USER-CONTROLLED KINETIC NETWORK GENERATION WITH INGEN

by

CRAIG ALLEN BENNETT

A Dissertation Submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in

Chemical & Biochemical Engineering

written under the direction of

Professor Michael T. Klein

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2009

ABSTRACT OF THE DISSERTATION

User-controlled Kinetic Network Generation with INGen

By Craig Allen Bennett

Dissertation Director:
Professor Michael T. Klein

A new kinetic model building software tool, INGen, was developed to allow, for the first time, large and detailed kinetic models to be built by chemists and chemical engineers without specific expertise in advanced computer programming languages. The creation of INGen put this power in the hands of chemists and chemical engineers by allowing any user the ability to model complex kinetic networks via a logical interface. The advanced theory of chemical reaction mechanisms and the need for extensive programming were thus incorporated into the interface instead of being required to be user supplied. In its tangible form, INGen represents a universal hydrocarbon model builder capable of producing pathways and mechanistic models for acid, metal, and free radical type chemistries. Mechanistically inclined pathways offer users more information and control over carbenium ion transitions for the PCP isomerization and cracking of paraffins. The user interface lies within the Microsoft Excel framework for quick tabulation and analysis of results, and ties strongly to CambridgeSoft's ChemDraw software for ease of species creation and structure analysis. INGen thus represents a paradigm shift in the development of molecule based modeling techniques.

ACKNOWLEDGEMENTS

My Rutgers experience has been anything but typical. I first learned of Professor's Michael Klein's research while I was an undergraduate at the University of Delaware, and it fascinated me even then. After relocating to Iselin, NJ for a job, I decided to begin night school and work towards an advanced degree. I was thrilled that Dr. Klein, who had since moved to Rutgers, accepted me into his research group. Ten years later, I'm finally finishing what I started.

I want to thank Dr. Klein, whom I consider one of the brightest, most eloquent advisors anyone could ever hope for. His patience, direction, teaching, and brainstorming helped me immeasurably over the years, and I will always treasure my time as his student.

I would also like to thank the graduate staff of the Chemical and Biochemical Engineering Department, with special kudos to Ursula Wolfe. They really helped with all the red tape details involved with being a graduate student, especially for so long.

In addition, I'd like to acknowledge Joyce Creteau and all the Bureau of Engineering Research staff for all their help behind the scenes in managing our research funding. I hope everyone realizes just how important you guys are.

I'd like to thank my family for keeping me going, even when the end was so far from sight, and my friends for assuming I would finish, even when I wasn't sure.

Finally, of course, I need to thank my wonderful wife Kristina for her emotional, psychological, spiritual, and technical support through the years. You're the tops.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

Chapter 1. Introduction

1.1 World changes motivate molecular modeling

After almost a decade of the new millennium, the world still relies heavily on easy to obtain and process light oils for its energy needs. As these finite resources diminish, countries and their respective companies will need to re-evaluate the way they handle those precious commodities. Lighter, easier to process oils will rise in price, and therefore need to be efficiently handled to reduce any opportunity cost that may come about due to a non-optimized selectivity of a process.

In addition, governments around the world are beginning to enable stricter policies aimed at preserving the environment. The United States, for example, has recently labeled carbon dioxide as a pollutant. Shocking, and absurd, as it may be, how many processes account for the accurate measurement of carbon dioxide (or any other specific pollutant)?

At the crux of these two problems, lies the solution of molecular kinetic modeling. "Modeling" itself contains problems and tradeoffs of its own. Real world reaction systems, especially those found in the hydrocarbon industry, are huge; sometimes containing tens of thousands of species! Building, tuning, and solving such a model can be impossibly tedious by hand, but with the right tools, it can be accomplished with the use of the computer.

1.2 Decreasing the complexity of modeling

Complexity has always been the bugbear of accurately modeling the kinetics of a chemical processes. First, there is the logistical complexity of the system: useful, real-world chemistries often contain hundreds if not thousands of species and reactions.

Secondly, there is a chemical complexity in establishing the size and scope of a model: how many lumps/reaction families are needed, what kind of data are available, etc. And finally, there is a complexity to the programming itself: creating models often requires a level of programming skill that is often beyond the scope of most users. These complexities can sometimes leave modeling in the hands of a small subset of engineers and chemists. This work ultimately strives to overcome these complexities and place the power of its modeling techniques in the hands of a larger audience.

Real world problems, especially in the hydrocarbon or petrochemical industry, have thousands of species, each reacting in tens of different ways. The pure nightmare of building the network and keeping track of the composition can only be abated with the use of a computer and/or an aggregation technique. Computer-based network generation and accounting techniques have, for the most part, existed for the past 15 years, but they have not seen wide spread use because of their own complexity. Lumping and aggregation techniques have been around for a lot longer, but have limitations that do not address the more chemically complex needs of today's engineer or scientist.

It is the intention of this work, and the software described herein, to allow uninitiated users full access to powerful kinetic network building tools and methodologies. The ultimate accomplishment was for a new user to create and evaluate a representative network with little to no outside help.

1.3 Kinetic modeling background and literature review

Industrial processes consist of numerous chemical reactions that involve the interactions of a large set of molecular and intermediate species. Every chemical process takes a

sub-set of species, known commonly as the feed, and through reaction, produces a different sub-set of species known as the products. The initial concentrations of the feed along with the conditions of the process, such as contact time, temperature, and pressure, drive the reaction and ultimately produce the resultant products.

It is often the job of the engineer to force the products to conform to a set of necessary standards. This task can be accomplished by varying said conditions, but whereas this is often easily accomplished on a laboratory scale, it becomes much harder as the process is scaled up. Changing the pressure or temperature of an industrial process is often an expensive endeavor, either in equipment, energy or even opportunity cost. The engineer is not often afforded the opportunity to adjust these conditions unless proof has been provided that the results will be economically viable.

Whenever a process cannot be experimented with on its full scale, the problem must be modeled. Modeling, in its simplest definition, is the process of representing something by using something else in its place. The equations that comprise most engineering problems are models of the behavior of the system. The parameters and level of the modeling are defined by the assumptions of the engineer. For instance, many chemical engineering problems can assume adiabatic or isothermal conditions; whereas this may not be strictly true, the representation may provide accurate enough information.

A large part of the process of modeling deals with determining exactly what information is necessary, available, and useful. If octane number is the most useful property of the product stream, the engineer must design the model around accurately determining it. If heat capacity of the same product stream is not important, then the modeler can ignore it insomuch as it does not affect the octane number. However, it is often the case that the engineer has to deal with a scarcity of data when designing the model. Balancing

number of parameters to the amount of available information can become a tricky task for the modeler. Sometimes, it is necessary for the engineer to take the available information and create a side model that will turn less information into more. By doing such, the modeler creates a finer granularity through which the model can be controlled.

Returning to the industrial example, one of the first options that an engineer or chemist has available is to represent the industrial process on a much smaller scale. Modeling the system on a laboratory scale, especially with a reasonable design of experiments, can quickly provide information that will help pinpoint possible condition ranges that will satisfy outlet concentrations. Whereas the laboratory scale model provides good information on the kinetics of reaction, there are many other issues that do not scale up well, such as heat and mass transfer. Good experimental results can sometimes lead to unforeseen problems on the industrial level.

A better solution is to take the results of the laboratory design of experiments and create an experimental plan to follow on a pilot plant scale. The pilot plant data will provide more insight into scale up hardships. Still, the pilot plant model may have scale up issues between pilot plant and industrial scales, though the problems are often more foreseeable. Additional problems arise in that pilot plants are few and far between. They are often in a central location catering to many different plants. Because of scheduling, it often becomes difficult to perform the necessary experiments in a timely matter. Instead, generalized reaction rules-of-thumb are gained, and industrial process engineers are afforded the opportunity to play with conditions within those ranges.

The next option available to the engineer is to model the system, not physically, but through mathematics. Balance equations can be created to represent the conservation of energy, momentum, and mass of the species. As mentioned above, certain assumptions

must be made, and various modeling parameters must be specified. For instance, one can model the kinetics of reaction using the Arrhenius equation to find the kinetic coefficient of the reaction equation. The equation requires the A and E0 values to be specified, and for most reaction systems, these values will not be available. The values must be "tuned" from process conditions and stream sampling on the industrial and pilot plant scale, as well as experimental results on the pilot plant and lab scale.

Tuning is a process of systematically changing the parameters of the model such that the predicted products match those that were determined experimentally. Various tuning methodologies exist, but each one will require that the model is computed hundreds if not thousands of times depending on the number of reactions. Obviously, tuning the model by hand will be extremely tedious for reaction networks with a large number of reactions. Solving and tuning the model numerically on a computer becomes the only reasonable option. However, simply representing a large scale kinetic network on the computer becomes an intense exercise in bookkeeping, and the tuning of said model may still take days of computation time. These two problems have been solved in a variety of ways.

The most common methodology is to combine the model species into "lumps" [1-15]. Mosby and fellow Amoco co-workers developed a resid conversion model via lumping the feed and the products each into three separate lumps [9]. The feed was broken into "hard" resid (hard-to-remove 1000+ ^{o}F material), "easy" resid (easy-to-remove 1000+ ^{o}F material), and gasoil (1000- ^{o}F material). The products were lumped into gases, naphtha, and distillate. In addition, an intermediate gasoil lump was also used. Interconversion between feed, intermediate, and product lumps was modeled using a single kinetic rate constant for each reaction pathway. Each was normalized to the reaction of hard resid to intermediate gasoil. Their results correlated well with performance data, but there is a great lack of detail in the product composition.

Another methodology was developed by Quann and Jaffe [16]. They developed a system of modeling by representing individual hydrocarbon molecules as vectors of 22 structural increments. They called their modeling approach Structure-Oriented Lumping (SOL). The structural increments used include three aromatic increments, six naphthenic increments, a CH_2 side-chain increment, a branch descriptor, a methyl descriptor, a hydrogen increment for unsaturated species, a biphenyl bridging increment, two sulfur increments, three nitrogen increments, and three oxygen increments. The vector for a species contains the number of each increment required to rebuild the molecule.

The original SOL program was later adapted to include two new increments, Nickel and Vanadium to represent certain porphyrins [17]. In addition, large multi-core structures were made possible by encoding extra information into the biphenyl bridging increment and the side-chain increment.

SOL uses ordered sets of reaction rules to transform reactant increments into product increments recursively. Once the products have been created, SOL uses its extensive database of properties and correlations to recreate overall physical properties of the products. It has been shown to work very well and is the industry standard for a large scale industrial modeling project.

Whereas lumping methods provide a simple network structure, and few parameters to tune upon, the output of such models does not provide the detail that may be required in today's industrial setting. Growing environmental concerns require that models be able to predict harmful product concentrations. Diminishing resources require that models be able to handle new feed stocks, and provide detailed information on products such that

desirable products are predicted in as much detail as possible.

Broadbelt, Stark and Klein [18] created the NetGen software to solve the problem of creating detailed large scale kinetic networks automatically on a computer. Molecules are represented as atomic connectivity graphs, and reactions occur through a process of applying universal reaction matrices to the reactants. Klein's research group has developed numerous NetGen implementations over the years [18-25], but each was written for a specific chemistry, and the rules of reaction are deeply imbedded the code. In order to apply it to a new chemistry, a new NetGen had to be created.

The output of NetGen could then be carried into the other programs of the KMT suite, or, after the work of Wei Wei at Rutgers, into KME [24]. ODEGen and CodeGen take the network of kinetic reactions and automatically create the model ODEs and wrapper code in the C programming language. KME expanded the functionality of KMT's ODEGen by allowing the user to specify reactor type, run mode, tuning algorithm, and other model specifications.

The user controlled nature of KME was a great boon to the KMT package, allowing end users to create, tune, and run kinetic models quickly. KME allowed users to enter in a series of reactions through a simple "A + B → C + D" language, and within a single button press, KME would compile an executable model for the reaction network. KME was limited in scope because the reactions needed to be entered by hand.

Creating small scale networks from scratch, especially those that were already well documented, became the focal point of KME modeling. Large scale networks could still be handled, but only with the same rigorous manual logistics that, as covered earlier, proved to be an almost insurmountable obstacle to molecule based modeling of large

scale systems. If a large scale model already existed, and its network could be converted to the KME grammar, KME could be used to add and subtract reactions without going back to the original model building method. Thus, a logical break formed between KME and the historic model builders of the KMT software. KME became a run-time modeling environment, capable of making specific small changes to the reaction networks, but it had no part in helping to automatically generate the networks.

NetGen and its many model builders would still need to be used for model creation. Unfortunately, the simplicity of the interface that made KME a useful tool to non-programmers, was completely absent from NetGen. In addition, there were many model building programs under the NetGen title, each aligned to a specific chemical process. A graphical interface could not simply be placed upon the existing NetGen framework; the entire program would need to be retooled and redesigned.

It was with this task in mind that the INGen project began. The ultimate goal of which was an interface as simple to use as that of KME, which would allow for the rapid generation of kinetic networks by any user familiar with the chemistry and not the programming. As a companion piece to KME, INGen would provide the missing automatic network generation step, thereby creating a cradle-to-grave suite of easy-to-use modeling tools. The following chapters will tell the story of the INGen success.

1.4 Thesis overview

This thesis is broken into two major logical sections, the first of which deals with the behind the scenes changes to the NetGen methodology and chemistries. The second consists of the creation of the user interface and the application of such to specific chemistry problems.

Chapter 2 will specifically delve into the issue of the expansion of NetGen's controllability and the logic associated with model building. In addition, various smaller fixes and improvements will be discussed.

Chapter 3 will cover the addition of new paraffin isomerization and cracking pathways that were built with an awareness of carbenium ion intermediates.

Chapter 4 will deal with the creation of the user interface and intermediate analytic tools and processes.

Chapter 5 will discuss the use of the tool to develop a pathways level paraffin isomerization and cracking network

Chapter 6 will similarly comprise the use of the tool in creating a specific hydro-processing network.

Chapter 7 will conclude the work, present a summary of relevant results, and propose further research.

1.5 References

[1] Kmak, W.S. (1971). A Kinetic Simulation Model of the Powerforming Process. AIChE Nat. Meet., Houston.

[2] John, T.M. and B.W. Wojciechowski (1975). "On Identifying the Primary and Secondary Products of the Catalytic Cracking of Neutral Distillates." J. Catal. **37**: 348.

[3] Van Damme, P.S., S. Narayanan, et al. (1975). "Thermal Cracking of Propane and Propane-Propylene Mixtures: Pilot Plant Versus Industrial Data." AIChE J. **21**: 1065-1073.

[4] Jacob, S.M., B. Gross, et al. (1976). "A Lumping and REaction Scheme for Catalytic Cracking." AIChE J. **22**: 701-713.

[5] Sundaram, K.M. and G.F. Froment (1977a). "Modeling of Thermal Cracking Kinetics." Chem. Eng. Sci **32**: 601-608.

[6] Sundaram, K.M. and G.F. Froment (1977b). "Modeling of Thermal Cracking Kinetics - II." Chem. Eng. Sci **32**: 609-617.

[7] Sundaram, K.M. and G.F. Froment (1978). "Modeling of Thermal Kinetics. 3. Radical Mechanisms for the Pyrolysis of Simple Paraffins, Olefins and Their Mixtures." Ind. Eng. Chem. Res. **17**: 174-182.

[8] Marin, G.B. and G.F. Froment (1982). "Reforming of C6 hydrocarbons on a Pt-Al2O3 Catalyst." Chem. Eng. Sci. **37**(5): 759-773.

[9] Mosby, J.F., R.D. Buttke, et al. (1986). “Process Characterization of Expanded-Reactors in Series.” Chem. Eng. Sci. **41**(4): 989-995.

[10] Ramage, M.P., K.R. Graziani, et al. (1987). "KINPTR (Mobil's Kinetic Reforming Model): A review of Mobil's industrial process modeling philosophy." Adv. Chem. Eng. **13**(193).

[11] Astarita, G. and R. Ocone (1988). "Lumping Nonlinear Kinetics." AIChE J. **34**: 1299.

[12] Aris, R. (1989). "On Reactions in Continuous Mixtures." AIChE J. **35**: 539-548.

[13] Gray, M.R. (1990). "Lumped Kinetics of Structural Groups: Hydrotreating of Heavy Distillates." Ind. Eng. Chem. Res. **25**: 505-512.

[14] Mudt, D.R., T.W. Hoffman, et al. (1995). "The Closed-Loop Optimization of a Semi-Regenerative Catalytic Reforming Process." AIChE paper-w51.

[15] Laxminarasimhan, C.S., R.P. Verma, et al. (1996). "Continuous Lumping Model for Simulation of Hydrocracking." AIChE J. **42**(9): 2645-2653.

[16] Quann, R.J. and S.B. Jaffe (1992). "Structure Oriented Lumping. Describing the Chemistry of Complex Hydrocarbon Mixtures." Ind. Eng. Chem. Res. **31**(11): 2483-2497.

[17] Jaffe, S.B., H. Freund, and W.N. Olmstead (2005). "Extension of Structure-Oriented Lumping to Vacuum Residua." Ind. Eng. Chem. Res. **44:** 9840-9852.

[18] Broadbelt, L.J., S.M. Stark, et al. (1994a). "Computer Generated Pyrolysis Modeling: On-the-Fly Generation of Species, Reactions and Rates." Ind. Eng. Chem. Res. **33**: 790-799.

[19] Broadbelt, L.J., C. Lamarca, et al. (1995). Chemical Modeling Analysis of Poly(Aryl Ether Sulfone) Thermal-Stability through Computer-Generated Reaction-Mechanisms. Industrial & Engineering Chemistry Research. **34:** 4212-4221.

[20] Broadbelt, L.J., S.M. Stark, et al. (1996). Computer generated reaction modelling: Decomposition and encoding algorithms for determining species uniqueness. Computers & Chemical Engineering. **20:** 113-129.

[21] Hou, G. and M.T. Klein (1999). Molecular modeling of gas oil hydrodesulfurization. Abstracts of Papers of the American Chemical Society. **218:** U610-U611.

[22] Joshi, P.V. (1998). Molecular and Mechanistic Modeling of Complex Process Chemistries. Chemical Engineering, University of Delaware.

[23] Joshi, P.V., S.D. Iyer, et al. (1997). Computer assisted modeling of gas oil fluid catalytic cracking (FCC). Abstracts of Papers of the American Chemical Society. **214:** 80-Petr.

[24] Wei, W. (2005). The Interface of Chemical Engineering and IT in Kinetics Models. Chemical and Biochemical Engineering, Rutgers University

Chapter 2. Core INGen Model Building Functionality

2.1 Model building fundamentals and background

There exists a set of fundamental obstacles which any model building software must overcome. First, the software must be able to accurately interpret and represent the structure of a molecule. Second, the software must be able to convert one structure to another by using chemical reactivity knowledge. Third, the growth of the network must be controlled. Fourth and finally, the resultant information must be useful. Without a clear-cut solution to any one of these, the model builder will not be useful.

2.2 Conceptual species representation and interpretation

2.2.1 Background

A molecule is defined by its atomic makeup, connectivity, and structure. From early in our chemistry training, we are taught to visualize molecules as ball and stick figures: the balls being atoms, and the sticks the bonds between them. We are given kits where the balls have geometrically arranged holes to represent valence electrons, and bendable, variable length sticks that can be used to represent single, double, and triple covalent bonds. The ball-and-stick model of spatial molecular representation is conceptually a good one, and the concepts are of great importance when trying to describe a molecule to a computer.

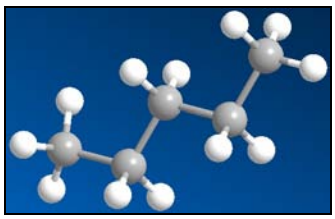


Figure 2.1 - Ball and stick representation of pentane

If we simply change the words ball to vertex, and stick to edge; we instantly have described a molecule in the mathematical terms of a graph. Graph theory has been a specialty mathematic science for well over a century, and has a number of pre-existing data structures and algorithms.

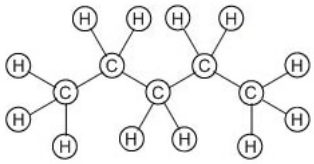


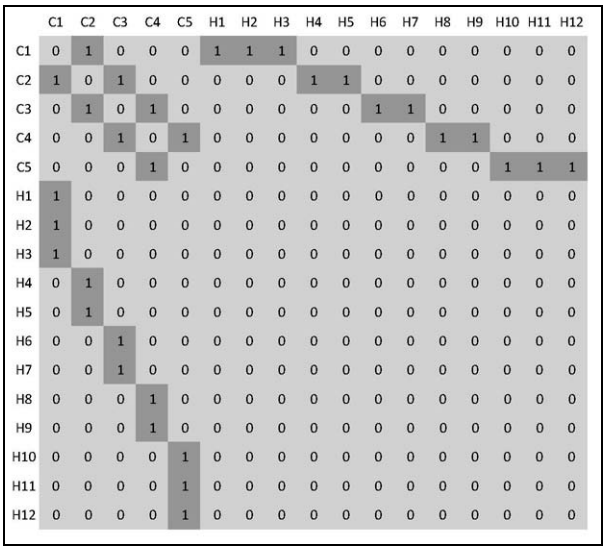
Figure 2.2 - Graph representation of pentane

2.2.2 Graph theory data structures and algorithms

Conceptually, the core structure for a molecule is that of the adjacency matrix. An adjacency matrix is an *n x n* matrix, where n is the number of vertices, or in this case atoms. The non-diagonal entries a_{ij} represent the number of edges connecting vertex *i* to vertex *j*. Putting this in terms of atoms and bonds, the matrix element a_{ij} is the bond order between atom *i* and atom *j*, where 0 represents no bond. Strictly speaking, the diagonal elements of an adjacency matrix represent edges that connect a vertex unto itself. Two graph theory conventions exist in that a_{ii} represents either the number of loop edges or twice that number (because of the two edge connections that are entailed), but neither convention is useful for the implemented molecular representation. Instead, the diagonal elements are used to

represent unpaired electrons (the empty holes in the ball-and-stick model).

Consider that in nearly all hydrocarbons each carbon bonds to at most four other atoms, and the hydrogen atoms bond to only one other atom. This limitation will lead to a sparse matrix structure as the size of the molecule increases. In addition, as the molecular composition increases linearly, the memory requirements of an adjacency matrix increase exponentially. Therefore, the adjacency matrix is an inefficient data structure for modeling molecules.



	C1	C2	C3	C4	C5	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12
C1	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
C2	1	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0
C3	0	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0
C4	0	0	1	0	1	0	0	0	0	0	0	0	1	1	0	0	0
C5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1
H1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H6	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H8	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
H9	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
H10	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
H11	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
H12	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2.3 - Sparse adjacency matrix of pentane

So, instead of an adjacency matrix, an *adjacency list* is used. In an adjacency list, each vertex has a list of the vertices it is adjacent to and their edge weights (bond orders). Although there is redundancy for each bond (it is listed as A connects to B, as well as B connects to A), the adjacency list is much more efficient for storing a sparsely connected molecular structure. The capabilities of the adjacency list were expanded to include atomic species identification (including radical specification). By creating these adjacency lists and saving them as .dat files, a modeler will be able to import molecular structures into the model building program. Simplified creation of these adjacency lists

will be handled in the a later section within this chapter, and it will be touched upon again in Chapter 4.

```
//pentane
17
0 0 C : {1, 1}, {5, 1}, {6, 1}, {7, 1};
1 0 C : {0, 1}, {2, 1}, {8, 1}, {9, 1};
2 0 C : {1, 1}, {3, 1}, {10, 1}, {11, 1};
3 0 C : {2, 1}, {4, 1}, {12, 1}, {13, 1};
4 0 C : {3, 1}, {14, 1}, {15, 1}, {16, 1};
5 0 H : {0, 1};
6 0 H : {0, 1};
7 0 H : {0, 1};
8 0 H : {0, 2};
9 0 H : {0, 2};
10 0 H : {0, 3};
11 0 H : {0, 3};
12 0 H : {0, 4};
13 0 H : {0, 4};
14 0 H : {0, 5};
15 0 H : {0, 5};
16 0 H : {0, 5};
```

Figure 2.4 - Adjacency list ".dat" file for pentane

Connectivity alone is not enough information to be useful. Depending on the choice of atomic labeling, it is possible for a single molecular species to be represented by numerous adjacency lists. Therefore, to be able to determine the equality of two molecules a canonical representation of the molecule must be established. This is accomplished by performing a depth-first search of the molecule in order to establish a spanning tree, as well as a list of back edges.

The depth-first search begins with a single connected vertex. The vertex is logged into the spanning tree, and then one of the edges of that vertex is chosen. If the adjoining vertex has not been visited, the edge is traversed and is noted as being a tree edge. That newly visited vertex becomes the current vertex, and the process repeats itself. If the adjoining vertex has been visited before (and the edge is not already marked as a tree edge), the edge become listed as a back edge, and another edge of the current vertex is evaluated. If all the edges of a vertex connect to already seen vertices, the parent of the current vertex becomes the current vertex again and its other edges are evaluated.

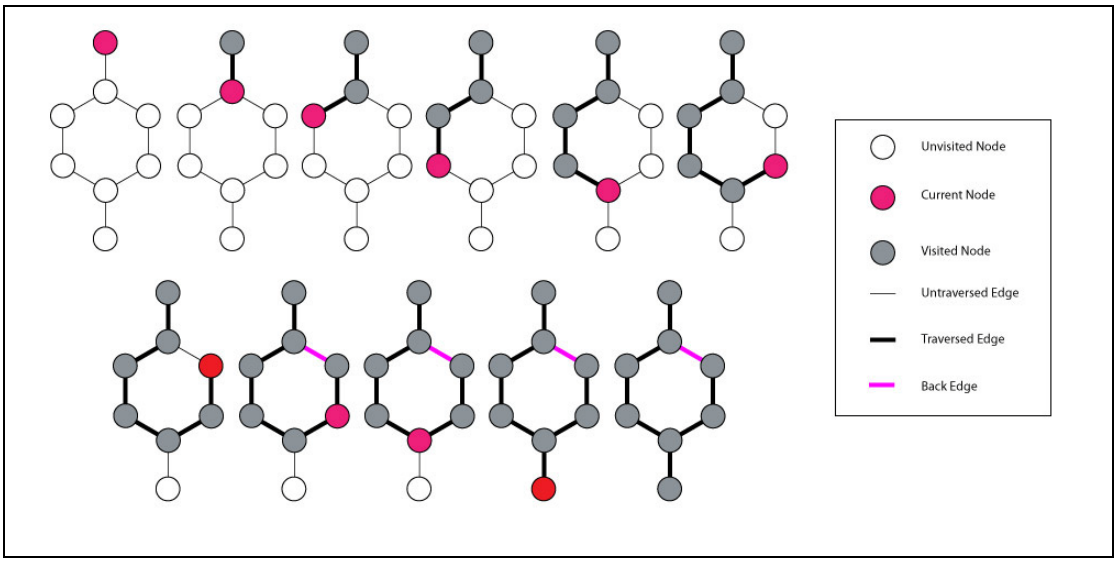


Figure 2.5 – Depth-first search example

At the end of the depth-first search, a single tree that spans all the vertices of the graph will be described. A tree by definition is a connected acyclic graph. If the molecule did have cycles or rings, our back edge list will not be empty. The union of the spanning tree and the back edge list will recreate the original graph, thus insuring no information is lost.

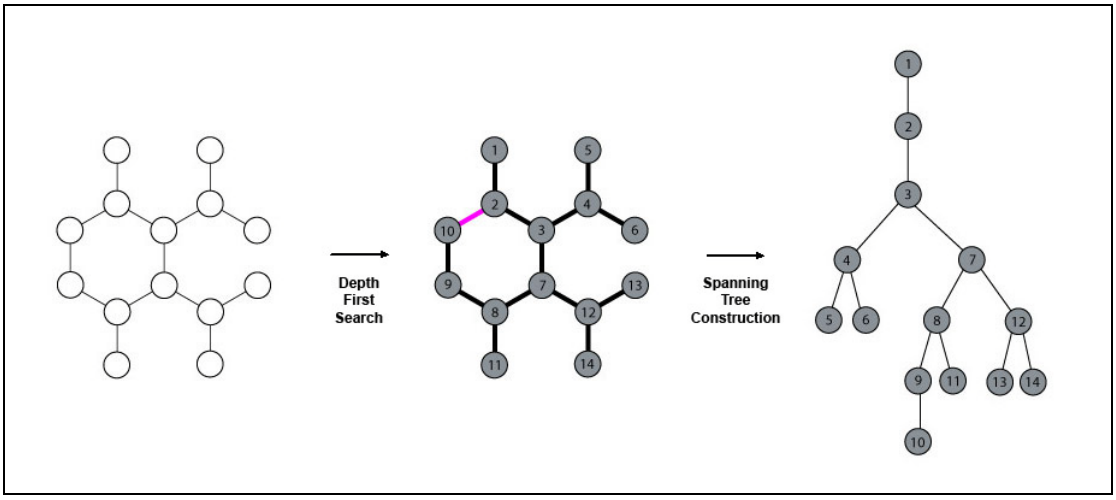


Figure 2.6 - Spanning tree example

The back edge list is used as a starting point for cycle detection. The two connected vertices defined by the back edge are members of the same biconnected component. By definition, a biconnected component is a maximal biconnected subgraph or our initial graph. A biconnected subgraph is a collection of vertices that, if any edge connecting two of those vertices were removed, the graph would still be connected.

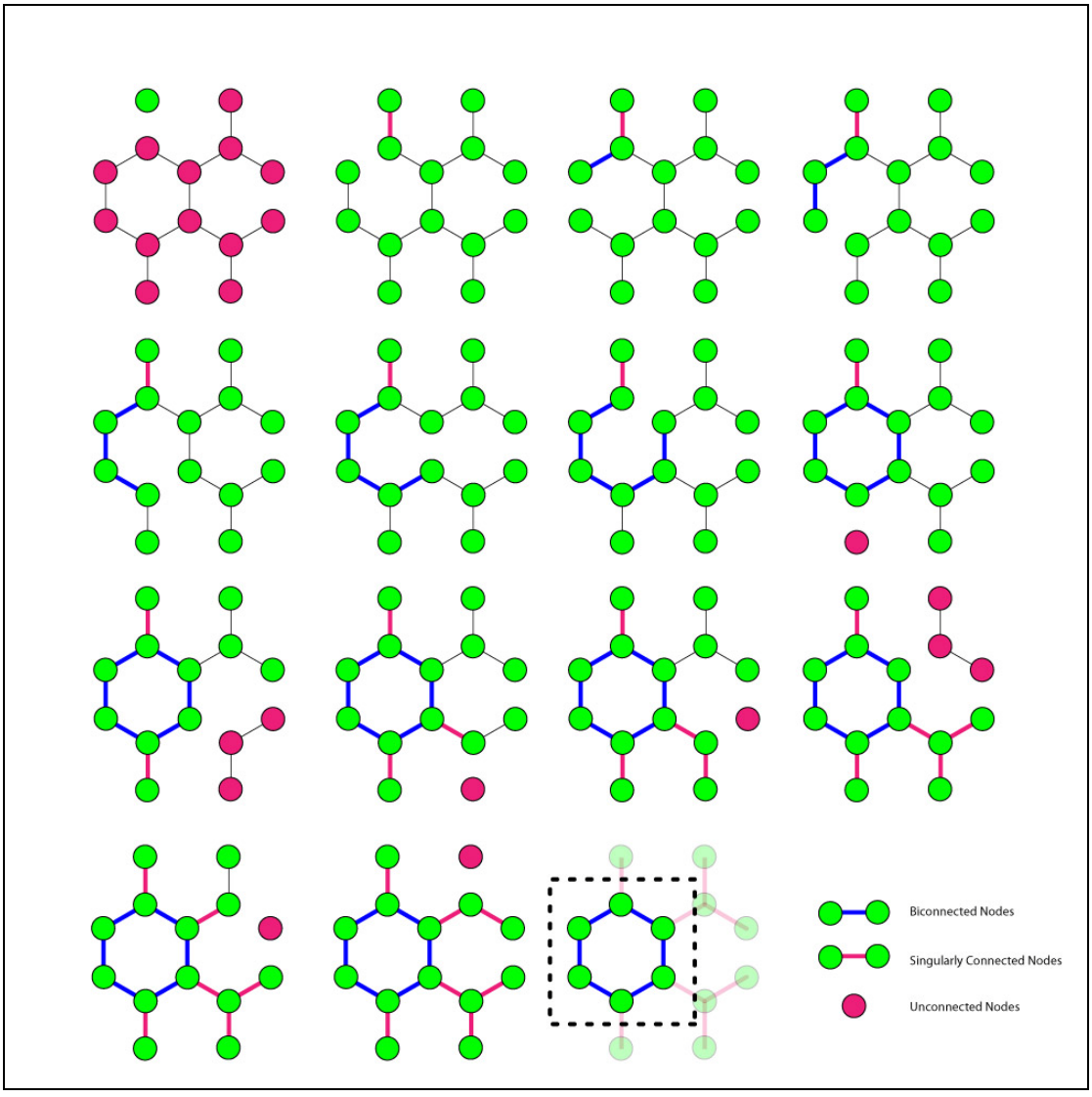


Figure 2.7 - Finding a biconnected subgraph

If a graph has more than one biconnected component, the biconnected components will be connected by chains of one or more articulation points. In general, an articulation point is

a vertex that, when removed, will split the graph into two unconnected graphs. It is important to note, and will become essential later, that terminating vertices are not articulation points. When they are removed, the rest of the molecule is still whole. Molecularly, the easiest example of articulation points would be any of the carbons within a normal paraffin. If any of those carbon atoms were removed, the chain would be split into multiple graphs. If an end carbon is removed, the rest of the carbon chain remains, along with three separate single-vertex graphs, namely the individual hydrogens that were connected to the removed carbon.

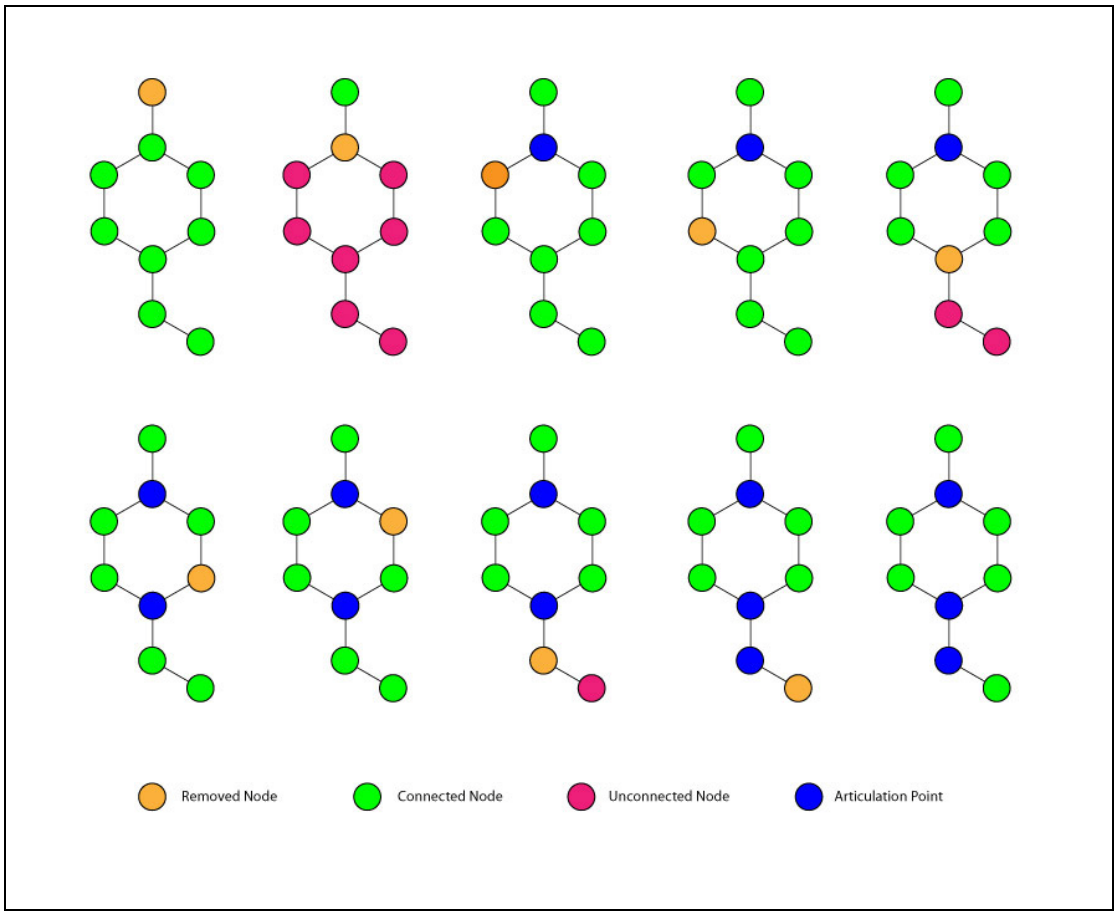


Figure 2.8 - Articulation point example

After establishing which vertices belong to which biconnected components, the minimal biconnected subgraphs of the biconnected component can be found. These minimal

subgraphs will define the chemically relevant cycles (or rings).

For each biconnected subgraph, a random node is chosen and a depth-first search is performed within the biconnected subgraph. Path information is stored for each visited node in the DFS. If a back edge is detected, a cycle is defined by retracing the path information to the far node of the back edge. This cycle is checked versus other existing cycles for duplication of routes as well as new shortest path information. If the cycle does include new shortest path information for a previously found cycle, the larger cycle is disregarded and the two new smaller cycles are noted.

