# An Algorithm for Assigning Unique Keys To Metabolic Pathways

Fang Fang
*Dept. of Computer Science*
*Univ. of Illinois at Chicago*
*ffang2@uic.edu*

Robert L. Grossman
*Laboratory for Advanced Computing*
*Univ. of Illinois at Chicago*
*frank@lac.uic.edu*

Xiangjun Liu
*Laboratory for Advanced Computing*
*Univ. of Illinois at Chicago*
*grossman@uic.edu*

## Abstract

*Different databases of metabolic pathways assign pathways different keys. For this reason, it is difficult to automatically compare pathways across databases. We introduce an algorithm called the Universal Pathway Key algorithm or UPK that assigns essentially unique keys to metabolic pathways. We show that the UPK algorithm assigns unique keys to the pathways in the MetaCyc database and can also be used to detect duplicate pathways. The UPK algorithm is a simple generalization of the UCK algorithm introduced in [7] that assigns essentially unique keys to chemical compounds.*

## 1. Introduction

Currently, there are many databases that contain collections of metabolic pathways. Commonly used databases of metabolic pathways include KEGG [1], BioCyc [2], CGAP [3] and UM-BBD [4]. MetaCyc [5] and EcoCyc [6] are two component databases in BioCyc. MetaCyc contains over 900 pathways from more than 900 different organisms whereas EcoCyc is the encyclopedia of *Escherichia coli* K12 genes and metabolism.

Most of these databases use their own mechanism for labeling pathways, making it difficult to compare pathways across databases. Adding to the problem, a pathway may even have been entered into the same database under different names.

In this paper, we consider graphs whose nodes and/or edges are labeled and are interested in algorithms that assign labels to the graphs as a whole. A label in this context is simply a string. If the map from graphs to strings is injective, we call the map a key. In this case, different labeled graphs are assigned different keys. If two graphs are assigned the same key, then they are identical as labeled graphs.

Assigning keys to graphs would be relatively easy if there were a natural order that can be assigned to the nodes in a graph. Unfortunately, in general an algorithm is not known for doing this, although in many special cases, such algorithms do exist.

In [7], we introduced an algorithm called the UCK or Unique Chemical Key algorithm that assigns essentially unique keys to graphs. In this paper, we show how this algorithm can be used to assign essentially unique keys to metabolic pathways.

We also describe the results of applying this algorithm to the metabolic pathways in the MetaCyc database.

By essentially unique, we mean that although the keys are not guaranteed to be unique by construction they do tend to be unique when used on naturally occurring graphs, such as those representing chemical compounds or pathways. The UCK algorithm depends upon a depth parameter d. As d increases, the keys are more likely to be unique. For example, when applied to the National Cancer Institute (NCI) database of 236,917 chemical compounds [8], all compounds had unique labels with d=3, and all except six compounds had unique labels with d=2.

We make two simplifying assumptions in this paper. First, we assume that the starting point of this algorithm is the representation of a metabolic pathway as a graph. This is a simplifying assumption since different scientists may view a given pathway as including more compounds or fewer compounds. The second simplifying assumption is that we consider only the primary substrates and primary products involved in the pathway's reactions.

**Acknowledgments.** This work was funded in part by the Chicago Biomedical Consortium with support from The Searle Funds at The Chicago Community Trust.

## 2. Graphical Models for Metabolic Pathways

### 2.1 Compound Graphs and Reaction Graphs

We begin with some definitions. By a metabolic pathway, we mean a series of two or more interconnected-enzyme mediated, or spontaneous, chemical reactions that take place in a cell. A

chemical reaction consists of one or more substrates (chemical compounds) transforming into one or more products (chemical compounds) via an enzyme (represented with an Enzyme Commission Number) [9].

Various types of graphs – including compound graphs, reaction graphs, bipartite graphs, and Petri nets – have been proposed for the analysis of metabolic networks [10, 11, 12, 13, 14 and 15]. In a compound graph, the nodes in the graph represent compounds and edges in the graph represent reactions/enzymes. In a reaction graph, the nodes represent reactions/enzymes and edges represent the compounds. In a bipartite graph, there are two classes of nodes – compound nodes and reaction nodes. There can be edges between compound nodes and reaction nodes, but not between nodes of the same class.

Petri nets also have two types of nodes – place nodes and transition nodes. In a Petri net, place nodes represent compounds and the transition nodes represent reactions.

