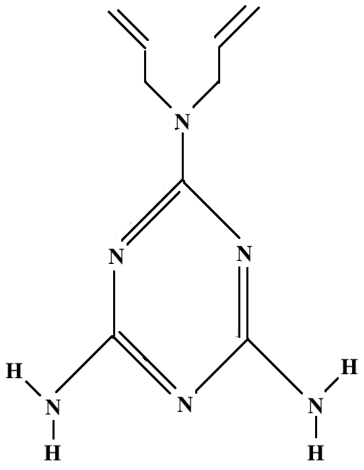

Fig. 5. Structural formula of N, N-Diallylmelamine with NSC id 4420

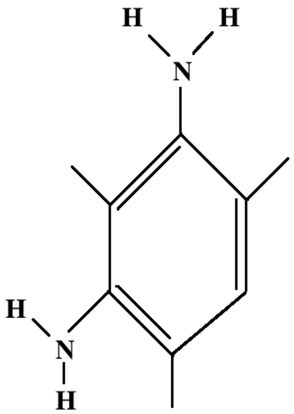

Fig. 6. Structural formula of 2,4-Mesitylenediamine with NSC id 10392

- CC1=C(N)C(=C(N)C(=C1)C)C
- CC1=CC(=C(N)C(=C1(N))C)C

See Figure 6 for the molecular graph. The unique key generated by UCK is F61473AE54FEC1737F7D15590650BBA2.

NSC ID 1889. Here are two different Unique SMILES strings for Pentamethylbenzene with NSC ID 1889.

- CC1=C(C)C(=C(C)C(=C1)C)C
- CC1=CC(=C(C)C(=C1(C))C)C

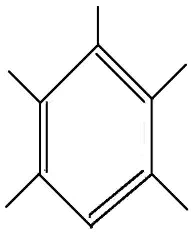

Fig. 7. Structural formula of Pentamethylbenzene with NSC ID 1889

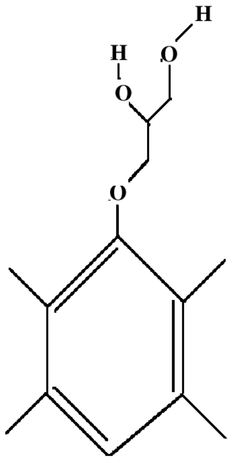

Fig. 8. Structural formula of 3-(3,5-dimethylphenoxy)-1,2-propanediol with NSC id 25239

See Figure 7 for the molecular graph. The unique UCK key generated by UCK: 1C5659F3ED5E10F02310455B56649849.

NSC ID 25239. Here are two different Unique SMILES strings for 3-(3,5-dimethylphenoxy)-1,2-propanediol with NSC id 25239.

- CC1=CC(=CC(=C1)C)OCC(O)CO
- CC1=CC(=CC(=C1)OCC(O)CO)C

See Figure 8 for the molecular graph. The unique key generated by UCK: AFD17D1BB28847F4FFAAD8C744A268AE.

8 Summary and Conclusion

Data integration involving chemical compounds is greatly aided by attaching unique IDs to chemical compounds. This is especially important when working with distributed bioinformatics data.

It has been recognized for some time that common names for chemicals, IUPAC names, CAS numbers, and general SMILES strings do not provide a good infrastructure for data integration. In this paper, we show that Unique SMILES strings [2] are also not a good foundation for data integration.

As the examples in the section above show, there are relatively simple chemical compounds that do not have Unique SMILES IDs. We have computed additional counter examples using our python script and these can be accessed at [6].

The CANGEN component of the Unique SMILES algorithm starts with a set of graph invariants and uses these to generate a canonical ordering of the nodes. This is then used as a basis for a depth-first traversal of the graph to generate the Unique SMILES string. Unfortunately, there is no set of invariants known that can distinguish all possible graph asymmetries that arise with the molecular graphs in common databases, such as the NCI database.

Although the Universal Chemical Key (UCK) algorithm [20] does not generate easy to interpret strings, it does generate unique keys for common databases such as the NCI database.

This suggests a strategy of using UCK like strings as keys to integrate distributed bioinformatics data, supplemented by SMILES-like strings that are easier to interpret.