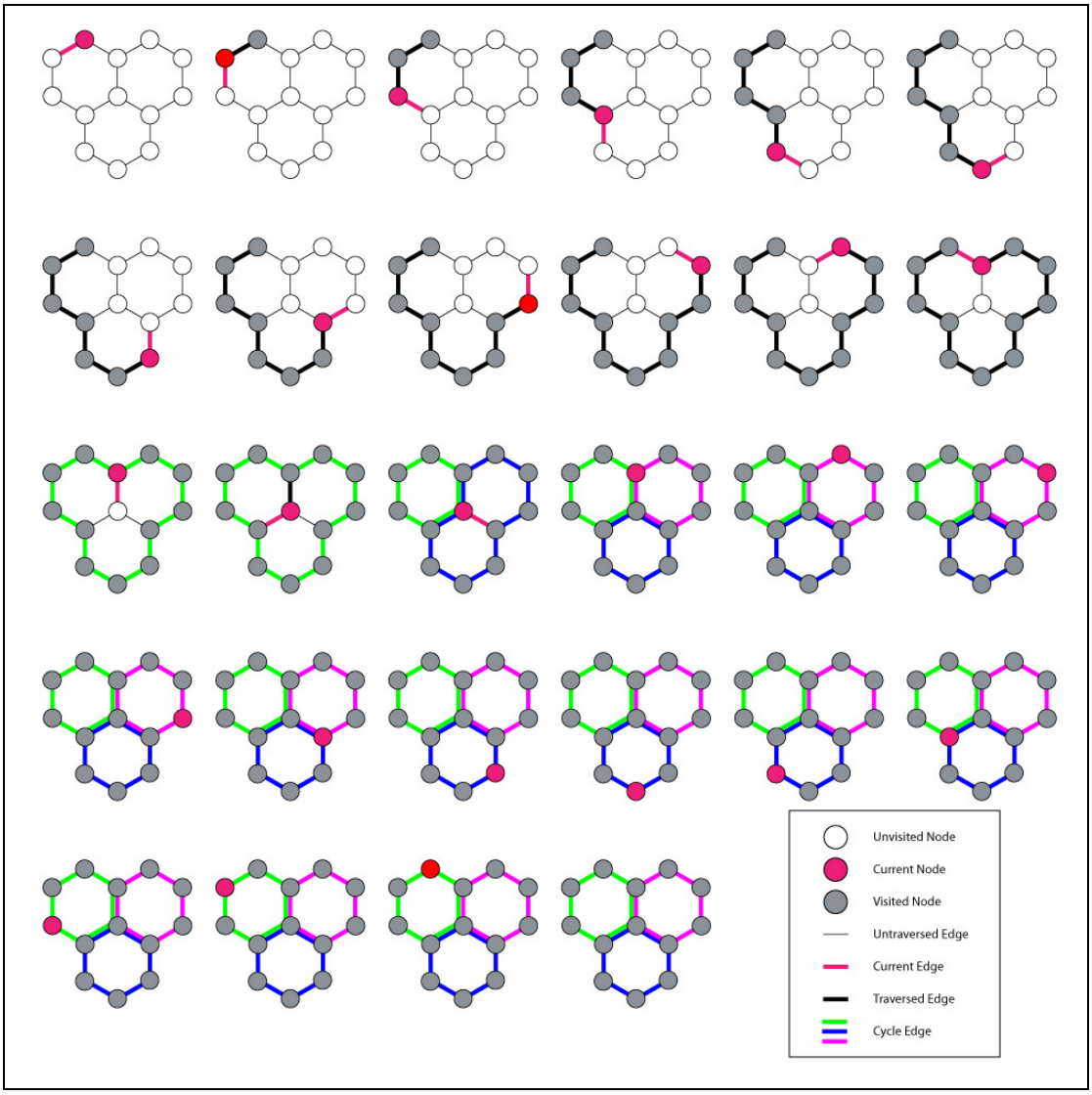


Figure 2.9 - Minimum cycle detection

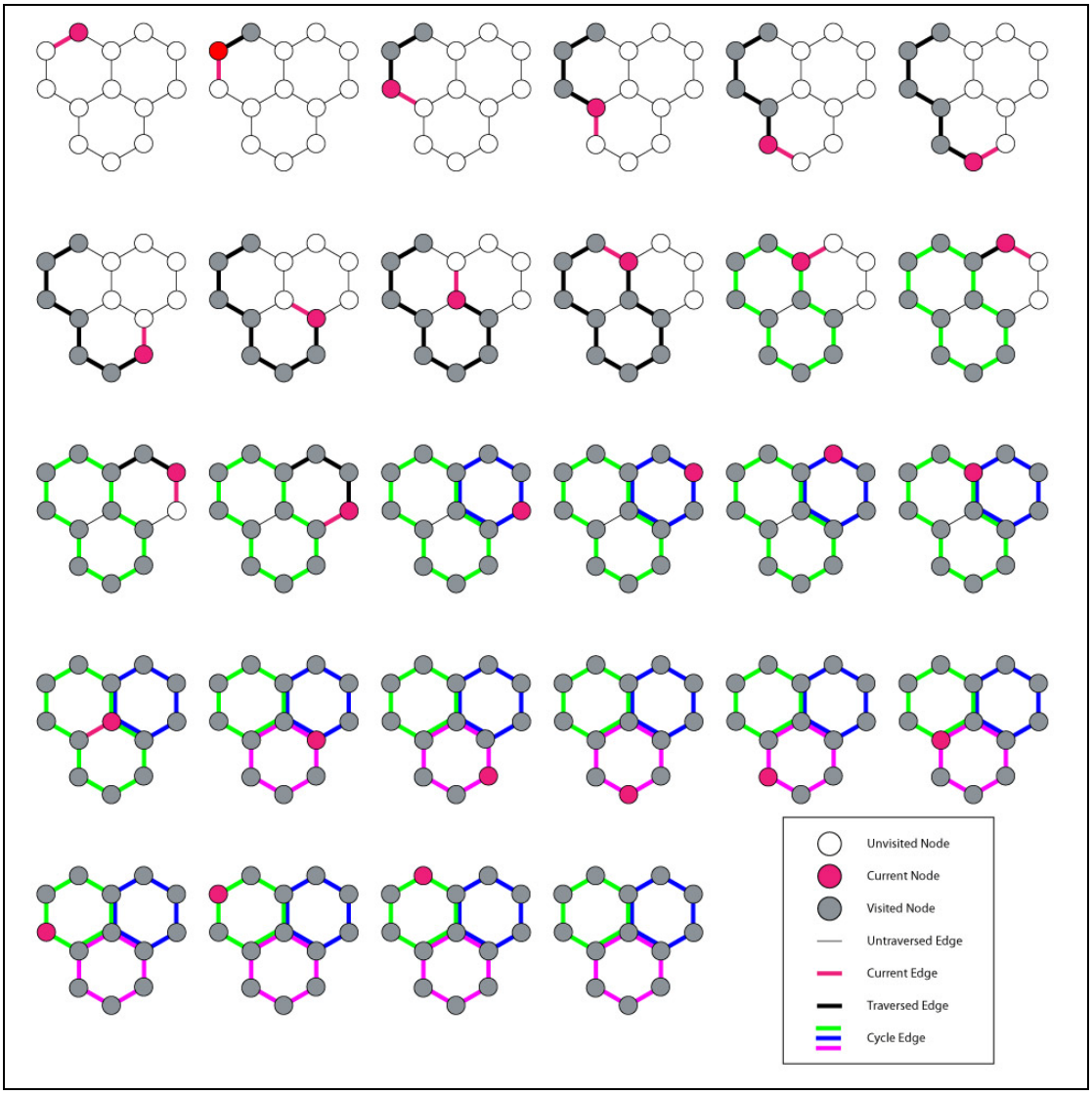


Figure 2.10 - Minimum cycle detection via different random path

Once the cycles have been identified, they can be labeled. Cycle identification can make use of the binary nature of computers. A register on a computer is comprised of a number of bits that have two states, on and off. Thinking in binary terms, the first bit position represents 2^0, the second is 2^1, the third 2^2, and so on. If each of the cycles is assigned a unique continuous number, the bit in that number's position can be used to assign an atom to a graph.

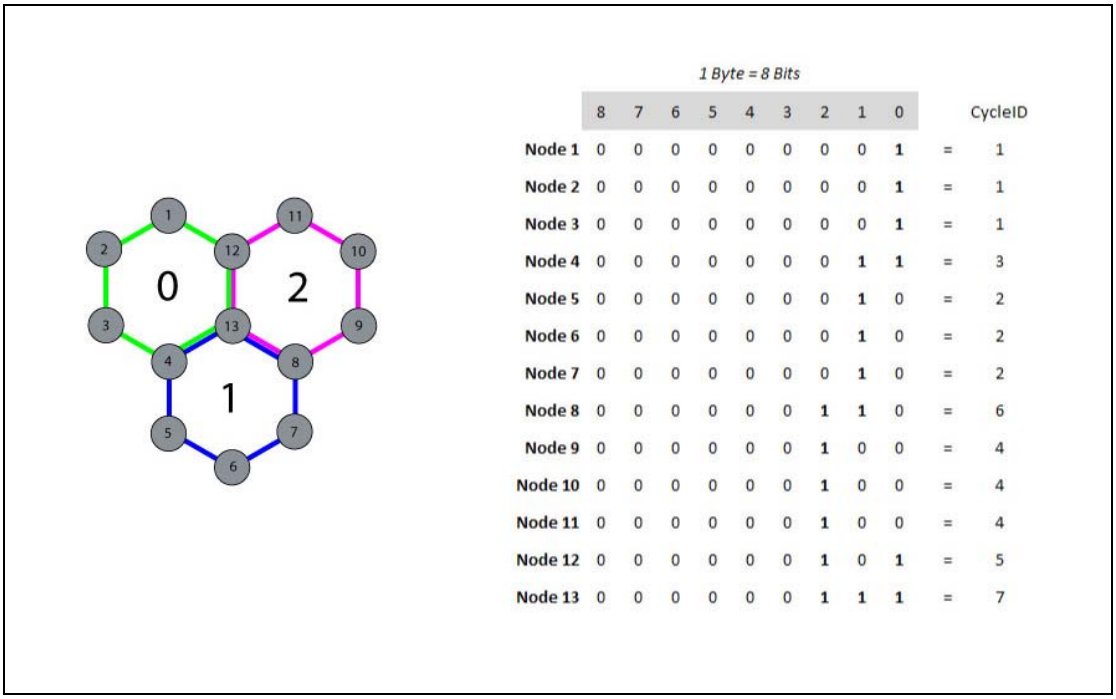


Figure 2.11 - Cycle bitset example

In this way, whether two atoms belong to the same cycle or not can be quickly established using the computers built in “bit wise and” operator (&). If one of the bit positions for both of the two atoms is set, the &-operator returns “true”, meaning that both atoms are in the same cycle.

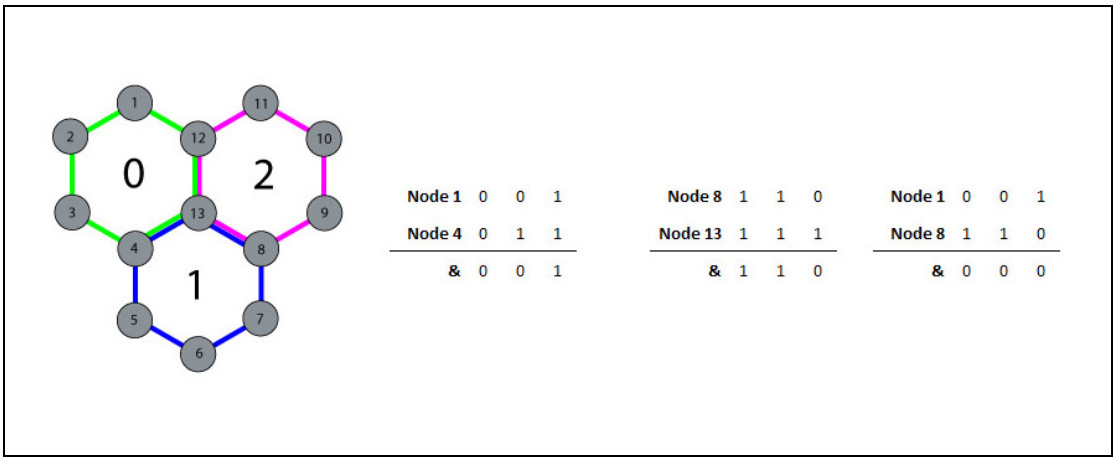


Figure 2.12 - Bitwise and-operator (&) examples

With the establishment of the number of cycles, which atoms belong to which cycles, and which atoms belong to no cycles, the canonical code for the graph can be created.

The canonical code will be a string of grouped vertices that will be unique for any given graph and its isomorphs. Canonical code construction begins with the creation of a decomposition tree. A decomposition tree is created by first finding the root of the tree via "lexographically minimal, similar degree" decomposition. Conceptually, decomposition is easiest to understand for acyclic graphs where only the similar degree aspect needs to be handled. Decomposition takes place by recursively eliminating all the nodes of a graph that are of a singular degree (only connected to one other active node) until only one or two atoms remain.

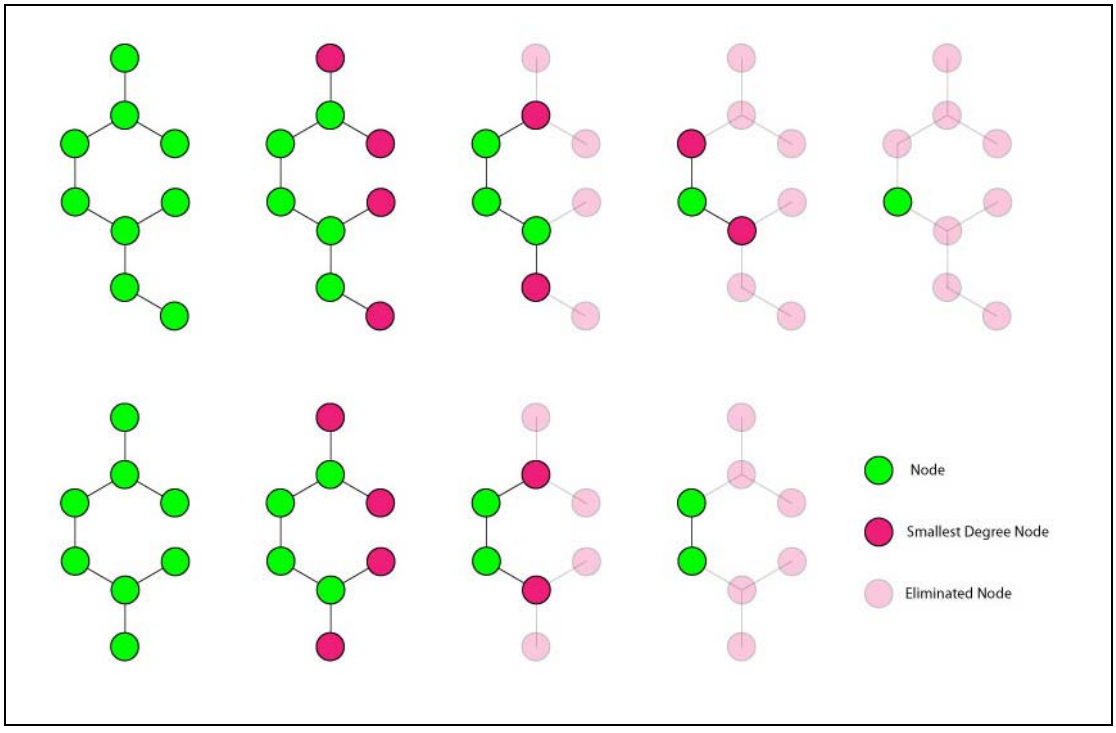


Figure 2.13 - Acyclic decomposition

If two nodes remain after decomposition, a temporary false node is used as the root of the

decomposition tree with the two remaining nodes as its children. Otherwise, the single remaining node is used. The tree is then constructed from the root in the normal fashion wherein each child node is listed a level below its parent node.

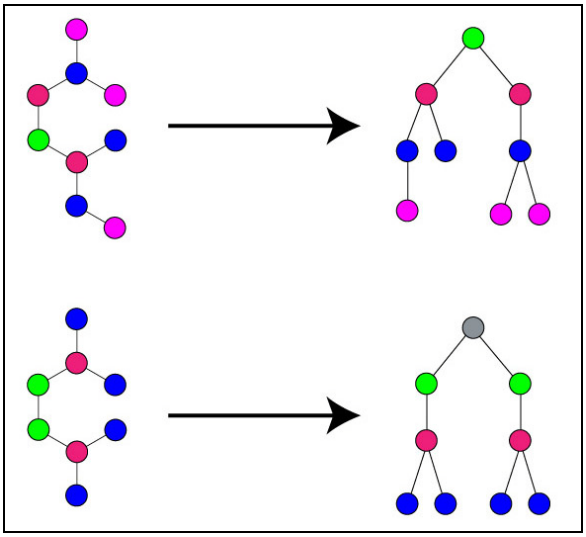


Figure 2.14 - Acyclic decomposition tree creation

The canonical code is created by beginning with the lowest parent node. Its children are lexicographically sorted by species type and placed in parentheses to the right of the parent. The parent of the parent is listed with its children's codes lexicographically sorted and placed in parentheses to the right of the parent. This process continues up the tree until the root node is reached. If the root node was a temporary node, it is not listed and its children are listed side by side without parentheses.

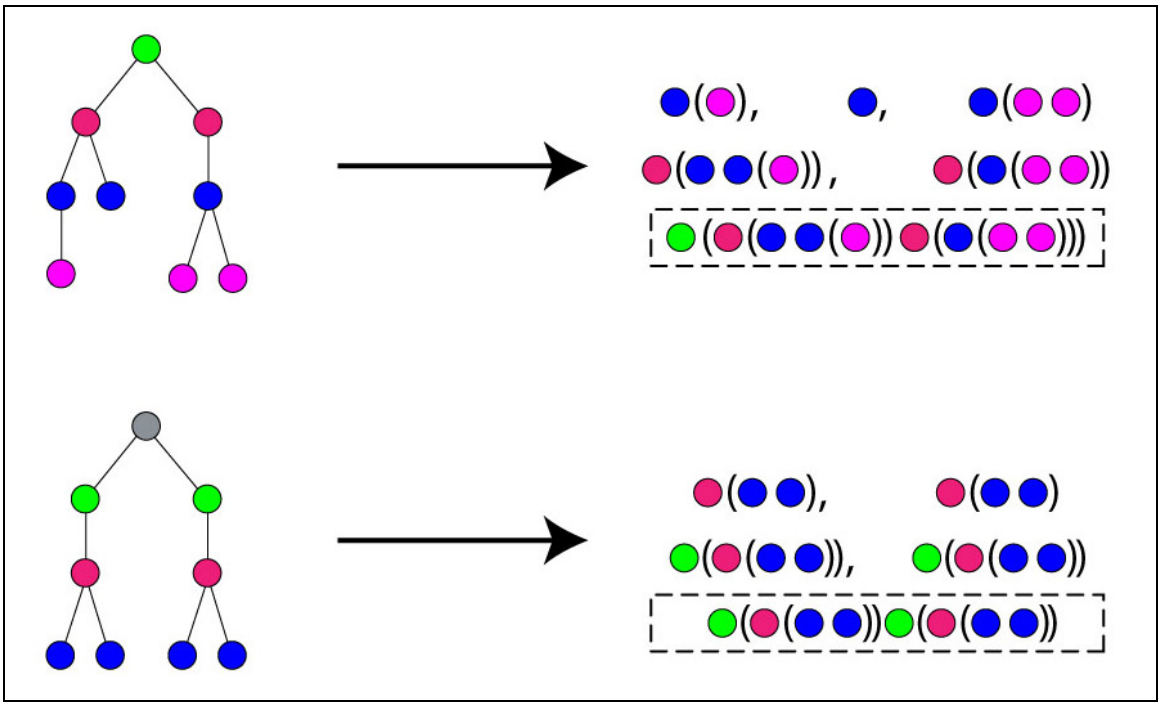


Figure 2.15 - Acyclic canonical code creation

Cyclic graphs require slightly different processing. The decomposition tree will be composed of both biconnected components and nodes. Decomposition will remove nodes as before, but eventually there will be no further singular degree nodes remaining. The biconnected components of the smallest degree are removed next. The degree of the biconnected component is defined by the number of active articulation points. Processing continues until only biconnected components (or nodes) of the same degree exist. Two biconnected components can exist at the end of processing if they are connected by a single edge.

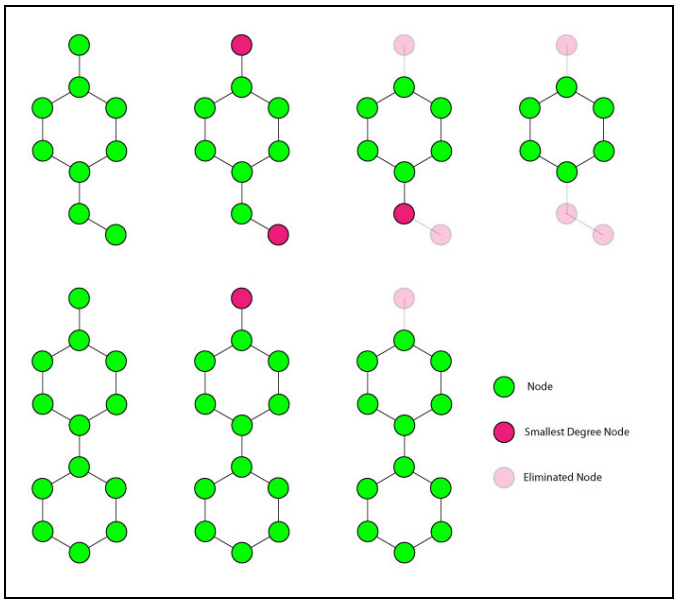


Figure 2.16 - Cyclic decomposition

After the root is found, the decomposition tree can be constructed by stepping back through the decomposition. The decomposition tree will also be populated with both nodes and biconnected components.

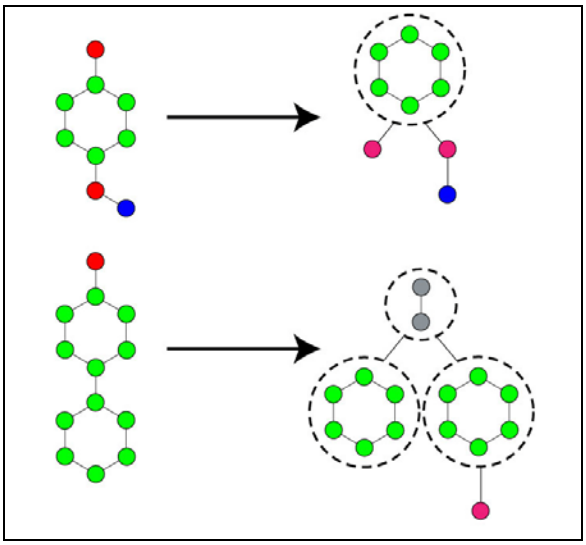


Figure 2.17 - Cyclic decomposition tree creation

The decomposition tree of a graph with biconnected components can be translated into a canonical code with special processing. For non-root biconnected components, the articulation point that connects the biconnected component to its parent acts as the entry node for the analysis. Canonical codes are recorded for each direction (via the neighbors of the initial node) around the biconnected component. Each path is considered simultaneously. If at any point one code is lexicographically shorter, that path is considered the canonical code and the other is dropped.

Similar processing takes place for root biconnected components, but there is no parent to define the initial node. The nodes with the highest degree within the biconnected component act as the initial points. Concurrent processing takes place for each initial point, and once again, only the lexicographical shorter code is kept.

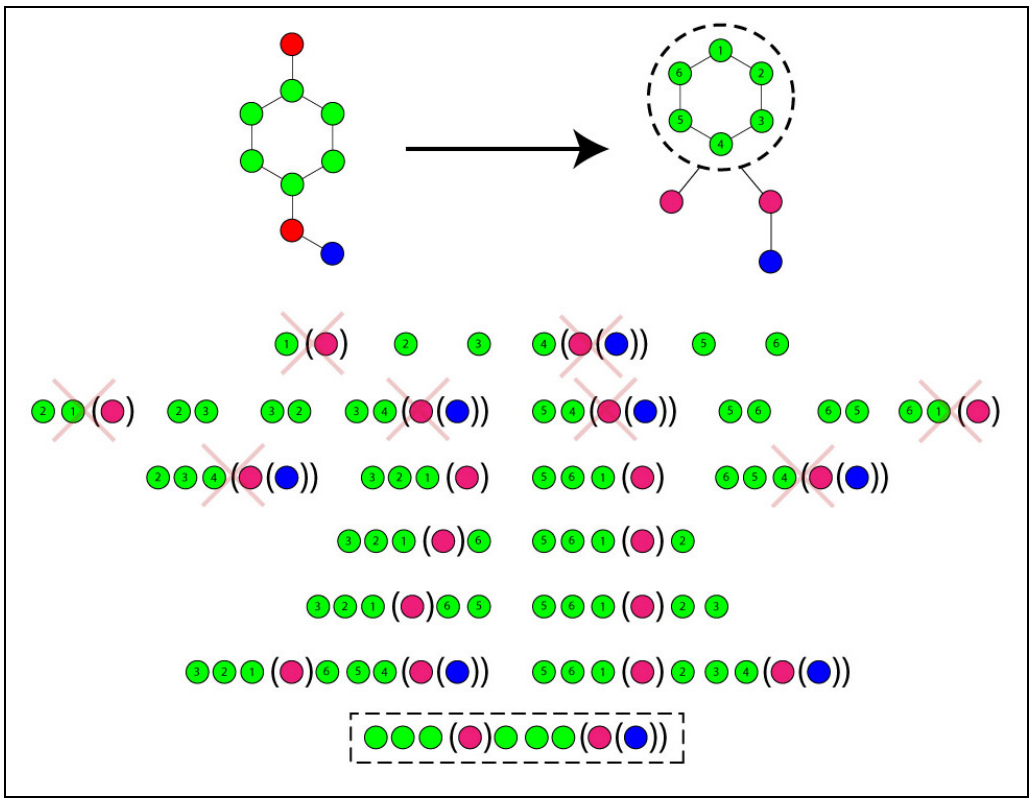


Figure 2.18 – Cyclic canonical code creation

A special case occurs when the biconnected component contains interior nodes. Interior nodes are those that cannot exist on the perimeter of a planar graph. The planar graph cannot contain edges that cross one another. In these special cases, the interior nodes are ignored during initial node selection and during traversal. After the perimeter canonical code has been created, the interior nodes (marked with a post fixed ":") are appended to the list and then lexicographically sorted. For chemical species, interior nodes are identified as those that belong to three rings.

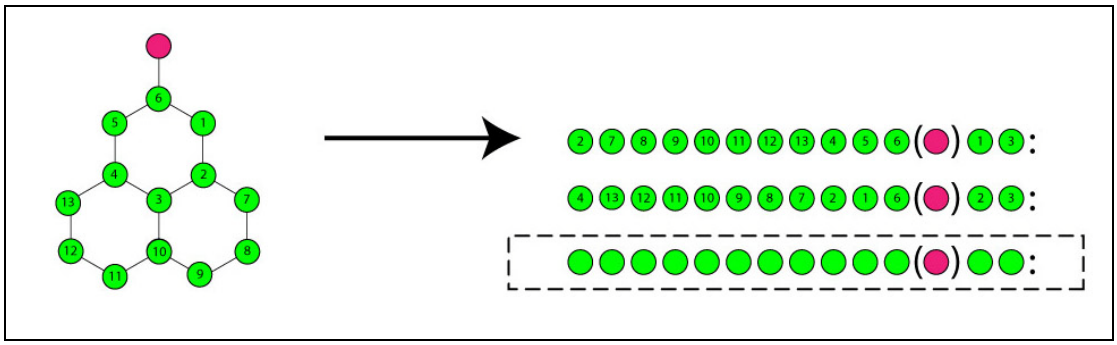


Figure 2.19 - Cyclic (with interior nodes) canonical code creation

2.2.3 Identification of species through chemical principles

Now that unique digital representations of the species have been formed, chemical principles can be applied. The first step in discovering the possible reactivity of a species is determining its type. At the fundamental level, there are three types of species handled: Molecules, Ions, and Radicals.

Molecules are identified as having the proper number of valence electrons for each of their atoms. For instance, in the adjacency matrix description, this would correspond to every row/column containing four bonds for a carbon atom with an $a_{ii} = 0$ on the diagonal (no

unpaired electrons). Ions, on the other hand, contain extra (or fewer) electrons, and therefore would have fewer actual bonds connecting them in the adjacency matrix. Radicals exist when there are unpaired electrons associated with the atom. In the adjacency matrix, radicals are identified by an $a_{ii} \neq 0$.

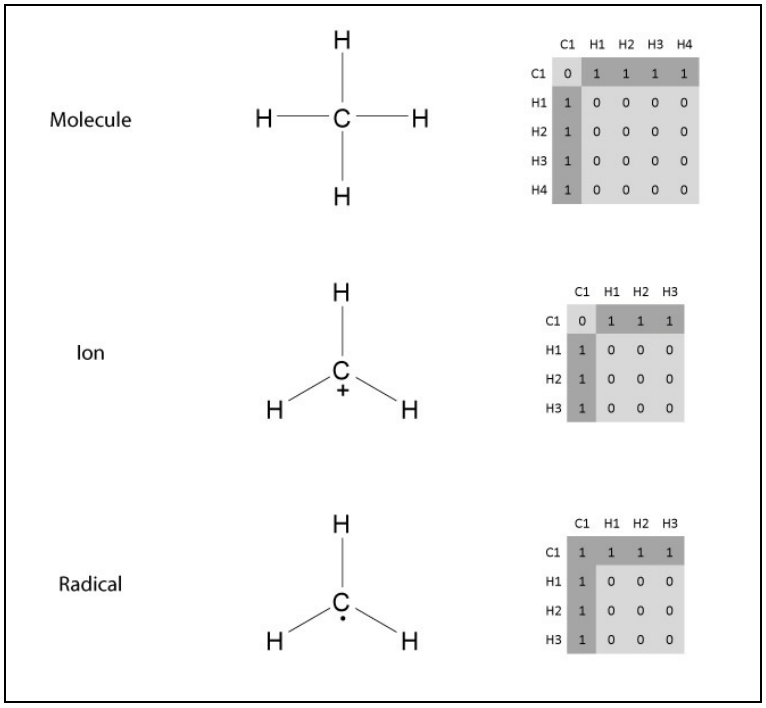


Figure 2.20 - Fundamental species types

Secondary (and even tertiary) species-type determination is then performed to break the larger species types into smaller, familiar chemical classes, such as iso and normal paraffins. Subtypes are determined by analyzing the species for the distinct characteristics of the class. Each atom is examined in reference to its edges, neighbors and cycle participation. When the defining characteristics of a class are broken, the algorithm excludes that type from consideration. For instance, a species with no cycles and at least one double bond between two carbons is classified as an olefin. A species with one six-carbon member ring, three alternating carbon-carbon double bonds in the ring, and any number of side chain atoms is considered an aromatic. Table 2.1 below lists the

molecular species types currently handled by INGen.

Table 2.1 – Secondary species type classifications for molecules

Hydrogen	H_2, HD, D_2
NParaffin	Contains only unbranched (normal) paraffins
IsoParaffin	Contains only branched (iso) paraffins
NOlefin	Contains only unbranched olefins
IsoOlefin	Contains only branched olefins
Nap5	Contains only 5-member naphthenic rings
Nap6	Contains only 6-member naphthenic rings
Nap56	Contains both 5-member and 6-member naphthenic rings
Aromatic	Contains one or more aromatic rings only
Nap5Arom	Contains both aromatic and 5-member naphthenic rings
Nap6Arom	Contains both aromatic and 6-member naphthenic rings
Nap56Arom	Contains aromatic, 5-member & 6-member napthenic rings
Sulfur	Contains a sulfur atom
Nitrogen	Contains a nitrogen atom

2.3 Species representation improvements specific to INGen

2.3.1 Creating adjacency lists

One of the key features of INGen is the ability for a user to quickly and efficiently model any given network. To this end, a large library of starting molecules has been included. Each species is a pre-created adjacency list ".dat" file and can be found in subdirectories based on its perceived molecular types (e.g. paraffin, olefin, aromatic, etc.).

Although many species are provided, not every starting species can be pre-determined. A user could create new adjacency list files by simply drawing out the molecule, labeling each atom numerically, and determining the connectivity. Such a procedure is time intensive, especially if numerous species must be created. Therefore, it was imperative to offer an alternative.

CambridgeSoft's commercial product ChemDraw is a well known CAD (computer aided design) type software for quickly drawing chemical species. If a modeler has a license for ChemDraw, creation of new adjacency lists becomes quite easy.

First the species is drawn in ChemDraw and saved as a CML (Chemical Markup Language) file. Second, the saved file is moved to the INGen converter's CML-In directory. Third, the DATGen converter is run. Finally, the newly formed .dat adjacency list file can be moved to the appropriate INGen adjacency list subdirectory.

2.3.2 Isomorphism detection for large species

The mechanism for species analysis remains similar to previous incarnations within NetGen, but certain corrections have been made for large molecules. Wei changed the cycle detection algorithm to accommodate species with more complicated ring structures, but support for large non-cyclic species was lacking. The problem presented itself as non-canonical codes were being created for Paraffins with more than 32 carbons. The problem was resolved by addressing the memory requirements used by the bitset definition of the bicompID. Each chained articulation point was receiving a unique bicompID, and as such, the availability of new IDs was running out for large non-cyclic molecules.

2.3.3 Memory management corrections

In addition, large-scale networks with large species were requiring too much memory space and eventually crashing out. This was solved in a twofold manner. First off, such memory problems are usually due to a memory leak. Leaks occur when memory space gets assigned to a specific variable, but that space is not deallocated when the variable is no longer used. In a highly repetitive program like INGen, such leaks can quickly add up.

The memory leak was due to original NetGen design wherein lists were not being deconstructed because of other memory problems. These cascading memory problems were reworked, and the primary memory leak was plugged.

The second solution was the JIT (just in time) creation of the matrices. As stated previously, the matrix representation is a waste of space due to its sparse nature, but NetGen still needed matrix representations in order to apply the row and column permutations as well as the reaction matrix. By creating the matrix only for the purpose of applying the reaction subroutine, it could be eliminated from memory immediately following reaction. When dealing with thousands of large species, such a plan saves a great deal of persistent memory.

2.3.4 Proposed changes to the biconnected component search

A new depth-first search algorithm was proposed for biconnected component detection wherein only the maximal biconnected components would be identified. Such a method also required a separate breadth-first search cycle detection algorithm to determine the minimal biconnected components within the maximal. Whereas these methods worked in themselves, the canonical ordering algorithm required information that was not expressly

available, and would need to be rewritten. Thus, the work was archived for later exploration.

2.3.5 Cascading species type definitions

Finally, INGen has combined separate model builders for mechanistic and pathways level modeling. As such, new routines were created that first determine if a species is a molecule, an ion, or a radical; the logic for which was previously described. Further analysis creates the subtypes of species. Previous versions of the model building software did not need to account for all three major types, nor concurrent subdivisions.

2.3.6 Summary of changes

All in all, the changes to the species representation will likely be transparent to the user but were quite important. The ability to create species with ChemDraw and almost immediately use it within a model is an outstanding feature that had been missing from previous versions of the NetGen software. In addition, computers have gotten faster and have more memory than in the days of NetGen, and because of this, INGen handles larger species and larger networks than were initially envisioned. Thus, INGen has made significant strides in improving the species representation and identification routines of the NetGen core.

2.4 Conceptual inter-conversion of species

2.4.1 Background

When molecules undergo chemical reaction there is a conservation of mass, and more

relevantly, a conservation of atoms. As a demonstrative example, the simple case of pathways level isomerization is considered. Isomerization is the reshaping of a chemical species while maintaining a conservation of connected atoms. The resultant product molecule will contain the same number and identity of atoms as the initial reactant molecule, but the bond distribution will have changed. In the adjacency matrix view, the a_{ij} values for certain *i*s and *j*s will change, but not the *i*s and *j*s themselves.

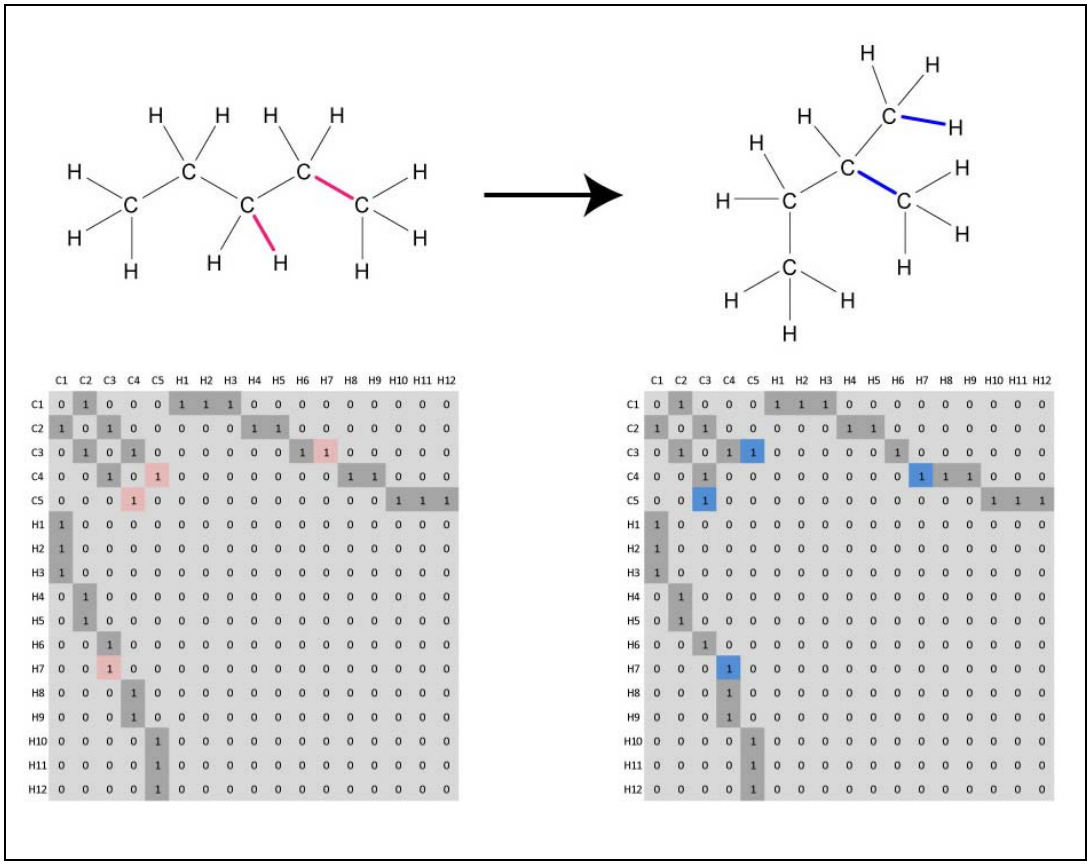


Figure 2.21 - Pentane isomerization

2.4.2 Reaction matrices

In the pentane isomerization example, two bonds are broken and two bonds are created. This reaction could be captured as a transformation of the initial matrix into the resultant

matrix, This “reaction matrix” would use a_{ij} values of “-1” to represent the breaking of a bond, and “1” values to represent the making of a bond. When the reaction matrix is added to the reactant matrix, the product matrix would form. Unfortunately, the reaction matrix would apply only to those specific *i* and *j* bonds. If the isomerization were to take place with different carbons and hydrogens than the example (which is likely), the transformation matrix would need to be different.