Note that in general the graphs associated with pathways are not uniquely labeled. Figure 2 gives an example. In this example, different reaction nodes have the same label (e.g., 2.5.1.36 or 1.14.13.85) because one enzyme can catalyze multiple reactions, which may occur simultaneously in one pathway.

There is no consensus yet about which of these graphical models, or other models, should be used to represent metabolic pathways.

In this paper, we represent pathways using bipartite graphs, although the algorithm we describe below for assigning keys to pathways also works with the various other representations.

### 2.2 Bipartite Pathway Graphs

As mentioned, in a bipartite graph, there are two classes of nodes – compound nodes and reaction nodes. A directed edge from a compound node to a reaction node denotes that the compound node is a substrate of the reaction, while an edge from a reaction node to a compound node denotes that the compound node is a product of the reaction. See Table 1 for some examples.

A labeled directed bipartite graph G can be associated with a metabolic pathway as follows:

1. The chemical compounds in a pathway are the compound nodes of the graph represented as ellipses.

2. The reactions in a pathway are the reaction nodes of the graph represented as rectangles.

3. An edge connects a compound node with a reaction node. A directed edge from a compound node to a reaction node denotes a substrate of the reaction, while a directed edge from a reaction node to a compound node denotes a product.

4. Assign labels to the compound nodes. This can be done in different ways, including using the Canonical SMILES string, which can be obtained from the PubChem database [20], or the UCK string, which can be computed using the algorithm described in [7].

5. Assign labels to the reaction nodes. This can be done using the EC number of the enzyme(s) that catalyze(s) the chemical reaction. We also consider the case below in which the reaction nodes are unlabeled, or equivalently, labeled using the empty string.

See Figure 1 for an example.

## 3. Unique Pathway Keys (UPK)

### 3.1 UCK Algorithm for Graphs

In this section, we recall the Universal Chemical Key (UCK) algorithm following [7]:

**1. Assign labels to each node in the graph.** For a depth d=1, 2, 3, …, we define a map $\lambda^{(d)}$ that labels each node b in G. The map $\lambda^{(d)}$ is defined recursively from $\lambda^{(d-1)}$ as follows:

Step 1a: Define $\lambda^{(0)}$ (b) to be the label of the node b itself.

Step 1b: For a node **a** with children $b_1 \cdots , b_k$, compute the k labels: $\lambda^{(d-1)}(b_1), \cdots , \lambda^{(d-1)}(b_k)$.

Step 1c. Lexicographically order the k labels and concatenate the result:

$$\lambda^{(d)}(a) = \lambda^{(d-1)}(b_{i1}) \cdots \lambda^{(d-1)}(b_{ik})$$

Notice that this is equivalent to considering all paths of length d from a, assigning the string obtained by concatenating the labels of the nodes along the path from **a**, lexicographically ordering the resulting strings, and then concatenating the result.

Since the label of **a** in $\lambda^{(d)}$ occurs once in each $\lambda^{(d-1)}(b_i)$, a simple variant is to remove all occurrences of **a**'s label from $\lambda^{(d-1)}(b_1), \cdots , \lambda^{(d-1)}(b_k)$, lexicographically order them to produce the string $\lambda^{(d-1)}(b_{i1}) \cdots \lambda^{(d-1)}(b_{ik})$, and then define $\lambda^{(d)}(a)$ to be:

$$\text{label}(a)\ \lambda^{(d-1)}(b_{i1}) \cdots \lambda^{(d-1)}(b_{ik})$$

**2. Assign labels to each pair of nodes in the graph.** Re-label G with labels $\lambda^{(d)}$ computed as above. For **a** and b two nodes in G, define the label

$$\mu(a,b) = [\lambda^{(d)}(a)\ n\ \lambda^{(d)}(b)],$$

where n is the length of shortest path from **a** to b and […] denotes the concatenation of the strings inside the brackets.

**3. Assign a label to the graph.** We now define the label of the graph as a whole as follows:

$$\mu(G) = [\ \text{Lex}(\ \mu(a,b) \mid a, b \text{ nodes in the graph G})\ ],$$

where Lex denotes the lexicographical ordering of the strings $\mu(a,b)$ for all nodes **a** and b, and, as above, [ … ] denotes the concatenation of the strings

inside the brackets. In general, this produces a long string, which is often hashed in order to shorten it. For example, the MD5 algorithm can be used to hash the string [16].

