International Journal of Bioinformatics and Biological Science: v.1 n.1 p.87-94. March, 2013

Graph transforms for modeling chemical reaction pathways

V. Yegnarayanan[1] and Gulshan Wadhwa[2]

[1]Senior Member IEEE & Senior Professor, Department of Mathematics, Velammal Engineering College, Chennai-600 066
[2]Joint Director, Department of Bio-Technology, Ministry of Science and Technology, New Delhi-110 003.

[*]Corresponding Author: Gulshan Wadhwa gulshan.dbt@nic.in

Received: 24 December 2012; Accepted: 03 January 2013

ABSTRACT

In this paper we discuss some general graph-theoretic formalism analogous to the chemical concepts of reactions and kits of reaction rules. We believe that this mathematical formalism may lead to improvements in Computer-assisted Organic Synthesis. Further we indicate that hypergraphs and its structure can be used to model the chemical reaction network.

Keywords: Graphs, Chemical Reaction Network, Metadigraphs, Hypergraphs, Stochastic Graph Transformation

INTRODUCTION

Chemical graphs are labeled graphs in which letters assigned to the vertices denote the atoms of a molecule and letters assigned to the edges denote the bonds. They are widely used in chemistry as representations of molecular structures. A chemical reaction type can be viewed as a transform which, when applied to a compound, generates new compounds. For example, benzene undergoes hydroxylation to form hydroxybenzene. This idea has been made algorithmically explicit in a variety of ways with respect to chemical graphs in computer-assisted organic synthesis(Barone and Chanon,1986, Koca *et al* ,1989 and Ugi *et al*,1979)and computer - assisted drug metabolism (Darvas, 1988 and Beck and Cowan,1978). The goal in these approaches is the construction of a kit of reaction rules that enables the computer to suggest chemical paths of compounds leading to or from a compound of interest. Koca *et al.* have re-examined the approach by Ugi *et al*. using graph theoretic concepts. Here we see some general graph - theoretic formalism analogous to the chemical concepts of reactions and kits of reaction rules, but freed from terminology and assumptions that limit the applications of this formalism to a particular scientific discipline. Such a mathematical formalism should eventually lead to improvements

in computer- assisted organic synthesis.

Chemical pathways are diverse. They may consist of a single reaction or a complex biochemical pathway. Figure 1 presents part of the "shikimic acid" metabolic pathway (Bu'lock ,1965). There we see cpd.2 was hydrorxylated to give cpd.3. Similarly, cpd.4 was hydroxylated to give cpd.5 and cpd.5 was further hydroxylated to give cpd.6. In our analysis, "hydroxylation", which performs a particular structural change, will correspond to the graph-theoretic concepts of a transform. In hydroxylation, a hydrogen atom is deleted and the atoms and bonds of a hydroxyl group are added. The graph elements (vertices or edges corresponding to the atoms and bonds) to be deleted by a transform will subsequently be indicated by assigning each such element $a\ e\ 1$, and the graph elements to be added will be indicated by assigning each such element $a + 1$. Note the asymmetry here. The reverse change, deleting a hydroxyl group and adding a hydrogen atom is called dehydroxylation. This asymmetry must be incorporated into our concepts of a transform. In addition, we see that a transform acts on some compounds and not on others. In Figure 1, compounds 2, 4 and 5 were hydroxylated but compounds 1, 3 and 6 were not. Those graph elements used to define the local environment where a structural change will take place, but which are neither to be added nor deleted, will subsequently be indicated by assigning each such element a 0. Finally, Figure 1 indicates that compound 2 is "deanimated" to give compound 5. The "deanimation" transform perform a different structural change than the "hydroxylation" transform. Thus, a solution to the problem of modeling chemical reaction pathways will require a specification of a broad class of possible transforms and a set of rules for deciding which transform is to operate where on each graph.

Figure 2 presents a graph theoretic counterpart of a chemical reaction pathway. This figure should be viewed as involving two diagraphs M_1 and M_2 whose vertices are themselves graphs. We shall refer to such diagraphs as metadiagraphs. The arcs of the metadigraphs are the graph-theoretic counterparts of chemical reactions. Those operations which convert one graph into another, the counterparts of "hydroxylation" and "deanimation", will be called transforms. A transform kit K will consist of a set T^i of inducing transforms and a set T^b of "blocking" transforms. Rules are given whereby an arc of a metadigraph can be considered to be an action of an "inducing" transform in T^i not blocked by a "blocking" transform in T^b. It will be seen that metadigraph M_1 in Figure 2 can be simply specified by an ordered pair $(\{P_1\})$ where K consists of a single inducing transform and a single blocking transform. It will also be shown that any metadiggraph M admits a specification of the form $(V(M), K)$ for some transform kit K where $V(M)$ denotes the set of graphs that comprise the vertex set of M.

Semitransforms and Semiactions

We now see rigorously the notions given in the introduction following the notation

in Chartrand and Lesniak [4]. A graph $C = (V, E)$ together with an assignment of the integers -1, 0, +1 to the elements (vertices and edges) of C, one integer per element, will be called a semitransform on C if zero edges (edges assigned zeros) are adjacent to only zero vertices, nonpositive (*de* 0) edges are adjacent only to nonpositive vertices, and n ve vertices.