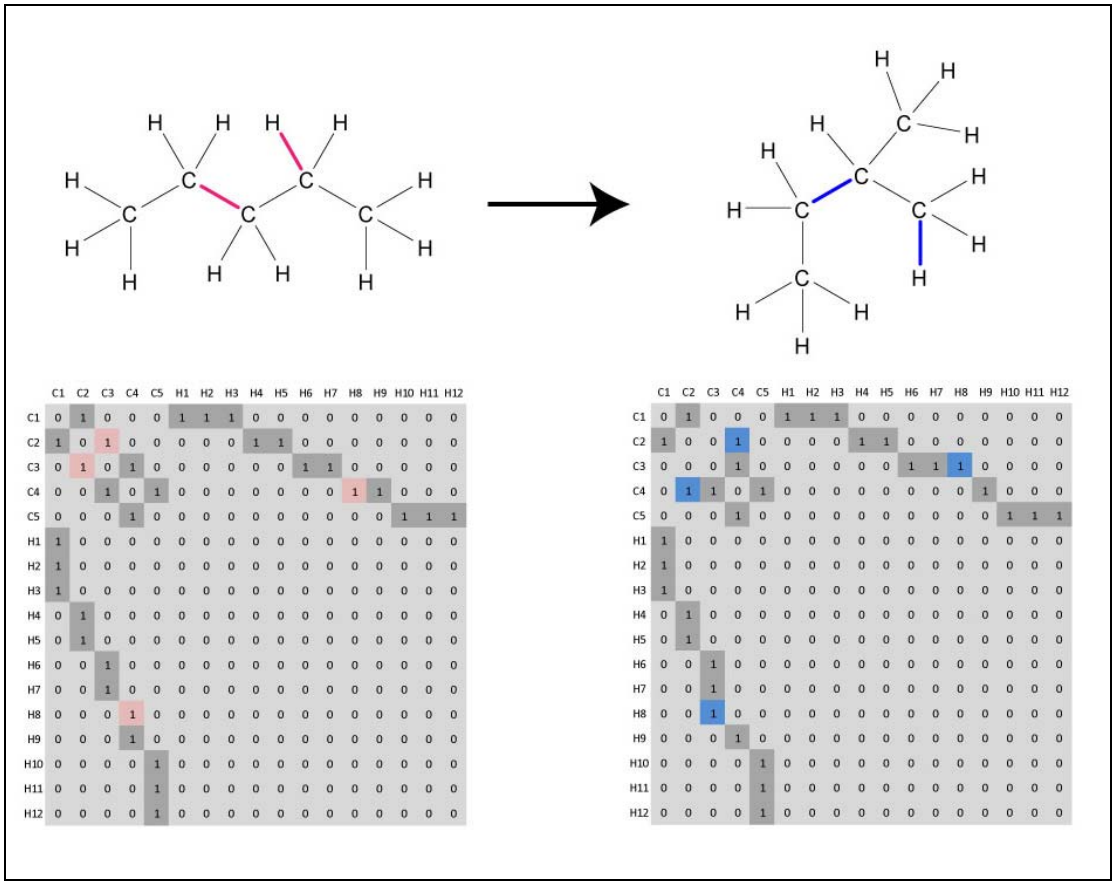


Figure 2.22 - Pentane isomerization at a different site

Fundamentally the operation is the same. Two bonds are broken, and two bonds are created. Therefore, the row and column permutation operations of linear algebra can be used to put the two sets of affected bonds (the four “active” atoms) in the upper left part of the matrix.

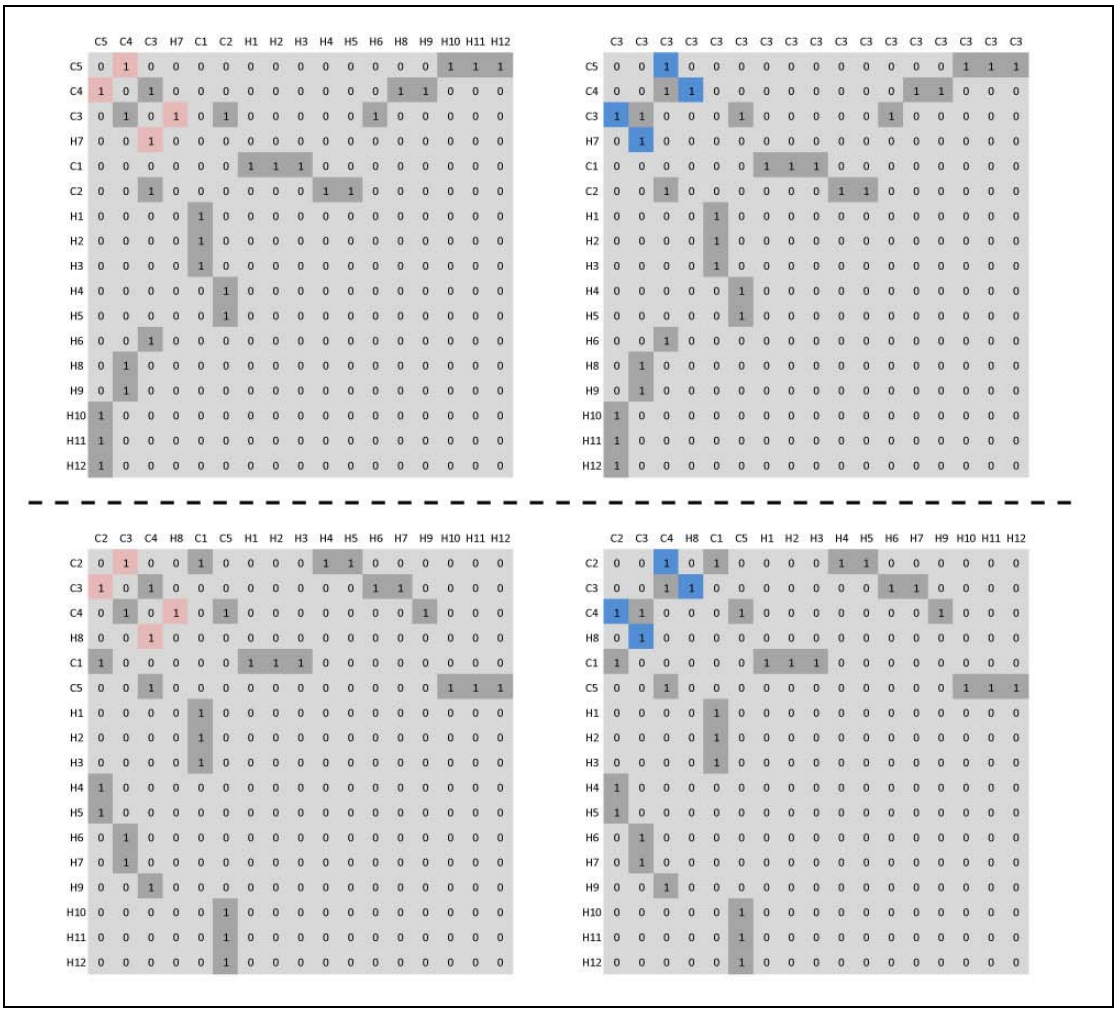


Figure 2.23 - Permuted matrices for pentane isomerization at two different sites

Now, only a single reaction matrix is required to represent both reactions. Because the rest of the reaction matrix is static, only the four active atoms need to be stored in a data structure. This simplification also allows this reaction matrix to work for any isomerization reaction regardless of carbon number. Only the four active atoms will be changed.

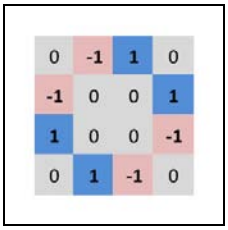


Figure 2.24 - Isomerization reaction matrix

2.4.3 Pre-reaction analysis

Now the problem of reaction lies in determining which rows and columns of the matrix should be permuted to the top left corner. This "site selection" is performed by searching the molecular structure for a series of atoms that follow a given set of rules. In the pentane isomerization example, reaction can occur wherever a hydrogen is connected to the end of a chain of three carbons.

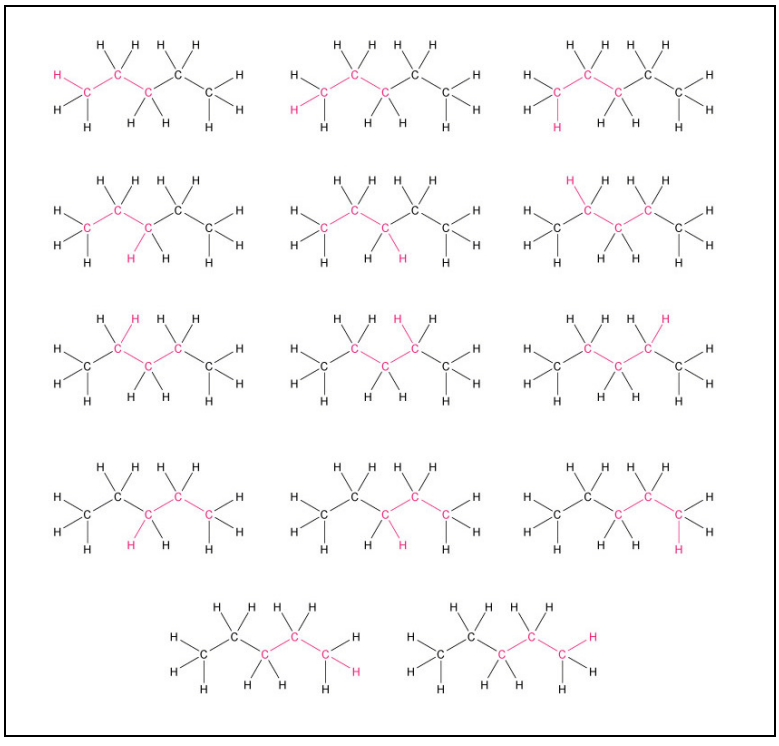


Figure 2.25 - Isomerization site selection

The above example has quite a number of possible sites, but it is important to realize that all of those sites will produce the exact same product. The atomic numbering and *i/j* labeling scheme is artificial, and as such, only the type of atom and its connectivity will produce unique species. The limiting of products and reactions will be covered later.

When performing the site selection routine, the identities of the atoms (the *i*/*j* value from the matrix representation) are recorded in a site data structure. Multiple sites can occur within a species, so the molecule is searched extensively and a list of all the site data structures is passed back to the program. In the isomerization example, the routine searches for a (C-C-C-H) chain, which is the same as looking for a (C-C-C) chain with a hydrogen tacked on to either end. The (C-C-C) chain is similarly nothing more than a (C-C) chain with a carbon tacked on to either end. And finally, the (C-C) is simply a carbon with another carbon attached to it. Such a recursive strategy is employed for each of the atoms in the species.

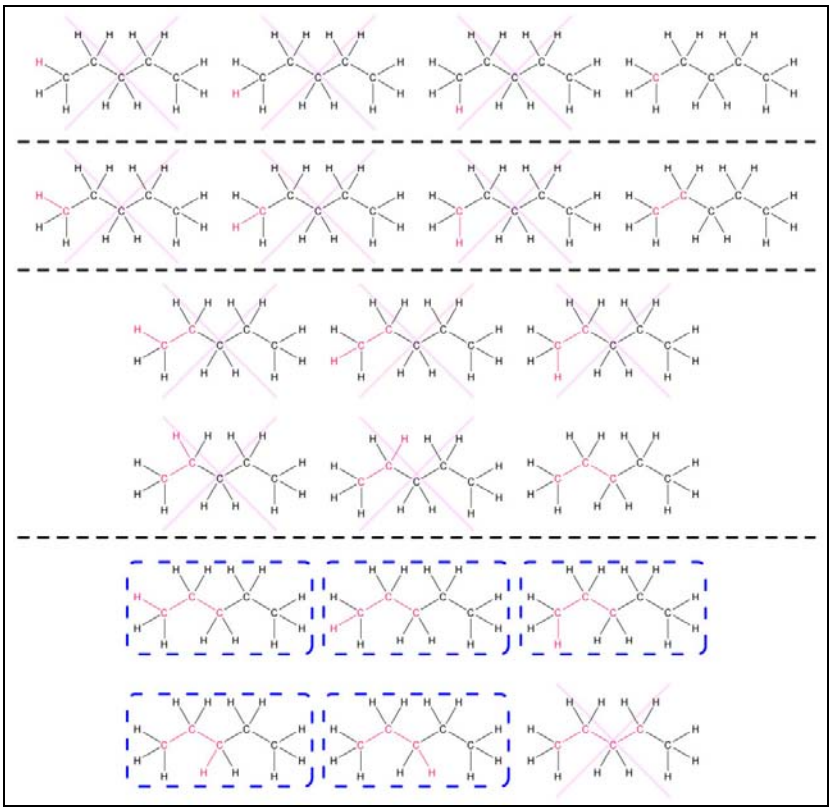


Figure 2.26 - Recursive site selection (partial)

After the creation of the list of site data structures via connectivity analysis, each site chain undergoes further rules analysis to determine unwanted sites. First and foremost, the bond order between the atoms is evaluated, and any sites that do not match the bond order criteria are discarded. As an example, consider two sites returned by applying the above isomerization site selection connectivity requirements to an olefin. Although both sites are returned, only the second site passes the bond order test.

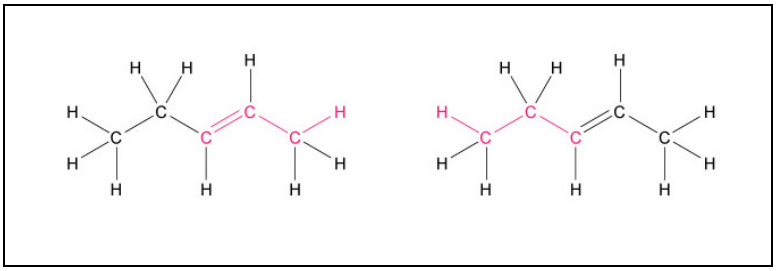


Figure 2.27 – Two isomerization site selection connectivity test results for an olefin

Next, all site atoms are checked versus cyclic requirements. For the isomerization example, no atom should belong to a cycle. If a reaction would not occur with atoms from a ring, then any site data structure with an atom that tests positive for a CycleID is removed from the list of sites. If ring atoms are allowed, sometimes it is necessary that all the atoms belong to the same ring. The bitwise "And" operator is applied to the CycleID of all atoms that should belong to the same ring. If the test returns false, the site date structure is removed from the list of sites.

Other examples of pre-reaction site selection paring will come in the next section of this chapter where the new paradigm for site selection is discussed. For now, we turn back towards the actual reaction.

2.4.4 Post-reaction analysis

Given a list of site data structures, the reaction function will permute the matrix appropriately for each site, and then apply the reaction matrix. The resultant product matrix must then undergo an identification procedure. If the product has been seen before, it is labeled with the appropriate SpeciesID number. Otherwise, it is given the next available new SpecisID number, and that product is added to the list of unreacted components.

After the products have been identified, the overall reaction needs to be checked for duplication. It is possible that the same overall reaction had occurred on a different site (such was the case for the isomerization of the pentane). In addition, when appropriate, the reverse of a reaction is automatically created. Duplication can occur if a forward reaction matches the reverse of a previously determined reaction. If the reaction is unique,

it is logged along with its reaction class (the type of reaction or reaction matrix applied).

2.4.5 Handling multi-molecular reactions

Up until this point, a simplistic mono-molecular example has been used, but the concepts apply to multi-molecular reactions as well. Consider a pathways cracking reaction for example. Reaction begins with a single molecule as before, but two product molecules will be formed. After permuting the reactant matrix, and applying the reaction matrix, a single product matrix remains. The key to analysis lies in the disconnectedness of the two graphs described by the one product matrix. The product identification algorithms will be able to determine the presence of two disconnected graphs when the number of atoms in the first product does not equal the number of columns and rows, *i*/*j*.

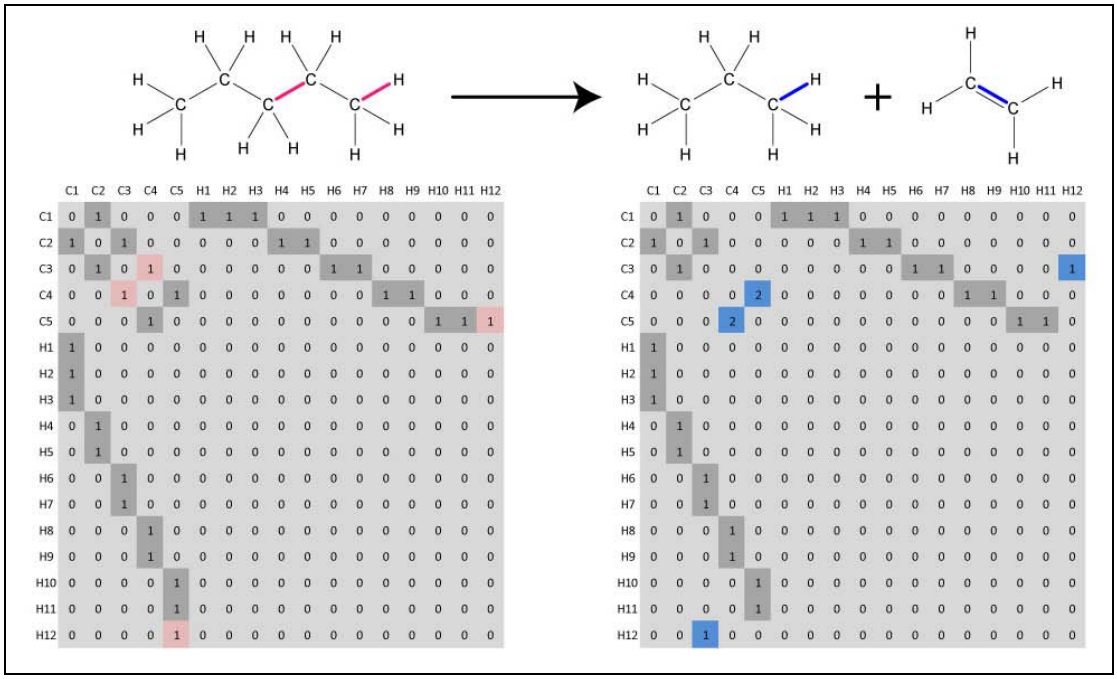


Figure 2.28 – An example of a multi-product reaction

By the same token, multiple reactants can be combined into a single augmented reactant matrix. When dealing with a multi-reactant system, the site data structure will house atomic identification information for both reactants. Because the reaction matrix will be set for a specific ordering of reactant atoms, the site data structure must follow the ordering requirements before permuting the columns and rows.

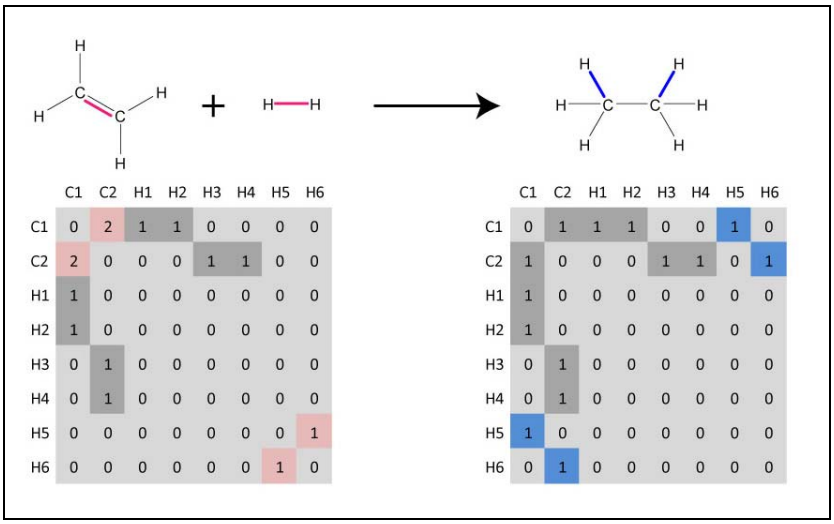


Figure 2.29 – An example of a multi-reactant reaction

2.5 INGen's approach to reactions

2.5.1 Underlying philosophy of reaction

The creation of INGen incorporated two key ideals into the conversion of molecular species via reaction. The first concept was that of the universal model builder, in which multiple chemistries can be applied within a single program execution. Secondly, INGen includes a major revision to the method of site selection that allows for more robust and accurate modeling.

As stated before, multiple versions of the NetGen model building software were designed

for different chemistries. The rules and reaction families of one chemistry would differ from those of another, and as such, were coded separately. In attempting to build user-friendly model building software, it was apparent that these restrictions should leave the back end world and become user controllable. By removing hard-coded, always-on rules, the structure for a single “universal” model builder was created.

The ultimate goal of a universal model builder is to be able to handle any type of reaction that the user sees fit. As a tangible intermediate goal, INGen set out to provide the acid, metal and thermal based catalytic chemistries around which the hydrocarbon and petroleum industries rely.

2.5.2 Acid chemistry

Acid catalysis occurs when the reactant species comes in contact with an available proton, signified as H^+. The proton, being highly elecrophilic, can attack dense electron clouds of the reactant species, usually in the form of a double bond. One of the bonds will break, transferring its electrons into a bond with the proton. Because of this, the reactant atom to which the hydrogen is not attached, but was originally part of the double bond, will now carry the positive charge. Assuming that the newly charged atom is a carbon, we call this a carbenium ion.

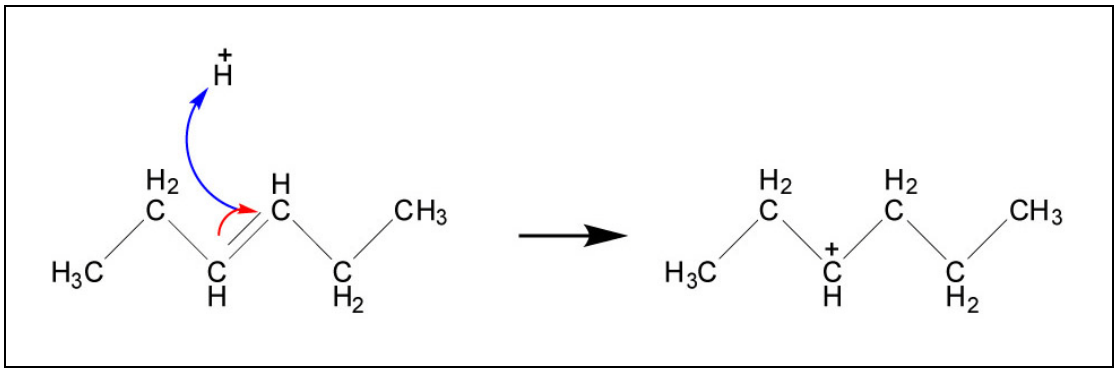


Figure 2.30 – An example of carbenium ion formation

There are four classifications of alkyl carbenium ions. The tertiary state is lowest in energy and consists of a carbon with three non-hydrogen atoms bonded to it. The secondary is next lowest in energy, with only two non-hydrogen atoms (and one hydrogen) bonded to the ionic carbon. These two states are found in most ionic catalysis, but the last two have high energy demands and are more uncommon. Primary carbenium ions occur when a carbon is only connected to one other non-hydrogen atom (and two hydrogens). The highest energy state is the Methyl carbenium ion where a carbon ion is connected to only three hydrogen atoms.

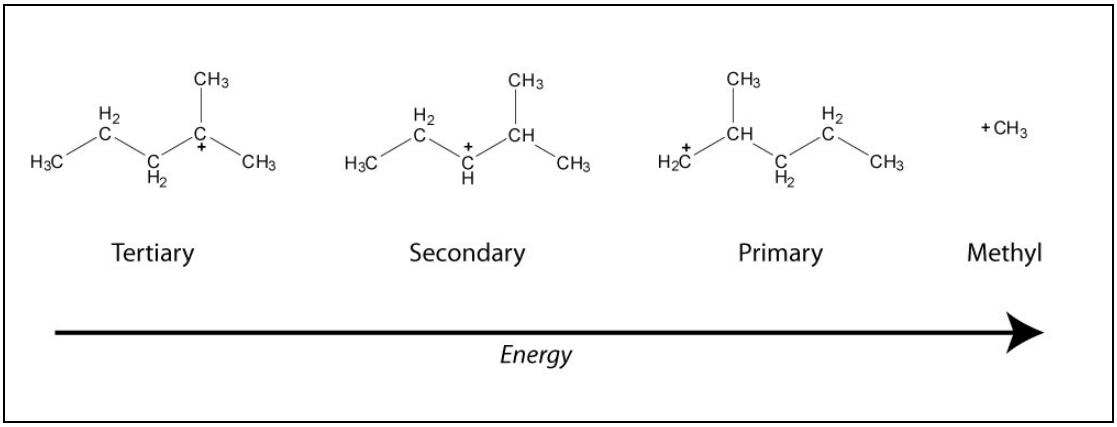


Figure 2.31 - Carbenium ion types

Carbenium ions drive further reaction. Electrons from other bonds within the species can be sapped to help fill the carbenium ion's open orbital. In a hydride shift reaction, the electrons that bond a hydrogen to a neighboring carbon can lose their affinity for that carbon, and shift to bond the hydrogen to what was the carbenium ion. Doing such leaves the originally bonded carbon as a new carbenium ion. Such a reaction is likely if the new carbenium ion is of a lower energy state. Similarly, methyl branches have been shown to also undergo a similar shift

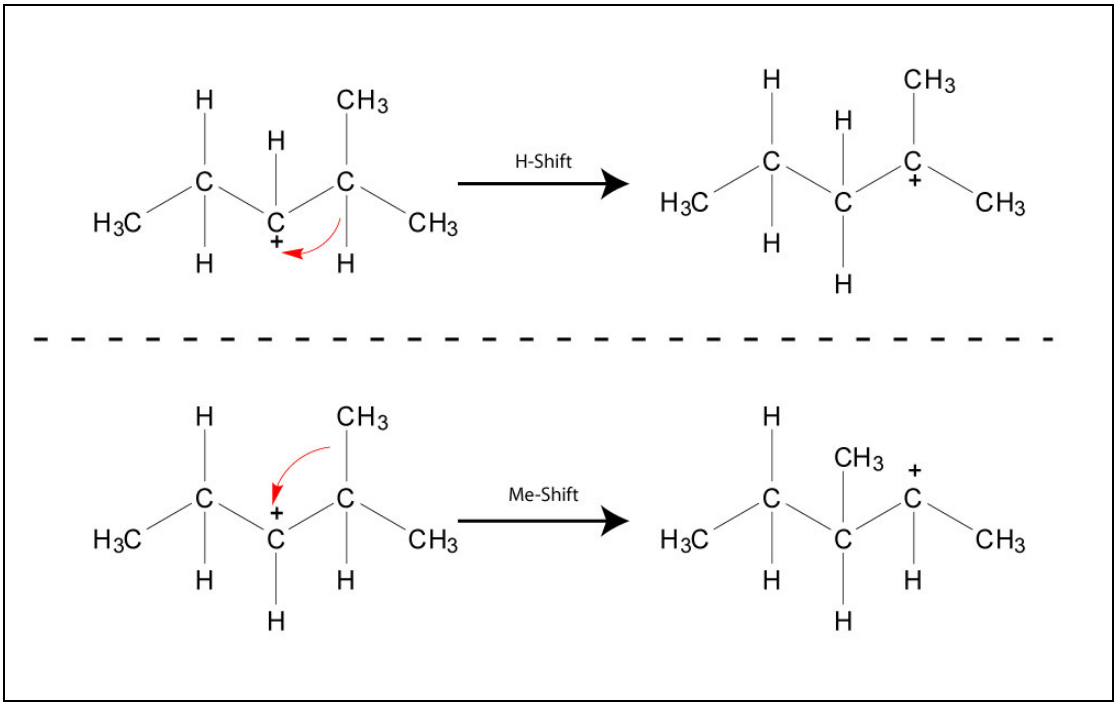


Figure 2.32 – Examples of hydride and methyl shift reactions

Another important reaction occurs when the hydrogen of a carbon that neighbors a neighboring carbon interacts with carbenium ion. Rather than simply shift the hydrogen, a bond is created between the two carbons. The electron density of the cyclopropane can create a tentative bond with the hydrogen's proton, thus creating a a protonated cyclopropane (PCP) [1]. The PCP is only metastable, and likely to return to a carbenium ion state. Each of the bonds between the three carbon atoms can break, rebonding the hydrogen to a single carbon, and thus recreating a carbenium ion.

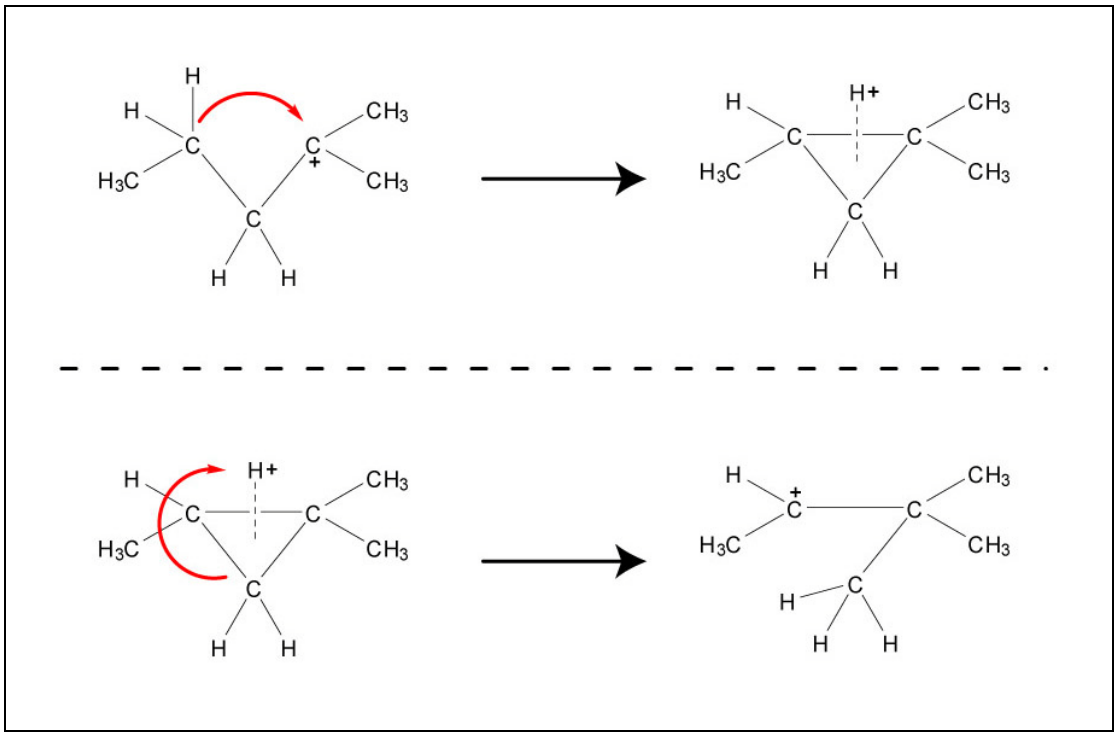


Figure 2.33 – An example of PCP formation and breaking

Each bond of the PCP can break in two ways, putting the hydrogen on one of the two carbons and the charge on the other. One of the six breaking paths will return the molecule to its pre-PCP state. Any restructuring of the species from hydride and methyl shifting or PCP formulation is called isomerization.

In addition to isomerization, carbenium ions can break (or crack) a molecule into two via a process known as β-scission. Whereas the carbenium ion attacks the bond between a carbon neighbor's neighboring carbon and its hydrogen in PCP formation, β-scission involves the bond between the carbon neighbor and its carbon neighbor. Doing such severs the bond between those carbons, and creates a new double bond between the neighboring carbon and the erstwhile carbenium ion. The charge is passed to the carbon neighbor's previously neighboring carbon, thereby creating a new carbenium ion.

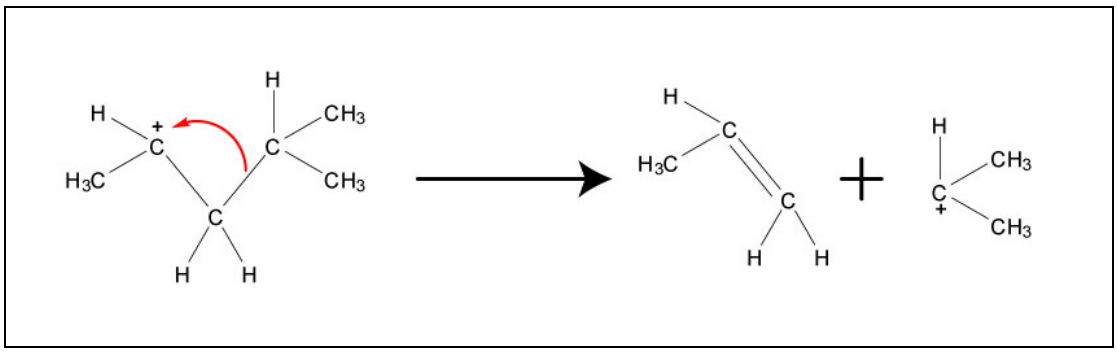


Figure 2.34 – An example of β-scission

Tracking the state of the carbenium ion is the key to understanding the kinetics and rate of the reaction. Pathways that necessitate high energy carbenium ions proceed slower than those with low energy carbenium ions.

Ultimately, the carbenium ions are only intermediates. If the carbenium ion attacks the bond between a neighbor and its hydrogen, and forms a bond with the carbon not the hydrogen, and double bond is formed between the original carbon and its neighboring carbon. The acidic proton is returned to solution, capable of catalyzing another series of carbenium ion reactions.

2.5.3 Metal chemistry

Metal based chemistries are those that capitalize on the transition states of metal atoms to form and break bonds. Metal catalysis differs from acid catalysis in that the metal can hold more charge than a single proton. Theory suggests that the charge remains in metal's transition state, never transferring to the attached hydrocarbon species. Therefore the metal drives all the bond breaking and rearrangement reactions. After the hydrocarbon is initially bonded to the metal, other atoms of the hydrocarbon also become bonded to the metal as it changes transition state.

Paraffins can undergo deyhdrogenation over an oxide supported metal catalyst. Dehydrogenation is the process of creating a double bond between two adjacent carbons by removing their respective hydrogen atoms. Dehydrogenation can begin with the absorption of a paraffin to a metal catalyst [2].

Theory suggests that electrons from a hydrogen-carbon bond are drawn into a new metal-carbon bond, while electrons from the oxide support form an oxygen-hydrogen bond with the freed hydrogen. Next, the hydrogen of a carbon adjacent to the metal-bonded carbon bonds with the metal, while the electrons of the metal-carbon bond are shifted to form a double bond between the two carbons. The hydrocarbon is no longer attached to the metal and is free to enter solution. Finally, a bond is formed between the hydrogen atom attached to the substrate and the hydrogen attached to the metal, thus releasing H_2 and regenerating the metal catalyst.

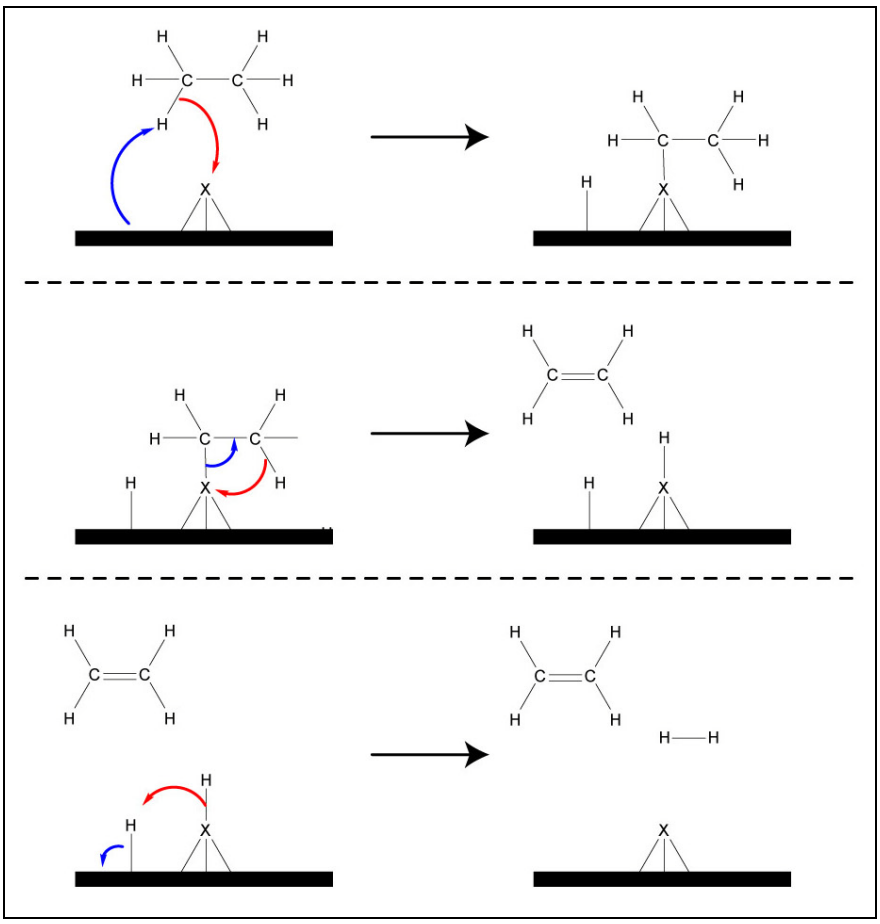


Figure 2.35 – An example of dehydrogenation over metal

Hydrogenation (the reverse of dehydrogenation) can also take place over a metal catalyst. Hydrogen, H_2, in solution can become attached to the metal and the oxide support in the reverse of its release in the dehydrogenation mechanism. The electron density of the double bond absorbs the olefin to the metal. The electrons transfer from the metal-hydrogen bond into a carbon-hydrogen bond while the carbon-carbon double bond breaks attaching the other carbon to the metal. Finally, the electrons bonding the hydrocarbon to the metal transfer to the adjacent oxide bonded hydrogen atom, thereby releasing the paraffin and regenerating the catalyst.

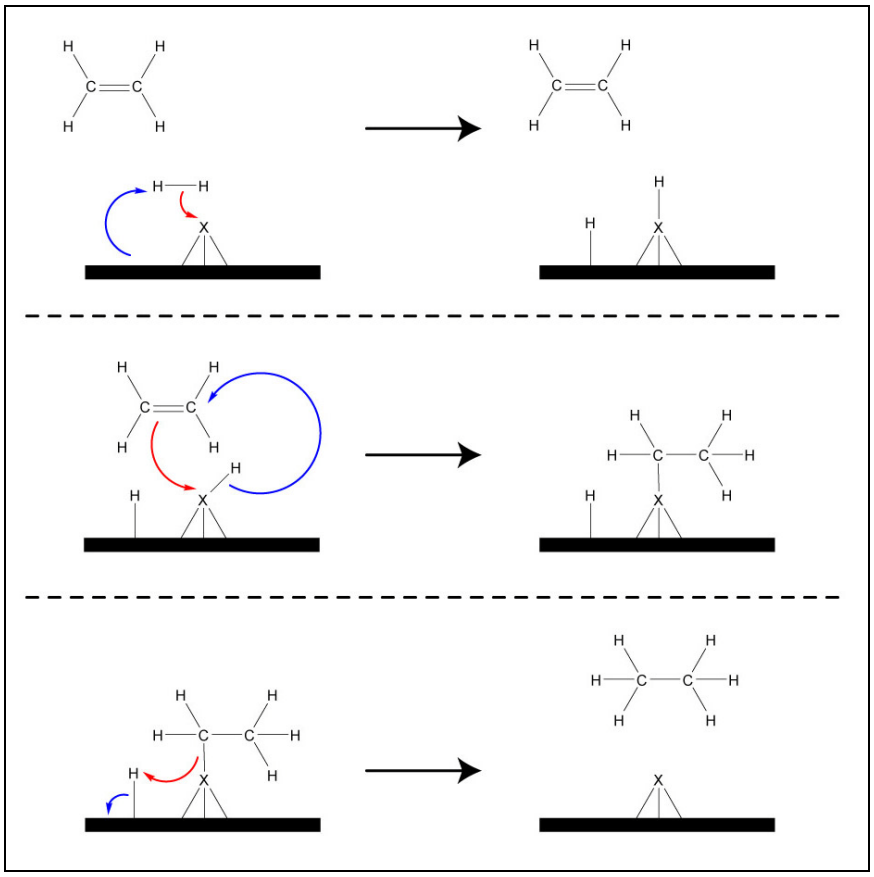


Figure 2.36 – An example of hydrogenation over metal

Another important hydrocarbon reaction that takes place over a metal catalyst is hydrogenolysis. In the presence of a metal catalyst, elemental hydrogen, H_2, can break single bonds within hydrocarbons. Theoretically, elemental hydrogen, H_2, bonds to the metal complex as before in the hydrogenation mechanism. One of the carbons of the hydrocarbon adsorbs to a free metal site, releasing a hydrogen atom to a secondary oxide site on the catalyst. An adjacent carbon atom also bonds to a different nearby metal site, also losing its hydrogen atom to the substrate. The electron density of the carbon-carbon bond then gets pulled away by a third metal site thereby creating a new carbon-metal bond on one of the carbons. At the same time, one of the nearby hydrogens rebonds to the singly metal-bonded carbon. The carbon-carbon bond is now split, and desorption takes place with locally attached hydrogen atoms.