### 3.2 Unique Pathway Key (UPK) Algorithm

Note that the UCK algorithm can be applied to the bipartite pathway graph defined in Section 2.2 above. Call the resulting key obtained the Unique Pathway Key or UPK. Also, note that the UCK algorithm can also be used if the pathway is represented as a graph in other ways.

As an example, the UPK for depth d=1 for UDP-glucose conversion, a metabolic pathway in MetaCyc [17], is shown in Appendix A. The MD5 hash of the UPK in Appendix A is

D056FD6068D74E68E4C777C5352959B1.

## 4. Experimental Results

The experimental results reported below use the Version 10.6 of MetaCyc [17, 18 and 19]. This version contains 984 pathways. Among them are 107 super-pathways (a super-pathway contains combinations of individual pathways to show the relationships between them. Individual pathways that separate from one another or that merge are combined into super-pathways).

Because the substrates or products or reaction directions are missing in the reactions of some pathways, and because the canonical SMILES strings of about six hundred chemical compounds in some pathways cannot be found on PubChem, there are a total of 714 metabolic pathways associated with unique keys in MetaCyc, excluding the super-pathways.

Among the 714 metabolic pathways that have been UPK, there are five pairs of pathways in the MetaCyc, shown in Table 3 and Table 4, that have been found to have the same unique keys when the depth d in the UCK algorithm is set to 1 or 2. The experimental results have shown that the five pairs of metabolic pathways are the same. (Although the pictures of these metabolic pathways on the MetaCyc website may look different at the first glimpse, they are actually the same pathways because the linked pathways in the pictures are not part of the metabolic pathways.)

If enzymes are ignored among all pathways by labeling the reactions nodes with empty strings, then there are eleven pairs/triple/tetrad of pathways associated with same keys in the MetaCyc database when d is set to 1 or 2. Table 5 and Table 6 give the pathways associated with the same keys when enzymes are ignored and when the depth is set to 1 or 2, respectively.

## 5. Conclusion

In this paper, the bipartite graph based UCK algorithm has been applied to generate unique keys for metabolic pathways. The algorithm views the structure of a metabolic pathway as a labeled directed bipartite graph. The algorithm was experimentally tested on 714 metabolic pathways in the MetaCyc database and generated unique keys for all the uniquely identified structures. The experimental results have shown that five identical pairs of metabolic pathways appear twice in MetaCyc. The unique keys generated by the UCK algorithm can be successfully used to determine if a metabolic pathway has already existed in one database. With this approach using UPKs, it is quite simple to compare metabolic pathways across databases and within a database.

## References


[1] See www.genome.ad.jp/kegg.

[2] See www.biocyc.org.

[3] See cgap.nci.nih.gov.

[4] See umbbd.msi.umn.edu/search

[5] See metacyc.org

[6] See ecocyc.org

[7] Robert L. Grossman, Pavan Kasturi, Donald Hamelberg, Bing Liu, "An Empirical Study of the Universal Chemical Key Algorithm for Assigning Unique Keys to Chemical Compounds", Journal of Bioinformatics and Computational Biology, Volume 2, Number 1, 2004, pages 155-171, 2004.

[8] See cactus.nci.nih.gov

[9] See for example, www.chem.qmul.ac.uk/iubmb/enzyme/

[10] Goto S., Bono H., Ogata H., Fujibuchi W., Nishioka T., Sato K., & Kaneisha M., "Organizing and Computing Metabolic Pathway Data in Terms of Binary Relations", In Pacific Symposium on Biocomputing'97(ed. Altman R.B., Dunker K., Hunter L., & Klein T.E.), World Scientific Publishing, Singapore. 1996, pp. 175-186.

[11] Peter D. Karp and Suzanne M. Paley, "Representations of Metabolic Knowledge: Pathways", Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, 1994.

[12] Yves Deville, David Gilbert, Jacques van Helden and Shoshana J. Wodak, "An Overview of Data Models for the Analysis of Biochemical Pathways", BRIEFINGS IN BIOINFORMATICS, 4(3): pp. 246–259, 2003.

[13] Jeong, H., Tombor, B., Albert, R. et al, "The Large-Scale Organization of Metabolic Networks", Nature, Vol. 406, pp. 651–654, 2000.

[14] Van Helden, J., Wernisch, L., Gilbert, D. and Wodak, S. J., "Graph-based Analysis of Metabolic Networks", Bioinformatics and Genome Analysis, Springer-Verlag, Germany, pp. 245–274. 2002.