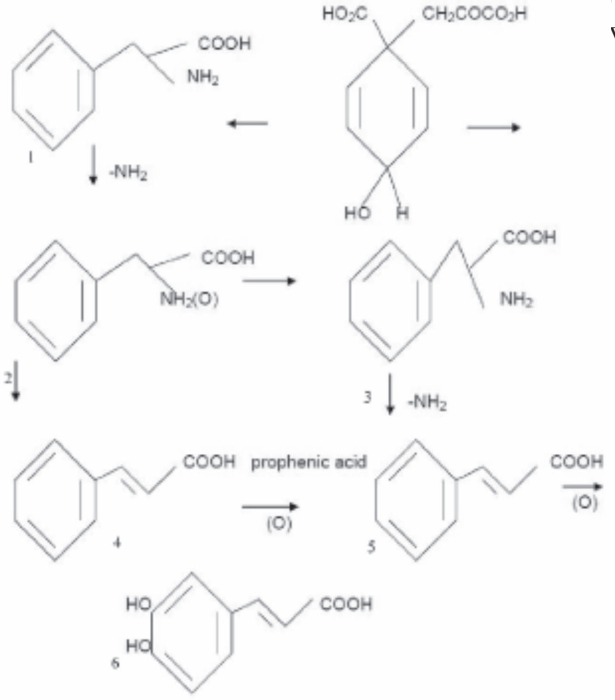

Figure 1: Part of the shikimic acid pathway

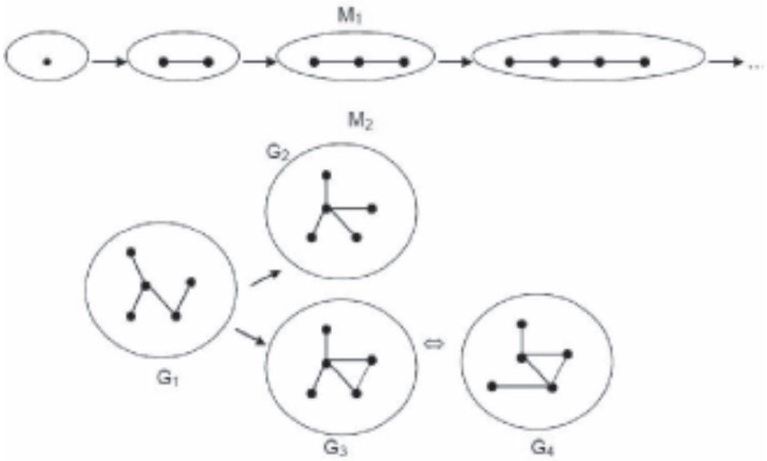

Figure 2: Two Metadigraphs

A semitransform will be illustrated by indicating negative, zero and positive edges with dashed, solid and dotted lines, and indicating negatives, zero and positive vertices with x's, solid dots and circles, respectively, Figure-3 shows three

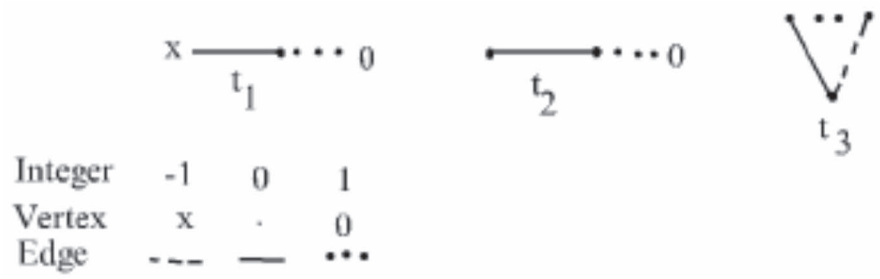

Figure 3: Some Semitransforms

semitransforms.

Let *t* be a semitransform on *C*. Four graphs will be commonly associated with *t*. Graph *C* is the covering of *t*. The nonpositive elements of *t* define the ingraph *I* of *t*; the nonnegative elements define the outgraph *O* of *t*; and the zero elements define the linking graph *L* of *t*. The restrictions on the assignment function assure that *I*, *O* and *L* are well-defined. In Figure 3, the covering graph of t_3 is the cycle C_3, the ingraph and outgraph are both P_3 paths. The linking graph is the union of K_1 and P_2. The graphs *C, I, O* and *L* will be called the defining graphs of *t*. These letters, with appropriate superscripts and subscripts, will denote the defining graphs of a semitransform with corresponding superscripts and subscripts. Clearly, if any three of the defining graphs of a semitransform, when viewed as labeled subgraphs of *C* are known or if the ingraph and outgraph are known, all of the defining-graphs are completely determined. Defining the cardinality $|G|$ of the graph $G = (V, E)$ by $|G| = |V| + |E|$. We obviously have $|I| + |O| = |C| + |L|$ (1) ; $p(I) + p(O) = p(C) + p(L)$ (2); and $q(I) + q(O) = q(C) + q(L)$ (3); where $p(G)$ and $q(G)$ denote the order and size of graph *G*.

There are many ways to form a semitransform having *G* for its ingraph and G^2 for it's outgraph. To see this, let *C* be any common supergraph of *G* and *G'*. Let *I* and *O* denote labeled subgraphs of *C* that are isomorphic to *G* and *G'*, respectively. Then the semitransform having *I* and *O* as the ingraph and outgraph is such a transform. Note that *C* will be the covering graph of that transform only if *I* and *O* cover *C*, i.e., every element of *C* is an element of either *I* or *O*. Let *t* be a semitransform on $C = (V, E)$ with assignment function $g: V^{*"} E' ! \{ "1, 0, 1\}$ and let *t'* be a semitranform on $C' = (V', E')$ with assignment function $h: V'^{*"} E' ' ! \{ "1, 0, 1\}$. Then *t* and *t'* are isomorphic and we write $t H" t'$, if there exists an isomorphism $f : V ' ! V'$ between *C* and *C'* such that $g(v) = h(f(v))$ for all $v " V$ and $g(uw) = h(f(u)f(w))$ for all $uw " E$.

Let *t* be a semitransform on *C* with assignment function *g*. Let *C'* be a labeled subgraph of *C* and let *g'* be the restriction of *g* to the elements of *C'*. The graph *C'* together with g^2 is a semitransform *t'* called a subtransform of *t*. If in addition, *t'* has as many nonzero elements as *t*, then *t'* is a reduction of *t*. If *t*" is any other semitransform isomorphic to *t'*, then *t*" is also called a subtransform (or reduction if such is the case) of *t*. The semitransform t_2 is reduction of the semitransform t_3 in Figure 3. If *t'* is a reduction of *t*, then *t* is an extension of *t'*. Clearly, if *t'* is a reduction of *t*, then |*X*| " |*Y* | = |*X'*| " |*Y'*| (4); *p*(*X*) " *p*(*Y*) = *p*(*X'*) " *p*(*Y'*) (5); *q*(*X*) " *q*(*Y*) = *q*(*X'*) " *q*(*Y'*) (6); Where *X* and *Y* denote any pair of defining graphs of *t* and *X'* and *Y'* denote the corresponding pair of defining graphs of *t'*.

Let (*G, G'*) be an ordered pair of graphs, and let *t* be a semitransform. We shall say (*G, G'*) is a semiaction *t* if there exists an extension *t'* of *t* such that *G* and *G'* are isomorphic to the ingraph and outgraph of *t'*. The extension *t'* is called a semioverlay of (*G, G'*) with respect to *t*. Clearly, every semitransform *t* is a semioverlay of (*I, O*) with respect to *t* itself where *I* and *O* denote the ingraph and outgraph of *t*. Let (*G, G'*) be any semiaction of *t* having ingraph *I* outgraph *O*. It follows directly from equations (5) and (6) that *p*(*G'*) = *p*(*G*)+*p*(*O*)"*p*(*I*) (7); *q*(*G'*) = *q*(*G*)+*q*(*O*)"*q*(*I*) (8); |*G'*| = |*G*|+|*O*|"|*I*| (9);

Example 1

Let (*G, G'*) be any semiaction of t_1 in Figure 3. Let *t* be any overlay of (*G, G'*) with respect to t_1 and let *C, I, O, L* denote the defining graphs of *t*. Then the linking graph *L* of *t* contains all the vertices and edges in *I* and *O* except for one edge, say *uy*, of *I* where *u* " *V* (*L*) and *y* ∉ *V* (*L*) and one edge of *O* adjacent to *u*, say *uz* where *z V* (*L*). Define the function *f: V* (*I*) *'! V* (*O*) by *f* (*v*) = *v* if *v* = *y* and *f*(*v*) = *z* if *v* = *y*. Clearly, *f* defines an isomorphism between *I* and *O*. Thus if (*G,G2*) is a semiaction of *t*, then *GE" G2 :* It might be pointed out that equations (7) and (8) imply that *G* and *G2* have identical orders and sizes since *I* and *O* have identical orders and sizes.