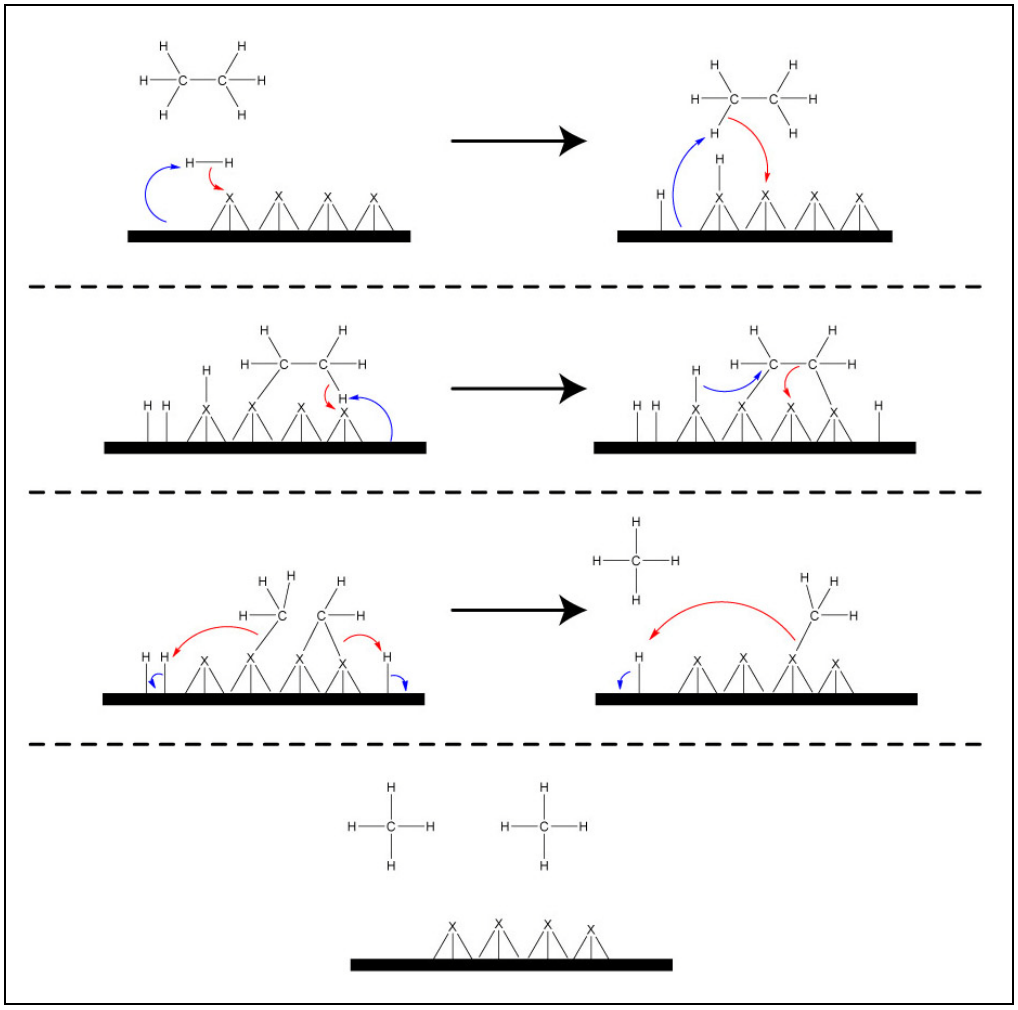


Figure 2.37 – An example of hydrogenolysis over metal

2.5.4 Free radical chemistry

Finally, free radical chemistry occurs in a high temperature environment where enough energy exists to split the bond energy between two atoms in a step known as bond fission. Each atom keeps one of the two bond electrons and is known as a radical. The unmatched electrons in the radical's bonding orbital make the species highly reactive.

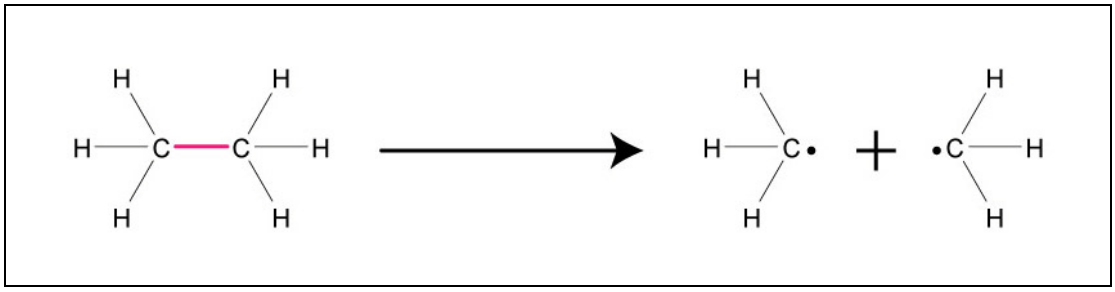


Figure 2.38 - An example of bond fission

Radical reaction continues over a series of hydrogen abstraction, β-scission, and radical addition steps. Hydrogen abstraction takes place when the radical attacks the electrons that bond a hydrogen atom to another atom. The hydrogen forms a bond with the attacking radical atom. Its previous neighbor is left with a unpaired electron, thereby conserving the radical on the attacked species.

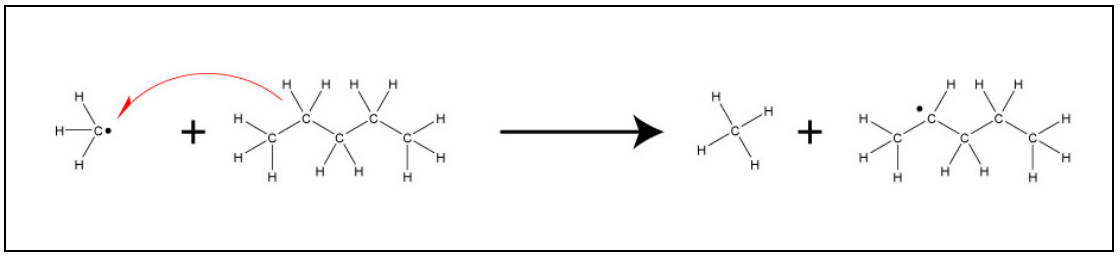


Figure 2.39 - An example of hydrogen abstraction via free radical chemistry

Radicals can undergo β-scission similar to the ionic β-scission discussed previously. The bond "beta" to the radical is stripped of an electron, thereby leaving two fragments. One fragment would have a new double bond, the other would maintain the radical.

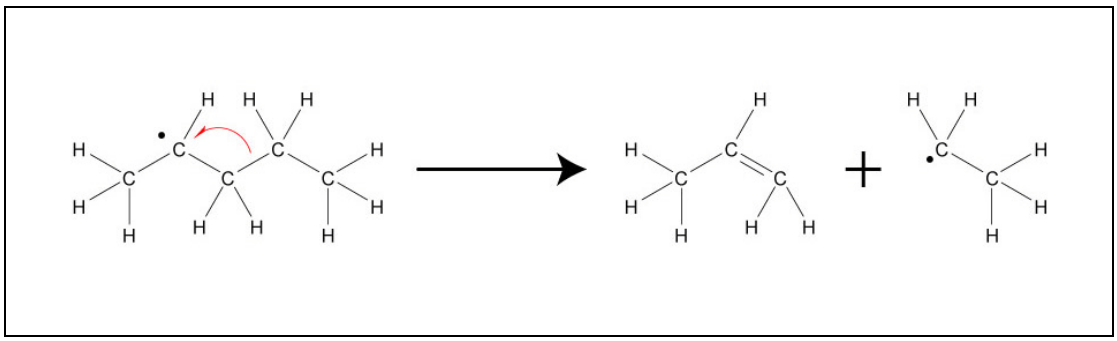


Figure 2.40 - An example of β-scission via radical chemistry

Radical species can also perform addition type reactions to olefins. The radical steals an electron from the olefin's double bond, thereby forming a new bond from the radical species to one of the previously double bonded carbons. The other carbon is left with an unpaired electron thereby maintaining the radical.

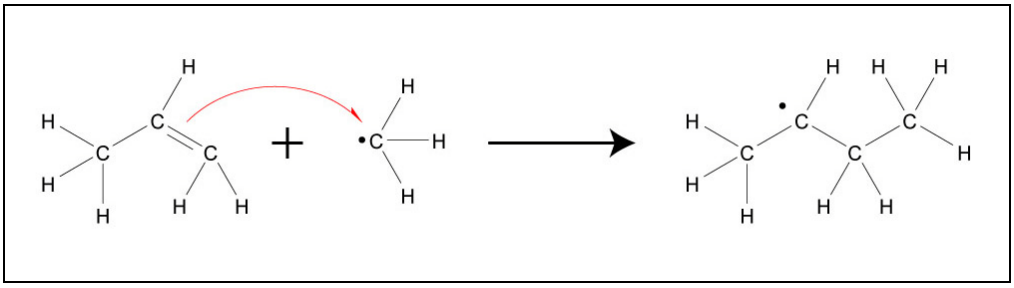


Figure 2.41 - An example of addition via radical chemistry

Finally, when two radicals are able to find one another they can reform a single bond between them, thereby removing the radical nature from both. This type of reaction is known as recombination.

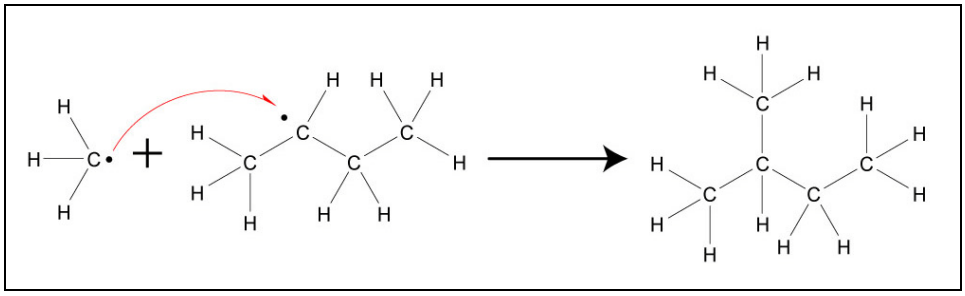


Figure 2.42 - An example of radical recombination

2.5.5 Process chemistry modeling

These three classes of chemistries represent the bulk of hydrocarbon reactions prevalent in industrial processes. As such, INGen attempts to fuse the specific reactions within each of the three types into a single package. By selecting the appropriate reactions, any industrial hydrocarbon process can be modeled. Table 2.2 below lists example processes which can be modeled using INGen

Table 2.2 - Process modeling examples

Thermal	***Oxidation***
C2, C3, C4 (olefins)	C1-C4 Alcohols in SCW
Naphtha	Naphtha (Octane Number)
Gas Oil	Cyclohexane
Resid (Visbreaking, Coking)	Lube Oil
Asphaltene	
Acid Cracking	***Hydro-processing***
Pure Components (olefins)	Naptha HDS
Naphtha	HydroIsomerization
Gas Oil (FCC)	
Hydrocracking	***Catalytic Reforming***

Gas Oil
Resid
Alkane Dehydrogenation

2.6 Mechanistic and pathways modeling with INGen

There are two major categories of reaction models: mechanistic and pathways. Mechanistic level modeling is a lean pure science that accounts for intermediates where each reaction represents a singular elementary action. Pathways level modeling is a hearty engineering science in that much knowledge and many elementary steps are packed into a single equation or reaction. INGen combines the two philosophies into a single shell so that advanced heterogeneous modeling techniques can be accomplished.

Mechanistically, INGen is set to handle a number of ionic, molecular, and radical stepwise reactions [3-10]. Table 2.3 below lists the currently implemented reaction steps housed within the INGen source code.

Table 2.3 - Mechanistic reactions

Free Radical Reactions	***Ion Reactions***	***Molecular Reactions***
Bond fission	Isomerization	Hydrogenation
Hydrogen Abstraction	Hydride Shift	Dehydrogenation
β-scission	Methyl Shift	Hydrogenolysis
Addition	β-scission	
Termination	Hydrogen Abstraction	
	Protonation/Deprotonation	
	5-Member Ring Closure	
	Ring Expansion	
	Addition	

In addition to the mechanistic reaction models, INGen also handles a number of pathways

level reaction families. Below, in Tables 2.4-2.7, the implemented reaction families are listed along with the species types to which they apply.

Table 2.4 - Pathways reaction families for species with no ring types

No Ring Types	
Paraffin	Isomerization Cyclization Hydrogenolysis Cracking
IsoParaffin	Isomerization Cyclization Hydrogenolysis Cracking
Olefin	Hydrogenation Double Bond Shift
IsoOlefin	Hydrogenation Double Bond Shift
Sulfur	Hydrodesulfurization Sulfur Saturation
Nitrogen	Denitrogenation

Table 2.5 - Pathways reaction families for species with only one type of ring

1 Ring Type	
Aromatic	2H-Saturation 4H-Saturation 6H-Saturation Side Chain Hydrogenation Dealkylation Side Chain Cracking Ring Closure
5-Member Naphtha	Ring Opening Ring Dealkylation Side Chain Cracking Ring Closure

6-Member Naphtha	Hydrogenation Ring Opening Ring Isomerization Ring Dealkylation Side Chain Cracking Ring Closure

Table 2.6 - Pathways reaction families for species with two types of rings

2 Ring Types	
5-Member Naphtha and Aromatic	2H-Saturation 4H-Saturation 6H-Saturation Ring Opening Dealkylation Side Chain Cracking Ring Closure
6-Member Naphtha and Aromatic	2H-Saturation 4H-Saturation 6H-Saturation Ring Isomerization Dealkylation Side Chain Cracking Ring Closure
5-Member Naphtha and 6-Member Naphtha	Ring Opening Ring Isomerization Ring Dealkylation Side Chain Cracking Ring Closure

Table 2.7 - Pathways reaction families for species with three types of rings

3 Ring Types	
5-Member Naphtha, 6-Member Naphtha and Aromatic	2H-Saturation 4H-Saturation 6H-Saturation Ring Opening Ring Isomerization Dealkylation Side Chain Cracking Ring Closure

INGen allows the mixing of both mechanistic and pathways level models. Each species type is given a list of reaction types through which it can occur. Each combination can be turned on or off, thereby allowing the user to control the reaction level specifics of how the network should progress. In addition, the carbon ranges to which each applies can be set by the user, thereby allowing specific management between mechanistic and pathways level modeling (as well as acting as a method of network size control).

2.7 Network control

2.7.1 Background

The final necessity for network generation software is the inclusion of a set of controls that will help to limit either the number of reactions or the number of species in the network. As previously stated in Chapter 1, real world problems can have thousands of species, but it is not always appropriate to model each and every one. Not only is there a tradeoff between model resolution time and model completeness, but there can also be a disparity between model size and the availability of data.

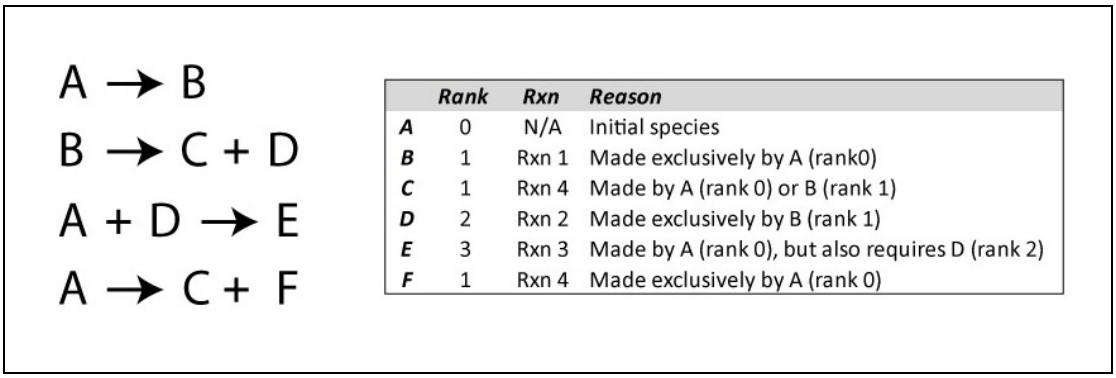
The general trend of lumping based models is to build a few discrete lumps that will represent large classes of molecules. Such methods have been shown to work well, but

fundamentally there is a loss of information. The lumps are chose because of their ability to recreate a wanted property, but as such, they may not be capable of accurately producing a property that may be wanted in the future. By using molecular modeling, such unknown properties can be aggregated after the fact with no changes to the model.

Although molecular modeling can make use of LFER type relationships to narrow the number of parameters tied to a model between carbon numbers, there are still a number of isomers for any given carbon number [11]. The LFER proposal states that there is a linear relationship between the thermodynamic properties of species within a certain reaction family, and the carbon number of said species. So, one key of molecular modeling is to curtail unnecessary isomers from forming. This goal can be accomplished by limiting the number of similar type reactions from occurring. For instance, only allowing ten isomerization reactions at any given carbon number will greatly cut down on the number of isomers at large carbon numbers.

Another method of isomer limitation involves only allowing a maximum number of branches in any given product species. By limiting such systems to 0 branches, only non-branched rings and normal type paraffins would be produced via reaction. Such a restriction is harsh, but would produce a much smaller modeling footprint. Depending on the conditions of the physical system, modeling networks may produce better results with maximum branching set between one and three.

Another class of model limitation is rank limitation. Rank, or Delplot rank, is the number of steps a species is away from an initial species [12].



	Rank	Rxn	Reason
A	0	N/A	Initial species
B	1	Rxn 1	Made exclusively by A (rank0)
C	1	Rxn 4	Made by A (rank 0) or B (rank 1)
D	2	Rxn 2	Made exclusively by B (rank 1)
E	3	Rxn 3	Made by A (rank 0), but also requires D (rank 2)
F	1	Rxn 4	Made exclusively by A (rank 0)

Figure 2.43 - Rank calculation example

By limiting the rank of a formed product, the reaction network can be controlled by curtailing reactions that rely on previous reaction steps. This leads directly into the concept of "advanced seeding". Seeding is the selection of the beginning molecular species which will be tested for reaction. The seed selection needs not mirror the input to the actual physical process. First off, if a molecule in the physical feed can be created via reaction according to the model builder's rules, than that species does not need to be seeded; it will be created.

More importantly, a species that does not exist in the physical feed may be included in the model building seed. By doing such, the modeler imposes new initial species from which the rank limitations can take hold. Joshi et al showed that for a rank-9 limited n-Heptane network, the number of species exploded to 5000 [13]. By seeding a few appropriate midrange isomers and limiting the network to rank-3, the network contained only 500 species. After tuning the kinetic parameters to available data, the two models provided similar good results.

Another important result of rank limitation and advanced seeding is the rank-0 network. A rank 0 network will not produce any species that is not initially seeded. In this manner, if the modeler already knows specifically which species are found in the product stream,

those species can be seeded. The model builder will simply draw the arrows between the species.

2.7.2 INGen's philosophy on reaction limitation

INGen incorporates all the network limiting techniques described above, but changes the scope from the entire network to specific species type/reaction type combinations. NetGen used rank, carbon range, and branch count as overall limiting factors, but such limiting was too coarse for certain reaction networks. By separating the limiting rules into separate species/reaction combinations, certain aspects of a reaction network can be concentrated upon.

Another novel network limitation approach within INGen is the specific limiting of the number of reactions that can occur. These numeric limitations are set for each species/reaction combination, and they apply a maximum to the number of reactions that occur for a given carbon number. In addition, in order to create a duality of precision, a "carbon cut" number can be established.

The carbon cut is a specific carbon number under which the reaction counts apply to specific species/reaction combinations, carbon number, and branch count. Therefore, more reactions are allowed for the carbon number in total, but it spreads them out amongst isomers with different numbers of branches. Any species above the carbon cut does not consider the number of branches.

The carbon cut can also be used in conjunction with reaction specific rules. One such is the limitation of the generic cracking routine which states: above a certain carbon number, cracking may only take place in the middle of the carbon chain.

By increasing the ability to control specific reactions, INGen greatly adds to the adaptability of the model builder. A chemically inclined user can sculpt the appropriate network within a few tries.

2.7.3 INGen's implementation of reaction controllability

In implementing the reaction/species combination controllability, INGen changes the logic of the model builder. After species identification, each reaction type is checked for "reactability" for the specific molecule.

The first step in the reactability check is whether the species/reaction combination has been allowed by the user. If not, the reaction type is skipped and the next reaction type is considered. Otherwise, the second step ensures that the species falls within the specified carbon range. If no range is given, it is assumed that range encompasses all carbon numbers. Like before, if the test fails, the reaction is no longer considered. Similarly if any subsequent test fails it will undergo the same treatment.

The third step in determining the reactability of the species/reaction combination is ensuring that the branch count of the species does not exceed the given maximum branch count. If no maximum is given, any number of branches are allowed.

The fourth and final step is determining whether the reaction count has already matched or exceeded the maximum allowed. The check follows the logic specified in the previous section. If no carbon cut number is given, every carbon number is assumed to be under the carbon cut. If no maximum number is given, it is assumed that the reaction will not be limited by the number of reactions that take place.

Reactability is only half the story of model limitation. After each product is created, it is

tested for “producibility”. Much like the measure of reactability, the producibility check tests each product against the user set carbon range, maximum branch count, and maximum rank of the reactant species/reaction combination. If any check fails, the products are removed from memory, and the reaction is ignored. If the products successfully pass the tests, they proceed on to the isomorphism checks (to determine if the species already exists) and ultimately the reaction is logged.

2.8 Model builder output

2.8.1 Background

Once the model builder has had a successful run, there are three key features that are given back to the user. The first, and most obvious, is the network itself. The second is the identity and properties of the species involved in the reaction network. Finally, the third important aspect is a set of network formation logistics. By studying these three facets, a chemically inclined modeler should be able to determine the acceptability of the network.

The network consists of the entire set of viable reactions based on the user’s choice of rules and limitations. The reactions are written in a standard form familiar to all who have taken a chemistry course with the exception that SpeciesID tags are used as placeholders for the species names or formulas.

Figure 2.44 - An example reaction network for the isomerization and cracking of n-hexane

In addition, each reaction took place via a certain type of reaction, and should be labeled as such. In the above example, the first two reactions are PCP isomerizations, and the third is a cracking reaction. Remember that the LFER assumption will have significance for species that undergo similar reactions.

Although the network is given in full detail, it is not useful until the species are defined. Each unique species in the network has a number, an adjacency list, and a canonical code. For the sake of readability, the network will use the representative numbers in its reaction delineations. The structure associated with those numbers can be backed out of the adjacency lists and the canonical codes. Other derived species information, such as carbon number, hydrogen count, ring counts, atomic weight, etc. can help in determining the basics of the reaction.

Finally, overall statistics are also important in determining the usability of the generated network. Species count and reaction count are the two most frequently used metrics as the network must comply with the user's size wishes. When determining the balance of the network, counts of the individual reaction types can play a helpful part. If the network is primarily composed of a single type of reaction, but should be balanced between two types, these statistics would provide valuable insight.

All in all, the model builder software should supply ample information for the modeler to be able to examine the run, and determine if the model should progress to its next stage, the generation of simultaneous differential equations.

Differential equation generation can be performed by hand, or automatically by the computer. For small scale networks, creating the mathematical model by hand may make sense if skeleton code is already available; but as the network grows larger, so too does the

importance of strict and accurate bookkeeping. Numerical solvers, in general, do not require the structural information of the species. The simple unique species numbers will be all that is useful. Therefore, only the reaction network itself is necessary in creating the equations.

2.8.2 INGen specifics

INGen strives to provide the user with the appropriate reaction network metrics and specifics that will prove essential in formulating a plan of generation. Species identification, network definition, and generation statistics are the three areas of concentrated effort.

INGen integrates with CambridgeSoft's ChemDraw and ChemScript software to provide more meaningful species identification methods. The species are listed in a dictionary that provides an IUPAC name for each. In addition, selection of the species will result in a popup windowed graphical version provided by ChemDraw. Additional information such as identified species type, carbon number, hydrogen number, atomic weight, counts of the ring types, and string code are also given regardless of the inclusion of the ChemDraw software. Every species has a ".dat" formatted adjacency list created in the model directory, as well as a ".cml" file. Checking the details of the reaction of "Species41 → Species 82 + Species 34" is now a simple task with INGen and ChemDraw.

INGen provides a list of reactions in three separate formats: the original NetGen's ".eqn" format, a ".kme" format that lists only reactions, and a ".csv" format that returns both the reactions and the reaction types. The comma separated values file is returned to the INGen interface for quick analysis.

For each reaction, the original ".eqn" format includes a line of string code to string code

conversion, as well as line for a speciesID to speciesID demarcation. The speciesID lines also include extra information for the ODEGen program to properly formulate its kinetic constant equation. Besides general clutter due to the string code lines, a down side of the ".eqn" format is that each reversible reaction is written as two separate one-way reactions.

```
/* C(C(C(H3)H2)H2)C(C(C(H3)H2)H2) -> C(C(C(H3)H2)C(C(H3)H2)C(H3)H) */
species1 -> 1k(PathCIsomerization) species2 ;
/* C(C(C(H3)H2)C(C(H3)H2)C(H3)H) -> C(C(C(H3)H2)H2)C(C(C(H3)H2)H2) */
species2 -> 1k(RePathCIsomerization) species1 ;
/* C(C(C(H3)H2)H2)C(C(C(H3)H2)H2) -> C(C(C(H3)C(H3)H)C(C(H3)H2)H2) */
species1 -> 1k(PathCIsomerization) species3 ;
/* C(C(C(H3)C(H3)H)C(C(H3)H2)H2) -> C(C(C(H3)H2)H2)C(C(C(H3)H2)H2) */
species3 -> 1k(RePathCIsomerization) species1 ;
/* C(C(C(H3)C(H3)H)C(C(H3)H2)H2) -> C(C(H3)C(H3)H2) + C(C(H2)C(H3)H) */
species3 -> 1k(PathCCracking) species4 + species5 ;
```

Figure 2.45 - An example ".eqn" format file

The ".kme" format overcomes these shortcomings. Reversible reactions are written only once but are given a two-ways double arrow to denote them. The string codes and encoded reaction types are removed completely, thereby allowing a user to simply cut and paste these values into the Kinetic Model Editor (KME).

```
species1 <-> species2
species1 <-> species3
species3 -> species4 + species5
```

Figure 2.46 - An example ".kme" format file

The ".csv" format is the same as the above ".kme" format except that on each line, a common separates the reaction as described above with its reaction type. Plans are in motion to have this file format importable into KME such that automatic assignment of reaction types can take place.

Finally, INGen's reaction statistics list information based reaction types and species. The

number of each type of reaction is tallied and reported for a model building run. In addition, each species lists the reactions under which it was processed as well as a total number of reactions it under went.

```
------------REACTANT STATS------------
species1
  PathCIsomerization = 2
  total = 2
species3
  PathCCracking = 1
  total = 1
------------PRODUCT STATS-------------
species2
  PathCIsomerization = 1
  total = 1
species3
  PathCIsomerization = 1
  total = 1
species4
  PathCCracking = 1
  total = 1
species5
  PathCCracking = 1
  total = 1
```

Figure 2.47 - An example ".stat" file

By careful monitoring of important reaction pathways and species, the reaction network can be successfully molded to conform to the user's needs. Species of high importance or fine grade analysis can be afforded more species than those which the user does not care about (and can consequently be lumped into representative isomers).

2.9 A summary of INGen's model building capabilities

As discussed throughout this chapter, there are three major obstacles to model building: the unique representation of species, the inter-conversion of species through reaction, and limitation of network size growth. INGen expanded upon the strategies laid forth in the NetGen program of the KMT software suite.

INGen extended the representation of species to include very large paraffins, and in the process fixed various memory management issues of the original data structures and algorithms. A cascading series of species type definitions was developed so that ionic, free radical and molecular chemistries could be performed concurrently through a single program. Improvements to the core graph theory routines were written and archived for further investigation. In addition, tools were built for interfacing CambridgeSoft's useful ChemDraw software to the model building routines. Creating and viewing species can now be performed nearly instantaneously.

INGen's fundamental approach to the implementation of species reaction was developed in the mindset of a universal model builder. The inclusion of acid, metal and free radical chemistries into a single program allows a user to create advanced hydrocarbon kinetic networks. Both mechanistic and pathways modeling methodologies were included into the INGen framework. Separate pathways for paraffin isomerization and cracking were written to represent the creation and transition of the different carbenium ion types. In addition, a new paradigm for site selection was established, wherein the reaction site can be an artificial data structure within a larger appropriate reaction environment. Both the new carbenium ion pathways and the site selection

Finally, INGen's reaction controllability was written such that any combination of species and reaction types can be controlled by the following options: valid carbon range, maximum number of branches, maximum rank, and maximum number of reactions at a given carbon number. These controls allow for fine grain management of the reaction network. Important reactions can be allowed a larger degree of freedom and thus produce more results.

All in all, the changes to the core functionality of INGen were made in preparation for the

creation of the user interface. The goal of a good user interface is to simplify complicated tasks in a logical manner. By expanding the core functionality, more useful options are given to the modeler. The user interface will be covered in detail in Chapter 4.

2.10 References

[1] Martens, J.A. and P.A. Jacobs (1990). “Evidence For Branching of Long-chain n-alkanes via protonated cycloalkanes larger than cyclopropane.” J. Catal. **124**(2): 357-366.

[2] Weckhuysen, B.M. and R.A. Schoonheydt (1999). “Alkane Dehydrogenation Over Supported Chromium Oxide Catalysts.” Catal Today. **51**: 223-232.

[3] G.E. Dolbear, “Hydrocracking: Reactions, Catalysts, and Processes,” in Petroleum Chemistry and Refining (J.G. Speigth, ed.), Taylor & Francis: Washington, D.C., 1998.

[4] V.A,. A.A. Filimonov, et al. “Th Rates of Reaaction of Individual Groups of Hydrocarbons in Hydrocracking,” International Chem. Eng. **12**(1), 1972, p. 21

[5] Girgis, M.J, and B.C. Gates (1991). “Reactivities, Reaction Networks, and Kinetics in High-Pressure Catalytic Hydroprocessing.” Ind. Eng. Chem. Res. **30**(9): 2021-2058.

[6] Gates, B.C. and H. Topsoe (1997). “Reactivities in Deep Catalytic Hydrodesulfurization: Challenges, Opportunities and the Importance of 4-methyldibenzothiophene and 4,6-dimethyldibenzothiophene.” Polyhedron. **16**(18): 3213-3217.

[7] Whitehurst, D.D., T. Isoda, and I. Mochida (1998). “Present State of the Art an Future Challenges in the Hydrodesulfurization of Polyaromatic Sulfur Compounds.” Adv. Catal. **42**: 345-471.

[8] Dente, M., E. Ranzi, et al. (1979). "Detailed Prediction of Olefin Yields from Hydrocarbon Pyrolysis through a Fundamental Simulation Model (SPYRO)." Computers and Chemical Engineering **3**: 61-75.

[9] Liguras, D.K. and D.T. Allen (1989). "Structural Models for Catalytic Cracking. 1. Model Compounds Reactions." Ind. Eng. Chem. Res. **28**(6): 665-673.

[10] Liguras, D.K. and D.T. Allen (1989). "Structure Modeling for Catalytic Cracking. 2. Reactions of Simulated Oil Mixtures." Ind. Eng. Chem. Res. **28**(6): 674-683.

[11] Mochida, I. and Y. Yoneda (1967). “Linear Free Energy Relationships in Heterogeneous Catalysis I. Dealkylation of Alkylbenzenes on Cracking Catalysts.” J. Catal. **7**(4): 386-392.

[12] Bhore, N.A., Michael T. Klein and Kenneth B. Bischoff (1990). “Species Rank in Reaction Pathways: Application of Delplot Analysis”, Chem. Eng. Sci. **45**(8): 2109-2116.

[13] Joshi, P.V., H. Freund, et al. (1999). Directed kinetic model building: Seeding as a model reduction tool. Energy & Fules. **13**: 877-880.

Chapter 3. Pathways Level Modeling of Paraffin Isomerization and Cracking

3.1 The benefits of pathways modeling

In the previous chapter, a distinction between pathways level modeling and mechanistic modeling was discussed from the point of view of network creation. In this chapter, the reasons behind choosing pathways modeling over mechanistic will be discussed, followed by the specific examples of how pathways level modeling can create kinetically meaningful models when applied to paraffin isomerization and cracking.

Mechanistic modeling attempts to model a chemistry or process at an elementary step level. Every theoretical step is considered independently, and as such, mechanistic models usually create intermediate species between each step. An intermediate species is usually meta-stable, or highly reactive and, as such, is not a measurable product. Therefore, the kinetics of the elementary reaction steps are not easily determined experimentally. Instead, experimentalists can determine the kinetics of collections of elementary steps that, when combined, create a reaction of measurable species to other measureable species. Theory helps shape the overall kinetics back into the modeled elementary steps. Besides the inability to directly measure intermediates, the sheer number of intermediates can cause a model to explode in size.

As discussed in the previous chapter, it is often necessary to limit the size of a model, especially in relation to the amount of data available. By definition, there are no intermediate concentration data available, but there could be thousands of intermediate species in a mechanistic model. The intermediates cannot simply be cut from the

mechanistic model, because the species they eventually react to are chemically significant. Therefore, the model can become bogged down in theoretical, unmeasured species. However, if the intermediates are removed from consideration by eliminating the necessity for elementary steps, the model becomes a pathways level representation.

By eliminating the elementary steps and intermediates, the pathways model lumps the kinetics of the mechanism into a single representative reaction. Whereas the kinetics can be measured distinctly in laboratory experimentation, the correct "form" of the kinetics equation is more difficult to ascertain. If one elementary step of the mechanistic pathway was rate controlling, the kinetics of the entire pathway should have a similar form to that rate determining step.

If the chemistry of the system is sufficiently understood or theorized, the formation of the rate law might not be an issue. With a kinetic model editor, such as KME, changing the automatic coefficient driven rate law of the overall pathway to one based on an intermediate rate determining step is an easy exercise, handled after network creation. Because of the caveat that the chemistry must be understood, pathways level modeling might not work for every system. In addition, pathways level modeling can sometimes conceal important mechanistic information and therefore combine unlike chemistries within the same pathway. Ionic paraffin isomerization and cracking is one such case.

Current theory hypothesizes that the mechanistic steps of both paraffin isomerization and cracking involve the formation of carbenium ions. As previously discussed, a carbenium ion is a carbon atom with one of its four bonding orbitals devoid of electrons, and therefore carries a positive charge. The stability of the ion is dependent on the number of hydrogen atoms connected to the central ionic carbon. The energy required to create each form varies. Both methyl and primary carbenium ions (three and two hydrogens respectively)

are absent from all but the most temperature rich processes because of their high energy demand. The paraffin isomerization and cracking system analyzed herein is not one such system. Only secondary and tertiary carbenium ion pathways will be allowed.

3.2 Isomerization

Branching paraffin isomerization is theorized to proceed through a protonated cyclopropane (PCP) mechanism as discussed in the previous chapter. Theoretically, the PCP will not form without an initial carbenium ion formation, probably due to some form of hydride abstraction. Given the constraints outlined above, the initial carbenium ion can only be of secondary or tertiary order, each with a different energy associated with its creation.

As described before, once the ion forms, it interacts with a carbon neighbor of a neighboring carbon; thereby, forming a tentative bond, and trapping the target carbon's hydrogen proton by the force of the cyclopropane's electron density. Such a state is known as a PCP, and because of the non-explict bondedness of the proton, it is metastable.

Any of the three PCP bonds can break, thereby attaching the hydrogen back to one of the two disconnected carbons, the charge "transferring" to the other. The carbon that holds the charge is now the carbenium ion, and could be secondary or tertiary depending on the structure of the molecule. Thus, four cases present themselves.

Case A: the positive charge from an initial tertiary carbenium ion returns to a tertiary carbenium ion after PCP bond breakage.

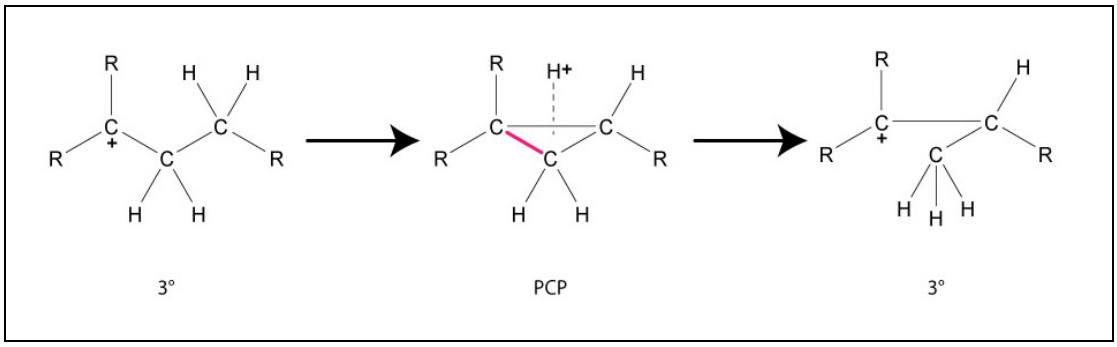


Figure 3.1 - An example of tertiary to tertiary PCP isomerization

Case B1: the positive charge from an initial secondary carbenium ion relocates to a lower energy tertiary carbenium ion after PCP bond breakage.

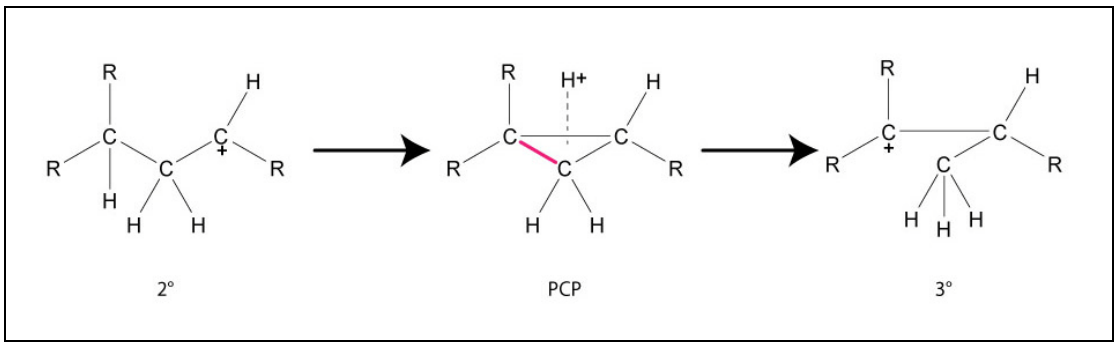


Figure 3.2 - An example of secondary to tertiary PCP isomerization

Case B2: the positive charge from an initial tertiary carbenium ion relocates to a higher energy secondary carbenium ion after PCP bond breakage.