[15] Forst, C. V. and Schulten, K., "Evolution of

metabolisms: A New Method for the Comparison of Metabolic Pathways", Proceedings of the Third Annual International Conference on Computational Molecular Biology, ACM Press, New York, NY, pp. 174–181. 1999.

[16] R. Riverst. The MD5 message digest algorithm, RFC1321, 1992.

[17] Karp, P.D., Riley, M., Paley, S., and Pellegrini-Toole, A., "The MetaCyc Database", Nucleic Acids Research, 30(1):59-61 2002.

[18] Krieger, C.J., Zhang, P., Mueller, L., Wang, A., Paley, S., Arnaud, M., Pick, J., Rhee, S.Y., and Karp, P.D., "MetaCyc: A Multiorganism Database of Metabolic Pathways and Enzymes", Nucleic Acids Research, 32(1), 2004.

[19] Caspi, R., Foerster, H., Fulcher, C.A., Hopkinson, R., Ingraham, J., Kaipa, P., Krummenacker, M., Paley, S., Pick, J., Rhee, S.Y., Tissier, C., Zhang, P., and Karp, P.D., "MetaCyc: A Multiorganism Database of Metabolic Pathways and Enzymes", Nucleic Acids Research, 34:D511-6, 2006.

[20] See: http://pubchem.ncbi.nlm.nih.gov

## Appendix A - Unique string for UDP-glucose Pathway

2.7.7.9C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)OOP(=O)(O)OP(=O)(O)O1C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O5.1.3.22.7.7.9C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OOP(=O)(O)OP(=O)(O)O1OP(=O)(O)OP(=O)(O)O3.6.1.12.7.7.9C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OOP(=O)(O)OP(=O)(O)O23.6.1.1OP(=O)(O)O2.7.7.9C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OOP(=O)(O)OP(=O)(O)O25.1.3.2C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O2.7.7.9C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OOP(=O)(O)OP(=O)(O)O3OP(=O)(O)O3C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O2.7.7.9C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OOP(=O)(O)OP(=O)(O)O3OP(=O)(O)O3.6.1.1OP(=O)(O)O1OP(=O)(O)O5.1.3.2C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O1C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O5.4.2.2C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O1C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O2.7.7.95.4.2.2C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O22.7.7.9C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OOP(=O)(O)OP(=O)(O)O5.4.2.2C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O3C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O5.1.3.25.4.2.2C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O3OP(=O)(O)OP(=O)(O)O3.6.1.15.4.2.2C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O43.6.1.1OP(=O)(O)O5.4.2.2C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O45.1.3.2C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O5.4.2.2C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O5C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O5.4.2.2C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O5OP(=O)(O)OC(C1C(C(C(C(O1)O)O)O)O)OP(=O)(O)O5.4.2.215.4.2.2C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)OC(C1C(C(C(C(O1)O)O)O)O)OP(=O)(O)O5.4.2.22C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O2.7.7.9C(C1C(C(C(C(O1)O)O)O)O)OP(=O)(O)O5.4.2.232.7.7.9C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OOP(=O)(O)OP(=O)(O)OC(C1C(C(C(C(O1)O)O)O)O)OP(=O)(O)O5.4.2.24C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O5.1.3.2C(C1C(C(C(C(O1)O)O)O)O)OP(=O)(O)O5.4.2.24OP(=O)(O)OP(=O)(O)O3.6.1.1C(C1C(C(C(C(O1)O)O)O)O)OP(=O)(O)O5.4.2.253.6.1.1OP(=O)(O)OC(C1C(C(C(C(O1)O)O)O)O)OP(=O)(O)O5.4.2.255.1.3.2C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OC(C1C(C(C(C(O1)O)O)O)O)OP(=O)(O)O5.4.2.26C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OC(C1C(C(C(C(O1)O)O)O)O)OP(=O)(O)O5.4.2.26OP(=O)(O)OC(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O2.7.7.912.7.7.9C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OOP(=O)(O)OP(=O)(O)OC(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O2.7.7.92C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O5.1.3.2C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O2.7.7.92OP(=O)(O)OP(=O)(O)O3.6.1.1C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O2.7.7.933.6.1.1OP(=O)(O)OC(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O2.7.7.935.1.3.2C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OC(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O2.7.7.94C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OC(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O2.7.7.94OP(=O)(O)OC1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O5.1.3.215.1.3.2C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OC1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O5.1.3.22C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OOP(=O)(O)OP(=O)(O)O3.6.1.113.6.1.1OP(=O)(O)OOP(=O)(O)OP(=O)(O)O3.6.1.12OP(=O)(O)O