References and Notes

1. David Weininger, SMILES, a Chemical Language and Information System 1: Introduction to Methodology and Encoding Rules, Medicinal Chemistry Project, Pomona College, 1988.
2. David Weininger, Arthur Weininger and Joseph L. Weininger, SMILES 2: Algorithm for Generation of Unique SMILES Notation, Daylight Chemical Information Systems, Irvine, California 92714, 1989. *Note* that although the Unique SMILES implementation has been changed by the Daylight Chemical Information System, this appears to be the most recent publication describing the algorithm.
3. David Weininger, SMILES 3: Depicting Graphical Depiction of Chemical Structures, Daylight Chemical Information Systems, New Orleans, Louisiana.
4. A SMILES to graph translation can be found at http://www.daylight.com/daycgi/depict.
5. A SMILES to UNIQUE SMILES translation can be found at http://cactus.nci.nih.gov/services/translate/.
6. More counter examples can be found at the web site http://ncdm171.lac.uic.edu/neglur/USMILES/USMILES.html.
7. NCI database, retrieved from http://129.43.27.140/ncidb2/ on March 2, 2005.

8. Sample adjacency list used -

```
{1:[['C',1,6,'0',3],[[1,2]]],
2:[['C',2,6,'0',2],[[1,1],[1,3]]],
3:[['C',4,6,'0',0],[[1,2],[2,4],[1,11]]],
4:[['C',3,6,'0',1],[[2,3],[1,5]]],
5:[['C',4,6,'0',0],[[1,4],[1,6],[2,8]]],
6:[['C',2,6,'0',2],[[1,5],[1,7]]],
7:[['C',1,6,'0',3],[[1,6]]],
8:[['C',3,6,'0',1],[[2,5],[1,9]]],
9:[['C',4,6,'0',0],[[1,8],[1,10],[2,11]]],
10:[['C',1,6,'0',3],[[1,9]]],
11:[['C',3,6,'0',1],[[1,3],[2,9]]]}
```

9. CANON Algorithm (Extract from Reference [2])-

```
(1) Set the atomic vector to initial invariants.
    Go to step 3.
(2) Set vector to product of primes corresponding to
    neighbors' ranks.
(3) Sort vector, maintaining stability over
    previous ranks.
(4) Rank atomic vector.
(5) If not invariant partitioning, go to step 2.
(6) On first  pass, save partitioning as symmetry classes.
(7) If  highest rank is smaller than number of nodes,
    break ties, go to step 2.
(8)... else done
```

10. See http://bioweb.dataspaceweb.org/chemicalKeys, retrieved on March 2, 2005.
11. T. Beyer and A. Proskurowski, Symmetries in graph coding, in Proceedings of Northwest 1976 ACM–CIPS Pacific Regional Symposium, 198-203 (1976)
12. C. B HM and A. Santolini, A quasi-decision algorithm for the p-equivalence of two matrices. ICC Bull. 8, 1 (1964), 57-69.
13. IUPAC, Nomenclature of Organic Chemistry, Pergamon Press, Oxford, 1979
14. Klin M. H., Lebedev O. V., Pivina T. S., Zefirov N. S., Nonisomorphic cycles of maximum length in a series of chemical graphs and the problem of application of IUPAC nomenclature rules. MATCH, 1992, 27, 133-151.
15. See http://www.iupac.org/projects/2000/2000-025-1-800.html, retrieved on March 2, 2005.
16. Milan Randic, Gregory M. Brissey, Charles L. Wilkins: Computer perception of topological symmetry via canonical numbering of atoms. Journal of Chemical Information and Computer Sciences 21(1): 52-59 (1981)
17. B. McKay, Practical Graph Isomorphism, Congr. Numer. 30, 45-87, 1981.
18. H. L. Morgan, The Generation of a Unique Machine Description for Chemical Structures – A Technique Developed at Chemical Abstracts Service. , J. Chem. Doc., 1965 , 5 , 107-113.
19. J. Braun, R. Gugisch, A. Kerber, R. Laue, M. Meringer, C. Rcker: MOLGEN-CID, A Canonizer for Molecules and Graphs Accessible through the Internet, Journal of Chemical Information and Computer Sciences 44, 542-548, 2004.
20. Robert Grossman, Donald Hamelberg, Pavan Kasturi, and Bing Liu, Experimental Studies of the Universal Chemical Key (UCK) Algorithm on the NCI Database of Chemical Compounds, Proceedings of the 2003 IEEE Computer Society Bioinformatics Conference (CSB 2003), IEEE Computer Society, Los Alamitos, California, pages 244-250