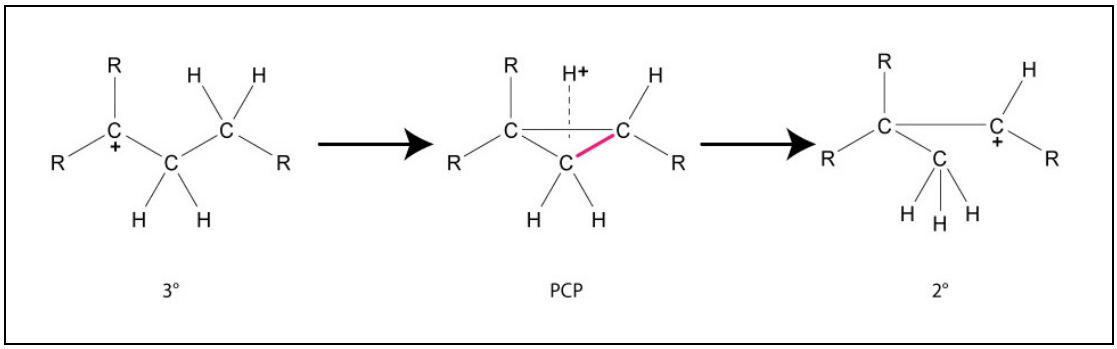


Figure 3.3 - An example of tertiary to secondary PCP isomerization

Case C: the positive charge from an initial secondary carbenium ion returns to a secondary carbenium ion after PCP bond breakage.

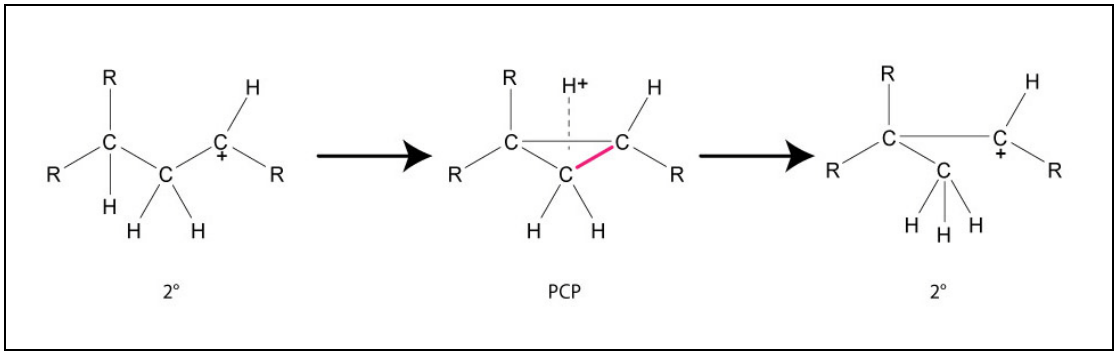


Figure 3.4 - An example of secondary to secondary PCP isomerization

Based on the differing reaction energy of the above cases, each of these different mechanisms may deserve its own kinetic consideration. Previously, all such isomerization reactions were grouped together into a single isomerization pathway. For some process systems, such a simplification may be sufficient, but if the kinetics of isomerization are to be truly studied on the pathways level, the separation into different cases is important.

INGen is capable of handling all four cases by creating sub-cases for different molecular

structures. Although pathways isomerization reactions occur only between four atoms, INGen examines a larger substructure of the molecule. Since no primary carbeniums are to be allowed, INGen first searches for a chain of five carbons in a row. The two end carbons give flexibility to the destruction of the PCP. Regardless of which bond PCP bond is broken, there exists the possibility of a resultant non-primary carbenium ion. Each of the three middle carbons will become part of the PCP, and are hereafter known as C1, C2, and C3. Each of the labeled carbons has their non-considered (not a labeled or end carbon) neighboring atoms analyzed, and the different combinations can lead to the four pathways outlined above.

For instance, if the first, second and third atoms have only hydrogen atoms attached, then the only possible path is that of A. An initial hydrogen abstraction creates a secondary carbenium ion, and after PCP formation, only secondary carbenium ions can be formed after bond breakage (disregarding any primary carbenium ions).

Listed over the next few pages is the "glossary" of the combinations of substituents on the three carbon chain and their respective possible pathways as written into the INGen code.

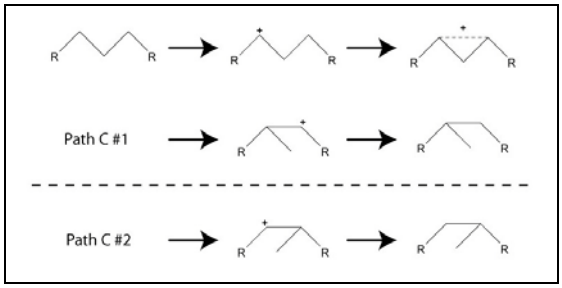


Figure 3.5 - PCP pathways for no substituents

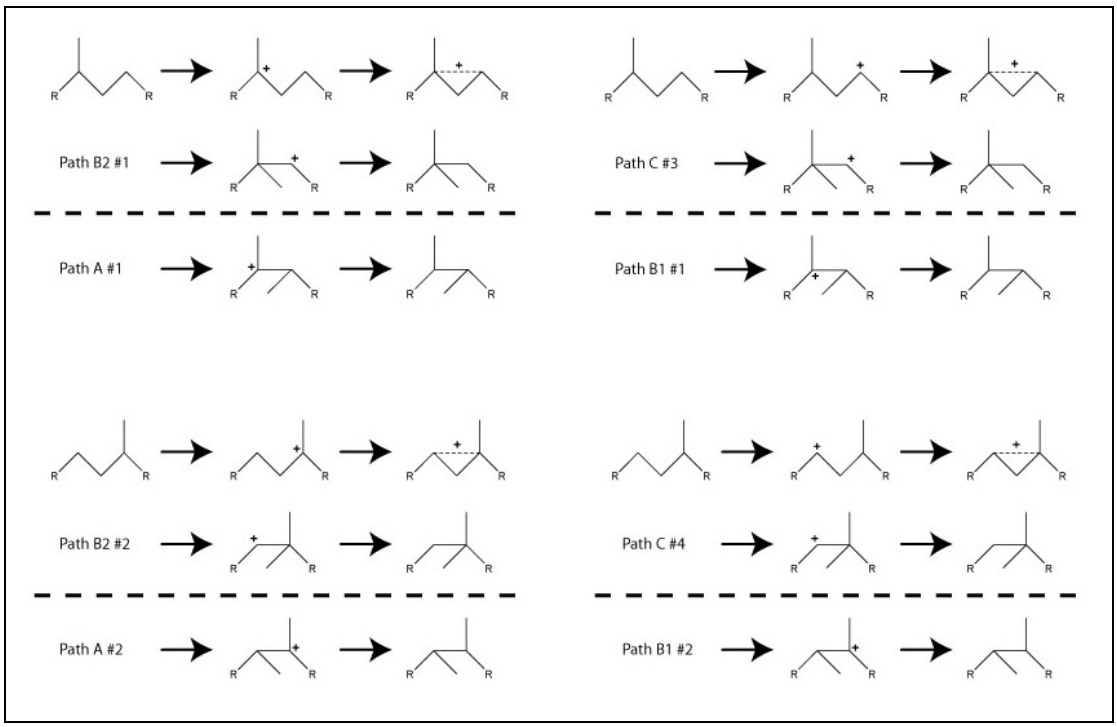


Figure 3.6 - PCP pathways for 1-methyl and 3-methyl substituents

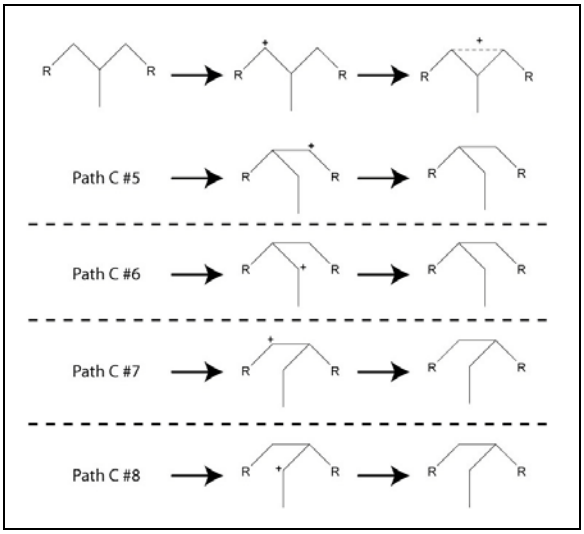


Figure 3.7 - PCP pathways for a 2-methyl substituent

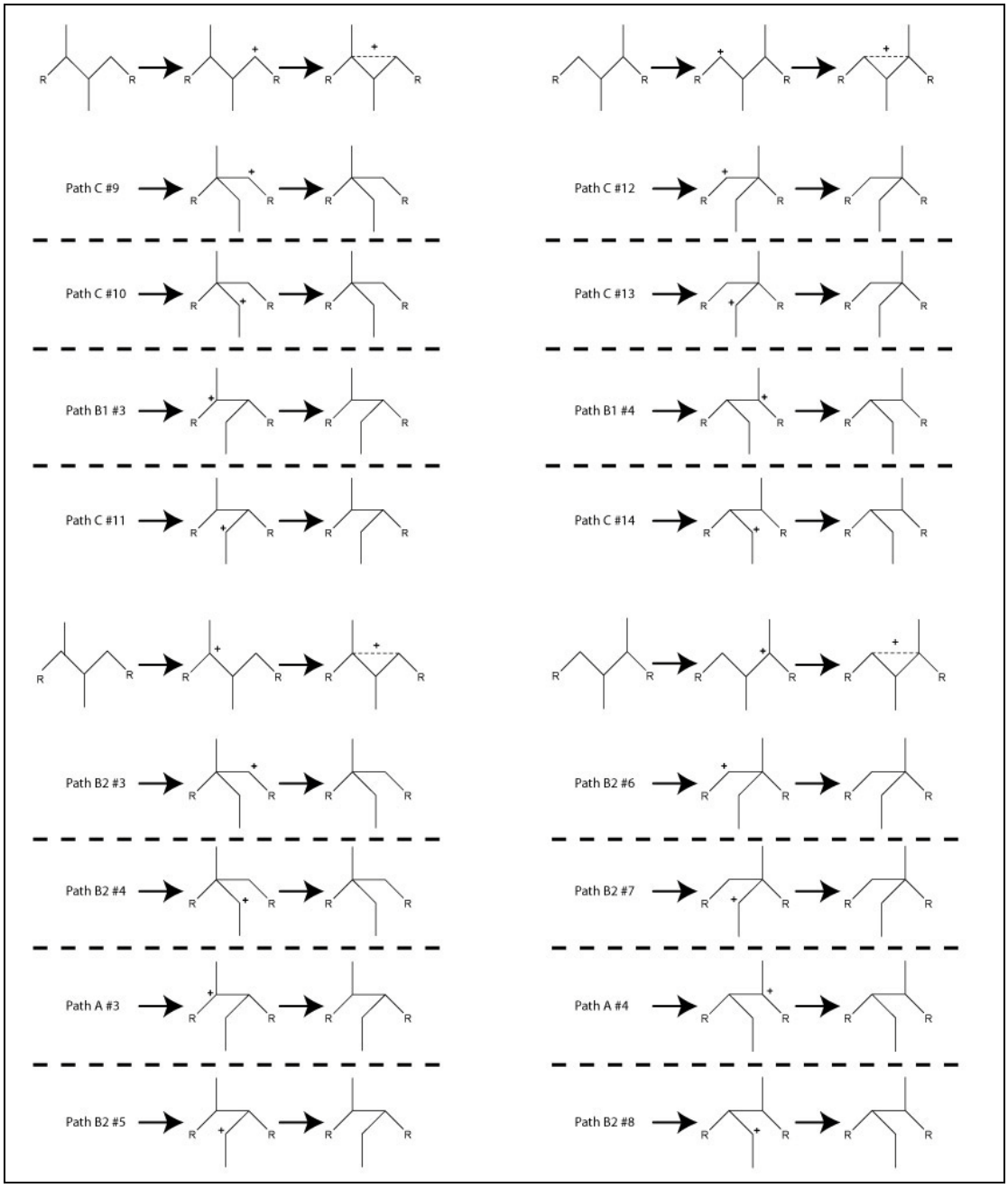


Figure 3.8 - PCP pathways for 1,2-methyl and 2,3-methyl substituents

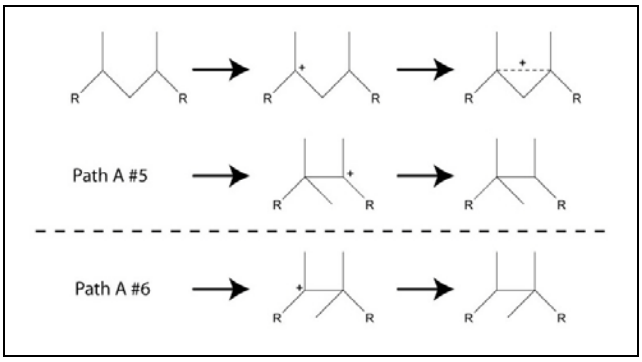


Figure 3.9 - PCP pathways for 2,4-methyl substituents

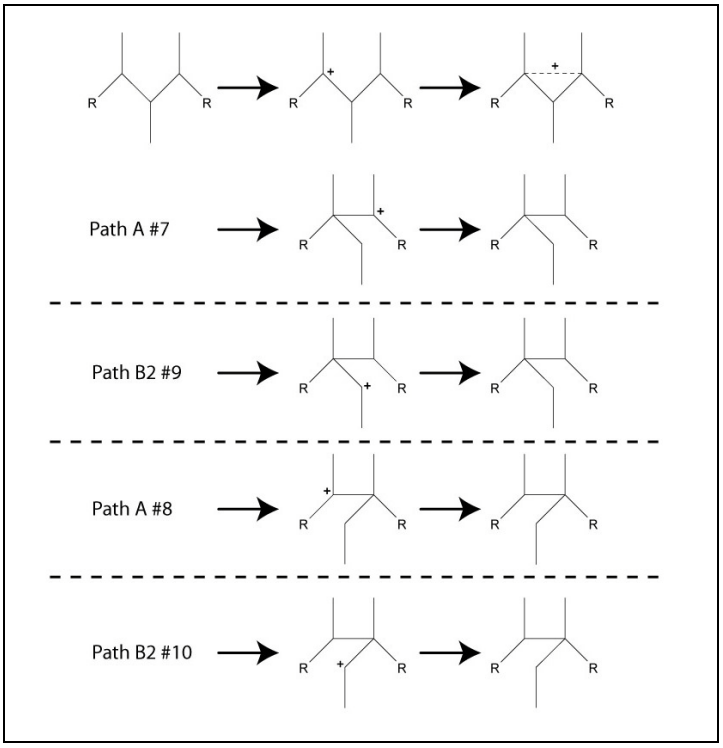


Figure 3.10 - PCP pathways for 1,2,3-methyl substituents

By using these separate pathways, it is possible that for a given reaction of A → B, multiple paths may be listed within a single reaction network. The reaction family for each will be different, based upon the mechanistic pathway taken. In this way, the isomerization process can be evaluated in a pathways manner, but maintain theoretical mechanistic significance.

3.3 Cracking

Much like the aforementioned paraffin isomerization mechanistic pathways, carbenium ions also play a role in cracking reactions. Cracking, as also discussed in the previous chapter, occurs when a carbon-carbon bond is broken in a paraffin such that two separate products are formed. The paraffin literally "cracks" like a twig into two separate pieces.

The mechanism for cracking also begins with an initial positive charge on a mid-chained carbon, probably due to hydride abstraction. The resultant carbenium ion can interact with the electrons in the bond between a neighboring carbon and its other carbon neighbor. The electrons of that bond are reallocated between the initial carbenium ion and its participating carbon neighbor, thus forming a double bond. Doing such eliminates all bonds to the neighbor's neighbor, and leaves that carbon with an empty bonding orbital, and therefore the positive charge. This process is known as β-scission, as it splits a "beta" bond (two away) from the charged atom. In this case, the molecule itself is cracked into two separate species.

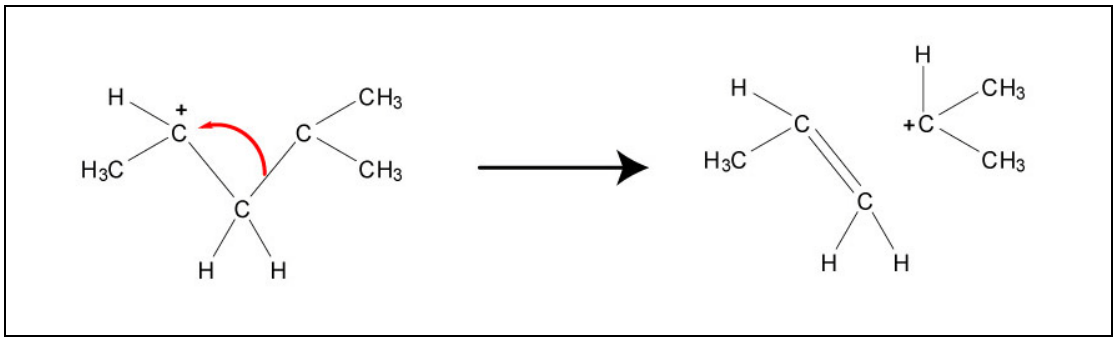


Figure 3.11 - β-scission

The species with the double bond is an olefin and can undergo further reaction such as hydrogenation or hydrogen saturation to once again become a paraffin. The ionic species can undergo further cracking or isomerization as outlined above, or can undergo a reverse

hydride abstraction to stabilize as a paraffin.

Again, it is important to realize that different degrees of carbenium ions are possibly formed during the mechanistic step. As was the case for isomerization, no methyl or primary carbenium ions were allowed to form in the INGen pathways routine. As before, four cases are described:

Case A: the positive charge from an initial tertiary carbenium ion relocates to a different tertiary carbenium ion after β-scission.

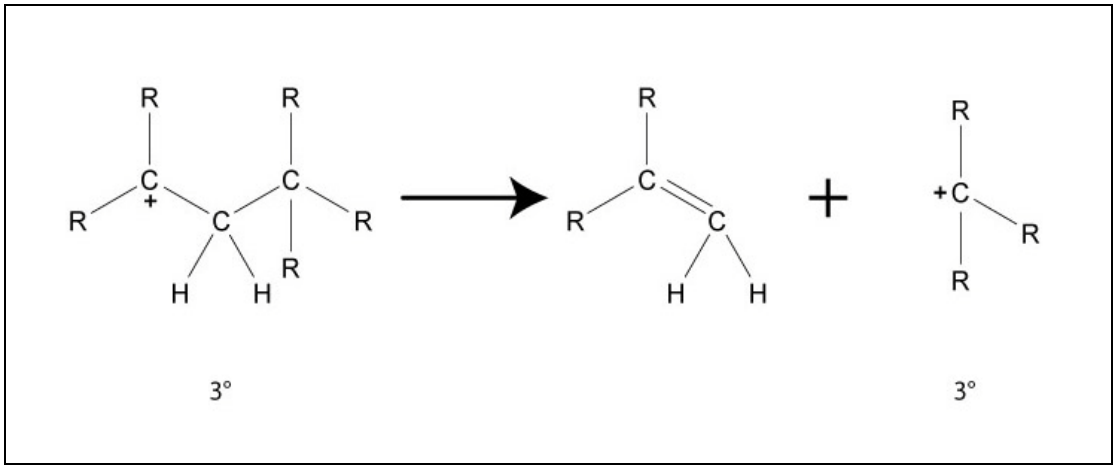


Figure 3.12 – An example of tertiary to tertiary β-scission

Case B1: the positive charge from an initial secondary carbenium ion relocates to a lower energy tertiary carbenium ion after β-scission.

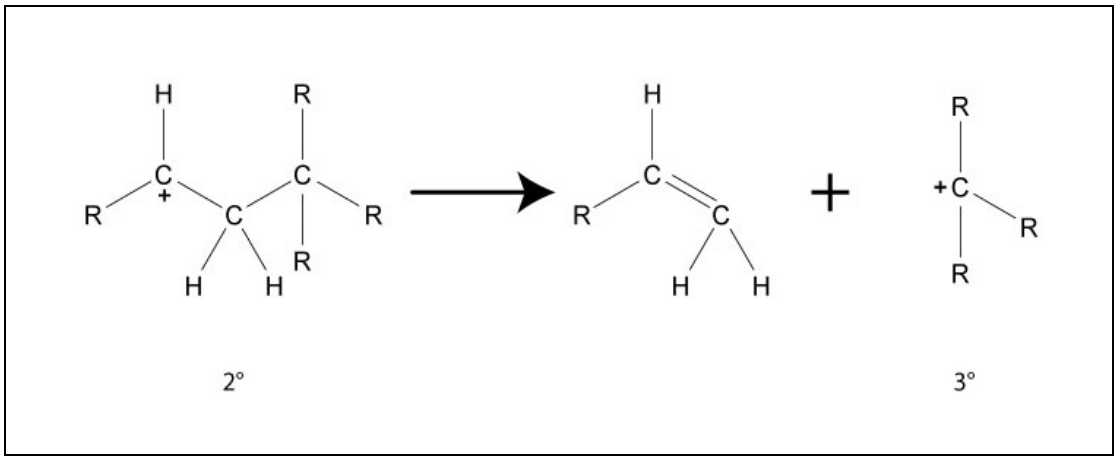


Figure 3.13 - An example of secondary to tertiary β-scission

Case B2: the positive charge from an initial tertiary carbenium ion relocates to a higher energy secondary carbenium ion after β-scission.

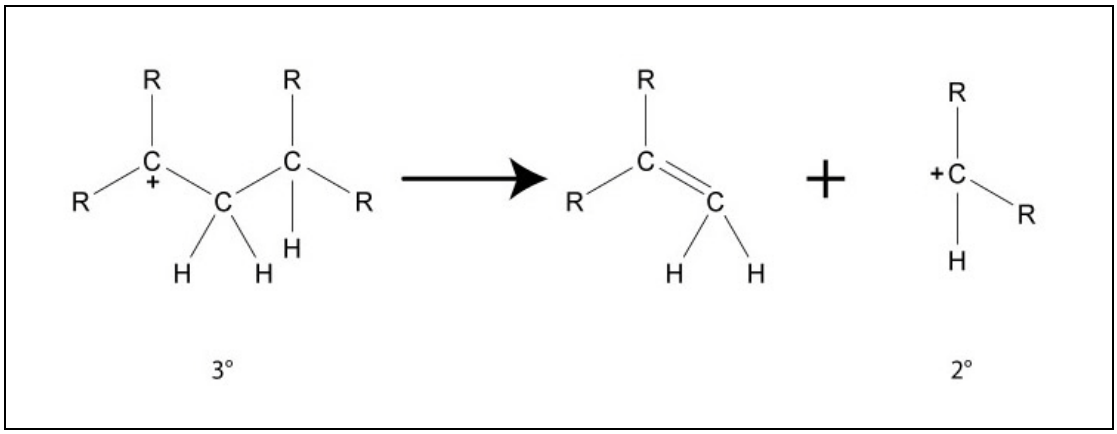


Figure 3.14 - An example of tertiary to secondary β-scission

Case C: the positive charge from an initial secondary carbenium ion relocates to a different secondary carbenium ion after β-scission.

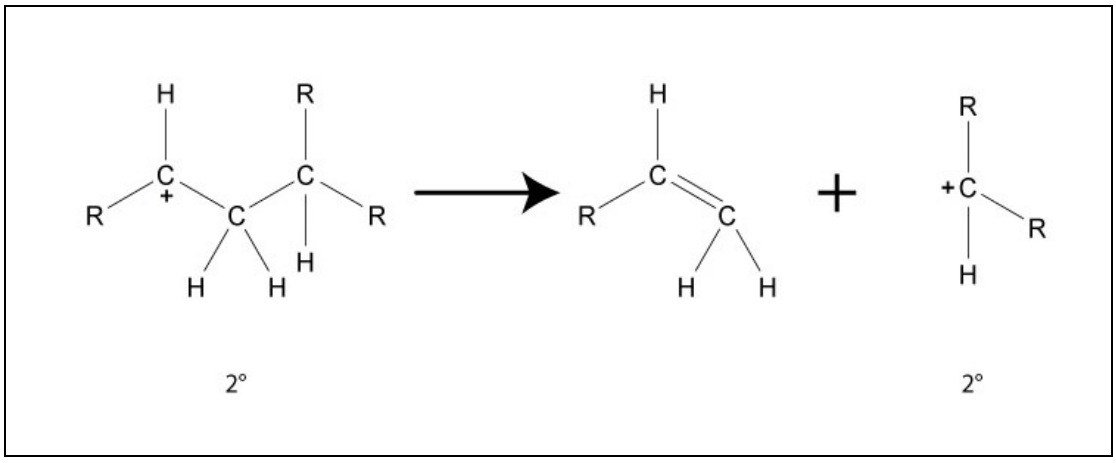


Figure 3.15 - An example of secondary to secondary β-scission

Because the energy state of each carbenium ion is different, INGen can now consider each of the four paths separately. A five carbon chain is the backbone for reaction, although only three carbons are actually involved in the reaction. The end carbons insure no primary carbenium ions need to be formed. In addition, there must be a branch on either the first or third of the three reactive carbons. Without this branch, a primary carbenium ion would form after β-scission.

Below is the dictionary of initial site and carbenium pathway definitions as handled by INGen. Only the first and third of the three interior carbons are addressed. The second carbon may have branch or hydrogen constituents.

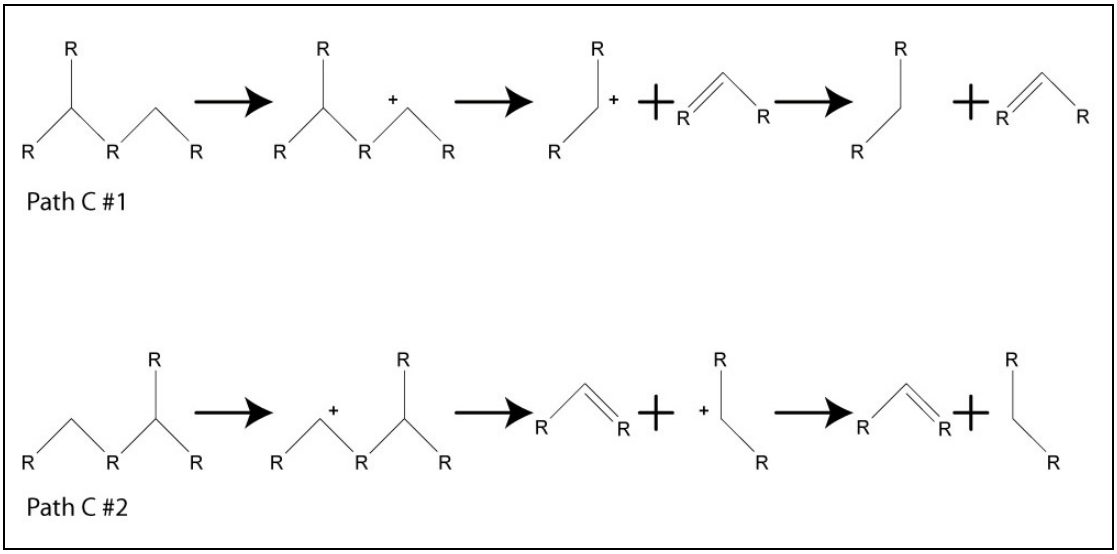


Figure 3.16 – β-scission pathways for 1-branch and 3-branch substituents

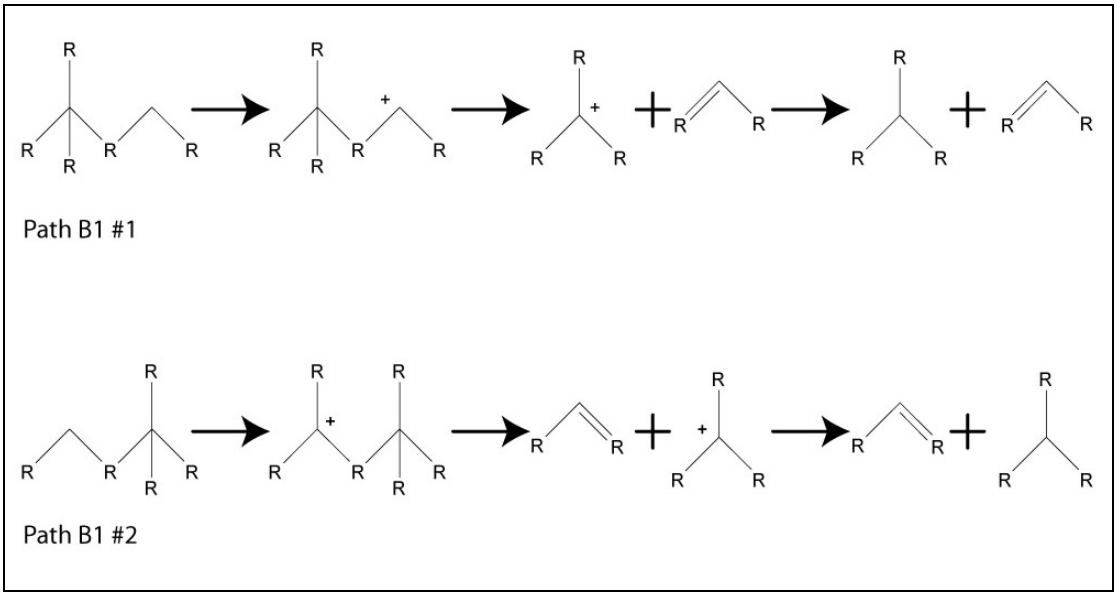


Figure 3.17 - β-scission pathways for 1,1-branch and 3,3-branch substituents

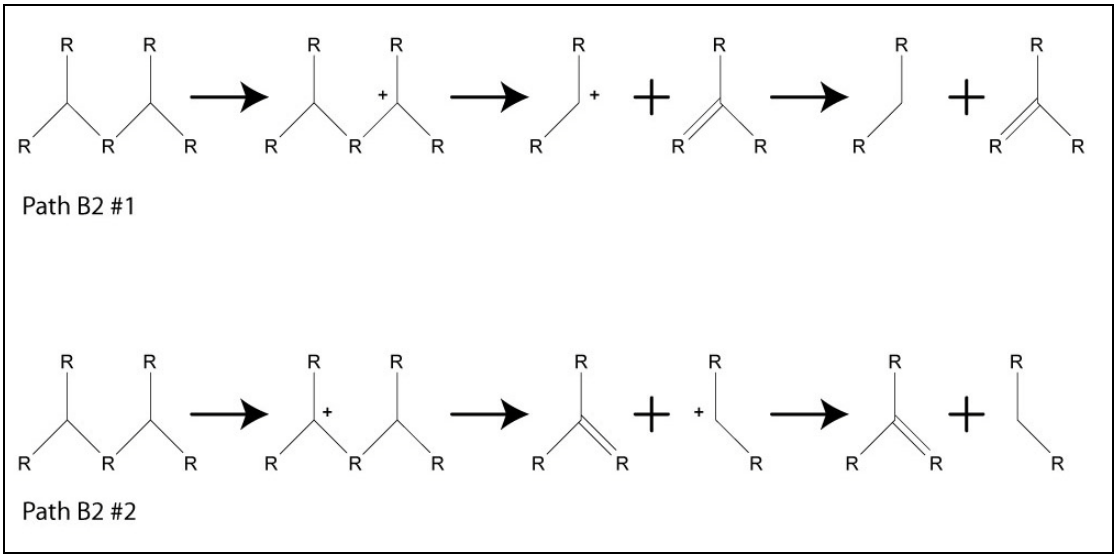


Figure 3.18 - β-scission pathways for 1,3-branch substituents

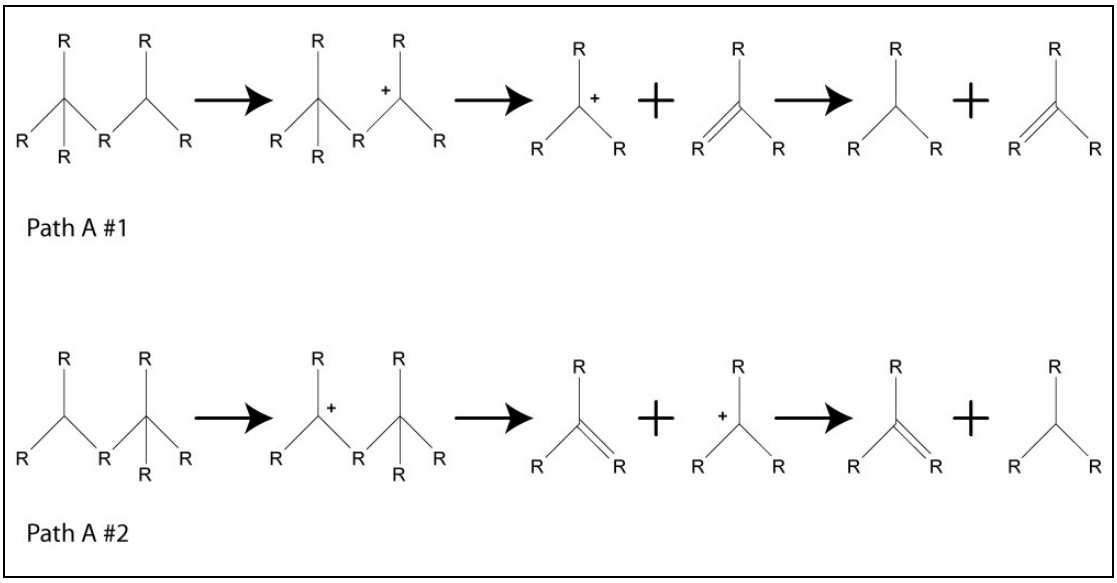


Figure 3.19 - β-scission pathways for 1,1,3-branch and 1,3,3-branch substituents

In addition to separating of cracking reactions into carbenium ion pathways, INGen also created parallel pathways through which a paraffin cracks into two paraffins, not a paraffin and an olefin. Theoretically, in a hydrogen rich environment, the olefins would saturate readily to reform paraffins, and may be considered intermediate species. Either set of

paraffin cracking pathways (olefin products or intermediates) are available to the user.

By expanding the ability of the user to categorize the underlying mechanism of a pathways reaction, INGen can greatly enhance the understanding of the kinetics while maintaining a pathways methodology.

3.4 A new paradigm of reaction site selection

The expansion of the past two reaction families to incorporate specific carbenium ion pathways led to a new paradigm for selecting appropriate sites for reaction. Site selection had previously been handled en mass by searching for chains of appropriate atomic species to which the reaction matrix could be applied. Bond order, cycle participation, and branch analysis were performed on the list of atomic chains. Post-processing was needed after reaction to determine the viability of the produced species.

With the inception of INGen, advanced site selection can now take place. A larger subset of atomic species can be recursively searched for and analyzed. The true site list can be artificially constructed from the larger atomic subgraph, and sent to the reaction routine. In this manner, complex branched sites, where there is a discontinuity in reacting atoms, can be created for use with existing reaction matrices and routines.

In addition, as demonstrated by the large number of PCP isomerization pathways, multiple sites and paths can be searched with a single given starting species. A single routine analyzes the appropriate constituents, ruling out cases that don’t match the structure. By incorporating a better site selection methodology, INGen makes coding new reactions much easier than the original NetGen.

3.5 Conclusion

Pathways level kinetic modeling can pack a great deal of information into a small equation. Although theoretical mechanisms can be used as the starting point for determining a pathway, sometimes information can be lost. As described above, a single theory can allow for multiple mechanistic paths that produce the same overall result. As such, pathways level modeling may lose scientific significance when determining the kinetics of a reaction system. Some overall reaction systems may not need a deeper understanding of the mechanistic kinetics, but others may. Therefore, it is important to give the modeler the choice of coarse or fine grain reaction families.

INGen provides a shapeable environment wherein mechanistically separate pathways can be distinguished for later modeling. As examples, 12 new reaction classes were created: paraffin isomerization paths A, B1, B2, and C; paraffin to paraffin/olefin cracking paths A, B1, B2, and C; and paraffin to paraffin/paraffin cracking paths A, B1, B2, and C. This novel approach to the legacy of NetGen pathways reaction building is an important step forward, and continued analysis of other multi-mechanistic pathways is urged.

Chapter 4. Frontend INGen

4.1 Background

As discussed in the first chapter, there can sometimes be a time consuming disconnect between modeler and programmer. When investigating new mechanisms and the like, a chemist or modeler often needs preliminary results quickly. If the proposed reaction selections do not create a viable model network, the modeler should be able to reconfigure the network generation and produce new results. Until the creation of INGen, the options available to a modeler using the NetGen technology were severely limited.

Controlling NetGen reactions was performed with a wide brush at the best of times, and in some cases, nonexistent. Each chemistry had its own model builder, with a fixed set of options, and possibly user controlled global variables like maximum carbon number. Fundamental changes to applied reaction families needed to be done on via code change. In some cases, the code change could be minor, but an understanding of the programming language, compilation, and Cygwin/Linus/Unix environment was still required.

NetGen and INGen share a common ancestry of C/C++ code, with the majority being in the lower level programming language C. One of the trickiest elements of the C language is the express handling of memory locations. Any error within the memory handling would simply return a cryptic “segmentation fault” error message, which sometimes required hours of debugging. Thus, it was not efficient for non-programmers or even novices to try to implement new rules or chemistries within NetGen. If changes needed to be made, the modeler or chemist would have to contact a programmer on the NetGen team and make a request for a new feature. Whenever work flow is divided amongst two people’s schedules there is bound to be lag.

Compilation of NetGen is performed through the use of the “make” project management system. Some of the code is written for a C compiler, some for a C++ compiler, and some for a Lex/Yacc compiler. The project manager knows how to compile each of the types of files, and compilation should not present a problem to any veteran programmer. Novices may not be familiar with the concepts involved in compilation, and may not realize that their code changes have not been implemented properly. All in all, multi-language compilation is another level of computer science abstraction that a chemist or modeler may not be familiar with.

Finally, the operating environment itself can cause problems for those not familiar with a Unix style command line interface. NetGen is compiled using the tools found within the GNU tool set, namely gcc and g++. GNU self-reflectively stands for “GNU’s Not Unix” because it is a set of tools and utilities that mimic those found within the Unix environment. Linux (and its various distributions) is written around these publicly available tools, and as such can be used for compiling NetGen. The problem is that Linux administration can be a difficult task for even the most computer savvy individuals. Most chemists and modelers are more likely to be running a Microsoft Windows environment, and therefore will need a non-linux solution for accessing the GNU toolkit.

Cygnus Solutions has provided such a solution with their Cygwin environment. Cygwin emulates a Linux-type environment within Windows. Although Cygwin is freely available to anyone, its installation as a NetGen compilation environment requires special knowledge to install properly. If all the tools used by NetGen are not loaded, the compilation and running will fail. In general, it is recommended that a user be familiar with Linux or Cygwin before attempting to perform any recompilation or command line actions.

On the subject of the command line, it should be noted that NetGen ran completely from command line interaction. A few file pointer flags and options were specified at run time, and the output files were created for later perusal or further processing. As such, global variables (applying to every type of reaction) for maximum/minimum carbon number, maximum branch count, and maximum rank were more efficient.

Overall, a chemist or modeler had little control over how the reaction network developed with a given model builder. The researchers behind NetGen were truly responsible for providing an almost preprogrammed network, not the network builder. If a modeler required changes to the network, changes would need to be made at the network builder level; changes that could take an unknown amount of time. Such inefficiencies hindered the adoption of NetGen. INGen seeks to overcome these drawbacks, and become the *de facto* standard in model building.