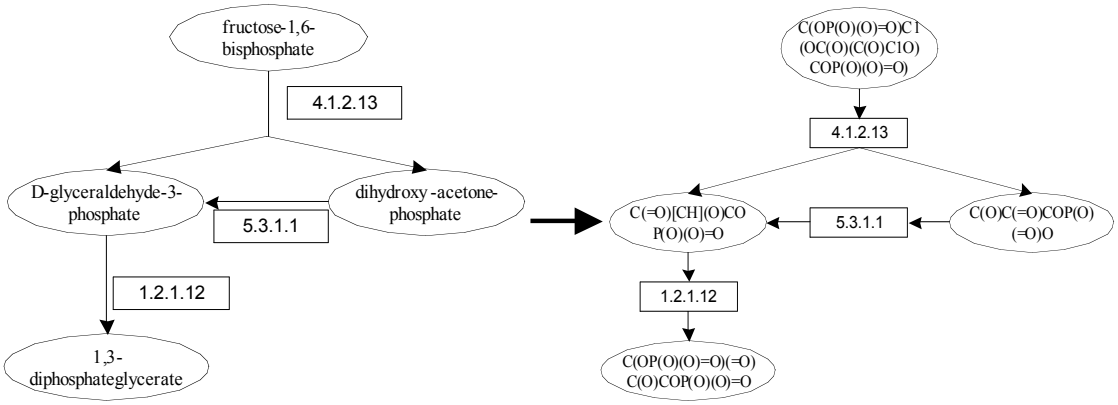



**Figure 1. Transformation of a metabolic pathway into the labeled directed bipartite graph.**

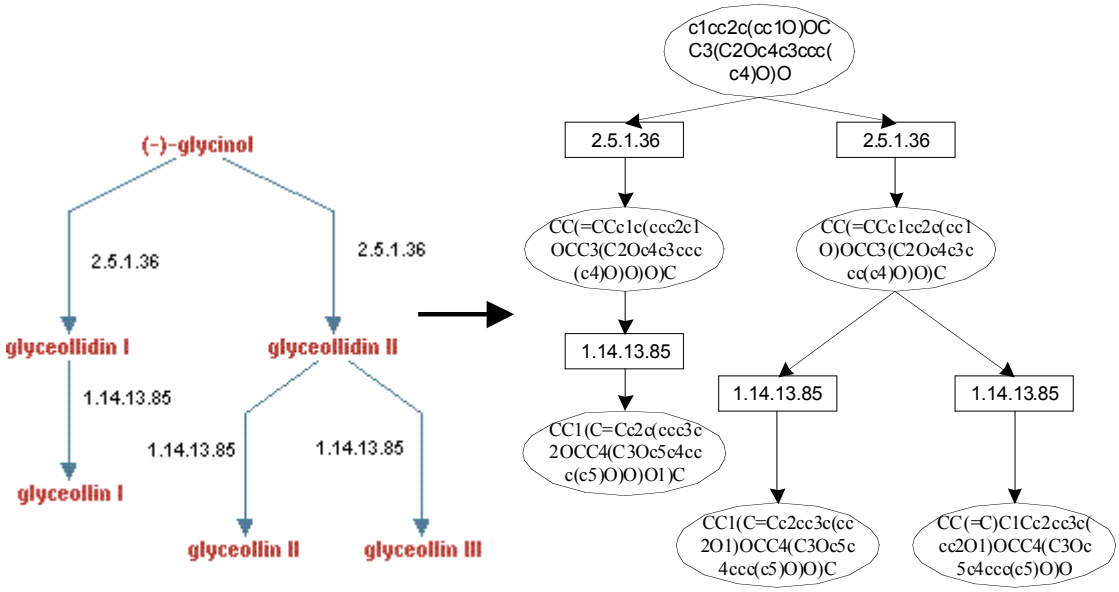
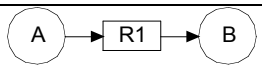
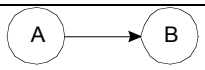