A graph *G* is transformable into *G'* by an edge rotation if *G* contains distinct vertices *u, v, w* such that *uv* e *E*(*G*)*; uw* *E*(*G*) and G^2 *He Geuv+uw* (5). An edge rotation is

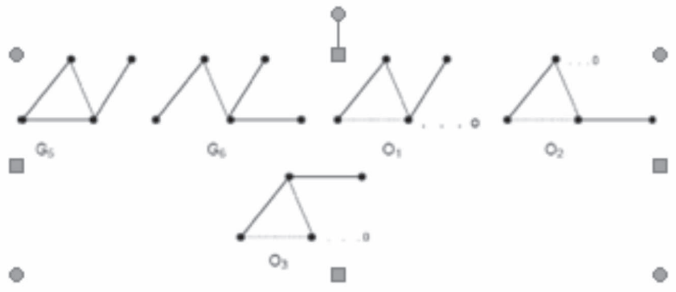

Figure 4. Various overlays of (G_5, G_6)

an edge shift, if in addition, *vw e E* (*G*) (9). It is a straightforward matter to show that *G* is transformable into *G'* by an edge rotation (edge shift) if and only if (*G, G'*) is a semiaction of t_2 (t_3) in Figure 4.

Transforms and Actions

Let *G* and *G'* be two graphs. A maximum common subgraph of *G* and *G'* is a graph *H* of maximum cardinality which is isomorphic to a subgraph of both *G* and *G'*. A minimum common supergraph of *G* and *G'* is a graph *H* of minimum cardinality which is isomorphic to a supergraph of *G* and *G'*. (See [8] for other uses of maximum common subgraphs and minimum common supergraphs, referred to there as maximum intersections and minimum unions, for comparing graphs of arbitrary order and size). A semi overlay *t'* of (*G, G'*) with respect to a semi transform *t* is an overlay of (*G, G'*) with respect to *t* if the linking graph of *t'* is a maximum common subgraph of *G* and *G'*. It follows directly from equation 1 that the linking graph of *t'* is a maximum common subgraph of *G* and *G'* if and only if the covering graph of *t'* is a minimum common supergraph of *G* and *G'*.

Figure 4 gives three overlays of graphs G_5 and G_6. Note that overlays O_1 and O_2 have isomorphic linking graphs but nonisomorphic covering graphs while overlays O_2 and O_3 have nonisomorphic linking graphs.

A semiaction (*G; G*^{2}) of a semitransform *t* is an action of *t* if *t* is a reduction of an overlay of (*G, G'*). A semitransform *t* is a transform if there exists an action of *t*. The following theorem establishes a consistency relationship between actions and overlays.

Theorem* 1:* A semioverlay of (*G, G'*) with respect to *t* is an overlay with respect to *t* if and only if (*G, G'*) is an action of t.

Proof*:* By definition, if *t' is* an overlay of (*G, G'*) with respect to *t*, then (*G, G'*) is an action of *t*. To establish the only if relation, assume (*G, G'*) is an action of *t* and *t'* is a semioverlay of (*G, G'*) with respect to *t*. By definition, there exists an overlay *t"* of (*G, G'*) with respect to *t*. Using equation (4) with *X* = *I* and *Y* = *L*, we have |*G*| = |*I'* | = |*L'*|+|*I'*|*e*|*L'* | = |*L'*|+ |*I*|*e*|*L*|; (10); and |*G*| = |*I"*| = |*L"*| + |*I"*| *e* |*L"*| = |*L'*| + |*I*| *e* |*L*|; (11). Equating the right hand sides of equations (10) and (11), we have |*L'*| = |*L"*|. Since *L'* is a subgraph of both *G* and *G'* of cardinality |*L*|, *L'* must be a maximum common subgraph.e

Not all semitransforms are transforms. In particular, the semitransforms given by t_1, of Figure 3 are not. To see this, suppose (*G, G'*) were an action of t_1. By definition, there exists an extension *t* of g_1 which is an overlay of (*G, G'*). Let *C, I, O, L* be the defining graphs of *t*. Then *I Ee G* and *G2 Ee O*. However, we showed in Example 1 that *G Ee G2* . Since *t* is an overlay, *L* is a maximum common subgraph of *I* and *O*, and consequently, *L* = *I* = *O*, i.e., *t* contains only zero elements. Since t_1 is a reduction of *t*, t_1 contains only zero elements, a contradiction. Thus t_1 cannot be a transform.

Let *ST, SO, T* and *O* denote the set of semitransforms, semioverlays, transforms and overlays. When defining semioverlays, it was shown that *ST = SO*. It has just been shown that *T* is a proper subset of *ST*. To see that *O* is s proper subset of *T*, note that the linking graph of t_2 in Figure 3 is the complement of K_3 while the maximum common subgraph of the ingraph and outgraph of t_2 is P_2 *e K_1. Thus, t_2 is not an overlay. To show that t_2 and t_3 of Figure 3 are transforms, let *GEe K* (1, 3) and *G2 Ee* P_4. Let *t* be defined by t_4 of Figure 5. Let *C, I, O* and *L* be the defining graphs of *t*. Since *G Ee I* and *G2 Ee O*, *t* is a semioverlay of (*G, G'*). Clearly *L Ee* P_3 *e K_1 is a maximum common subgraph of *G* and *G2* . Thus, *t* is an overlay. Now we simply

Figure 5: An overlay and a semioverlay with respect to t_3

note that both t_2 and t_3 are reductions of *t*.

It should also be noted that not all semiactions of a transform are actions of that transform. To see this, let *t'* be defined by t_5 of Figure 5. Clearly, *t'* is a semioverlay of (P_4, *P4*). Since t_2 and t_3 of Figure 3 are reductions of *t'*, (P_4, P_4) is a semiaction of both t_2 and t_3. However, the linking graph of *t'* is P_2 *" P_2; which is not a maximum common subgraph of P_4 and P_4. Thus, *t'* is not overlay of t_3, and consequently not of t_2. If follows from Theorem 1 that (P_4, P_4) is not an action of either t_2 or t_3.

Let *S*(*t*) and *A*(*t*) denote the sets of semiactions and actions of *t*. It would be nice to be able to prove that *A*(*t*) = *A*(*t2*) implies *t E" t2* .

Specification of Metadigraphs

A metadigraph *M* is a possibly infinite, diagraph *D* = (*V, E*) together with an injective function *g*: *V '!* that assigns a unique graph in 0 to each vertex of *D*. Thus a metadigraph can be considered to be a labeled digraph in which the set of labels is a set of unlabeled graphs. It follows that we can uniquely and unambiguously denote *M* by (E^g) where is the image set *g*(*V*) of *V* and E^g = {*g*(*u*)*g*(*v*) : *uv "E}* i.e., we can uniquely represent *M* by (E^g). Examples of two metadigraphs were given in Figure 2. Let *M'* be another metadigraph *D'* = (*V', E'*) with assignment function *h*, and let (E^h) denote its corresponding labeled graph representation. *ME"M'* if there exists

an isomorphism *f: V '! V'* between *D* and *D'* such that $g(v) = h(f(v))$ for *v " V*. Clearly *ME"M'* if and only if their corresponding labeled graph representations (E^g) and (E^h) are identical ie., and $E^g = E^h$. Thus we shall say *M* is a submetadigraph of *M'* if and E^g E^h.