4.2 INGen – Preparing the Backend for a Frontend

In the previous chapters, changes to the backend core of the model builder were discussed at length with regard to the principles of model building. This section will cover the several other changes that were made to the backend in preparation for a useful front end.

Conceptually, the first change was the inclusion of all the various chemistries present throughout the various incarnations of NetGen and its model builders. By including every reaction known to NetGen, INGen will allow the user to simply turn reaction families on and off. The reactions and implementation were covered thoroughly in Chapter 2, and they will only be lightly touched on again with the description of the control file.

The second major change, also covered in Chapter 2, was the dissolution of global control variables into species/reaction combination variables. By allowing finer control over reaction generation for specific families, a more sculpted network can be created. Because there were quite a number of combinations, the traditional command line options methodology would no longer work for controlling the reaction generation at run time. Instead, a control file tactic was instituted.

The control file is the third major change implemented in preparing for the INGen front end, and it encompasses the previous two aspects. By creating an XML type control file, the user can specify the values for any species/reaction combination variables. The backend of INGen was modified to read and process such a file into new species/reaction combination level control variables.

By default, all combinations are assumed to be disallowed, unless otherwise noted (allow = 1). If allowed, each combination may have a minimum and maxium carbon number, a maximum number of branches, a maxium rank, a carbon cut point, and a maximum number of reactions. Each option is discussed in detail in Chapter 2. All such options are assumed to be their most lenient, unless otherwise stated.

```
<OptionSet model=test>
  <Species Type=Aromatic>
    <Reaction Type=Saturation2H>
      <Option Type=Allow>0</Option>
      <Option Type=MaxRank>-1</Option>
      <Option Type=MaxCarbon>-1</Option>
      <Option Type=MinCarbon>-1</Option>
      <Option Type=MaxBranch>-1</Option>
      <Option Type=MaxCount>-1</Option>
      <Option Type=CarbonCut>-1</Option>
    </Reaction>
    <Reaction Type=Saturation4H>
      <Option Type=Allow>0</Option>
      <Option Type=MaxRank>-1</Option>
      <Option Type=MaxCarbon>-1</Option>
      <Option Type=MinCarbon>-1</Option>
      <Option Type=MaxBranch>-1</Option>
      <Option Type=MaxCount>-1</Option>
      <Option Type=CarbonCut>-1</Option>
    </Reaction>
    <Reaction Type=Saturation6H>
      <Option Type=Allow>0</Option>
      <Option Type=MaxRank>-1</Option>
```

Figure 4.1 - An example of an INGgen options file

The control file is a logical organization of information, but for systems with a large number of combinations, it can be a tedious chore to create. Thus, there is a need for more efficient control file creation through the front end.

4.3 INGen – More Tools for the Frontend

In addition to the changes to the backend, various programs and scripts were created to help convert input and output information into frontend friendly formats. As discussed in Chapter 2, molecules can be saved between runs in a ".dat" adjacency list format. The adjacency lists can be automatically created through an easy set of steps. First, the species can be created within ChemDraw by literally drawing the molecule with its unique set of tools. Second, the species is saved as a CML (Chemical Markup Language) type file. The file can either be directly saved into the CML-In conversion directory, or moved there afterwards. Next, the PERL language DATGen.plx script is run, thereby creating new .dat

format files in the DAT-Out directory. These DAT-Out files can be placed anywhere within the INGen Data directory. A similar process is automatically handled upon a successful model building run.

INGen outputs the adjacency list for every species in the model into the “Adj” directory within its appropriate model base directory. When the model is successfully built, a script turns each adjacency list file in the Adj directory into a CML format file in the “CML” subdirectory for that model. These CML files can be easily opened by ChemDraw in order to visualize the structure of the species.

The CML files will be further processed by the ChemScript “DictionaryGen.py” Python language program. The dictionary generator will determine the IUPAC name for each CML file and return the results as a list. In addition, a separate script coded directly into the INGen frontend will access the ChemDraw routines and automatically create graphical “.gif” files for each of the CML files when accessed through the frontend.

Also upon completion of a successful network build, a PERL script (eqn2kme.plx) will transform the NetGen language reaction network into the more readable KME language as described in Chapter 2. The script also creates the reaction family file (.fam), the combined output (.csv), and the network statistics file (.stat) as previously discussed.

Other Perl scripts are available that will convert the adjacency list .dat format files into “.mtx” matrix files, “.xml” XML files, and “.ct” connection table files. In Chapter 2, the matrix file was used for the mathematics of reaction, as well as an easy to understand conceptual model of the representation of the species. As such, seeing the actual matrix of a species may lend some benefit when analyzing its composition. The XML and connection table formats were included for possible integration with other systems.

The creation of the “middleman” scripting components allows both the backend and front end to remain relatively unchanged while changing data from one format to another. They are intended to simplify tedious tasks and allow for their automation at the front end level.

In the previous chapters, changes to the backend core of the model builder were discuss

4.4 INGen – The Frontend

Up to this point, the nature of the frontend has been discussed, but not the specifics therein. This section will reverse this trend by delineating all aspects of the front end, and giving specific instructions on how a user can simply and efficiently create a new reaction network. The application will be described as a series of sheets, pages or screens, fundamentally laying forth the logic of model building once again.

The INGen frontend was written to be a companion piece to the existing KME software, and as such, it was programmed similarly via Visual Basic for Applications (VBA) within the Microsoft Excel environment. The project includes three forms as well as multiple worksheets and modules. Execution begins when the INGen.xls workbook is loaded into Excel. The workbook contains macros and the modeler must first allow the macros to run.

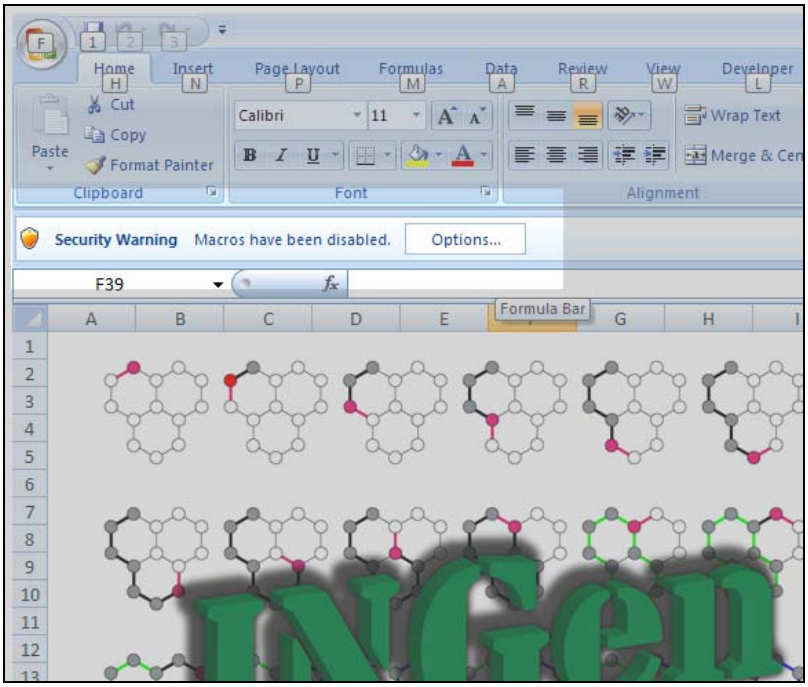


Figure 4.2 - INGen macro warning

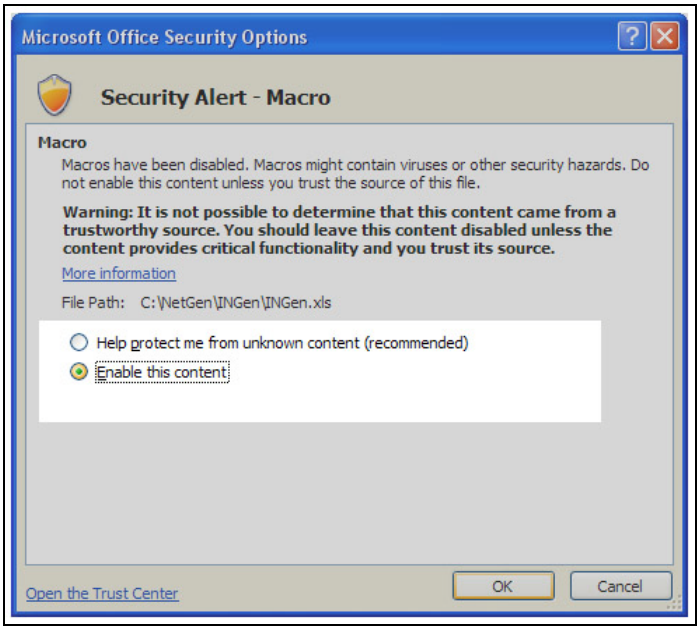


Figure 4.3 - Enable macros

Once the macro execution is allowed, INGen cleans the execution environment of any previous run. This process does not delete any previous models; they are still saved in

subdirectories and can be recalled easily. Because INGen also behaves like a normal Excel workbook, the user is presented with a "Save Changes?" prompt upon closing. It is possible that the options that were selected on a previous run were saved to the various options sheets. INGen corrects this upon opening, and presents the user with a pristine environment.

Once control has been returned to the user, he will find himself at the title page on the "*INGen*" sheet. The title page acts as logical starting point when beginning a model, and can be used to reinitialize the INGen environment at any time. The only other available worksheet tab should be the "*Models*" sheet.

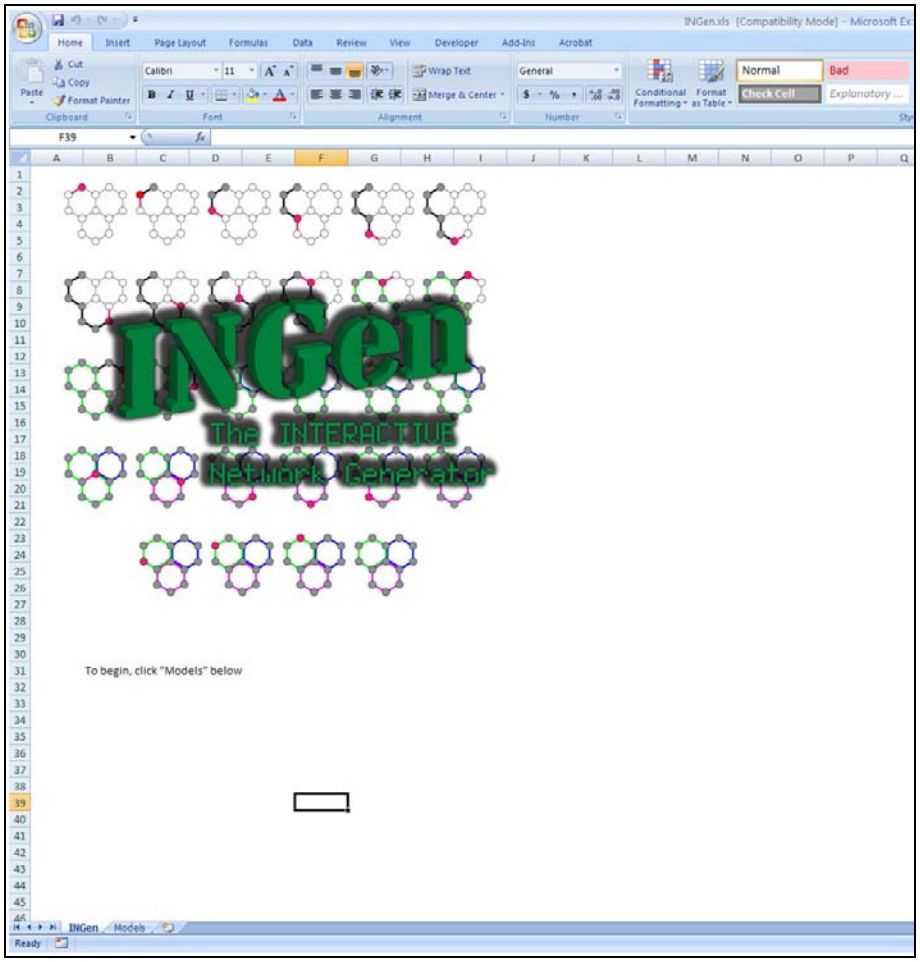


Figure 4.4 - INGen opening screen

When the *Models* tab is selected, a VBA form initializes and presents the user with a list of currently available models. Each model is actually housed in a subdirectory within the INGen folder. If a model is moved from one computer to another, it will be automatically displayed the next time the form initializes.

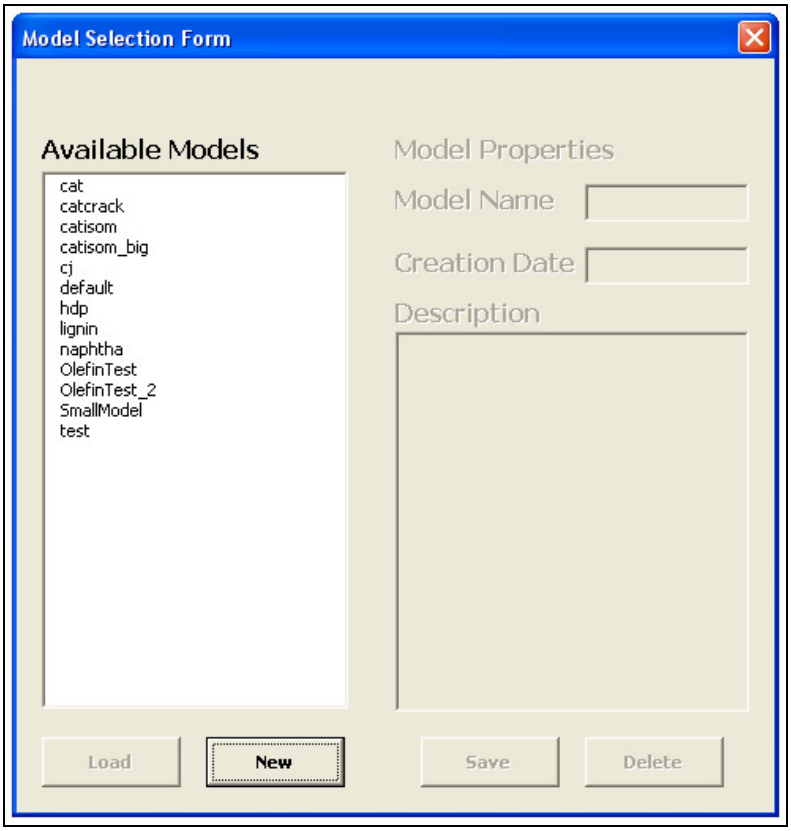


Figure 4.5 - Model selection form

When a model is selected, its properties (if given by the user) are displayed, and the "Load" button becomes active. If any of the properties are changed, the user should select the "Save" button to signify that a change has been made and that it should be saved for future reference.

If the modeler prefers to create a new model rather than work with an existing model, the

"New" button should be pressed. A blank list of properties will be shown, and the user will need to supply at least a model name. After the user has entered any property information, the "Save" button should be pressed. The "Save" option creates a model customized shell script (for the Cygwin environment) and batch file (for the Windows environment), both of which are necessary for successfully running the INGen backend.

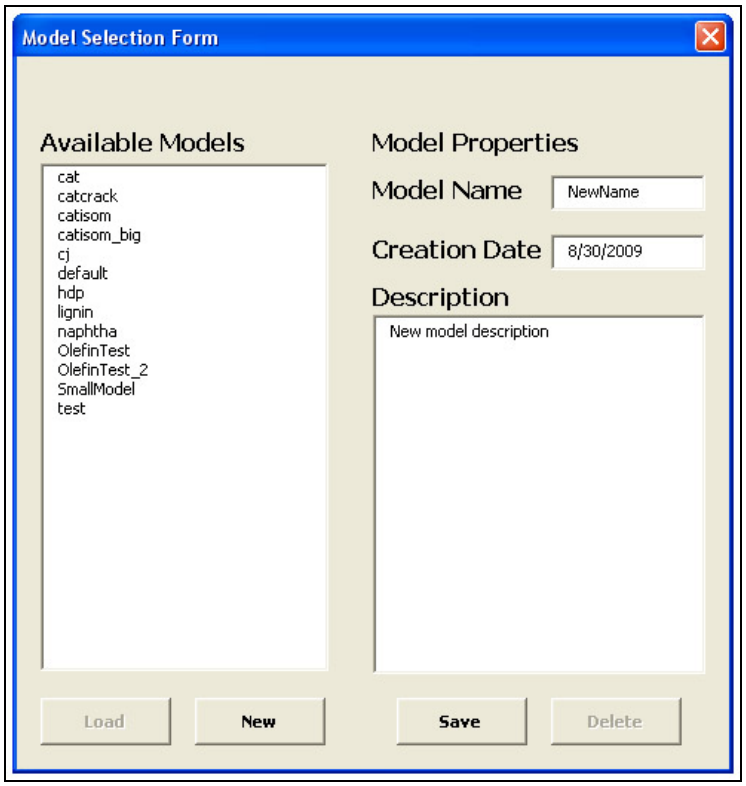


Figure 4.6 - New model information

After either the "Save" or "Load" button is pressed, a new "*Options*" tab is added to the worksheet, and the user is supplied with the options VBA form. The name of the model is written in bold type across the top of the form and should correspond to the value from the previous sheet.

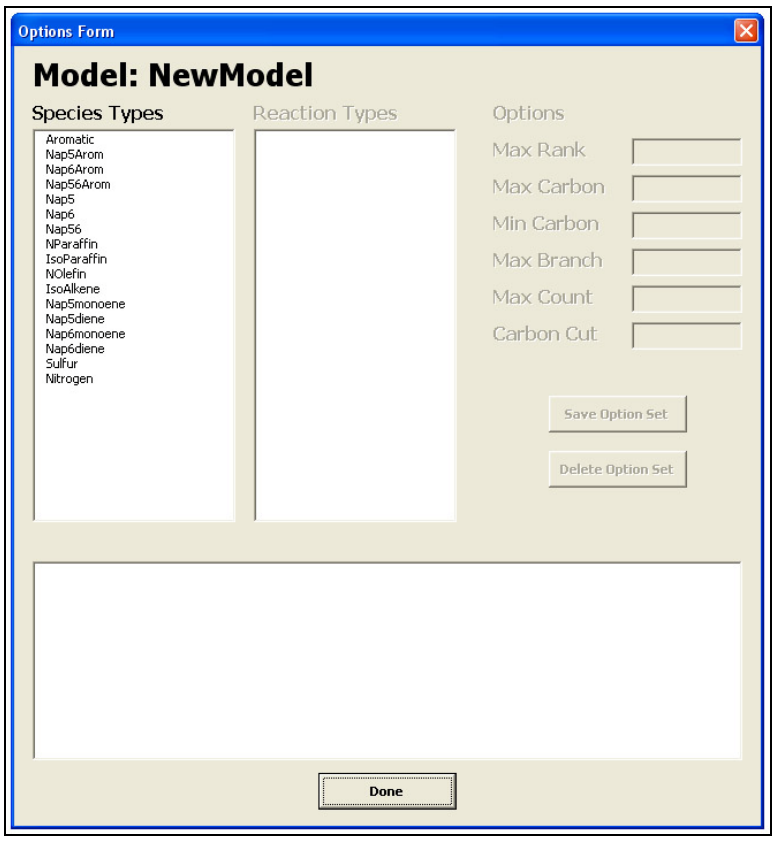


Figure 4.7 - Options form

The first column is labeled “Species Types” and lists all the types of species handled by INGen. Additions to the list are only available through further releases of the program. When a species type is selected, the “Reaction Types” column is populated with all the available reaction types for that species. Again, additions to the list are only available through further releases of the program. When a reaction type is selected, the modeler can enter values into the Options boxes. No option is required as each has an internal, non-limiting default value, but pressing the “Save Option Set” will signify that the species/reaction combination should be allowed. Saved option sets will then appear in the horizontal box below. The process can be repeated for every wanted species/reaction combination.

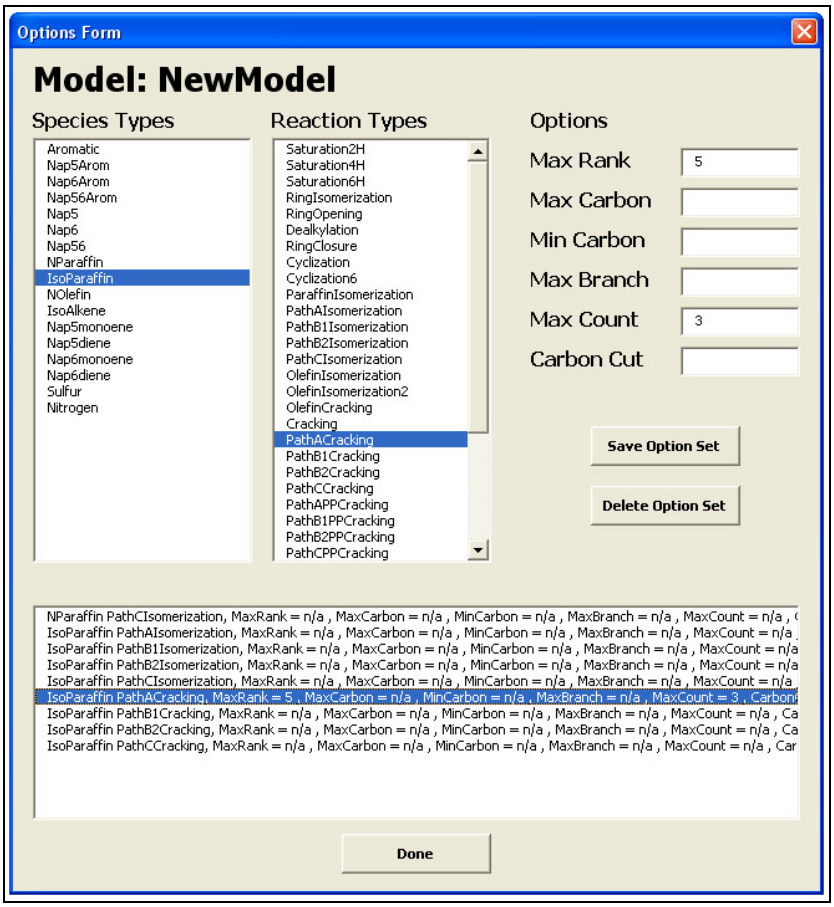
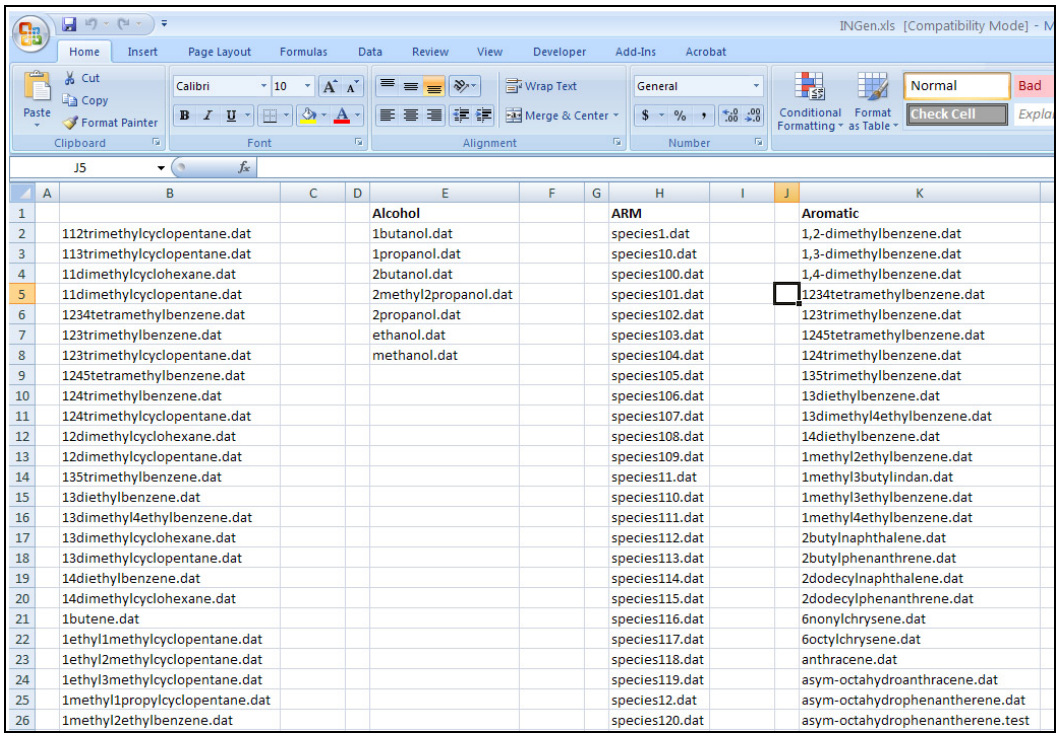


Figure 4.8 - Populated options form

If a previously entered combination is selected, the entered options are displayed, and the line is highlighted in the option set display. In addition, if a line in the option set is selected, its corresponding combination is selected in the species and reaction columns, and its options are displayed. Changes can be made to the options, by simply saving the new values with the "Save Option Set" button. If a species/reaction combination should be removed (not allowed), the "Delete Option Set" button should be pushed, thereby deleting it from the option set display (and the corresponding control file).

When all the species/reaction combination options have been entered, the modeler should select the "Done" button, signifying that the control file can be created based on the specified options. The modeler is taken to a new worksheet labeled "*Seeds*".

The *Seeds* sheet lists all the adjacency list files available to the user in the Data directory. Each column represents a different sub-directory, and in general, a special category of species. Users can create their own sub-directories and fill them with ".dat" format adjacency lists, the *Seeds* sheet will automatically refresh whenever it is initialized.



	A	B	C	D	E	F	G	H	I	J	K
1					**Alcohol**			**ARM**			**Aromatic**
2		112trimethylcyclopentane.dat			1butanol.dat			species1.dat			1,2-dimethylbenzene.dat
3		113trimethylcyclopentane.dat			1propanol.dat			species10.dat			1,3-dimethylbenzene.dat
4		11dimethylcyclohexane.dat			2butanol.dat			species100.dat			1,4-dimethylbenzene.dat
5		11dimethylcyclopentane.dat			2methyl2propanol.dat			species101.dat			1234tetramethylbenzene.dat
6		1234tetramethylbenzene.dat			2propanol.dat			species102.dat			123trimethylbenzene.dat
7		123trimethylbenzene.dat			ethanol.dat			species103.dat			1245tetramethylbenzene.dat
8		123trimethylcyclopentane.dat			methanol.dat			species104.dat			124trimethylbenzene.dat
9		1245tetramethylbenzene.dat						species105.dat			135trimethylbenzene.dat
10		124trimethylbenzene.dat						species106.dat			13diethylbenzene.dat
11		124trimethylcyclopentane.dat						species107.dat			13dimethyl4ethylbenzene.dat
12		12dimethylcyclohexane.dat						species108.dat			14diethylbenzene.dat
13		12dimethylcyclopentane.dat						species109.dat			1methyl2ethylbenzene.dat
14		135trimethylbenzene.dat						species11.dat			1methyl3butylindan.dat
15		13diethylbenzene.dat						species110.dat			1methyl3ethylbenzene.dat
16		13dimethyl4ethylbenzene.dat						species111.dat			1methyl4ethylbenzene.dat
17		13dimethylcyclohexane.dat						species112.dat			2butylnaphthalene.dat
18		13dimethylcyclopentane.dat						species113.dat			2butylphenanthrene.dat
19		14diethylbenzene.dat						species114.dat			2dodecylnaphthalene.dat
20		14dimethylcyclohexane.dat						species115.dat			2dodecylphenanthrene.dat
21		1butene.dat						species116.dat			6nonylchrysene.dat
22		1ethyl1methylcyclopentane.dat						species117.dat			6octylchrysene.dat
23		1ethyl2methylcyclopentane.dat						species118.dat			anthracene.dat
24		1ethyl3methylcyclopentane.dat						species119.dat			asym-octahydroanthracene.dat
25		1methyl1propylcyclopentane.dat						species12.dat			asym-octahydrophenantherene.dat
26		1methyl2ethylbenzene.dat						species120.dat			asym-octahydrophenantherene.test

Figure 4.9 - Seed selection sheet

Seeded species are selected by double clicking the cell to the left of the specie's name. A check mark should appear signifying its inclusion. If the box next to the bold faced column category is doubled clicked, all the species in the column will be selected for inclusion. Whenever a checked box is double clicked, it will return to a blank signifying exclusion of the species. Clicking a checked box next to the category will unselect every species in the category.

Figure 4.10 - Selected seed

After all wanted seed species have been selected, the modeler should proceed by continuing down the line of worksheet tabs to the "*Run*" sheet. As soon as the *Seeds* sheet is unselected, its included species are saved into the ".nam" seed file in the appropriate model directory.

The Run sheet will first test the model directory for a valid Option (control) file, Seed file, Shell file and Batch file. If all four files exist, INGen prompts the user to "Run the Model?"

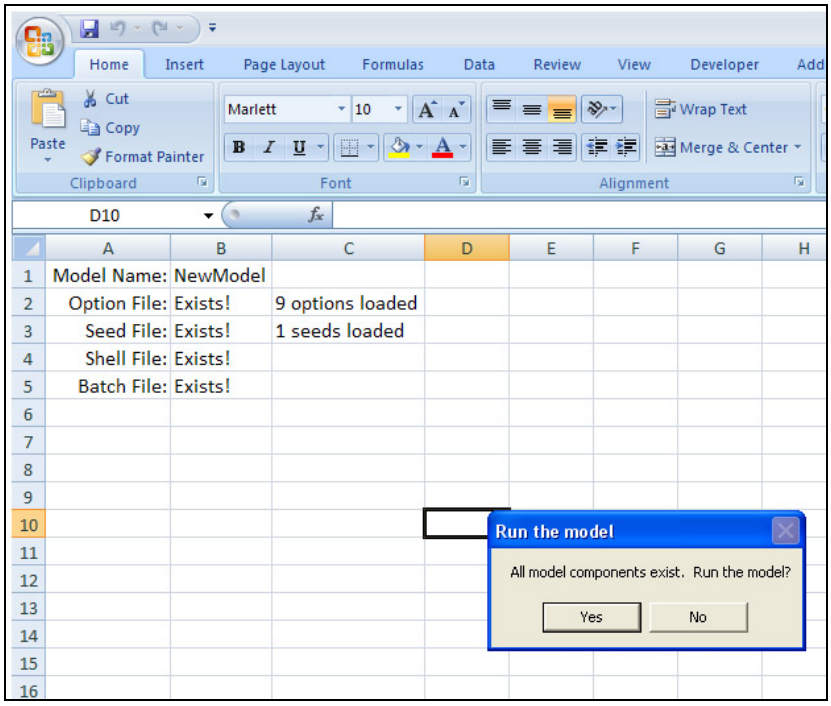


Figure 4.11 - Run confirmation

The modeler may select “No” in order to change options or seeds, or to wait until a later date to run the model. By reselecting the Run sheet, the modeler will be re-prompted to run the model.

If the modeler selects “Yes”, the batch script is run. When running, the modeler will see a command prompt window open with the various program calls and command line output of the programs. If output data already exists for a model run, it is first deleted.
Second, the shell script that runs the INGen backend is started (and completed) under the Cygwin environment. Third, the resultant network undergoes analysis and transformation via the eqn2kme.plx perl script. Fourth, the adjacency lists are converted to CML files via the CMLGen.plx perl script. And finally, if the modeler has ChemDraw and ChemScript available on the current machine, the DictionaryGen.py python script creates the list of IUPAC names.

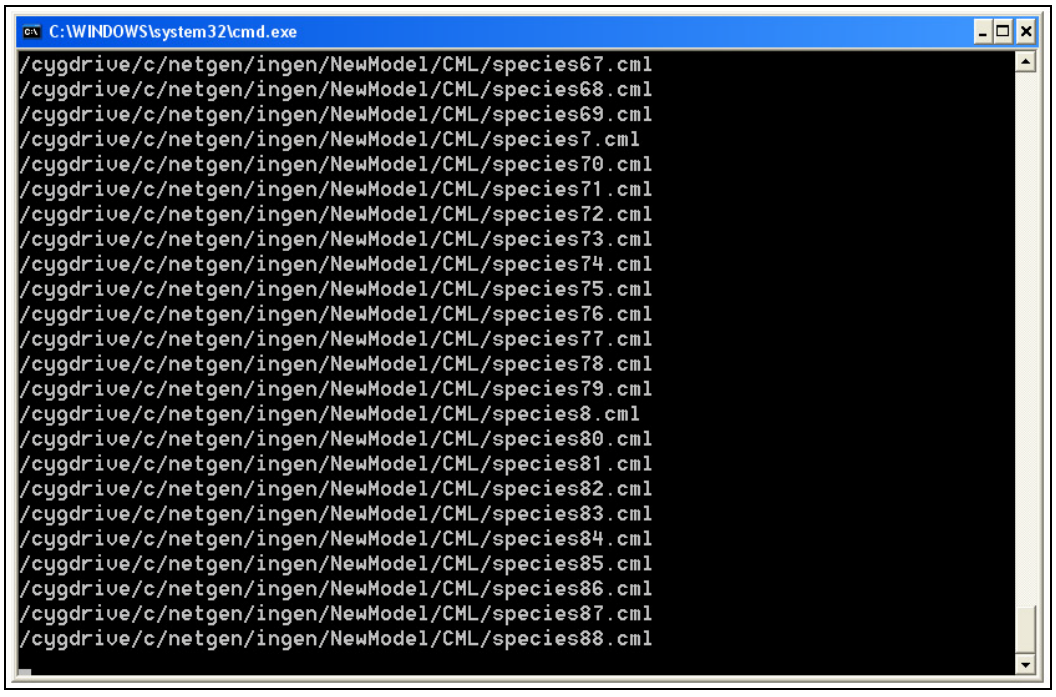
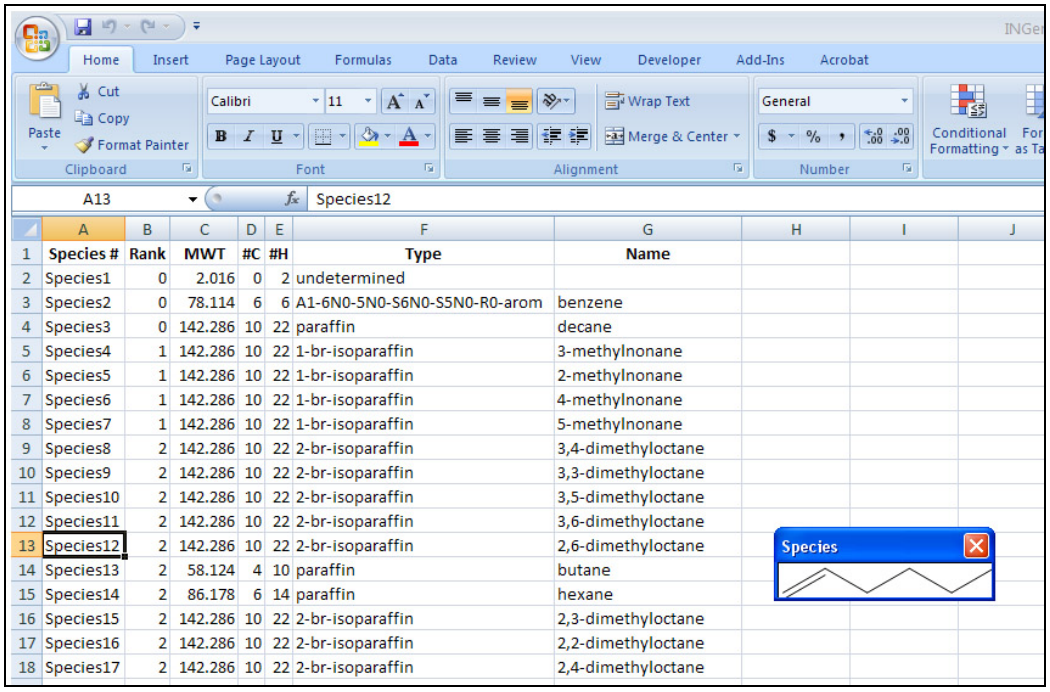


Figure 4.12 - Execution window

After a successfully completed run, three new sheets appear in the current workbook. The "*Species*" sheet contains information about the species. The "*Network*" sheet contains the kinetic network. The "*Statistics*" sheet contains information about the reaction pathways for the different species.

The *Species* sheet lists every species in the reaction network. Its rank, molecular weight, number of carbon atoms, number of hydrogen atoms, molecular type, and its IUPAC name (if ChemDraw was available) are also listed. In addition, if ChemDraw was available, clicking the speciesID will create a popup VBA form that displays the drawn structure of the species.

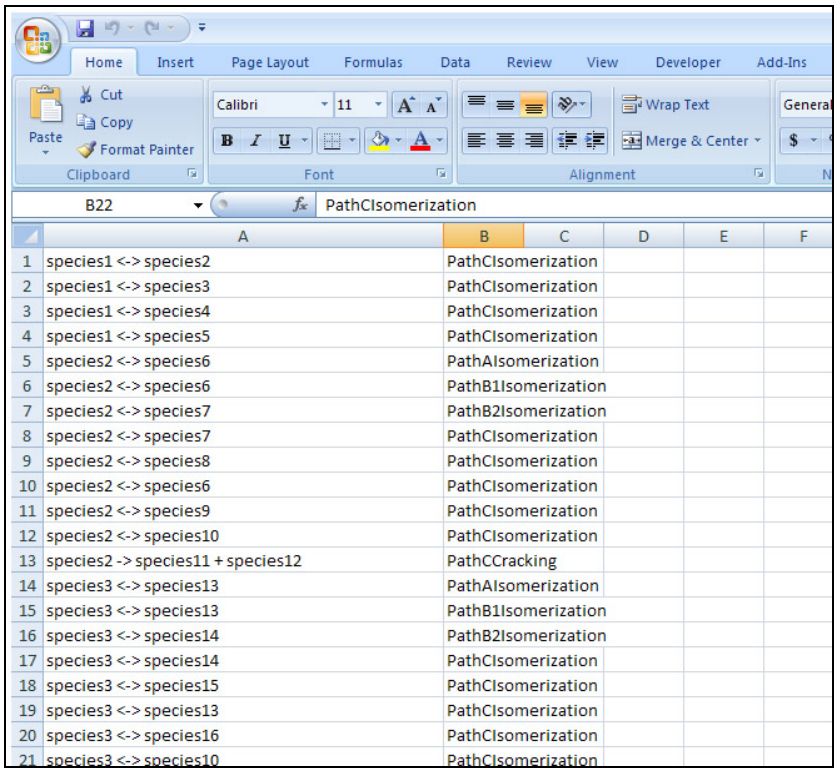


	A	B	C	D	E	F	G	H	I	J
1	**Species #**	**Rank**	**MWT**	**#C**	**#H**	**Type**	**Name**			
2	Species1	0	2.016	0	2	undetermined				
3	Species2	0	78.114	6	6	A1-6N0-5N0-S6N0-S5N0-R0-arom	benzene			
4	Species3	0	142.286	10	22	paraffin	decane			
5	Species4	1	142.286	10	22	1-br-isoparaffin	3-methylnonane			
6	Species5	1	142.286	10	22	1-br-isoparaffin	2-methylnonane			
7	Species6	1	142.286	10	22	1-br-isoparaffin	4-methylnonane			
8	Species7	1	142.286	10	22	1-br-isoparaffin	5-methylnonane			
9	Species8	2	142.286	10	22	2-br-isoparaffin	3,4-dimethyloctane			
10	Species9	2	142.286	10	22	2-br-isoparaffin	3,3-dimethyloctane			
11	Species10	2	142.286	10	22	2-br-isoparaffin	3,5-dimethyloctane			
12	Species11	2	142.286	10	22	2-br-isoparaffin	3,6-dimethyloctane			
13	Species12	2	142.286	10	22	2-br-isoparaffin	2,6-dimethyloctane			
14	Species13	2	58.124	4	10	paraffin	butane			
15	Species14	2	86.178	6	14	paraffin	hexane			
16	Species15	2	142.286	10	22	2-br-isoparaffin	2,3-dimethyloctane			
17	Species16	2	142.286	10	22	2-br-isoparaffin	2,2-dimethyloctane			
18	Species17	2	142.286	10	22	2-br-isoparaffin	2,4-dimethyloctane			

Figure 4.13 - Species properties

The *Network* sheet simply lists every reaction in the first column along with its reaction family type in the second column. Currently, the first column can simply be cut and

pasted in the KME environment in order to create the model equations, but plans for tighter integration (especially with the reaction families) are planned.