**Figure 2(a). MetaCyc pathway: glyceollin biosynthesis II**

**Figure 2(b). The labeled directed bipartite graph representation of glyceollin biosynthesis II**

**Figure 2. An example of non-uniquely labeled directed bipartite graph**

**Table 1. Bipartite graph representation of reaction example**

| Reaction Type | Reaction | Bipartite Graph | Compound Graph | Reaction Graph |
|---|---|---|---|---|
| Reversible | R1:A↔B |  |  |  |
| Irreversible | R1:A → B |  |  |  |

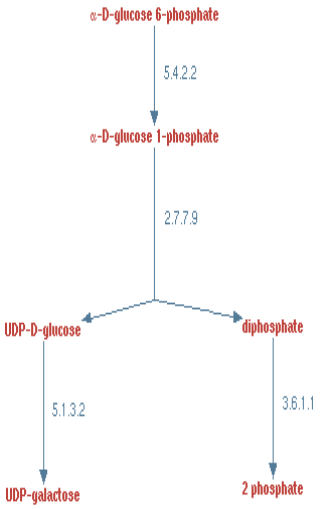



**Figure 3(a).**
**UDP-glucose conversion**

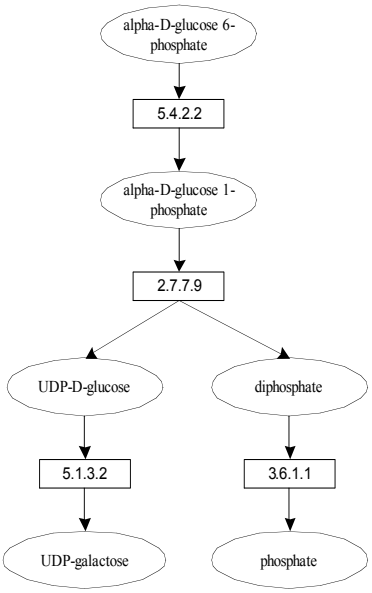



**Figure 3(b).**
**The corresponding bipartite pathway graph**

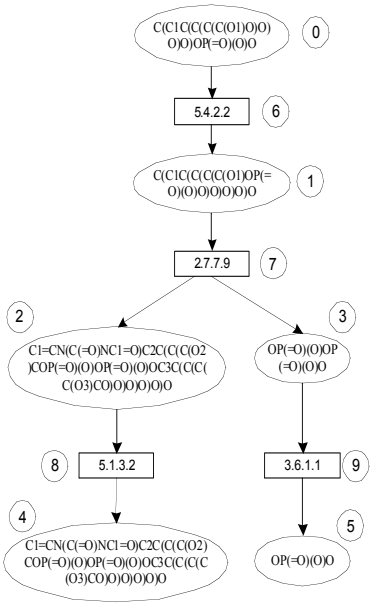



**Figure 3(c). The pathway graph labeled with SMILES strings and node numbers.**

**Table 2. The nodes labels $\lambda^{(1)}$**

| Node Index | Node Label $\lambda^{(1)}$ |
|---|---|
| 0 | C(C1C(C(C(C(O1)O)O)O)O)OP(=O)(O)O5.4.2.2 |
| 1 | C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O2.7.7.9 |
| 2 | C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O5.1.3.2 |
| 3 | OP(=O)(O)OP(=O)(O)O3.6.1.1 |
| 4 | C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O |
| 5 | OP(=O)(O)O |
| 6 | 5.4.2.2C(C1C(C(C(C(O1)OP(=O)(O)O)O)O)O)O |
| 7 | 2.7.7.9C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)OOP(=O)(O)OP(=O)(O)O |
| 8 | 5.1.3.2C1=CN(C(=O)NC1=O)C2C(C(C(O2)COP(=O)(O)OP(=O)(O)OC3C(C(C(C(O3)CO)O)O)O)O)O |
| 9 | 3.6.1.1OP(=O)(O)O |

**Table 3. Metabolic pathway pairs associated with identical keys (d = 1)**

| Pathway ID | Pathway ID | Common unique key for pathways in MetaCyc (d = 1) |
|---|---|---|
| PWY-3921 | PANTO-PWY | 80FC59FA114257D968DFB11340596A5D |
| PWY-40 | PWY0-823 | F5F6EE9D53346CE260E8BB9F2188D3B3 |
| PWY-43 | ARGDEG-III-PWY | 4CEDB573EB9EA21C96C5DCAA600981A5 |
| PWY-5374 | PWY-5375 | 4C27126F42F2F3615605B3CC5BB48845 |
| BETSYN-PWY | CHOLINE-BETAINE-ANA-PWY | 78345AA6F54BAA8C5042E08C13ED55EF |