An ordered pair (T^i; T^b) of possibly infinite, sets of transforms is a transform kit. An action (*G, G'*) of *t* is blocked by *t'* if (a) *t'* is an extension of *t*, (b) (*G, G'*) is an action of *t'* and (c) *t'" T*^{b}. Let *A*(*t*|T^b) denote the actions of *t* that are not blocked by *t'* for any *t'" T*^{b}.

Clearly, *A*(*t*|T^b) = *A*(*t*)", where the union is over those *t'* in T^b which are extensions of *t*. An ordered pair (*G, G'*) of graphs is an action of *k* = ($T,^i$ T^b) if (*G, G2*) " *A*(*t*|T^b) for some *t " T*^{i}. A graph *G2* will be said to be *K*-reachable from *G* if either *G'E" G* or there exists a sequence *G*1, G_2, *Gn* such that *GE"G*1, *G'E" Gn* and (G_i, *Gi*+1) is an action of *K* for *i* = 1, 2,…, *n"*1. Let be a set of graphs and let *K* be a transform kit. Define the metadigraph *M* (*K*) as the metadigraph (*E*) where is the set of graphs which are *K*-reachable from a graph in and where *E* is a subset of whose members are actions of *K*. Given a metadigraph *M*, we shall say that (*K*) is a specification of *M* if *M* (*K*) = *M*.

Example 2

Let t_4 and t_5 be defined by Figure 6. Let *K* = ({t_4}, {t_5}) and let *M* = (*E*) denote the metadigraph M_1 in Figure2, ie., = {P_n : *n* = 1,…} and *E* = {(P_i, P_{i+1}) : *i* = 1,…}. Then ({*P* 1}, *K*) is a specification of *M*. To see this, we simply note that for every graph *G* and every vertex *u " V* (*G*), (*G, G + u + uv*) is an action of t_4 where *u V* (*G*). Similarly, (*G, G + u + uv)* is an action of *t* if and only if the *deg* (v) > 1. It follows that if (*G*, G^2*)* is an action of *K*, then *GE"*^{2} *G + u + uv* where *deg* (*v*) < 2. Thus if *G* is a path P_i, then *G2 E"* P_i + 1. Since every path is *K*-reachable from P_1, it follows that

({P_1}, *K*) is a specification of *M*.

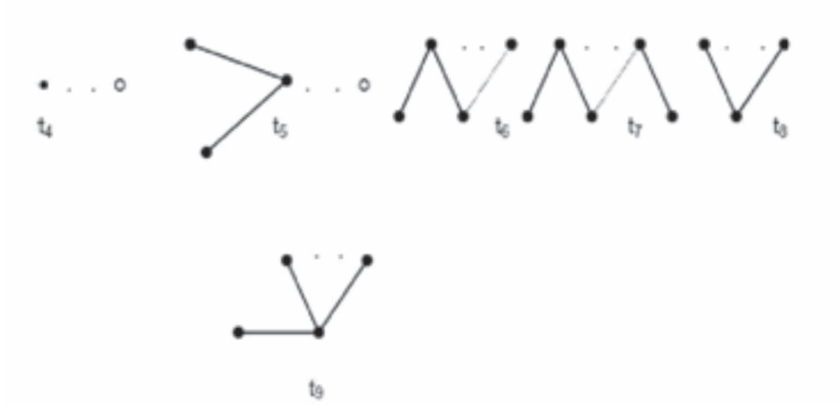

Figure 6: Transforms

Example 3

Let M be metadigraph M_2 in Figure 2, and let t_i i=6,..., 9 be transforms defined by Figure 6. Define $T^i = \{t_6, t_8\}$ and $T^b = \{t_7; t_9\}$ and let K be the transform kit $(T^i; T^b)$. Then $(\{G_1\}, K)$ is a specification of M. To see this, note that t_6 performs an edge shift of the form G " $uv + uw$ where v is adjacent in G to a vertex w with deg $(w) > 1$, while t_7 blocks those cases in which u is not a terminal vertex. Likewise, t_8 adds an edge between vertices u and v which are not adjacent, but which are both adjacent to another vertex w, while t_9 blocks those cases in which deg $(w) = 2$.

We now prove as a corollary to the following theorem that 'any metadigraph admits a specification for some K. First we prove a lemma.

Lemma* 1:* If t is a reduction of a semitransform $t2$ for which $IE'' I2$ and $OE'' O2$, then $tE'' t2$.

Proof*:* To see that the reduction condition is essential, let t be defined by transform t_5 in Figure 5, and let $t2$ be the transform for which $C2\ E'' P_4 E'' L'$. Since $LE'' P_2 *'' P_2$, t and $t2$ are not isomorphic. On the other hand, $IE'' OE'' P_4 E'' I2\ E'' O2$. Thus, the lemma fails without the reduction condition.

By definition, t is isomorphic to a reduction t" of $t2$ for which I" and O" are labeled subgraphs of $I2$ and $O2$. since I"E" $IE'' I2$ and O"E" $OE'' O2$, it follows that I" = $I2$ and O" = $O2$. Thus t" = $t2$. Since $tE'' t$", transitivity implies $tE'' t2$. Let T_o denote the set of all transforms. Theorem 2 and its corollary show that T_o can be used to prove that all metadigraphs admit a specification with respect to some transform kit."

Theorem* 2:* Let $K = (T^i, T^b)$ be the transform kit based on any partitioning $\{T^i, T^b\}$ of T_o such that every member of T^i is an overlay. Then (G, G^2) is an action of K if and only if there exists an overlay of (G, G^2) in T^i.

Proof*:* Assume that there exists an overlay t and T^i of $(G, G2)$, i.e., $IE'' G$ and $OE'' G2$. If $(G, G2)$ is not an action of K, then the action $(G, G2)$ must be blocked by some t^i " T^b. This implies that I is a subgraph of $I2$ and O is a subgraph of $O2$. By definition, $(G, G2)$ must be an action of $t2$ i.e., $t2$ is a reduction of an overlay of $(G, G2)$. This implies that $I2$ is a subgraph of G and that $O2$ is a subgraph of $G2$. It follows that $IE'' I2$ and $OE'' O2$. By Lemma, $tE'' t2$, a contradiction since T^i and T^b are disjoint sets.