	A	B	C	D	E	F
1	species1 <-> species2	PathCIsomerization				
2	species1 <-> species3	PathCIsomerization				
3	species1 <-> species4	PathCIsomerization				
4	species1 <-> species5	PathCIsomerization				
5	species2 <-> species6	PathAIsomerization				
6	species2 <-> species6	PathB1Isomerization				
7	species2 <-> species7	PathB2Isomerization				
8	species2 <-> species7	PathCIsomerization				
9	species2 <-> species8	PathCIsomerization				
10	species2 <-> species6	PathCIsomerization				
11	species2 <-> species9	PathCIsomerization				
12	species2 <-> species10	PathCIsomerization				
13	species2 -> species11 + species12	PathCCracking				
14	species3 <-> species13	PathAIsomerization				
15	species3 <-> species13	PathB1Isomerization				
16	species3 <-> species14	PathB2Isomerization				
17	species3 <-> species14	PathCIsomerization				
18	species3 <-> species15	PathCIsomerization				
19	species3 <-> species13	PathCIsomerization				
20	species3 <-> species16	PathCIsomerization				
21	species3 <-> species10	PathCIsomerization				

Figure 4.14 - Network sheet

The *Statistics* sheet lists the number of reaction family occurrences (and the total number of reactions) for each species as a reactant, and as a product. Both the reactant and product lists are given consecutively in the first column.

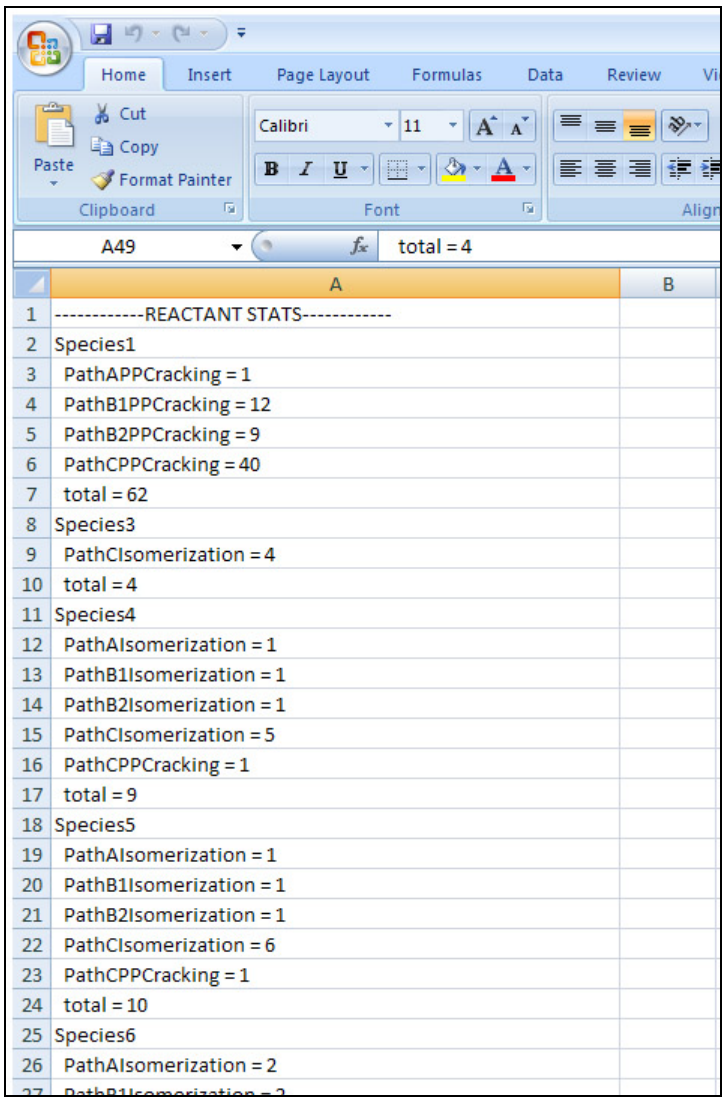


Figure 4.15 - Statistics sheet

Once the network has been analyzed, the modeler can return to any worksheet tab in order to change the options or rerun the model. Returning to the *INGen* tab will reset every page, and should be done only to create a new model.

4.5 Interface Conclusions

The process of creating a model kinetic network from scratch has never been as easy as with INGen. The concepts of model creation are presented in a logical, linear form

through the use of a user friendly interface. Customizability of species/reaction combination options, as well as seed selection (and creation/inclusion), allow a modeler to have great control over network generation. The interface and all automated steps allow the modeler to create and change the network quickly and easily.

Chapter 5. INGen Case Study – Paraffin Isomerization and Cracking

5.1 Background

In the preceding chapters, the inner workings of INGen were brought to light. The core functionality of a model builder, pared with the ease of use of a user interface led to the creation of a powerfully robust modeling tool. The tool was described in minute detail; its design based on theoretical situations and usefulness. This chapter and the following will attempt to refocus this work on solutions to real world scenarios by presenting two cases for network creation and analysis.

The first case study involves the catalytic isomerization and cracking of paraffins. A large scale kinetic network was proposed for the purpose of further thermodynamic analysis. The original intent was to create a comprehensive C70 (a maximum carbon number of 70) pathways system, with each species having a maximum of three branches. In addition, because the thermodynamics of the system were to be studied in detail, the pathways would need to reflect the different energy levels involved in carbenium ion creation and transition for each reaction.

As laid out in the previous chapters, the carbenium ion pathways were added to the INGen modeling system for both cracking and branching isomerization. Reactions that share both common reactants and products were allowed to be duplicated as long as two separate carbenium ion pathways were possible. Therefore, the number of reactions was likely to increase from the non-carbenium aware pathway, but not the number of species.

It should be noted that the required network would encompass only PCP type isomerizations (see Chapters 2 & 3); no methyl shifts were allowed. In addition, the number of branches was not to exceed three, and each branch length could not exceed two (an ethyl branch). As a continuing assumption, no primary or methyl carbenium ions were allowed at any step of the pathway. Finally, the conceptual starting point was the normal C70 paraffin.

A normal C70 paraffin cannot crack on its own without forming a primary carbenium ion, therefore isomerization was the necessary first step. The only isomerization pathway that a normal paraffin could undertake is Path C isomerization wherein a secondary carbenium ion is created, transferred to a different secondary carbenium ion, and then removed.

Therefore, normal paraffin species were checked for only the Path C Isomerization species/reaction combination. Isoparaffins, on the other hand, can encompass all four isomerization pathways as well as all four cracking pathways. The initial goal was to run the network with the nine stated species/reaction combinations unbounded except for branch number.

5.2 Results

The unbounded case, as described, presented multiple problems. Computational problems occurred with regards to memory allocation. Various backend changes were made to correct and limit the amount of memory a species could use, the sheer number of large species could not be handled on a typical computer system.

After the memory issue was ameliorated, allowing a tenfold additional growth to the network, the hugely isomeric nature of a C70 system was still beyond the capability of

computer (and human) analysis. Extrapolation based on a power law fit of C20 and under data shows that for a three branched network, the expected number of species is around 1 x 10^7 and the number of reactions is about 3 x 10^8.

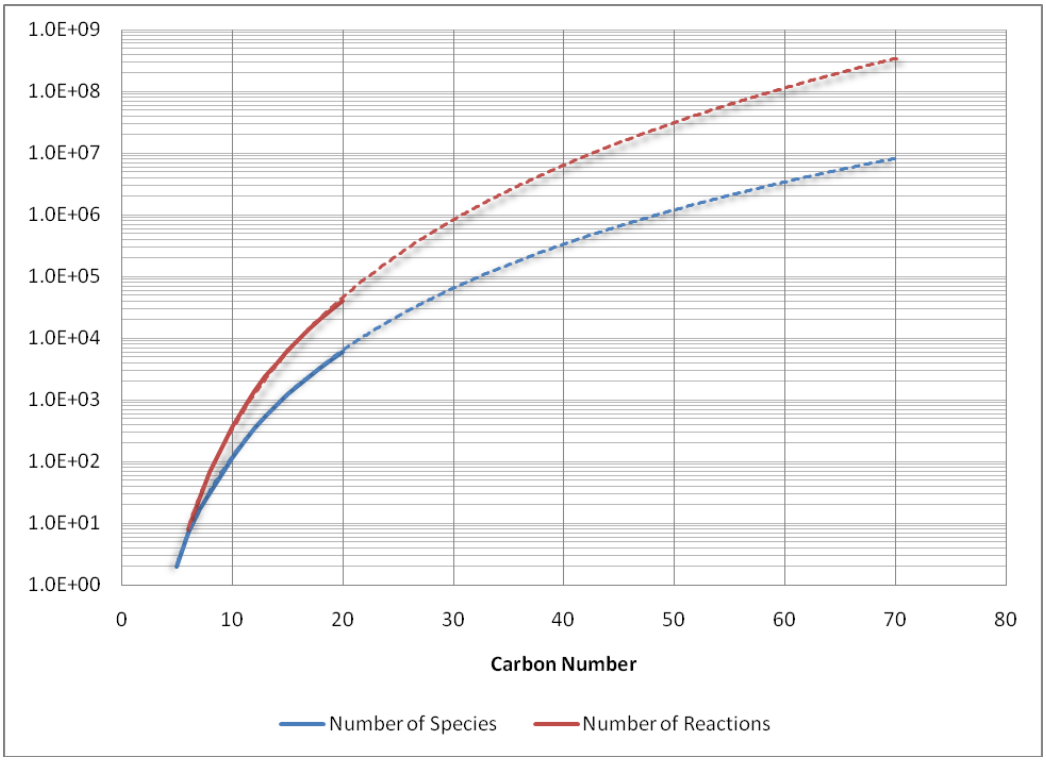


Figure 5.1 - Extrapolation of C70 network data for an isomerization & cracking network bound to a maximum of three branches per species

Therefore, the original modeling concept was modified it two different ways. In one method, changes were made to the limitations for both the number of reactions and the maximum rank of species. The other type of change would involve rewriting the problem to fit the situation; that is, build an in depth network based on a normal paraffin of shorter length.

Because network insights could be derived from a smaller network, the second methodology was chosen first. The first step involved maximizing the carbon number

before the computer ran out of addressable memory. Because the scope of the network had already been significantly decreased, the initial experiment was conducted with no branch limitations on the network.

It was found that the computer could handle a fully expounded C16 (maximum carbon number of 16) isomerization and cracking network. Further increases to the carbon number led to problems with the computer's memory. The projected number of species and reactions for an unlimited number of branches in a C70 isomerization and cracking network were about 3×10^{14} and 3×10^{11} respectively.

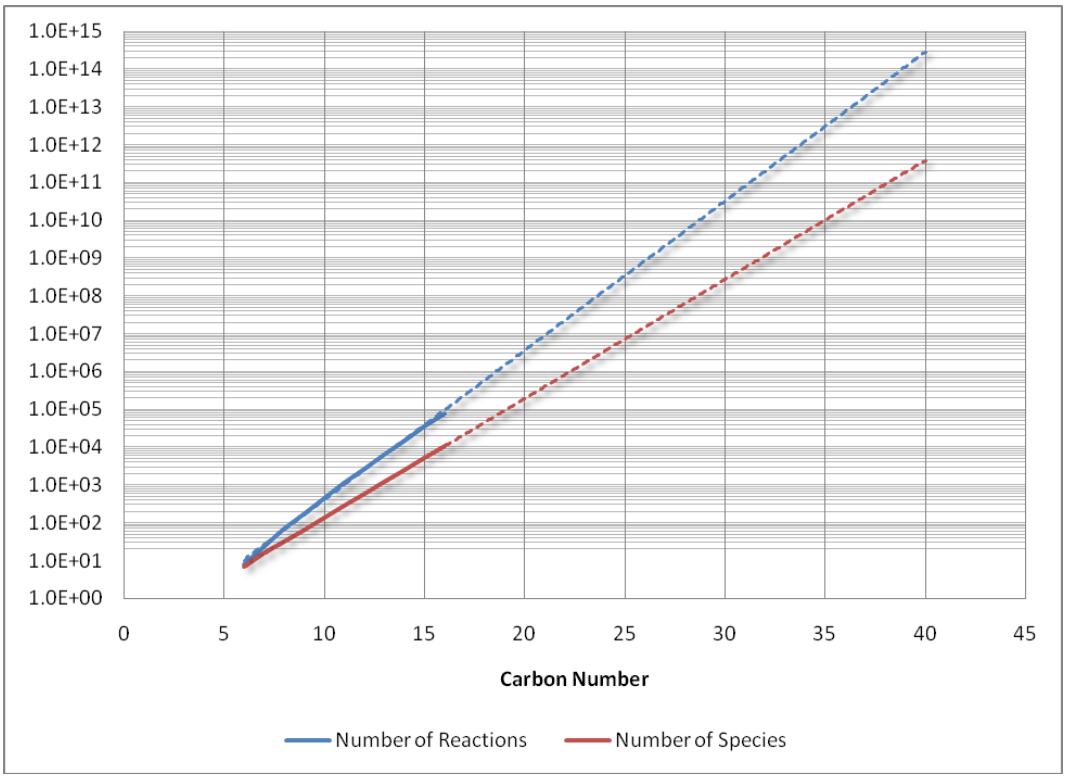


Figure 5.2 - Extrapolation of C70 network data for an isomerization & cracking network unbound by the number of branches per species

If each species took only a single Kilobyte of computer memory, it would take almost an Exabyte of memory to create all the species (almost a billion times the average amount of

memory in a modern computer in 2009).

Rather than attempt any further extrapolation, network analysis and limitation techniques were performed on the smaller network. Analysis of the “unbound” C17 network revealed consistent exponential increases in each of the eight reactions pathways with carbon number.

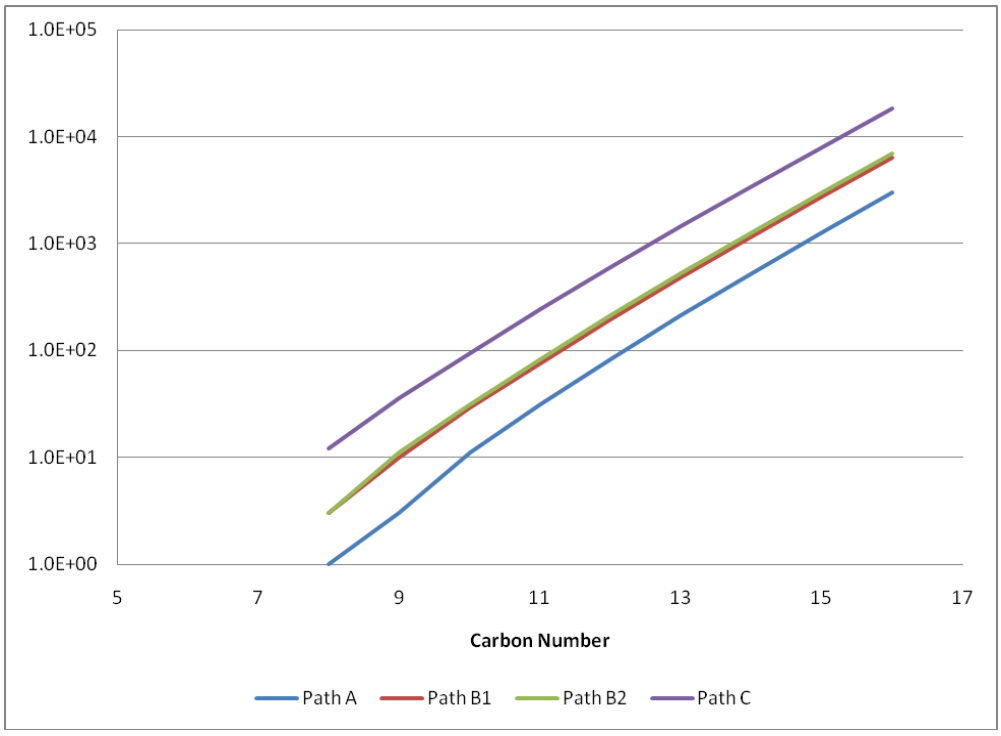


Figure 5.3 - The number of "unbound" cracking reactions as a function of carbon number

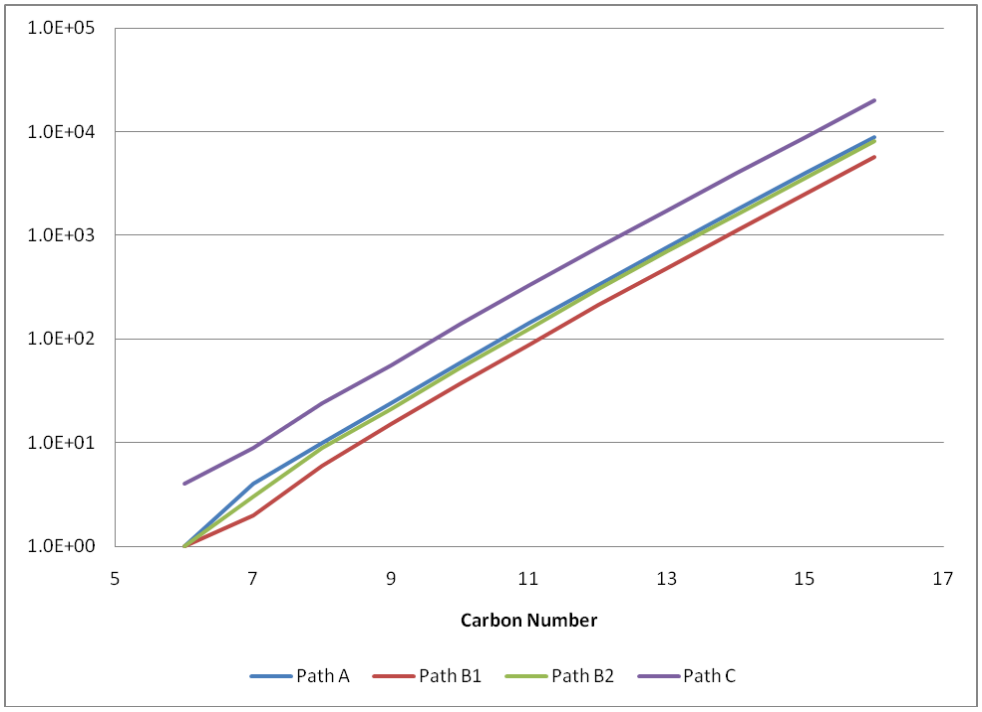


Figure 5.4 - The number of "unbound" isomerization reactions as a function of carbon number

The ratio of the number of each of the two major reaction pathways to the total number of reactions approached 50% with an increase in carbon number. Therefore, as the size of the initial paraffin increases, the size of the reaction network is not lopsided towards cracking or isomerization. Therefore limitation techniques should be administered to both cracking and isomerization pathways.

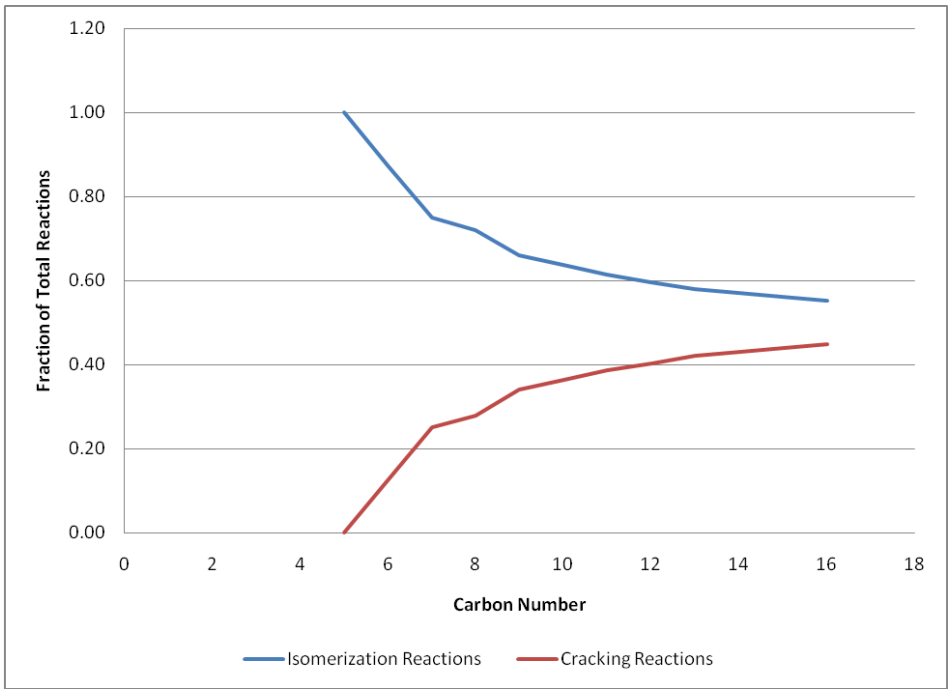


Figure 5.5 - Distribution of total reactions between the two major reaction families for an "unbound" network

Next the fractionalization of the number of total reactions with respect to carbenium ion pathways was studied. Each pathway that involves the post-reaction creation of a tertiary carbenium ion (paths A and B1) encompasses a 15% fraction of the total number of possible reactions. The transition of an initial tertiary to a secondary carbenium ion (path B2) had a slightly higher fraction at about 20%. The remaining 50% was comprised of path C reactions in which secondary carbenium ions were maintained before and after reaction. The large dominance of path C type reactions will also help govern the selection of reaction network limitations.

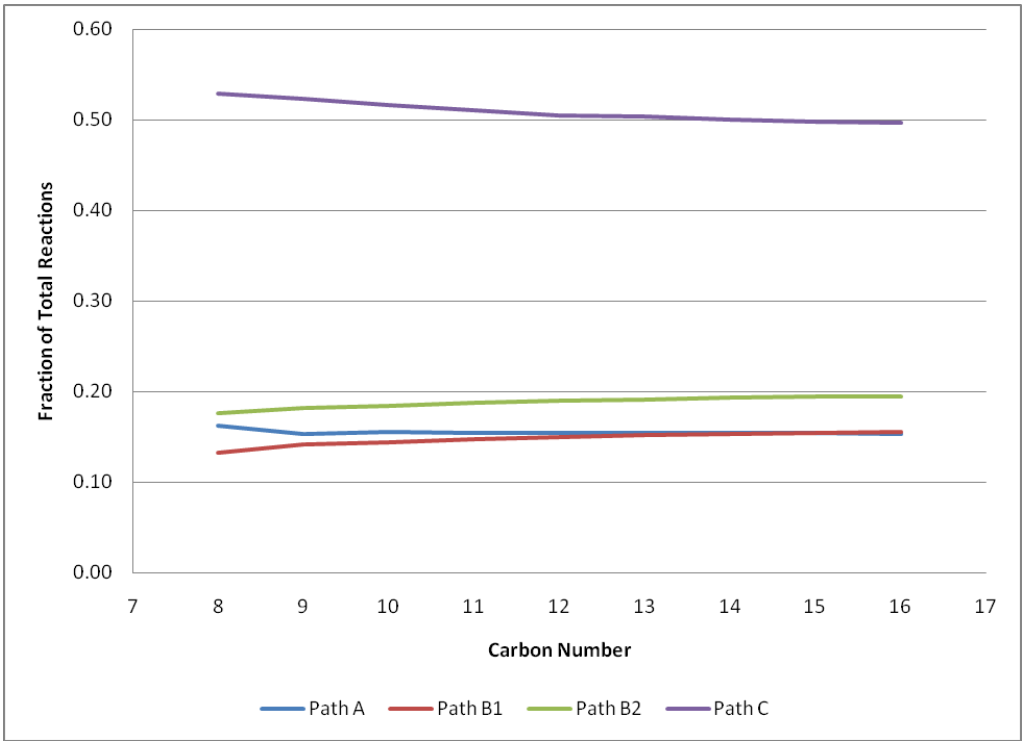


Figure 5.6 - Pathways fractionalization of total reactions

In the next stage, various network limitations were then placed on a C20 network in order to study their affect on the total number of species and reactions. A baseline for network control was given by limiting only the number of path C isomerization reactions that can occur for each unique carbon number/branch count combination. After adding limits to the path C cracking reactions, there was not much change in the number of species, but the number of reactions was noticeably decreased.

Limiting the isomerization along both paths C and B2 led to a slight decrease in the number of species, and the decrease in the number of reactions was about equivalent to that of the previous case. Placing additional limits on the C and B2 cracking pathways continued to slightly decrease the number of species and reactions.

A factor of ten decrease in species and reactions was obtained upon instilling limitations on

the resultant tertiary carbenium ion pathways (A & B1). Those lower energy pathways were weighted with double the number of possible reactions than those of the B2 & C reaction pathways. The ordinate of the graphs below is based on the path C (and B2) limitations.

Finally, for the purpose of methodology comparison, a limitation of at most three branches per species was imposed on the fully limited case given above. The resulting decreases to both number of species and reactions were noticeable, but not as large as the step changes observed when all four pathways were limited.

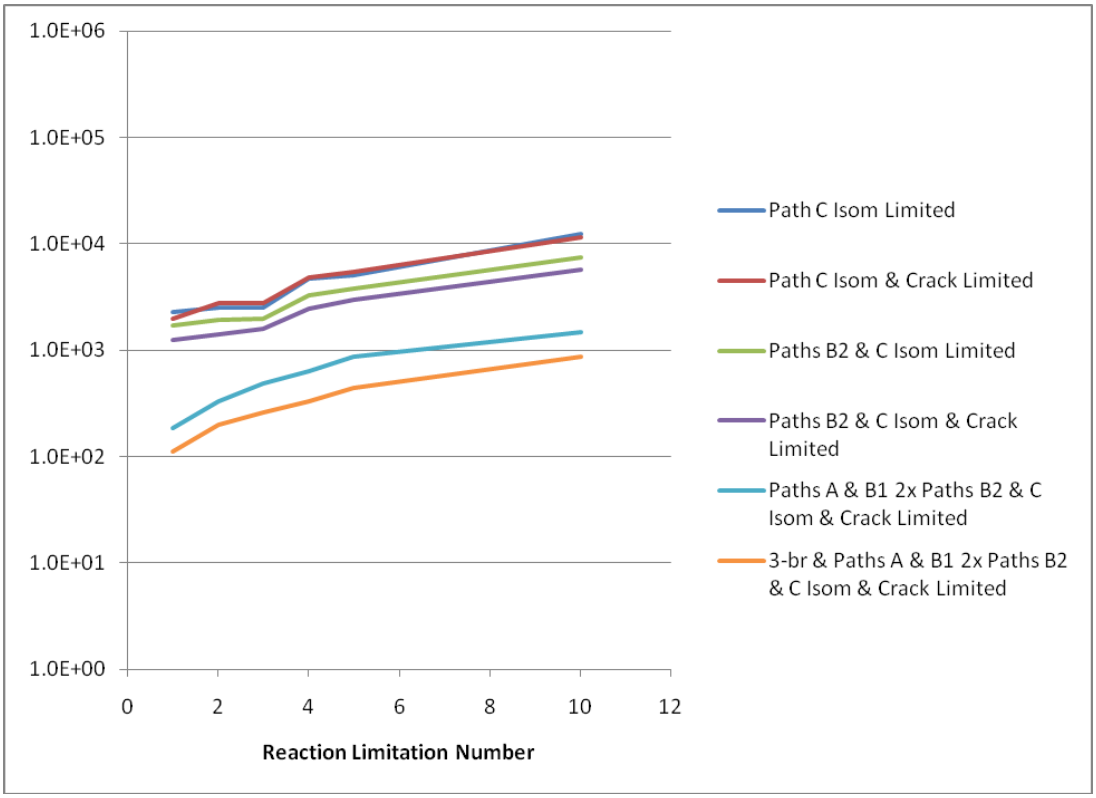


Figure 5.7 - Species number as a function of network limitations

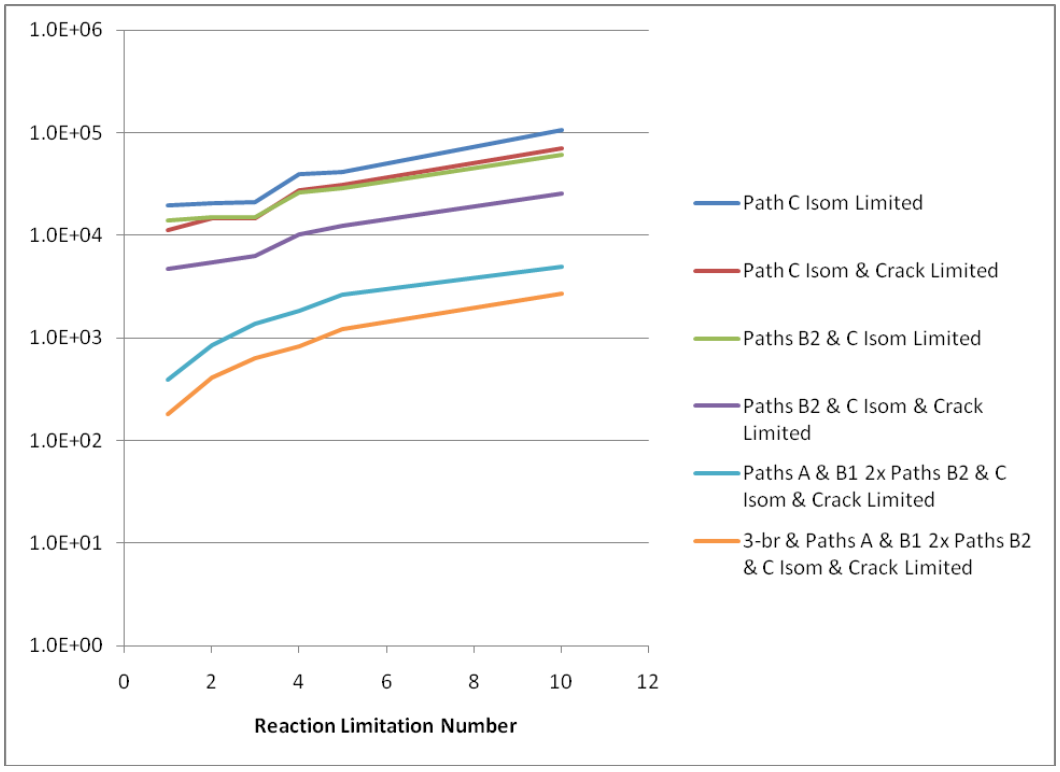


Figure 5.8 - Reaction number as a function of network limitations

Further branch analysis was performed by comparing the number of species and number of reactions over a range of carbon numbers for networks that were limited to species that contained one, two, three, or unbound number of branches. As expected, limiting the network by the number of possible species branches has a profound effect on the number of species, but may not describe the network in as much detail as would be desired. Current analytical techniques can distinguish up to three separate branches on a species, and therefore representative species should be available in the reaction network.

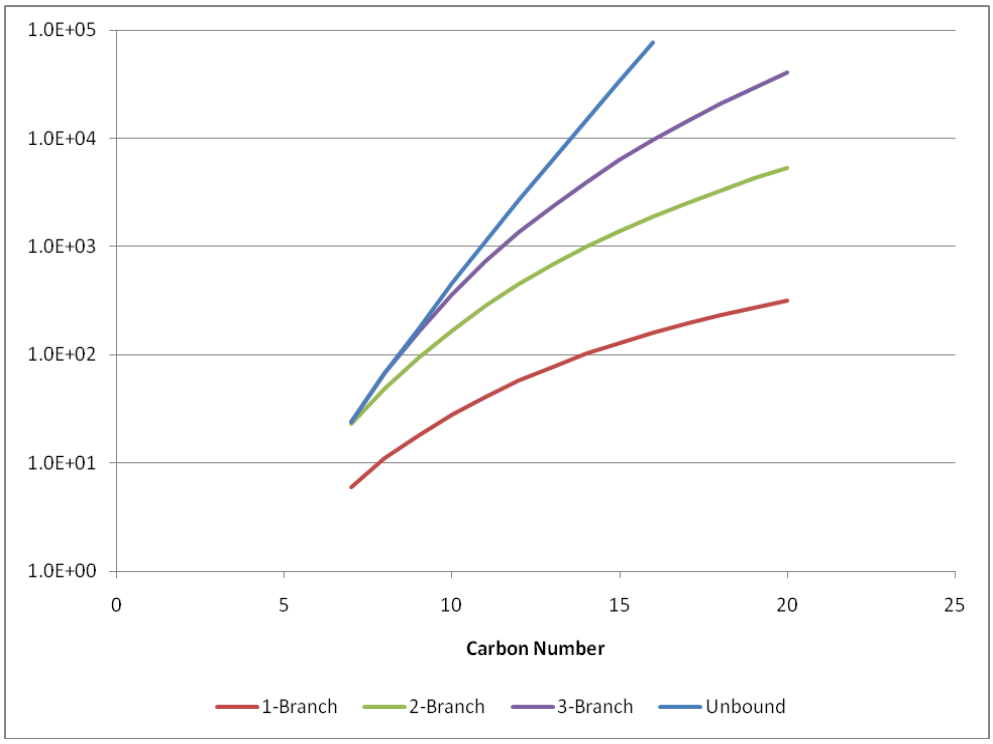


Figure 5.9 - The effect of branching limitations on the number of reactions

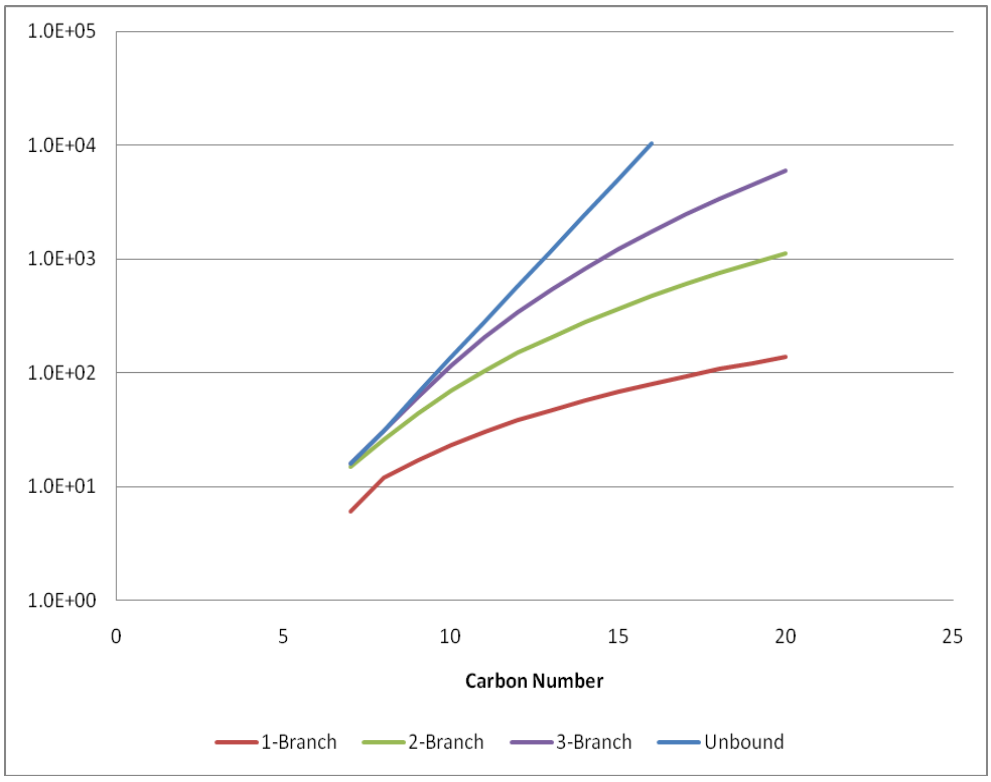


Figure 5.10 - The effect of branching limitations on the number of species

One final experiment was performed on the smaller carbon number network. Disregarding all cracking reactions, a study of isomers was conducted on a C20 system. The results shown below describe the number of total species if branching limitations are placed on the network. For the case of a single branch limitation, the number of isomers is linear. For a limitation of two branches, the result is a second order polynomial. Similarly, a three branch limitation resulted in a third order polynomial correlation.

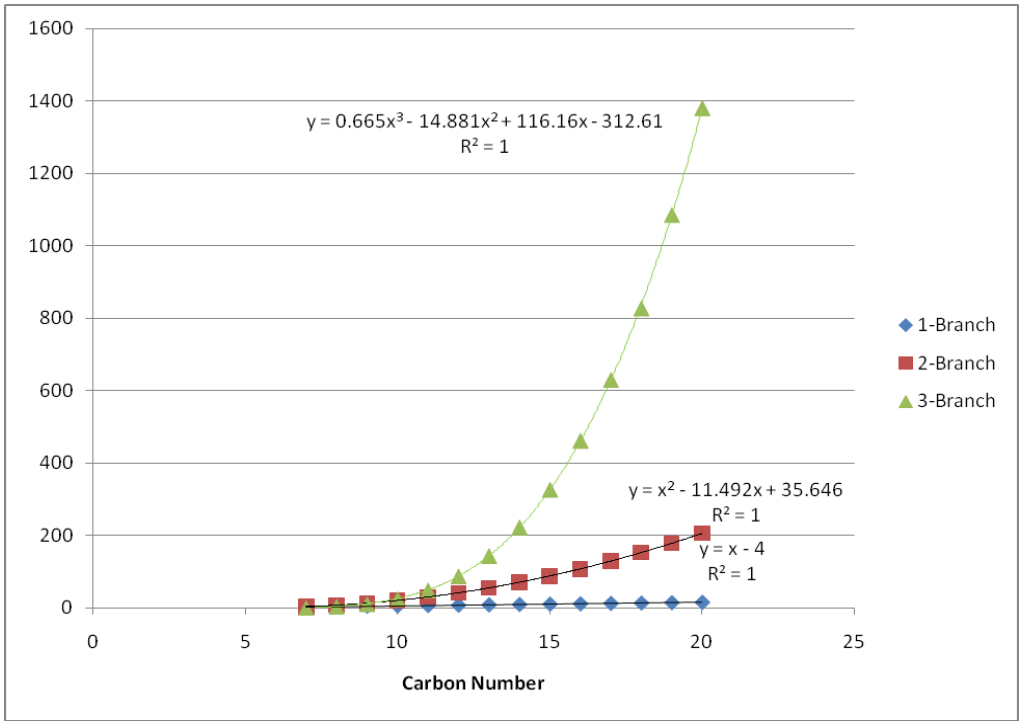


Figure 5.11 - Number of C20 paraffin isomers

With all small experiments conducted and results considered, focus shifted to obtaining results for a C70 network.