**Table 4. Metabolic pathway pairs associated with identical keys (d = 2)**

| Pathway ID | Pathway ID | Common unique keys for pathway in MetaCyc (d = 2) |
|---|---|---|
| PWY-3921 | PANTO-PWY | 83750D0FC3623277A99146907AD5EEB8 |
| PWY-40 | PWY0-823 | CA77B0D0110A16202BDE0945886D22F3 |
| PWY-43 | ARGDEG-III-PWY | 287E04E230CF0483DDB2AECB5CDBA462 |
| PWY-5374 | PWY-5375 | 6600BB174C00CC4F968E0D18505EA05A |
| BETSYN-PWY | CHOLINE-BETAINE-ANA-PWY | 5B070F35B2B63D1B530C51C341BBF88E |

**Table 5. Enzyme-ignored pathway pairs associated with identical keys (d = 1)**

| Pathway ID | Pathway ID | Pathway ID | Pathway ID | Common unique keys for pathway in MetaCyc (d = 1) |
|---|---|---|---|---|
| PWY-3921 | PANTO-PWY | | | 5AAF1CF673777506F76A031B68D23372 |
| PWY-40 | PWY0-823 | | | 161BF0B4C9543B339DBCE12C1B21A0B8 |
| PWY-43 | ARGDEG-III-PWY | | | C1C250FF74EC8BD01093E2D102185C44 |
| PWY-5374 | PWY-5375 | | | CEE4073DCC4658533A10832855C2790E |
| BETSYN-PWY | CHOLINE-BETAINE-ANA-PWY | PWY-3721 | PWY-3722 | EA9DC402F56AA034A14544132D1C5A3C |
| PWY-5336 | PWY-1164 | | | 3CFDA2E6332A5C29D38F065096AE42BA |
| PWY66-21 | PWY66-161 | PWY66-162 | | 53CE87E84250B07E00C8390B04E87F1A |
| PROSYN-PWY | PROLINE-SYN2-PWY | | | 7817F9E548C2CCEFD49DC03AF825006A |
| DARABITOLUTIL-PWY | LARABITOLUTIL-PWY | | | 9A188A17705A87F4F1350EEF70713A6A |
| GLUTSYN-PWY | GLUGLNSYN-PWY | | | C8C8DC34E66A47D3E54ED3E36AB486B8 |
| GLUTSYNIII-PWY | GLUTAMATE-SYN2-PWY | | | DF8F620DC6C8792AA11749034A6A6F4B |

**Table 6. Enzyme-ignored pathway pairs associated with identical keys (d = 2)**

| Pathway ID | Pathway ID | Pathway ID | Pathway ID | common unique keys for pathway in MetaCyc (d = 2) |
|---|---|---|---|---|
| PWY-3921 | PANTO-PWY | | | 191F3E22567F9AD5F0E62F6AEA59819B |
| PWY-40 | PWY0-823 | | | C3B2BB4A24799FA7BE60972789A8A359 |
| PWY-43 | ARGDEG-III-PWY | | | D3BE904C22F8D4D598F0AC113812024C |
| PWY-5374 | PWY-5375 | | | A1BD4B3FA61AA663DF8CE8E61D69312E |
| BETSYN-PWY | CHOLINE-BETAINE-ANA-PWY | PWY-3721 | PWY-3722 | A3CF06A35AFEA347DC3FF516AEF3DE7D |
| PWY-5336 | PWY-1164 | | | 5FA9CA171DCE1B1085CCC366D352E1AF |
| PWY66-21 | PWY66-161 | PWY66-162 | | 21ED45A4E3BDACD1D8F9134D7245B095 |
| PROSYN-PWY | PROLINE-SYN2-PWY | | | 416AEEB875646442A0C32D4CEB4B8627 |
| DARABITOLUTIL-PWY | LARABITOLUTIL-PWY | | | 41B0926C77C6E7F450873CCD229CF6CD |
| GLUTSYN-PWY | GLUGLNSYN-PWY | | | 7235332DD53801DD9D26FB4B60C1439A |
| GLUTSYNIII-PWY | GLUTAMATE-SYN2-PWY | | | C3485EB12F38ABFC4E7EE527FA1A7E3 |