On the other hand, assume (G, G^2) is an action of K. Then (G, G^2) " $A(t|T^b)$ for some t " T^i. Let t^2 be any extension of t such that (G, G^2) is an action of t^2. If t^2 " T^b, then (G, G^2) A $(t|T^b)$. Thus, all extensions of t having (G, G^2) as an action must be in T^2. But (G, G^2) " $A(t|T^b)$ implies the existence of an overlay t^2 of (G, G^2) with respect to t. As an extension of t having (G, G^2) as an action, t^2 must lie in T^i, i.e., there

exists an overlay of (G, G^2) " T^i. "

Corollary* 1:* For any metadigraph M, there exists a transform kit K such that (V (M), K) is a specification of M.

Proof*:* Write M = (E). For every edge $GG2$ " E, form an overlay of (G, $G2$) as noted in the section on semitransforms by letting the covering graph be a minimum common supergraph of G and G'. Let T^i denote the set of overlays so formed, and let K = (T^i; T_o "T^i). By Theorem 2, GG^2 is an action of K if and only if GG^2 is an edge of M. Since M and M (K) have identical vertex set, $M = M$ (K). As there generally exists many overlays of a pair (G, G^2) of graphs, there will generally be many kits K that satisfy the conditions of the preceding proof."

Also see (for more) [5], [10], [11], [12], [13], [14].

Graph Theoretical Interpretation of Chemical Reaction Networks

(Heckel ,2006)gives an excellent overview of basic graph transformation theory. It consists of an easy to-follow description of graph transformation systems including the representation of systems as graphs (type and instance graphs), rules and transformations, constraints and application conditions. To explain its most basic functionality, a rule first looks for a pattern match for the left side of the rule in the input graph. Then, edges and nodes that are not in the right hand side of the rule are deleted, while edges and nodes that are newly created in the right hand side are placed into the graph. This paper will be invaluable as a reference to basic concepts.

Modeling molecules and reaction networks using graphs is described as a very natural application by much of the literature (Benko *et al* , 2003, Clark,2004 and Rossello *et al* ,2005)discuss how a form of graphs is already a fundamental part of organic chemistry, when depicting molecules by their structural formula. Atoms can naturally be seen as nodes in a graph, with the bonds between them represented as bi-directional edges. Double bonds and triple bonds could be modeled as two edges and three edges respectively (as represented in Benko *et al* ,2003)) but this would only be useful if a reaction involved the breaking of one of the double bonds in one or more of its elementary steps (this may be useful for the esterification case study).

The problem with this basic approach, however, is a loss of information about spatial configurations i.e. cis and trans isomers and chirality. Furthermore, the valencies of individual atoms (the no. of other atoms it can bond to) are not automatically conserved(Ehrig *et al* ,2006) offers an alternative approach that should be much more intuitive for chemists. By modeling the atoms as hyperedges and the bonds between them as nodes, valency and chirality can both be incorporated into the model. Each hyperedge is typed by atom, allowing for different numbers of joined nodes, therefore the concept of valency is preserved. The outgoing bondnodes from a hyperedge are labeled in order to show the three-dimensional ordering (related

to D-glyceraldehyde), and preserving chirality. This is particularly important for reactions where one enantiomer is more reactive in an elementary step, or where only one enantiomer binds with a particular substrate. This is more prominent in biochemical systems. The example of Citrate binding with Aconitase is given in (Ehrig *et al* 2006).

From the representation of molecules as graphs, envisioning reactions between molecules as graph transformations is not a big step. Both (Rossello *et al*, 2005 and Yadav *et al* 2004) describe how reactions involve the breaking and creation of bonds in a molecule to transform it into a different molecule. In the graph, this would mean an elementary reaction step would involve the deletion of edges (bonds) and nodes not appearing in the right hand side of the rule, and the creation of edges and nodes that appear on the right hand side of the rule. As nodes and edges can have attributes, attributes may also be updated during the transformation. Such attributes could indicate energies of bonds, valencies or other useful information.

Another reason why graph transformation representations are so natural is their "inherent concurrency" (Ehrig *et al* ,2006). This is the ability for these systems to allow simultaneous reactions of different reactants, which is obviously what happens in real chemistry. Causal dependencies can be monitored along with conflicts, using critical pair analysis. (Rossello *et al* ,2005) attributes the strength of graph transformations in this field to their "pattern handling power". In any chemical reaction network, involving an arbitrarily large number of molecules, a multitude of simultaneous reactions are possible. However, graph transformation rules define the standard reactivity of certain functional groups.

Pattern matching provides a powerful search method for where these rules can be applied within the network. This will be explored in both our case studies and a more thorough explanation is giving in later sections of this report. (Ehrig *et al* ,2006) explains how graph transformations work in detail, by summarizing the so-called double pushout approach. The left side and right side of the rule span show how the molecule will change, whereas the gluing graph between the left and right hand side simply indicates which elements are involved ("read") in the rule, but not consumed. In the "toy" model of artificial chemistries given in (Benko *et al* ,2003) the essential idea is the same, but the gluing graph is labeled the context. The visualization is also much more attuned to the structural formula representation used by chemists - atomic/group nodes are just labeled by element symbols and the edges replaced by standard bond representations (with double and triple bonds preserved). Although this may be more intuitive for chemists, it does not allow us to easily show attributes of nodes or edges, and also may seem alien to the way the graphs will look in their eventual implementation using tools such as AGG. Using standard graph representations should be easy enough to understand so the further simplifications in the toy model are not necessary.

(Ehrig 2006) suggests 3 types of rules in any graph transformation system. Symmetry rules are useful for the hyperedge model. Although chiral molecules cannot rearrange their spatial configuration with respect to the other groups around the chiral carbon centre, the bond connected to this carbon can itself rotate, meaning the groups can change position. So that this is not considered a separate molecule, equivalence rules can be set up. Expansion rules allow us to expand atoms grouped together in one node/hyperedge (usually for simpler representation) into their full expanded graph. This is necessary if at some point in the graph the details of the group become important e.g. one of the atoms is involved in a reaction. Again, this contracted representation is familiar for chemists, who often contract large groups in structural formula when their exact spatial details are not contextually important. Reaction rules define the change of groups. Our model should definitely incorporate the second and third types of rules, and consider the symmetry rules if the chirality preserving hyperedge model is used.(Ehrig *et al*,2006)Ehprovides some other invaluable information about how graph transformation theory can be applied to reactions, and how atoms can be traced throughout the reaction and will serve as good reference material.

Stochastic graph transformation systems and rates of reaction

Heekel ,2005 and Heckel *et al*,2004 are excellent references explaining the extension of a graph transformation system to a stochastic graph transformation system. Given a start graph and a graph transformation system, a labeled transition system can be deduced which shows all the possible states (graphs) possible. Applied to a reaction network, each state would represent all the species (reactants, products, by-products and intermediates) that would result from applying the graph transformation rules. Labels from state to state can be assigned attributes such as a probability of reaction. This is analogous to the rate of the transformation from one step to the next, hence essential to our derivation of the overall rate of reaction.(Heckel ,2005) provides all the necessary definitions and procedures needed to convert a stochastic graph transformation system to a Continuous Time Markov Chain, which then allows the application of stochastic temporal logic to deduce long-term non-functional stochastic properties of the system. Equivalence rules can be set up which check for certain properties of the system at any point during the progress of the simulation, thereby allowing us to monitor the presence of molecules for example. Querying the system at regular intervals can then allow us to effectively measure properties proportional to the concentration of chemical species.