Numerous concurrent network limitations were necessary to obtain data for the C70 network. Combinations of reaction network control parameters were tried until a network could be created within the confines of the computer's memory. Using a C70 normal

paraffin seed, a viable network was generated using a maximum of five normal-parafin path C isomerization reactions per carbon number, 40 iso-paraffin path A and path B1 isomerization and cracking reactions at a maximum rank of 50, and 20 iso-paraffin path B1 and path C isomerization and cracking reactions at a maximum rank of 50.

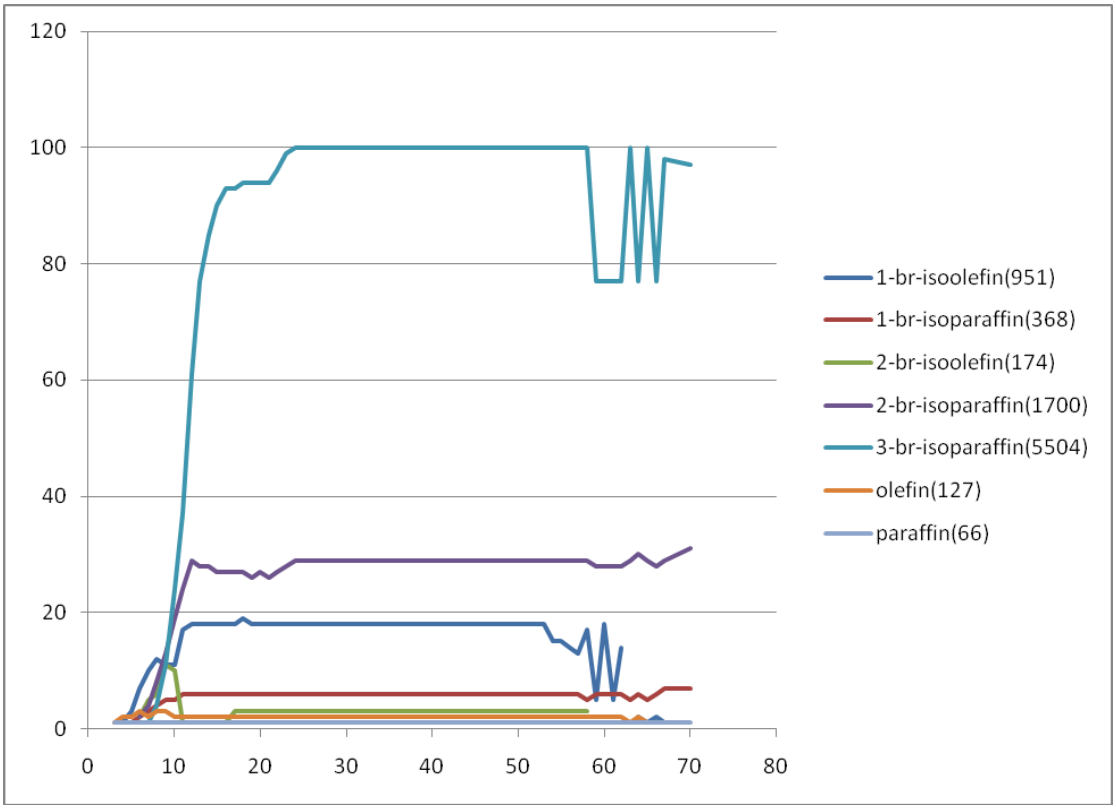


Figure 5.12 – Composition of the initial computationally viable C70 network

The network contained 8,890 species and 23,337 reactions. Whereas the network could be studied at this size, efforts were made to pare it down further.

By using advanced seeding techniques and severe network limitations, pared down versions of the C70 isomerization and cracking network were created. The networks were tested for robustness by examining the availability of each species type (zero, one, two, and three branched paraffins and olefins) for each carbon number. If the plots showed gaps,

the network was deemed unfit, and the control parameters changed.

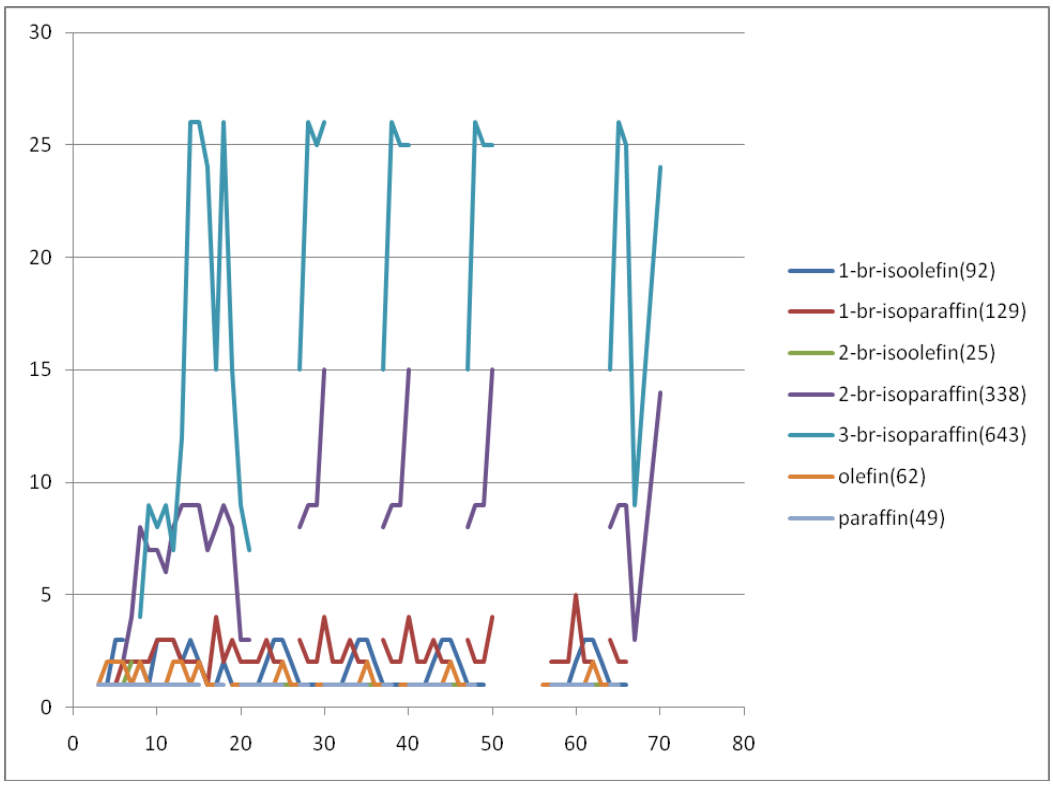


Figure 5.13 - An example of an unfit C70 network

Ultimately, a network with 2,698 species and 8,181 reactions was found using the following parameters: C70, C60, C50, C40, and C30 normal-paraffin seeds, a maximum of five normal-parafin path C isomerization reactions per carbon number, a maximum of ten path A, path B1, and path B2 iso-paraffin isomerization and cracking reactons with a maximum rank of 12, and a maximum of six path C isomerization and cracking reactions with a maximum rank of 3.

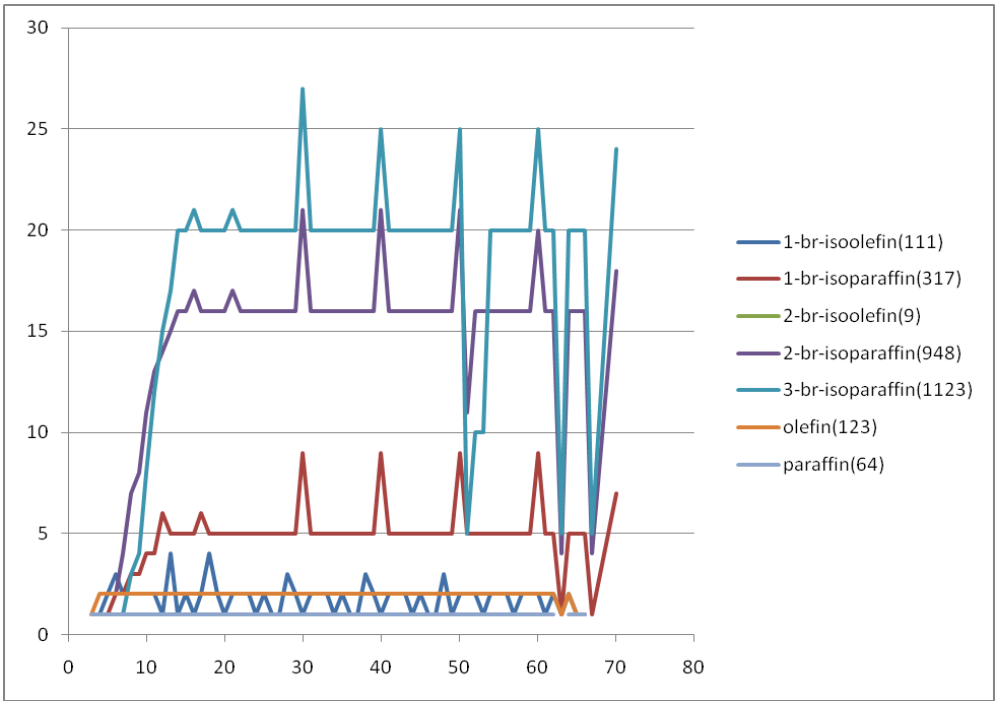


Figure 5.14 - An example of a small C70 network with all important species represented

5.3 Conclusion

INGen's ability to handle carbenium ion pathways allowed for the creation of detailed branching isomerization and cracking networks. Large carbon numbers still posed a daunting problem with respect to the number of species and reactions that were feasible. INGen made possible the rapid resolution of numerous studies in network limitation and analysis techniques. Without INGen, the network creation and analysis would need to be conducted largely by hand over a much longer time frame.

Ultimately, a relatively small C70 network was created by INGen that encompassed all important species. When paired with LFER approximations through KME or similar model equation building software, the nine selected reaction families will allow the thousands of species and reactions to be compressed into a much smaller set of parameters.

Chapter 6. INGen Case Study – Building a Specific Hydro-Processing Network

6.1 Background

In Chapter 5, the INGen tool was used to create a large network of reactions and species for a paraffin isomerization and cracking process. This chapter applies the INGen tool to the problem of creating a smaller, more controlled hydro-processing network. Both the reactants and the products of the network have undergone detailed experimental analysis, and the problem was posed to create a representative network that encompasses the chosen representative species.

The reactant feed was shown to contain 90% of the overall representative model species. The final 10% were to be created via reaction only. Two possible methods presented themselves. In the first, the reaction would be controlled much like the isomerization and cracking network mentioned in the previous chapter. The number of reactions per carbon number, the maximum rank, and the maximum number of branches would be used to control the network growth. Post generation analysis would be used to reclassify (into their representative species) or remove unwanted species.

The second method, would limit the reaction entirely to seeded species and therefore be considered a “rank zero” model. The high percentage of species found in both the feed and the model footprint throw credence to this methodology.

Both network limitation methods were studied, and their result presented below.

6.2 Results

The first step in generating the hydro-processing network was the creation of previously non-existent species that were to be included as seeds to the model builder. INGen's integration with CambridgeSoft's ChemDraw software changed the task from the extreme tedium of manual adjacency list generation to that of a simple exercise of drawing the species with the help of CAD software. All missing species were generated flawlessly in a few short hours with INGen and ChemDraw rather than the error-prone days that would be necessary for manual creation.

Included below is a list of all species measured in the feed broken into groups based on their type.

Table 6.1 - Normal-paraffin species in the hydroprocessing feed

Name	*Type*	*#C*	*#S*	*#N*
n-pentane	nP	5		
n-heptane	nP	7		
n-octane	nP	8		
n-nonane	nP	9		
n-decane	nP	10		
n-undecane	nP	11		
n-dodecane	nP	12		
n-tridecane	nP	13		
n-tetradecane	nP	14		
n-pentadecane	nP	15		
n-hexadecane	nP	16		
n-heptadecane	nP	17		
n-octadecane	nP	18		
n-nonadecane	nP	19		
n-eicosane	nP	20		
n-heneicosane	nP	21		
n-docosane	nP	22		
n-tricosane	nP	23		
n-pentacosane	nP	25		

Table 6.2 - Iso-paraffin species in the hydroprocessing feed

Name	*Type*	*#C*	*#S*	*#N*
2methylpropane	iP	4		
2methylpentane	iP	6		
2methylhexane	iP	7		
2methylheptane	iP	8		
2methyloctane	iP	9		
2methylnonane	iP	10		
2methyldecane	iP	11		
2methylundecane	iP	12		
2-methyl-dodecane	iP	13		
2-methyl-tridecane	iP	14		
3-methyl-tetradecane	iP	15		
3-methyl-pentadecane	iP	16		

Table 6.3 - Single aromatic ring species in the hydroprocessing feed

Name	*Type*	*#C*	*#S*	*#N*
pyridine	A	5		1
aniline	A	6		1
benzene	A	6		
toluene	A	7		
ethylbenzene	A	8		
metaxylene	A	8		
orthoxylene	A	8		
paraxylene	A	8		
123trimethylbenzene	A	9		
1methyl3ethylbenzene	A	9		
propylbenzene	A	9		
1-methyl-2-(1-methylethyl)-benzene	A	10		
1-methyl-3-propyl-benzene	A	10		
1234tetramethylbenzene	A	10		
13diethylbenzene	A	10		
2-ethyl-1,4-dimethyl-benzene	A	10		
2-methylpropyl-benzene	A	10		
1,3-diethyl-5-methyl-benzene	A	11		
1,3-dimethyl-5-(1-methylethyl)-benzene	A	11		
1-ethyl-3-(1-methylethyl)-benzene	A	11		

1-methyl-4-(2-methylpropyl)-benzene	A	11
ethyl-1,2,4-trimethyl-benzene	A	11
pentamethyl-benzene	A	11
1,3,5-trimethyl-2-propyl-benzene	A	12
1,4-dimethyl-2-(2-methylpropyl)-benzene	A	12
1-(1-methylethenyl)-3-(1-methylethyl)-benzene	A	12
1-(2-butenyl)-2,3-dimethyl-benzene	A	12
3-methyl-2-phenyl-2-pentene	A	12
hexylbenzene	A	12
1-ethenyl-4-(1-methylbutyl)-benzene	A	13
1-methyl-3-hexyl-benzene	A	13
heptylbenzene	A	13
1-methylheptyl-benzene	A	14
nonylbenzene	A	15
decyl-benzene	A	16

Table 6.4 - Two fused aromatic ring species in the hydroprocessing feed

Name	*Type*	*#C*	*#S*	*#N*
naphthalene	AA	10		
1-methyl-naphthalene	AA	11		
2-methyl-naphthalene	AA	11		
1,3-dimethyl-naphthalene	AA	12		
1,4-dimethyl-naphthalene	AA	12		
1,5-dimethyl-naphthalene	AA	12		
1,7-dimethyl-naphthalene	AA	12		
1-ethyl-naphthalene	AA	12		
2,3-dimethyl-naphthalene	AA	12		
2,6-dimethyl-naphthalene	AA	12		
2,7-dimethyl-naphthalene	AA	12		
2-ethyl-naphthalene	AA	12		
1,4,5-trimethyl-naphthalene	AA	13		
1,4,6-trimethyl-naphthalene	AA	13		
1,6,7-trimethyl-naphthalene	AA	13		
1-methyl-7-isopropyl-naphthalene	AA	13		
1-propyl-naphthalene	AA	13		
2,3,6-trimethyl-naphthalene	AA	13		
2-isopropyl-naphthalene	AA	13		
1,2,3,4-tetramethyl-naphthalene	AA	14		
1,4,5,8-tetramethyl-naphthalene	AA	14		

5-ethenyl-1,4-dimethyl-naphthalene	AA	14

Table 6.5 - Two non-fused aromatic ring species in the hydroprocessing feed

Name	*Type*	*#C*	*#S*	*#N*
biphenyl	A-A	12		
3-methyl-biphenyl	A-A	13		
1,1-diphenyl-ethane	A-A	14		
2-ethyl-biphenyl	A-A	14		
4,4'-dimethyl-biphenyl	A-A	14		
1-benzyl-3,5-dimethyl-benzene	A-A	15		
3-isopropyl-biphenyl	A-A	15		
4'-methyl-stilbene	A-A	15		
4,4'-dimethyl-diphenylmethane	A-A	15		
4-isopropyl-biphenyl	A-A	15		
3,3',5,5'-tetramethyl-biphenyl	A-A	16		

Table 6.6 - Species with single fused aromatic and napthenic rings in the hydroprocessing feed

Name	*Type*	*#C*	*#S*	*#N*
indan	AN5	9		
1-methyl-indan	AN5	10		
5-methyl-indan	AN5	10		
1,6-dimethyl-indan	AN5	11		
2,2-dimethyl-indan	AN5	11		
2-ethyl-indan	AN5	11		
4,7-dimethyl-indan	AN5	11		
1,1,5-trimethyl-indan	AN5	12		
1,2,2-trimethyl-indan	AN5	12		
1,5,7-trimethyl-indan	AN5	12		
4,5,7-trimethyl-indan	AN5	12		
4-propyl-indan	AN5	12		
1-methyl-tetralin	AN6	11		
5-methyl-tetralin	AN6	11		
1,3-dimethyltetralin	AN6	12		
1,5-dimethyl-tetralin	AN6	12		
6-ethyl-tetralin	AN6	12		
1,1,6-trimethyl-tetralin	AN6	13		
2,5,8-trimethyl-tetralin	AN6	13		

Table 6.7 - Species with three fused rings in the hydroprocessing feed

Name	*Type*	*#C*	*#S*	*#N*
phenanthrene	AAA	14		
2-methyl-phenanthrene	AAA	15		
2,5-dimethyl-phenanthrene	AAA	16		
fluorene	AN5A	13		
2-methyl-fluorene	AN5A	14		
2,3-dimethyl-fluorene	AN5A	15		
9,10-dihydro-1-methyl-phenanthrene	AN6A	15		
9,10-dihydro-2-methyl-anthracene	AN6A	15		

Table 6.8 - Species with special ring structures in the hydroprocessing feed

Name	*Type*	*#C*	*#S*	*#N*
9,9-dimethyl-1,4-dihydro-1,4-methano-naphthalene	A(N5)N6	12		
4,6,8-trimethyl-azulene	A7A5	13		
7-ethyl-1,4-dimethyl-azulene	A7A5	14		
quinoline	ANit6	9		1
indole	ANit5	8		1
1a,9b-dihydro-1H-cyclopropa[l]phenanthrene	AN6(N3)A	15		
2-phenyl-naphthalene	AA-A	16		
benzothiophene	AS5	8	1	
dibenzothiophene	AS5A	12	1	
3-methyldibenzothiophene	AS5A	13	1	
46-DiMeDBT	AS5A	14	1	

In addition to the molecules found in the feed, the following tables list species that were experimentally found in the products. These species will need to appear in the traditionally limited model, and will need to be seeded in the rank zero model.

Table 6.9 - Species found in the hydroprocessing products

Name	*Type*	*#C*	*#S*	*#N*
1-(1,1-dimethylethyl)-4-ethenyl-benzene	A	12		
1-methylheptyl-benzene	A	14		
ethyl-indene	AA5	11		
1,2,3,4-tetrahydrophenanthrene	AAN6	14		

5-methylindan	AN5	10
1-6-dimethylindan	AN5	11
tetralin	AN6	10
5-methyltetralin	AN6	11
1,5-dimethyltetralin	AN6	12
1,1,6-trimethyltetralin	AN6	13
ethylcyclopentane	N5	7
isopropylcyclopentane	N5	8
butylcyclopentane	N5	9
methylcyclohexane	N6	7
ethylcyclohexane	N6	8
1-ethyl-4-methyl-cyclohexane	N6	9
butylcyclohexane	N6	10
1,2,3,4,5,6,7,8-octahydrophenanthrene	NAN	14
1,2,3,4,5,6,7,8-octahydro-2-methylphenanthrene	NAN	15

The first method of building the network revolved around the multiple case studies of different reactant/species combination options. If an expected product was not found, certain options were loosened and the network recreated. Much like in the isomerization study of the previous chapter, if the reaction created too many species, network control options were tightened to help control the growth of the model.

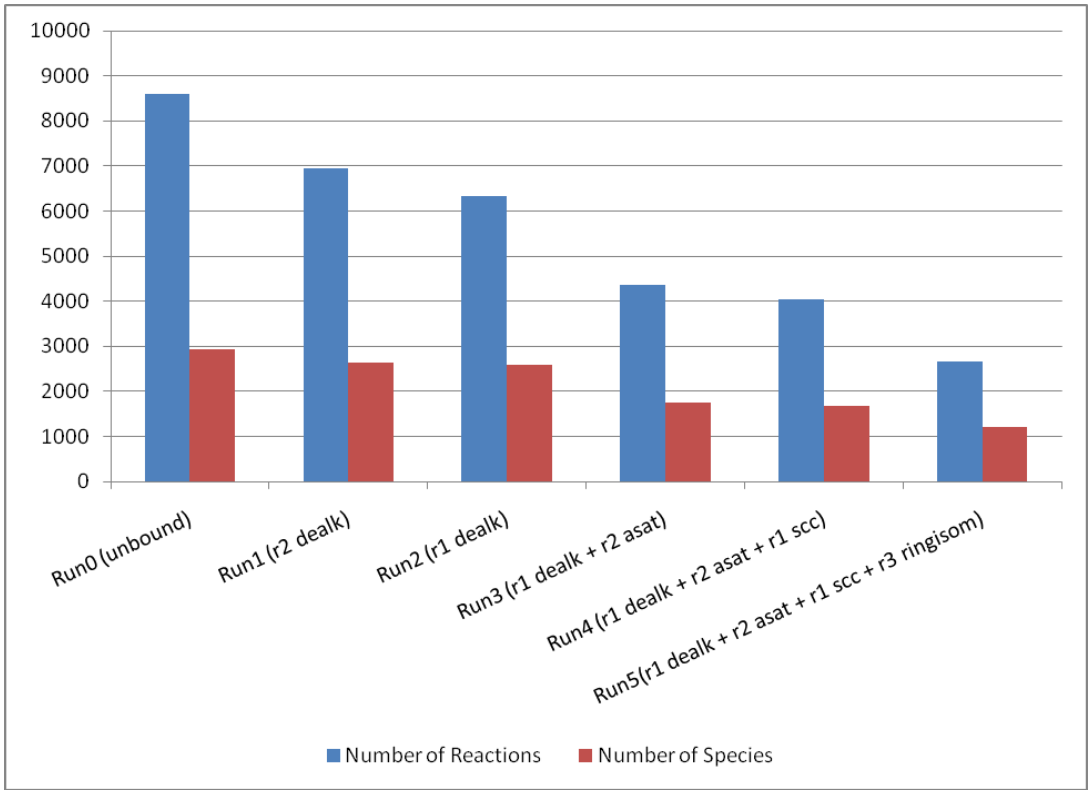


Figure 6.1 - Results of rank limitations for a hydropessing network

Within five trial runs, the total number of network reactions was decreased by about 70% and the number of species was reduced by about 60%. Limitations were placed based on an understanding of the chemistry coupled with the need for fewer reactions of any given type.

For each run, the number of forward reactions was compared for each reaction family, and a rank limitation was place upon those with the greatest number of reactions. The first limitation was placed on dealkylation reactions, but a second order rank limitation was not sufficient in curbing the reaction family. Another run was then performed with a first order rank limitation on dealkylation reactions and the results were better.

The third run attempted to cut down on the number of ring saturation type reactions by limiting all ring saturation types (2H, 4H, and 6H) to the second rank. A rank-two

limitation would allow initial seeded species to dealkylate, and otherwise react, before becoming saturated.

The fourth step in limiting the network was curtailing the side chain cracking of ring species by instilling a rank-1 limitation. Much like the limitation to dealkylation, seeded species were allowed to undergo side chain cracking, but not species otherwise produced within the reaction network.

The final step was the placement of a third rank limitation ring isomerization reactions. The ring isomerization reactions were of great importance in the study of the network, but as with all isomerization reactions, too many species were produced. By limiting the reaction to rank-three, isomerizations could occur throughout most of the network.

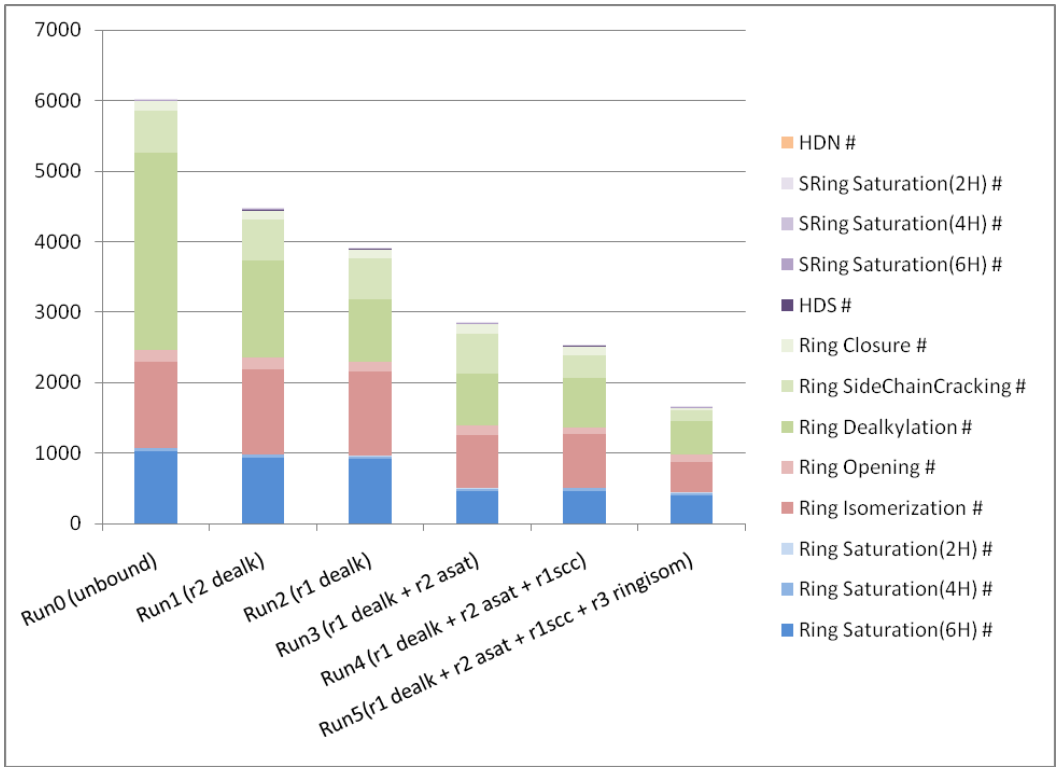


Figure 6.2 - Reaction family breakdown of the total number of forward reactions

Overall, the reaction network produced the desired results, but the size of the network was still a bit large for the study to be conducted. All the species of the limited network needed to be analyzed and re-combined into representative species. In hopes of avoiding that step, the zero-order network growth approach was taken.

The second methodology, that of the rank zero network, initially used all the species that were determined experimentally in the feed as seeds. The network generation options for any species/reaction combination were only limited to rank zero; no further limitation was allowed. After processing of the network, 35 reactions were allowed amongst the 140 species.

INGen was the re-seeded with all the feed molecules and the species only found in the product stream. The number of reactions increased to 35, largely due to increases in the number of ring saturation reactions.

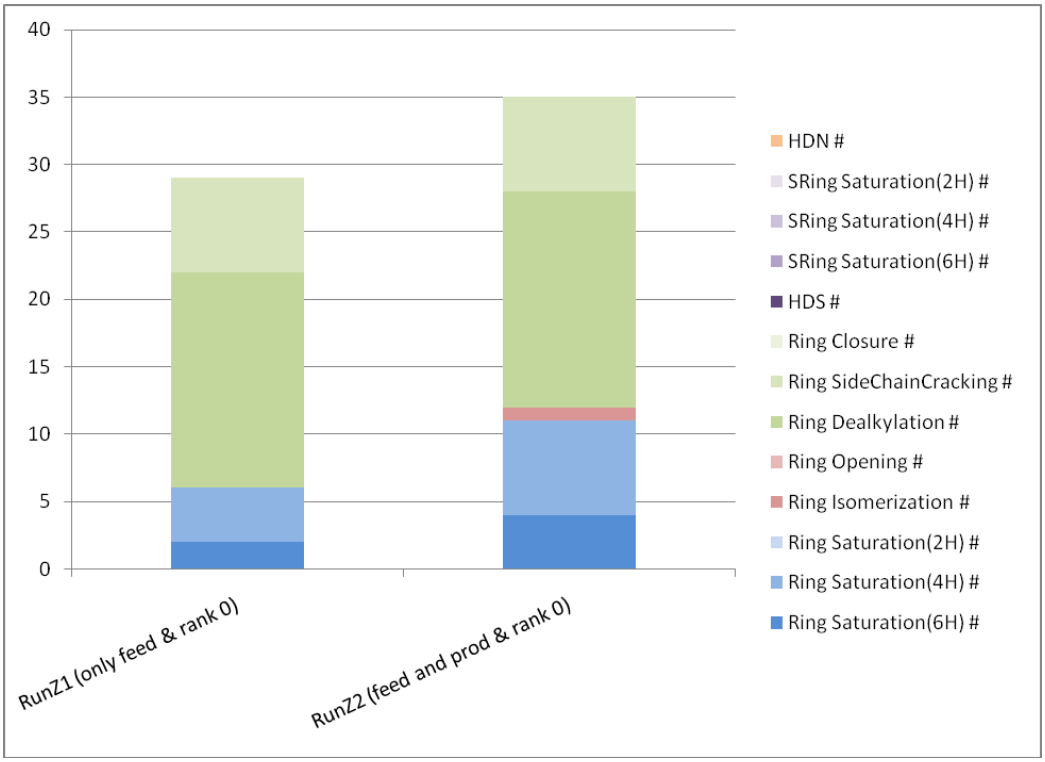


Figure 6.3 - Reaction family breakdown of the total number of forward reactions for a zero order network

The total number of reactions and species for the zero-order network were only a small fraction of those for the unbound case study. If the network included all important species and reactions, the smaller network would indeed be easier to tune to laboratory data. Unfortunately, simply including products in the seed does not ensure that they will be produced. "Intermediate" species could have been consumed in full upon creation, or were simply not expressly found in the product analysis. Such missing links will not allow a rank-zero limitation to generate a representative network without determination of their identities. Re-seeding with those intermediates would allow the zero-order network to be generated properly, but require the pathways to be known ahead of time.

6.3 Conclusion

In conclusion, two valid methods for creating the hydro-processing network were established. In one, the size and shape of the network was controlled using modeler supplied parameters. The remaining discontinuity between model and experimental species was alleviated by massaging unmeasured isomeric information into representative species.

The second methodology required that representative feed and product species, as determined by experimental analysis, to be seeded into INGen. No species that were not seeded were allowed to form. Reaction processes that include unmeasured intermediates will not produce a viable network without the re-seeding of those intermediate species. Determination of the intermediate species requires fore-knowledge of the network structure. Therefore, a new hybrid methodology is formulated.

By creating a network via the traditional first methodology, specific pathways can be studied for the creation of each analyzed product. Any intermediate species that are not measured in the feed or the products are noted, and their adjacency lists are added to the list of seeds for the second methodology. With a complete set of reactants, appropriate intermediates, and products, a rank-zero limited network would then keep the number of reactions and species to a bare minimum.

INGen's ability to quickly model and analyze networks proved to be a great boon in the study of the hydroprocessing network. New starting species were created in a fraction of the time, multiple networks were created almost instantly, and their results could be easily tabulated without leaving the Excel environment. The addition of certain ring-type pathways to the core functionality of INGen ensured the proper production of certain species. Finally, the ability of INGen to allow species/reaction combination limiting

techniques and purely seed driven network generation, or a combination thereof, shows the extensibility of applying INGen to solve this “real-world” hydroprocessing challenge.

Chapter 7. Summary and Conclusions

7.1 Summary

Research in anticipation of the world's impending energy crisis includes the development of large molecule-based kinetic models for the reactions of complex mixtures of complex molecules. INGen, the Interactive Network Generator, is the first model-builder tool that allows chemists and chemical engineers without specialty computer programming knowledge to build these kinetic models easily and quickly. Prior to INGen, modeling at the molecular level has not had wide spread use in the energy industry because of its complexity. Although the overlying principles of molecular modeling are easy to understand, historic implementations required a deeper knowledge of programming than chemistry. INGen refocuses the problem, allowing non-programmers to have access to a large set of molecule based modeling techniques.

INGen helps develop environmentally friendly, sustainable solutions by allowing modelers to de-lump harmful chemicals from their traditional models. Separate model species for each harmful molecule allow for a more accurate prediction of concentrations. In addition, as newer, more environmentally friendly processes are being developed, INGen will allow for the careful modeling of the reaction network in order to maintain conversion and selectivity (direct profit!) and lessen environmental impact (fewer fines and less spent on disposal = profit!).

INGen is equipped to handle a wide variety of new feed stocks. Some new feed stocks will come from renewable resources such as biomass, others will come from the still plentiful coal, tar-sands and shale resources. INGen provides the ability to include new species as quickly as they can be drawn. When using a molecule-based modeling

methodology, the chemical principles and reaction families underlying the processing of hydrocarbons remain independent of the feedstock.

INGen accomplished its goal of becoming both a powerful and useful molecular modeling tool in a two-fold manner. The chemistry behind INGen represents the bulk of hydrocarbon processing techniques: acid, metal, and free radical reaction families. INGen allows for combinations of the three families to be built using mechanistic and pathways level modeling. INGen's implementation of pathways level modeling encompasses both "blind eye" pathways and mechanism-specific pathways. In this manner, INGen can be considered the first universal hydrocarbon model builder.

INGen's second approach to achieving acceptance was the creation of a graphical user interface. INGen used the power behind Microsoft's Excel to help create a familiar environment through which results can quickly and easily be tabulated and analyzed. The interface was designed to logically lead a user through the tenets of model building. With INGen, a new user can quickly build complicated networks in a manner of hours. This is a feat that no other molecular model builder has achieved.

7.2 Applications

INGen was used to successfully model the branching isomerization and cracking of a normal-paraffin through carbenium ion specific pathways. Studies were conducted on determining the number of species for a fully determined C70 system, $O(10^{11})$. In attempting to model such a system, smaller experiments were performed to analyze the various techniques available to limit the number of reactions and species. A C20 network was successfully created using a combination of rank limitations and changes to the maximum number of reactions that can be performed on a species per carbon number.

Path C type carbenium ion reaction pathways were observed as a primary target for limitation. Limitations to paths A, B1 and B2 were handled as a unit, separately from those of Path C. In addition, for large carbon numbers the number of isomerization reactions approached the number of cracking reactions.

The C70 network was created using this knowledge, and visualizations of the continuity of species. A fully represented network was created that contained only 2,698 species and 8,181 reactions.

In addition, INGen was successfully used to create a hydroprocessing network using specific feed and product species identities. New seed species were easily created using ChemDraw and then quickly imported into INGen. First, a fully specified network was created encompassing 3,000 species and 8,500 reactions. By analyzing the breakdown of reaction types, the network was successfully limited to about a third of its size. Additional work was done to create a zero-order network wherein only seeded species were allowed to be generated. The reactant and product species of the process were known via analytical experimentation, and each was given as a seed to INGen. The resultant network had about 153 species and only 140 reactions. The lack of reaction pathways that led to expected products meant that either unmeasured intermediates were formed, or critical isomer differentiation of the products failed.

Overall, the power and usefulness of INGen was proven in the rapid creation of the networks and the ease of analysis.

7.3 Conclusions

INGen allows non-programmers to model complex kinetic networks with a minimally

tedious effort. Networks for the study and modeling of hydrocarbon processes can be created within minutes, their network composition results analyzed within hours, and a finalized model can be had within days. INGen represents an outstanding step forward in the promotion of molecule based modeling.

By creating a seamless interface between the user and the underlying code, INGen will greatly expand the audience of this technology in hopes of solving some of the world's current and future energy and environmental problems.

7.4 Future work

INGen represents the basis for a truly universal model builder. The reaction families included represent the lion's share of hydrocarbon processing technologies, but there are the remaining application specific (and not as generally useful) pathways and mechanisms that should eventually be included. In addition, innumerable reaction families lie outside of the hydrocarbon industry and can be incorporated as the need arises.

Pathways built with specific mechanistic knowledge were developed and differentiated for only a few reaction families. Additions could be made for methyl shift isomerizations, fused ring isomerizations, as well as a number of other pathways. Further development of free radical mechanistic network generation monitoring and limitations would also prove useful.

New reactions should be entered using the paradigm of site selection outlined in this thesis dissertation, or by developing a proposed model-builder builder. The further abstraction of automated model building leads to the unexplored theory of model builder building. A reaction is defined by its site selection and its reaction matrix. Creating a reaction matrix

(and the surrounding code) are conceptually easy. The site selection could be accomplished by first creating a sub-graph adjacency list for the site environment and using established graph theory to locate the sub-graph within the molecular graph. A labeling system could be used to identify the specific atoms within the sub-graph as those that will be the true reaction sites. In this way, reactions could be added by defining a new adjacency list and reaction matrix.

Energy calculations should be added for integration with GAMESS or similar software. By incorporating limits to heats of reaction, the network could be automatically pared based on energy principles. In addition, further modeling with KME or a similar package would benefit greatly from pre-calculated values.

Such changes would enhance the immediate utility of INGen, and make further development easier.

CURRICULUM VITA

Craig Allen Bennett

Professional Experience

1999-2008 Rutgers, The State University of New Jersey, Piscataway, New Jersey
Associate Technical Director – School of Engineering

1998-1999 Engelhard Corporation, Iselin, New Jersey
Associate Engineer – Research and Development

Colleges Attended

2009 Rutgers, The State University of New Jersey, Piscataway, New Jersey
Doctor of Philosophy in Chemical and Biochemical Engineering

2005 Rutgers, The State University of New Jersey, Piscataway, New Jersey
Master of Science in Chemical and Biochemical Engineering

1998 University of Delaware, Newark, Delaware
Bachelor of Chemical Engineering

Publications

Zhen Hou, Craig A. Bennett, Michael T. Klein, and Preetinder S. Virk, Approaches and Software Tools for Modeling Lignin Pyrolysis, Energy Fuels XXXX, XXX, 000–000.

Darren M. Campbell, Craig A. Bennett, Zhen Hou, and Michaele T. Klein, Attribute-based Modeling of Resid Structure and Reaction, Ind. Eng. Chem. Res., Vol. 48, No. 4, 2009, 1683-1693.

Wei Wei, Craig A. Bennett, Ryuzo Tanaka, Gang Hou and Michael T. Klein, Detailed Kinetic Models for Catalytic Reforming, Fuel Processing Technology, 89 (2008) 344–349.

Wei Wei, Craig A. Bennett, Ryuzo Tanaka, Gang Hou, Michael T. Klein Jr. and Michael T. Klein, Computer Aided Kinetic Modeling with KMT and KME, Fuel Processing Technology, 89 (2008) 350–363.

Wei Wei, Craig A. Bennett, Ryuzo Tanaka, Gang Hou and Michael T. Klein , DETAILED KINETIC MODELS FOR CATALYTIC REFORMING, Prepr. Pap.-Am. Chem. Soc., Div. Fuel Chem. 2005, 50 (4), p. 386