Paper (Heckel, 2005)demonstrates the application of stochastic graph transformations to P2P networks. Although the model is simpler (in that the rates are more easily assigned) the example will be invaluable in understanding how stochastic systems are applied. As the theory behind stochastic graph transformations (Continuous Time Markov Chains, Qincidence matrices, Stochastic Temporal Logic) can be quite difficult to grasp, both of these papers will be useful to return to. In

deciding the rates of elementary steps, paper (Benko *et al* ,2003) provides a very good starting point.

As discussed earlier, the Arrhenius equation relates the change in energy between reactants and products with a rate of reaction. (Benko *et al* ,2003) discusses a refinement where instead of entire molecules, just the energy of hybridized orbitals involved in the reaction are considered. For complex reactions such as case study 2 we can follow the methodology described in (Benko *et al* ,2003) and automate the procedure of assigning energies by looking at recurring sections of molecules (characterized by different hybridized orbitals) rather than entire molecules. This generalization step would avoid having to manually determine and assign energies for the high number of possible species involved in the reaction (i.e. chains of many different lengths). It can incorporate more complicated electronic distributions accurately into the model, such as -stabilization where adjacent -bonding orbital can overlap in a molecule, lowering their overall energy and making them more resistant to breaking and therefore reaction. (Benko *et al* ,2003) also describes how this method accounts for regioselectivity within a molecule. i.e. if there are two places where a rule can be applied, calculating energies would force the rate of reaction at one site to be realistically higher than the other. (Benko *et al* ,2003) actually goes on to suggest that the calculation of energies could be used for stochastic simulations of reaction networks using the Gillespie algorithm.

There are other methods in the literature that are used to model stochastic systems. (Cardelli ,2008)and (Keeler ,1999) both use process languages to specify the reaction rules and to derive useful properties. Cardellis paper (Cardelli ,2008) is particularly useful as it sets out to achieve what this project does using stochastic process algebra, namely CCS and CGF. While this approach is probably more intuitive and precise for computer scientists, it loses its appeal for chemists due to its technical complexity. The visual graph representation is much more useful in this respect because the components are easily recognizable to chemists. The content of the paper is quite technical and without a background in logic and automata course requires substantial background reading to understand fully. The paper is therefore currently of limited use. Nevertheless, it would be highly advisable to understand Cardellis approach, in order to note his assumptions and the way in which he assigns rates to elementary reaction steps. For this reason, a quick overview of -calculus (provided by Milner 2007)) and subsequent research into CCS still needs to be undertaken.

Implementation tools

AGG is widely used by the graph transformation academic community. A brief description of its utilization to a chemical reaction setting is given in (Ehrig *et al* ,2006). AGG does have several limitations for our purposes though. Firstly, it cannot be used for stochastic simulations. To obtain stochastic data, a combination of PRISM and GROOVE (as described in (Heckel 2005) and Heckel *et al* 2004)) could be used. (Erhard *et al* ,2008)also provides a very thorough presentation of FERN, a

Java framework that can be used for stochastic simulation. The API seems fairly straightforward. It does not however provide its own visualization module. AGG also has its own Java API and a combination of the two tools could be coordinated to fit the needs.

Further investigation of (Erhard *et al*) will be necessary to achieve this. An appealing feature of (Erhard *et al*,2008) is that it provides implementations of several stochastic simulation algorithms, namely the Gillespie algorithm, extended Gillespie algorithm and a tau-leaping algorithm. The framework applies the most appropriate algorithm depending on the speed of the reaction, and can even change dynamically during runtime. Other limitations of the software should also be considered before using AGG such as minimum requirements, known bugs and elements of graph transformation theory that are not implemented (such as the ability to input hyperedges). The user manuals, bug reports and examples which can be found at should be reviewed before/while using the tool.

Consideration for more complete modeling

Despite being an overview paper that has little relevant technical content, Yadav *et al* ,2004)does illuminate some interesting points to consider if we are to progress to a more complete model of real chemistries. In particular, "Global Context Sensitivity" is discussed. This states that physical properties play an important role in chemical reactions, such as temperature, solvent, viscosity, catalysts and radiation to name just a few. A graph transformation model may be limited in its ability to incorporate such factors. This should be taken into account in the final stages of the project. "Local Context Sensitivity" considers for example the "three dimensional conformation of reactive groups" of a molecule and how this affects reactivity. Large groups for example may block collision with incoming molecules, thereby hindering reaction rates.

The graph structure of the chemical reaction network is restated explicitly in terms of graph theory, precisely hypergraph theory. For example what is the graphical interpretation of the null space of the stochiometric coefficient, and what about its orthogonal complement? To that end we review the theory of flows on graphs. We then consider the possibility of such a theory for flows on hypergraphs. A few formal results are obtained. We also pose a number of problems for future consideration.

Hypergraph Basics and the Model

A weighted hypergraph is an ordered triple *H* (*V, E, 5Ø—ß*) (or just *H*). *V* or *V* (*H*)

H^{σ} admits the incidence matrix representation $S \in M_{m,n}(\mathbf{Z})$:

$$S_{ij} = \begin{cases} w_{ij}, & \text{if } v_i \in h_j \ (i = 1,\ldots,n;\ j = 1,\ldots,m). \\ -w_{ij}, & \text{if } v_i \in t_j \end{cases}$$

.....(A)

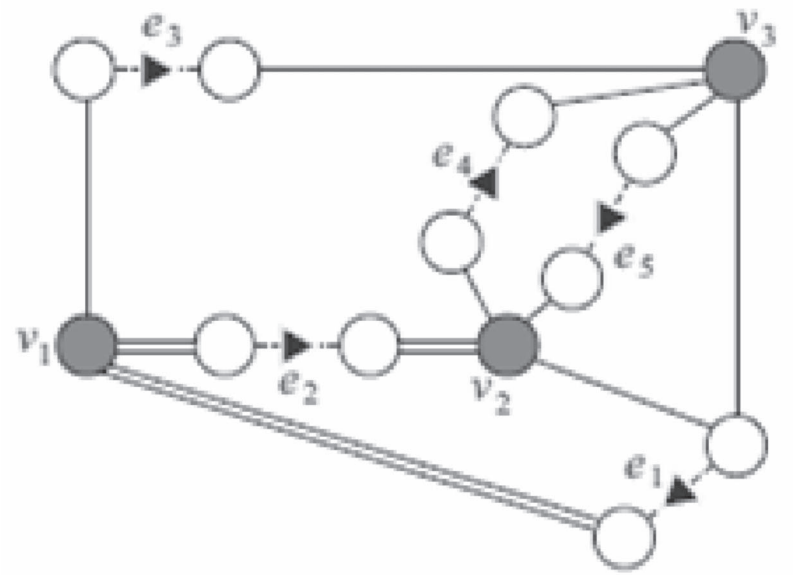

Figure 7: A directed weighted hypergraph, represented according to the Sinanoglu formalism. Weights appear in the figure as multiple connections.

is a set of vertices, and | V |= m. E or $E(H)$ is a set of edges, subsets of V each of which contains at least two vertices, and | E |= m; $w : V \times E$ '! N^+ is a function which assigns a weight w_{ij} to each v_i *e* e_j . If v_i e_j, then w_{ij} is by default zero. Note that H contains no self loops. A regular hypergraph is just a weighted hypergraph with all weights set to zero or one. Note that a graph is a hypergraph G such that *e*e_j *e* E, | e_j |= 2.

A directed weighted hypergraph is a weighted hypergraph H together with an orientation ó, and is denoted $H : : E$ '! $H \times T$ is a function which partitions every e_j " E into a head set h_j and a tail set t_j .

$H^{5\emptyset ß}$ admits the incidence matrix representation S"$M_{m,n}$(**Z**):

$$\text{Figure 7 is } \begin{pmatrix} 2 & -2 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 & 1 \\ -1 & 0 & 1 & 1 & -1 \end{pmatrix}$$

Of course, H admits an incidence matrix representation given by | S |. An example of a directed weighted hypergraph appears in Figure 7. Precisely, the figure is a Sinanoglu representation of a hypergraph (this formalism provides the clearest graphical depiction of the type of hypergraphs presently under consideration). According to this formalism, a directed edge is represented by a pair of white nodes

connected by a dashed line. The error along the dashed line points from the 'tail set node' of the edge to the 'head set node' of the edge. Of course, when a vertex is connected to a tail or head set node, the vertex is said to be in that set. The weight associated with that vertex of the edge can either appear as a numerical label on its connection or graphically as multiple connections (the second formalism is used in Figure 7). The incidence matrix associated with Figure 7 is

We relate the chemical reaction network and the directed weighted hypergraph by making the associations between vertices and chemical species, edges and reactions, head and tail sets and reaction products and reactants, respectively, and weights and stoichiometric coefficients. Under this association (in so far as the given definition of a hypergraph allows), catalysts are ignored. It is evident that the incidence matrix is precisely the stoichiometric matrix of the reaction network.

The theory of hypergraphs has not been elaborated in the great detail that is has for the theory of graphs. However, a few important references do exist. For a somewhat technical introduction to the field, see [15].

Cuts and Flows on Hypergraphs

The material of this subsection is adapted with modification from Chapter 14 in [(Godsil and Royle ,2001). It also draws upon (Cardelli 2008)

Let $\mathbf{R}^E$ denote the real vector space with coordinates indexed by the edges (precisely, their indices) of *H*. The row space of *S*, that is the subspace of $\mathbf{R}^E$ spanned by the rows of *S*, is known as the cut space of $H^{5\emptyset ß}$. And the orthogonal complement of the cut space is called the flow space of *H*. So the cut space is the set of all vectors *x* " $\mathbf{R}^E$ satisfying *Sx* = 0. By an abuse of notation we will simply refer to the cut and flow space of *H* with the understanding that is fixed. In the following, these terms

$$z_j \equiv \begin{cases} 0, & \text{if } e_j \notin C \\ \sum_{i:v_i \in e_j \cap V(+)} S_{ij}, & \text{if } e_j \in C \end{cases} \quad \text{.....(B)}$$

will be justified.

The cut space

If (*U, V*) is a partition of *V* (*H*) (the vertex set of *H*), into two nonempty subsets. The set of edges *e* " *E* (*H*) with *e)"U* `"Õ and *e )" V* `"Õ is a *cut*, denoted *C*. We shall call *U* and *V* the *shores* of a cut. A nonempty cut that is of minimal size is called a *bond*.

An *oriented* cut is a cut with one shore declared as positive *V* (+), and one shore declared as negative *V* (“). Using the orientation of *H*, that is using $H^{5\emptyset ß}$, an oriented cut *C* determines a vector *z* $e\mathbf{R}^{E}$ as follows:

We refer to *Z* as the *signed Characteristic vector* of the oriented cut *C*; we will have reason to introduce the more complete notation *z*(*C*). For a given oriented cut, reversing the direction on the direction of the edges of *H* changes the sign of *z*. However, it is not necessarily the case that the signed characteristic vector is invariant if all signs in Equ (B) are switched, that is, if we define the sign characteristic vector in terms of *V* (“) instead of *V* (+).

We call an edge e_j balanced if

$$\sum_{i:v_i \in e_j} w_{ij} = \sum_{i:v_i \notin e_j} w_{ij}.$$

We speak of a *balanced hypergraph* every edge *H* is balanced. The components of the sign characteristic vector associated with balanced edges are invariant with respect to the two definitions of the characteristic vector described above. So, in a balanced hypergraph the sign characteristic vector is invariant with respect to the two definitions.

Each vertex v_i determines an oriented cut $C(v_i)$ with positive shore $\{v_i\}$ and negative shore $V(H) \setminus v_i$. The i^{th} row of *S* is the signed characteristic vector of the cut $C(v_i)$, so these vectors lie in the cut space of *H*.

***Theorem* 3**: If *H* is a balanced hypergraph, then for every cut *C*

$$\text{for every cut } C\ z(C) = \sum_{v \in V(+)} z(C(v)) \qquad \ldots\ldots(C)$$

Proof: We consider the sum on the RHS of Eq. (C). Let *p a”*| *V* (+) | and *I* = (i_1,…, i_p) an index set for the elements of *V* (+). Then, by the definition of the signed characteristic vector

If $ej \subseteq V(-)$ (so $e_j \notin C$), then $S_{i,j} = 0$ (*k* = 1,…, *p*) and the prior sum evaluates to zero. Otherwise,

$$= \sum_{i:v_i \in e_j \cap V(+)}^{p} S_{ij}.$$

If e_j *V* (+) (so e_j *C*), then by the fact that *H* is balanced, the prior sum evaluates to

zero. Otherwise, e_j "*C*. The desired result follows from the definition of the signed characteristic vector."

$$\sum_{v \in V(+)} z_j(C(v)) = \sum_{k=1}^{p} S_{i,j}.$$

Corollary* 2:* If *H* is a balanced hypergraph, then the signed characteristic vector of each cut lies in the cut space of *H*.

I suspect that more can be said of the cut space. In particular, there should exist upper and lower bounds on the dimension of the cut space, if not a precise value, that can be determined from basic topological properties of the hypergraph. And, some minimal forest on a hypergraph should act as an alternative basis for the cut space. This basis would have the advantage that it more accurately describes "fundamental" transport structures of the hypergraph. These issues will be considered in the future work.

The flow space

The flow space of *H* is the orthogonal complement of the cut space, so consists of all vectors $x \; e \; R^E$ such that $Sx = 0$. We would like to extend the results from the theory of flows on graphs just we have done for cuts of graphs. However the manner by which to proceed becomes less clear than in the previous case. Whereas in the previous section we were able to extend the definition of a cut, in this case we must appropriately restrict the definition of a cycle. The precise definition of a cycle, which in its present form will seem like less of a restriction than it aught, is quite cumbersome and requires additional formalism. Lacking follow up results to justify such formalism, we refrain from said tedium.

Other Aspects in the Theory of Hypergraphs

We define the linearization of *H* to be the directed weighted graph (*H*) obtained by collapsing tail sets and headsets to single nodes of a graph. This construction then admits analysis according to the usual techniques of graph theory. It should be noted that this linearization resembles in some ways the technique of principle component analysis in graph theory.

A few obvious remarks can be made of this constructive. First, the time that it takes to perform the linearization ostensibly scales with the number of edges in the hypergraph. Second the dimension of the flow space of (*H*) provides an immediate upper bound on the dimension of the flow space of *H*. It would also be nice if there were a simple algebraic matrix operation that could immediately yield matrix of (*H*).

REFERENCES

Barone R. and M. Chanon., 1986. "Computer- Aided Organic Synthesis (CAOS), In Computer

Aids to Chemistry, Ed. by G.Vernin and M.Chanon, John Wiley and sons,19-102.
Beckett A.H. and D.A.Cowan., 1978. "Pitfalls in Drug -Metabolism Methodology", In Drug Metabolism in Man, Ed. by J.W Gorrod and A.H. Beckett, Taylor and Francis Ltd., London237-257.
Bu'lock., J.D. 1965. "the Biosynthesis of Natural Products: An Introduction to Secondary Metabolism", McGrawHill Pub. Co., New York.
Chartrand G. and L. Lesniak., 1986. "Graphs and Digraphs", Second Edition, Wads worth and Brooks, Monterey.
ChartrandG. , F.Saba and H.B.Zou., 1985. "Edge Rotations and Distance between Graphs, Casopis *Pro Pest.Mat* **110:**87-91.
Darvas, F. 1988. "Predicting Metabolic Pathways By Logic Programming" J.Mol.Graphics. **6** :80-86.
Diestel, R. 2005. Graph Theory (Springer-Verlag, 2005), electrical edition ed.
Fujita, S. 1986. "Description of Organic Reactions Based on Imaginary Transition Structures, 1. Introduction of New Concepts, "J.Chem.Inf.Comput. *Science* **26** 205-212.
Godsil, C. and G.Royle, Algebraic Graph Theory, vol. 207 of Graduate Texts in Mathematics (Springer-Verlag, 2001).
Johnson, M. 1985. "Relating Metrics, Lines and Variables Defined on Graphs to Problems in Medicinal Chemistry, "In Graph theory and its applications to algorithms and computer science, ed- by Y.Alavi, G.Chartrand, L.Lesniak, D.,R.Lick and C.E.Wall, John Wiley and Sons Inc.457-470.
Johnson, M. 1987. "An Ordering of some Metrics Defined on the space of Graphs",*Czech. Math. J.***37:**75-85.
Koca, J., M. Kratochril, V. Kvasnicka, L. Matyska and J.Pospichal., 1989. "synthon Model of Organic Chemistry an synthesis Design, " springer verlag, Berlin, 207p.
Ugi, I., J.Bauer, J.Brandt, J.Friedrich, J.Gast0eige, C.Jochum and W. Schubert., 1979. "New Applications of Computers in Chemistry", Angew.Chem.Int.Ed.Engl., **18:**111-123.
Varkony, T.H. Smith, D.H. and C.Djerassi, 1978. "Compter-ssisted Structure Manipulation: studies in the Biosynthesis of Natural Products", Tetrahedron, **34:** 841-852.
Berge, C. Hypergraphs, vol.45 (North-Holland Mathematical Library, 1989).
Benko, G., and Flamm, C. and Stadler, P.F., 2003. A Graph-Based Toy Model of Chemistry. *Journal of Chemical Information and Computer Sciences* **43:** 1085-1093.
Cardelli, L., 2008.From Processes to ODEs by Chemistry. IFIP International Federation for Information Processing, **273:** 261-281.
Clark, J., 2006. The Mechanism for the Esterification Reaction. http://www.chemguide.co.uk/physical/catalysis/esterify.html, hemguide, last accessed 08/04/09, last updated 2004.
Ehrig, H., Ehrig, K., Prange, U., Taentzer, G. Fundamentals of Algebraic Graph Transformation. EATCS Monographs, Springer.
Ehrig, K., Heckel, R., Lajios, G., 2006. Molecular Analysis of Metabolic Pathway with Graph Transformation. International Conference on Graph Transformations 2006, Lecture Notes in Computer Science **4178:** 107-121.
Erhard, F., Friedel, C.C., Zimmer, R., FERN - 2008. A Java framework for stochastic simulation and evaluation of reaction networks. BMC Bioinformatics **9:** 356-368.
Heckel, R., Embedding of Conditional Graph Transformations. Proc. 2005. Colloquium on Graph Transformation and its Application in Computer Science. Technical Report B-19, Universitat de les Illes Balears.
Heckel, R., 2006. Graph Transformation in a Nutshell. Electronic Notes in Theoretical Computer Science **148:**187-198.
Heckel, R., 2006. Molecular Pathway Analysis: From Graph Transformation to Stochastic Models. International Conference on Graph Transformations, presentation slides.

Heckel, R., 2005. Stochastic Analysis of Graph Transformation Systems: A Case Study in P2P Networks. International Colloquium on Theoretical Aspects of Computing 2005, *Lecture Notes in Computer Science* **3722:** 53-69.

Heckel, R., Lajios, G., Menge, S. 2004. Stochastic Graph Transformation Systems. International Conference on Graph Transformations 2004, Lecture Notes in Computer Science **3256:** 210-225.

Heckel, R., Engel, G., Graph 2000. Transformation and Visual Modeling Techniques. Workshop Summary and HowTo, Bulletin of the EATCS **72:** 69-76.

Keeler, J., 1A Chemistry 1999. Kinetics of Chemical Reactions. University Of Cambridge, Department of Chemistry.

Kwiatkowska, M., Norman, G., Parker, D., 2008. Using Probabilistic Model Checking in Systems Biology. ACM SIGMETRICS Performance Evaluation Review, **35**(4):14-21, Association for Computing Machinery.

Milner, R., 2007. Communicating and Mobile Systems: the calculus. Cambridge University Press, Cambridge, UK.

Palacz, W., 2008. Practical Aspects of Graph Transformation Meta-Rules. Intelligent Computing in Engineering, 70-77.

Rossello, F., Valiente, G., 2005. Graph Transformation in Molecular Biology. Formal Methods (Ehrig Festschrift), *Lecture Notes in Computer Science* **3393:**116-133.

Yadav, M.K., Kelley, B.P., Silverman, S.M., 2004. The Potential of a Chemical Graph Transformation System. International Conference on Graph Transformations 2004, Lecture Notes in Computer Science **3256**: 83-95.

http://tfs.cs.tu-berlin.de/agg/, The AGG Homepage, October 2008.